ORIGINAL PAPER

Support vector machine guided reproducing kernel particle method for image-based modeling of microstructures

Yanran Wang[1] · Jonghyuk Baek[1] · Yichun Tang[2] · Jing Du[2] · Mike Hillman[3] · Jiun-Shyan Chen[1]

Abstract

This work presents an approach for automating the discretization and approximation procedures in constructing digital representations of composites from micro-CT images featuring intricate microstructures. The proposed method is guided by the Support Vector Machine (SVM) classification, offering an effective approach for discretizing microstructural images. An SVM soft margin training process is introduced as a classification of heterogeneous material points, and image segmentation is accomplished by identifying support vectors through a local regularized optimization problem. In addition, an Interface-Modified Reproducing Kernel Particle Method (IM-RKPM) is proposed for appropriate approximations of weak discontinuities across material interfaces. The proposed method modifies the smooth kernel functions with a regularized Heaviside function concerning the material interfaces to alleviate Gibb's oscillations. This IM-RKPM is formulated without introducing duplicated degrees of freedom associated with the interface nodes commonly needed in the conventional treatments of weak discontinuities in the meshfree methods. Moreover, IM-RKPM can be implemented with various domain integration techniques, such as Stabilized Conforming Nodal Integration (SCNI). The extension of the proposed method to 3-dimension is straightforward, and the effectiveness of the proposed method is validated through the image-based modeling of polymer-ceramic composite microstructures.

Keywords Image-based modeling · Support vector machine · Reproducing kernel particle method · Weak discontinuity · Microstructures

1 Introduction

In recent years, a variety of non-destructive imaging techniques, such as micro-X-ray computed tomography (micro-CT), have emerged as powerful alternatives to obtain detailed information about the microstructure and internal deformation of composite materials [1–4]. Nevertheless, modeling microstructures remains challenging owing to their geometrical and topological complexities and heterogeneity, making the body-fitted mesh generation for mesh-based methods extremely tedious and time-consuming, especially in the three-dimension model construction. An example of a 2D

micro-CT image slice of a polymer-ceramic composite specimen (polymer matrix reinforced by ceramic particles) is shown in Fig. 1a with a resolution of 8 μm, and a body-fitted FEM mesh is generated for a selected Region of Interest (ROI) of 200 by 200 pixels in Fig. 1b to demonstrate the meshing complexity. As can be seen, body-fitted meshing requires significant mesh refinement near material interfaces of inclusions with complex geometries. This adds complexity to mesh generation and demands extensive mesh density in the discretization, yielding 37,454 elements and 112,538 nodes in this demonstration example.
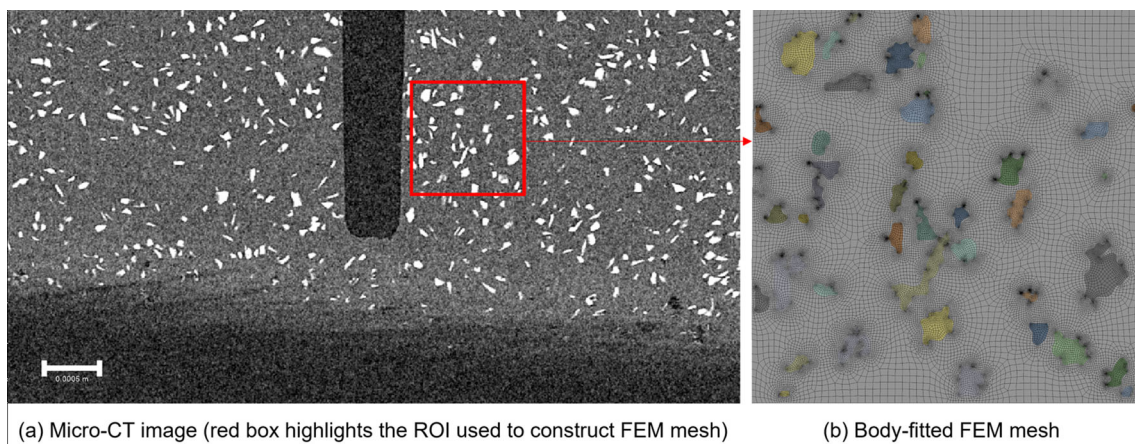
Various image segmentation techniques have been developed over the past several decades, including region-based and classification-based methods [5]. Global and local thresholding [6] is a simple region-based method that uses a threshold value to separate objects from the background, but it can lead to poor results if the threshold is not chosen correctly. The region growing method [7] is another region-based approach that relies on user-selected seed pixels and offers advantages over thresholding, but the numerical

✉ Jiun-Shyan Chen
  js-chen@ucsd.edu

1   Department of Structural Engineering, University of
    California San Diego, La Jolla, CA 92093-0085, USA

2   Department of Mechanical Engineering, Pennsylvania State
    University, University Park, PA 16802, USA

3   Karagozian and Case, Inc, Glendale, CA 91203, USA

(a) Micro-CT image (red box highlights the ROI used to construct FEM mesh)   (b) Body-fitted FEM mesh

Fig. 1 Micro-CT image of a polymer-ceramic composite microstructure and its corresponding body-fitted finite element mesh

results can be sensitive to the selection of initial seed points. Clustering-based methods [8], such as K-mean clustering, hierarchical clustering, and Gaussian Mixture Models, are region-based, unsupervised algorithms that partition images into local regions or clusters based on the similarity of their attributes. They can perform image segmentation directly with image pixel information without labeled data, but appropriate selection of features and number of clusters are essential to the effectiveness of many clustering methods. On the contrary, classification-based methods generally adopt a global approach for image segmentation, whereby an automatic pattern recognition process is utilized in the context of supervised learning based on manually segmented training datasets. K-Nearest Neighbor is a simple, non-parametric supervised learning model that makes predictions based on the k-nearest neighbors in the training data, but it usually requires a large amount of training data to suppress high variance problems [9]. Tree-based algorithms, including decision trees and random forests, are another widely used supervised learning techniques in which the training data is partitioned into smaller subsets without much data pre-processing and with high interpretability and computational efficiency. However, these methods may have limited ability to predict unseen data, restricted decision boundary expressiveness, and can be sensitive to imbalanced data [10]. Recently, deep learning algorithms have enabled to develop state-of-the-art image segmentation methods, especially those based on convolutional neural networks, which can automatically learn features from raw images with minimal human interaction. However, these methods require large amounts of labeled datasets with extensive training and are mathematically more challenging to interpret due to the highly non-linear relationships between input features and output labels [11, 12]. Other techniques besides machine learning algorithms have also been widely adopted for image segmentation. The level set method originated from Osher and Sethian [13] uses an

auxiliary function to represent and track the evolution of interfaces in images [14]. While it is capable of handling irregular shapes and complex topologies, identification of zero level set can be time consuming. Fast Fourier Transform explores the frequency domain of images where quick computations can be performed on frequency components of images. This algorithm is highly efficient as it operates in low-dimensional frequency domain. However, Fast Fourier Transform provides limited direct spatial localization information of images features, and it works more optimally for certain image processing task like image registration [15] and periodic patterns reconstruction and segmentation [16].

The present work employs the Support Vector Machine (SVM) algorithms as the image segmentation method to guide the numerical model generation. SVM is a classification-based machine learning algorithm built on solid mathematical foundation and optimization frameworks [17, 18]. Compared to other supervised algorithms, SVM is advantageous because it generates a unique maximum-margined global hyperplane for separating training datasets, providing a global solution for data classification. Additionally, it is not sensitive to the underlying probabilistic distribution of the training dataset, ensuring high performance for limited, noisy, or imbalanced datasets [19]. One apparent limitation of the standard SVM is that it requires $O(l^3)$ operations, where l is the length of the training dataset, to solve a complex quadratic programming problem (QPP) with inequality constraints. Various approaches have been proposed to overcome this limitation, such as the training decomposition method [20] and the reduced support vector machine algorithm [21], which significantly improves SVM's training speed. Additionally, more efficient formulations of SVM have been introduced, such as the Least Square SVM algorithm [22] and the Lagrangian SVM algorithm [23]. The Least Square SVM algorithm optimizes a dual

problem directly using a least-square loss function, replacing the hinge loss function in the original SVM's formulation to reformulate the complex QPP as a linear system of equations. In contrast, the Lagrangian SVM algorithm utilizes an implicit Lagrangian for the dual of the standard quadratic program of a linear SVM, leading to the minimization of an unconstrained differentiable convex function in the space of dimensionality equal to the number of training datasets. Both mentioned algorithms eliminate the necessity of complicated programming problem solvers, making them feasible for classifying large datasets. In addition to the binary SVM classifier, extensive research has been done to extend SVM to multi-class classification. The one-vs-all method, one-vs-one method, error-correcting output codes, and directed acyclic graphs are among the most widely used approaches to handle multi-class classification with SVM [24]. The traditional binary SVM algorithm is adopted in this work for its effective applicability to the two-phase materials.

Numerical modeling of heterogeneous materials remains challenging for both mesh-based methods discretized with body-fitted discretization and meshfree methods formulated with smooth approximations. For the Finite Element Method (FEM), incomplete handling of discontinuities in mesh construction can lead to suboptimal convergence [25], and aligning meshes with interfaces is a non-trivial task for composites with complex microstructures and significant variations in constituent moduli. The Finite Cell Method is a high-order embedded domain technique [26] that provides simple yet effective modification of traditional FEM to bypass the necessity of exhaust body-fitted meshing for geometrically and topologically complex microstructures. Korshunova et al. [27, 28] presented image-based numerical characterization and validation of additively manufactured structures using Finite Cell Method and numerical homogenization. Special numerical integration schemes need to be considered to differentiate between inside and outside the physical domain for the Finite Cell Method. The meshfree methods utilize point-wise discretization instead of carefully constructed body-fitted meshes. However, meshfree methods such as element-free Galerkin (EFG) [29] and reproducing kernel particle method (RKPM) [30–32] typically suffer from Gibb's-like oscillation in the approximation when modeling weak continuities in composite materials, as their smooth approximation functions with overlapping local supports fail to capture gradient jump conditions across material interfaces [33]. Considerable effort has been dedicated to developing effective techniques for dealing with interface discontinuities. Since the proposed work is under the Galerkin meshfree framework, the review of methods developed based on mesh-based context to address interface discontinuities is omitted here. Reviews on some key non body-fitted FEM developments for interface discontinuities can be found in [34, 35].

Two primary approaches in meshfree methods have been proposed for handling material interface weak discontinuities. The first approach involves introducing discontinuities in the meshless approximation function. Krongauz and Belytschko proposed two types of jump enrichment functions into the conventional Moving Least Squares or Reproducing Kernel (RK) approximation of the field variables [33]. The enrichment functions introduce discontinuous derivatives into solutions along material interfaces, but additional unknowns must be solved in this method. Chen et al. [36] introduced the jump enrichment functions into the RK shape function based upon enforcing the consistency conditions, which is termed the interface-enriched reproducing kernel approximation. However, coupling interface-enriched RK shape functions with the standard RK shape function requires duplicated unknowns. In addition, Masuda and Noguchi introduced a discontinuous derivative basis functions to replace the conventional polynomial basis function used in Moving Least-Squares approximations [37]. Another class of methods introduces modifications to the weak formulation to consider the effects of discontinuity in a weak sense. Codes and Moran treated material interface discontinuities by a Lagrange Multiplier technique so that the approximations are disjoint across the interfaces while the Lagrange Multiplier imposes the interface continuity constraints into the variational formulation of the meshfree discretization [38]. This approach introduces additional degrees of freedom to be solved associated with the Lagrange Multiplier, and stability conditions need additional attention. On the other hand, the discontinuous Galerkin formulation has also been considered, where the continuity of a field variable and its resulting interface flux or traction across interfaces are imposed in the weak form [39, 40]. Wang et al. proposed a Discontinuous Galerkin reformulation of the EFG and RKPM to address interface discontinuity problems of composite materials [41]. This approach avoids duplicated unknowns, and by decomposing the domain into patches, the gradient jump of the dependent variable is captured by the boundary of the adjacent patches while the continuity condition is realized weakly through an augmented variational form with associated flux or traction crossing material interfaces. Additionally, other meshfree methods have also been proposed for non-body-fitted discretization of heterogeneous media, such as the immersed methods [42, 43]. However, these methods require special care of interface oscillations due to the employment of volumetric constraints on the foreground and background discretization.

The current work introduces a novel Interface-Modified Reproducing Kernel Particle Method (IM-RKPM) to properly handle weak discontinuities in composite materials across material interfaces. The proposed approach utilizes signed distance functions obtained from SVM classified

micro-CT images to introduce regularized weak discontinuities to the kernel functions for arbitrary interface geometries. No duplicated unknowns, special enrichment functions, or complicated reformulation of the RK shape functions are required in the proposed approach, offering automated model construction capabilities for modeling complex microstructures.

The remainder of the paper is organized as follows. Section 2 provides basic equations for the model problem and the Reproducing Kernel Particle Method, and the associated numerical domain integration techniques are also discussed in this section. A brief introduction of the SVM formulation and the proposed SVM and RK-guided procedures for image segmentation of heterogenous materials are introduced in Sect. 3. Section 4 presents an interface-modified kernel function and the Interface-Modified Reproducing Kernel Particle Method formulation to introduce weak discontinuities in the image-based modeling of composite microstructures. In Sect. 5, two numerical examples for image-based modeling of microstructures are demonstrated, and the paper concludes with a discussion and summary in Sect. 6. Throughout the paper, the following abbreviations and symbols have been introduced:

- SVM: Support Vector Machine
- IM-RKPM: Interface-Modified Reproducing Kernel Particle Method
- SCNI: Stabilized Conforming Nodal Integration
- micro-CT: micro-X-ray Computed Tomography
- ROI: Region of Interest
- QPP: Quadratic Programming Problem
- FEM: Finite Element Method
- EFG: Element-Free Galerkin
- RKPM: Reproducing Kernel Particle Method
- SVM-RK: Support Vector Machine Guided Reproducing Kernel
- B2: Quadratic B Spline Kernel Function
- B3: Cubic B Spline Kernel Function
- SEM: Scanning Electron Microscopy
- EDS: Energy Dispersive X-ray Spectroscopy
- PK: Power Kernel Function
- $\mathbb{S}^{RK}$: SVM-RK discretization node set containing total number of NP nodes
- $\Psi_I(\boldsymbol{x})$: RK shape function with support centered at node $\boldsymbol{x}_I$ and evaluated at point $\boldsymbol{x}$
- $\phi_a(\boldsymbol{x} - \boldsymbol{x}_I)$: RK kernel function with a compact support a defined over a node I's associated subdomain Ω_I
- $\boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)$: Vector of monomial basis functions to the order of n
- $\boldsymbol{M}(\boldsymbol{x})$: Moment matrix
- $\widetilde{\nabla}\Psi_I(\boldsymbol{x}_L)$: Smoothed RK shape function gradient evaluated at nodal integration point $\boldsymbol{x}_L$

- $\widetilde{\boldsymbol{B}}(\boldsymbol{x}_L)$: Gradient matrix associated with smoothed nodal RK shape function gradient
- $\mathbf{D} = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^l$: Labeled SVM training data set containing l pairs of training data $\boldsymbol{x}_i \in \mathbb{R}^d$ and corresponding response label $y_i \in \{-1, 1\}$
- $\boldsymbol{w}$: d-dimensional weight vector
- b: A scalar bias
- $h(\boldsymbol{x}; \{\boldsymbol{w}, b\})$: d-dimensional separating hyperplane
- $\{\boldsymbol{x}_i^{SV}\}_{i=1}^{N_{SV}}$: The set of support vectors with a total number of N_{sv} data points
- ξ^*: Margin
- $h^*(\boldsymbol{x})$: Optimal separation hyperplane that maximizes the margin
- $\nabla\epsilon_i$: Slack variable corresponding to the ith data point
- C: Penalty weight parameter
- $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$: Kernel function used in non-linear SVM formulation
- γ: Gaussian radial basis kernel scale
- $S(\boldsymbol{x})$: SVM classification score function
- $\mathbb{S}^0$: A set of training data points located at image pixel centroids with a total number of NP_0 points
- $\mathbb{S}^+$: A subset of $\mathbb{S}^0$ that contains elements with non-negative classification scores
- $\mathbb{S}^-$: A subset of $\mathbb{S}^0$ that contains elements with negative classification scores
- $\{\boldsymbol{x}_K^+, \boldsymbol{x}_K^-\}_{K=1}^{NP_{IF}}$: Interface-searching node pairs with $\boldsymbol{x}_K^\pm \in \mathbb{S}^\pm$ as a near-interface master/slave node
- d_K^*: A scaler line search step for identifying interface node $\boldsymbol{x}_K^*$
- $\boldsymbol{R}_K$: Interface node line search direction for identifying interface node $\boldsymbol{x}_K^*$
- $\mathbb{S}^{IF}$: Interface node set, containing a total number of NP^* identified interface node $\boldsymbol{x}^*$
- $\mathbb{S}^{C+}$: Master candidate node set, a subset of $\mathbb{S}^+$ containing near-interface master nodes
- $\mathbb{S}^{C-}$: Slave candidate node set, a subset of $\mathbb{S}^-$ containing near-interface slave nodes
- $\widetilde{S}(\boldsymbol{x})$: RK shape function interpolated classification score function
- s_I: SVM classification score value for node $\boldsymbol{x}_I \in \mathbb{S}^0$
- $\Psi_I^0(\boldsymbol{x})$: RK shape function centered at node $\boldsymbol{x}_I \in \mathbb{S}^0$
- $\widetilde{H}$: Regularized Heaviside function for interface modification
- $\overline{\phi}_a(\boldsymbol{x} - \boldsymbol{x}_I)$: Modified RK kernel function with a compact support a
- $\overline{\xi}(\boldsymbol{x})$: A distance measure normalized with respected to the nodal spacing
- $\overline{\boldsymbol{M}}(\boldsymbol{x})$: Modified moment matrix
- $\overline{\Psi}_I(\boldsymbol{x})$: Interface modified reproducing kernel shape function
- $\{\boldsymbol{x}_K^{*+}, \boldsymbol{x}_K^{*-}\}$: Mirrored node pair for the interface node $\boldsymbol{x}_K^*$

2 Basic equations

2.1 Model problem

Let a model elasticity problem be defined on a domain Ω with its boundary assigned as $\partial\Omega = \partial\Omega_g \cup \partial\Omega_t$, $\partial\Omega_g \cap \partial\Omega_t = \varnothing$, where the subscripts g and t denote the Dirichlet and Neumann boundaries, respectively. The strong form for heterogeneous elastic media can be described as:

$$
\begin{aligned}
\nabla \cdot \boldsymbol{\sigma} + \boldsymbol{s} &= 0 \quad \text{in} \quad \Omega \\
\boldsymbol{u} &= \overline{\boldsymbol{u}} \quad \text{on} \quad \partial\Omega_g \\
\boldsymbol{n} \cdot \boldsymbol{\sigma} &= \boldsymbol{t} \quad \text{on} \quad \partial\Omega_t
\end{aligned}
\tag{1}
$$

where $\boldsymbol{u}$ represents the unknown displacement field, $\boldsymbol{\sigma}$ is the Cauchy stress tensor, $\boldsymbol{s}$ is the body force vector, $\overline{\boldsymbol{u}}$ and $\boldsymbol{t}$ are the prescribed displacement vector and the applied surface traction vector on the Dirichlet and Neumann boundaries, respectively, and $\boldsymbol{n}$ is the unit outward normal of the Neumann boundaries. The elastic constitutive relationship for heterogeneous materials is represented as:

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

Here $\boldsymbol{C}(\boldsymbol{x})$ is the elasticity tensor defined as:

$$
\boldsymbol{C}(\boldsymbol{x}) = \begin{cases} \boldsymbol{C}^1, & \boldsymbol{x} \in \Omega^1 \\ \boldsymbol{C}^2, & \boldsymbol{x} \in \Omega^2 \end{cases}
\tag{3}
$$

where Ω^i are material sub-domains to be segmented by the SVM classification of microstructure image pixels.

The weak formulation is to find $\boldsymbol{u}(\boldsymbol{x}) \in U \subset H_g^1$, such that for all weight function $\boldsymbol{v}(\boldsymbol{x}) \in V \subset H_0^1$,

$$
\int_\Omega \boldsymbol{\varepsilon}(\boldsymbol{v}) : \boldsymbol{\sigma}(\boldsymbol{u}) d\Omega = \int_\Omega \boldsymbol{v} \cdot \boldsymbol{s} d\Omega + \int_{\partial\Omega_h} \boldsymbol{v} \cdot \boldsymbol{t} d\Gamma
\tag{4}
$$

The Galerkin formulation seeks the trial solution function $\boldsymbol{u}^h \in U^h \subset U$, so that for all weight function $\boldsymbol{v}^h \in V^h \subset V$,

$$
\int_\Omega \boldsymbol{\varepsilon}\left(\boldsymbol{v}^h\right) : \boldsymbol{\sigma}\left(\boldsymbol{u}^h\right) d\Omega = \int_\Omega \boldsymbol{v}^h \cdot \boldsymbol{s} d\Omega + \int_{\partial\Omega_h} \boldsymbol{v}^h \cdot \boldsymbol{t} d\Gamma
\tag{5}
$$

2.2 Reproducing kernel approximation

Let a closed domain $\overline{\Omega} = \Omega \cup \partial\Omega \subset \mathbb{R}^d$ be discretized by a set of NP nodes denoted by $\mathbb{S}^{\text{RK}} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{NP} | \boldsymbol{x}_I \in \overline{\Omega}\}$, and let the approximation of a field variable $\boldsymbol{u}(\boldsymbol{x})$ in $\overline{\Omega}$ be denoted by $\boldsymbol{u}^h(\boldsymbol{x})$. The RK approximation of the field variable $\boldsymbol{u}(\boldsymbol{x})$ based on the discrete points in the set $\mathbb{S}^{\text{RK}}$ is formulated as follows:

$$
u_i^h(\boldsymbol{x}) = \sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) d_{iI}
\tag{6}
$$

where Ψ_I denotes the RK shape function with support centered at the node $\boldsymbol{x}_I$ and d_{iI} is the nodal coefficient in i^{th} dimension to be sought. Moreover, let a node I be associated with a subdomain Ω_I, over which a kernel function $\phi_a(\boldsymbol{x} - \boldsymbol{x}_I)$ with a compact support a is defined, such that $\overline{\Omega} \subset \bigcup_{I \in \mathbb{S}^{\text{RK}}} \Omega_I$ holds. The RK approximation function is constructed as [30–32, 44]:

$$
\Psi_I(\boldsymbol{x}) = C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I) \phi_a(\boldsymbol{x} - \boldsymbol{x}_I) = \left(\sum_{|\alpha| \leq n} (\boldsymbol{x} - \boldsymbol{x}_I)^\alpha b_\alpha(\boldsymbol{x}) \right) \phi_a(\boldsymbol{x} - \boldsymbol{x}_I)
$$
$$
\equiv \boldsymbol{H}^{\text{T}}(\boldsymbol{x} - \boldsymbol{x}_I) \boldsymbol{b}(\boldsymbol{x}) \phi_a(\boldsymbol{x} - \boldsymbol{x}_I)
\tag{7}
$$

$$
\boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I) = [1, x_1 - x_{1I}, x_2 - x_{2I}, \ldots, (x_3 - x_{3I})^n]
\tag{8}
$$

where α is a multi-index notation such that $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_d)$ with a length defined as $|\alpha| = \alpha_1 + \alpha_2 + \cdots + \alpha_d$, and $\boldsymbol{x}^\alpha \equiv x_1^{\alpha_1} \cdot x_2^{\alpha_2}, \ldots, x_d^{\alpha_d}$, $b_\alpha = b_{\alpha_1 \alpha_2 \cdots \alpha_d}$. The term $C(\boldsymbol{x}; \boldsymbol{x} - \boldsymbol{x}_I) = \boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I) \boldsymbol{b}(\boldsymbol{x})$ is called the correction function of the kernel $\phi_a(\boldsymbol{x} - \boldsymbol{x}_I)$ designed to introduced completeness to the RK approximation. The terms $\{(\boldsymbol{x} - \boldsymbol{x}_I)^\alpha\}_{|\alpha| \leq n}$ form a set of basis functions, and $\boldsymbol{H}^T(\boldsymbol{x} - \boldsymbol{x}_I)$ is the corresponding vector of basis functions to the order n. The vector $\boldsymbol{b}(\boldsymbol{x})$ is the coefficient vector of $\{b_\alpha(\boldsymbol{x})\}_{|\alpha| \leq n}$ and is solved by enforcing the following discrete reproducing conditions [45]:

$$
\sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) \boldsymbol{x}_I^\alpha = \boldsymbol{x}^\alpha, \quad |\alpha| \leq n
\tag{9}
$$

or equivalently,

$$
\sum_{I=1}^{NP} \Psi_I(\boldsymbol{x})(\boldsymbol{x} - \boldsymbol{x}_I)^\alpha = \delta_{0\alpha}, \quad |\alpha| \leq n
$$

Or

$$
\sum_{I=1}^{NP} \Psi_I(\boldsymbol{x}) \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) = \boldsymbol{H}(\boldsymbol{0})
\tag{10}
$$

where $\boldsymbol{H}(\boldsymbol{0}) = [1, 0, \ldots, 0]^T$ according to Eq. (8). After inserting Eq. (7) into Eq. (10), $\boldsymbol{b}(\boldsymbol{x})$ is obtained as:

$$
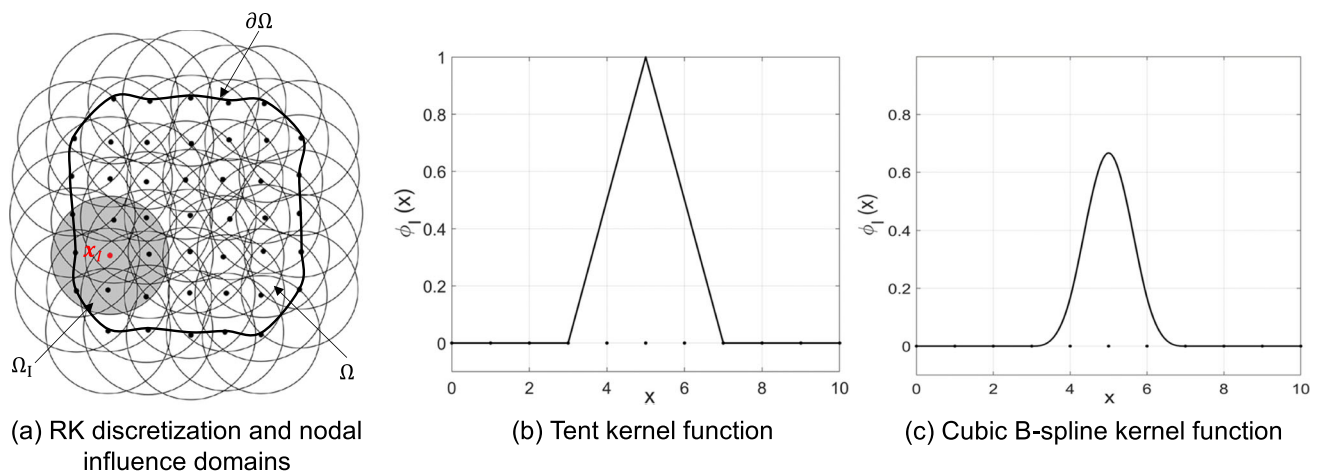\boldsymbol{b}(\boldsymbol{x}) = \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{H}(\boldsymbol{0}) \phi_a(\boldsymbol{x} - \boldsymbol{x}_I)
\tag{11}
$$

where $\boldsymbol{M}(\boldsymbol{x})$ is the moment matrix and is formulated as:

$$
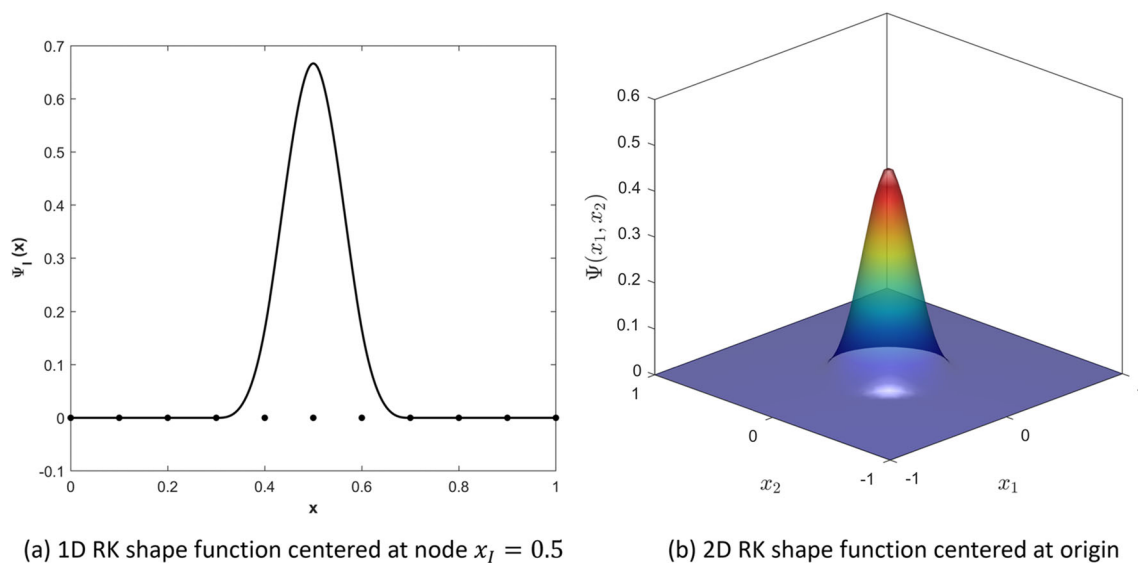\boldsymbol{M}(\boldsymbol{x}) = \sum_{I=1}^{NP} \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) \boldsymbol{H}^{\text{T}}(\boldsymbol{x} - \boldsymbol{x}_I) \phi_a(\boldsymbol{x} - \boldsymbol{x}_I)
\tag{12}
$$

Finally, the RK shape function is obtained as:

$$
\Psi_I(\boldsymbol{x}) = \boldsymbol{H}^{\text{T}}(\boldsymbol{0}) \boldsymbol{M}^{-1}(\boldsymbol{x}) \boldsymbol{H}(\boldsymbol{x} - \boldsymbol{x}_I) \phi_a(\boldsymbol{x} - \boldsymbol{x}_I)
\tag{13}
$$

(a) RK discretization and nodal influence domains

(b) Tent kernel function

(c) Cubic B-spline kernel function

Fig. 2 RK domain discretization and examples of kernel functions

(a) 1D RK shape function centered at node $x_I = 0.5$

(b) 2D RK shape function centered at origin

Fig. 3 Examples of 1D and 2D RK shape functions constructed based on the cubic B-spline kernel function and linear basis functions

Examples of 1-dimensional kernel functions are shown in Fig. 2, and the 1- and 2-dimensional RK shape functions constructed based on cubic B-spline kernel function and linear basis functions are shown in Fig. 3. The locality and the smoothness of the RK approximation functions are determined by the kernel function, while the order of completeness in the approximation is determined by the order of basis functions n. Interested readers are referred to [30–32, 44, 46, 47] for basic properties of reproducing kernel approximation.

2.3 Numerical domain integration

Due to the rational nature and arbitrary local supports, introducing RK approximations in the Galerkin weak form requires special attention. The conventional Gauss integration on background integration cells leads to a sub-optimal convergence unless significantly high-order quadrature rules

are used, which is computationally infeasible especially in three-dimension [44]. More recent quadrature integration methods have been developed to handle broken integration cells commonly encountered in non-geometrically conforming meshes, such as the moment fitting based methods [48, 49] and smart octree based methods [50]. Those quadrature methods can achieve similar accuracy with low order quadrature points compared to the conventional Gauss integration and handle complex and irregular integration domains. However, it would be natural for meshfree methods to use nodal-based numerical integrations. Several nodal-based integration techniques have been proposed, such as stabilized conforming and non-conforming nodal integrations [51–56] and variationally consistent integration [57], along with various stabilization methods [58–60]. The stabilized conforming nodal integration methods is utilized in this work and is summarized in this section.

Fig. 4 Voronoi tessellation of domain and representative nodal cell
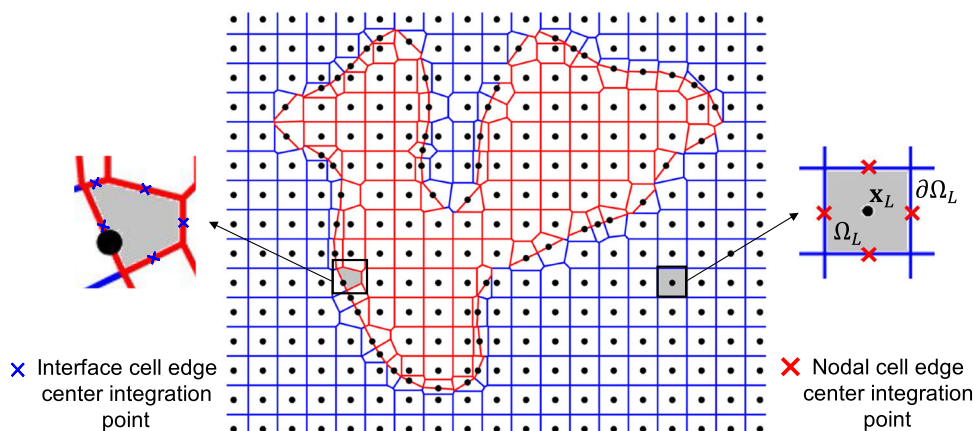
2.3.1 Stabilized conforming nodal integration

One solution that significantly eases the computational cost of Gauss domain integration is to use the discretized nodes as integration points [58], referred to as the direct nodal integration. The direct nodal integration technique is appealing because of its simplicity and efficiency, as it does not require a background integration mesh, which makes numerical approximation truly "mesh-free." Nevertheless, since it is similar to a one-point quadrature rule, the under-integration of the weak form results in improper zero energy modes in most situations [51, 58].

Chen et al. [51] introduced the Stabilized Conforming Nodal Integration (SCNI) method as an enhancement of the direct nodal integration by fulfilling the consistency conditions between the approximations and the numerical integration of the weak form known as the integration constraints. The SCNI method is formulated to exactly meet the first-order integration constraint and also to remedy the rank deficiency in the direct nodal integration method by introducing the following smoothed gradient in the Galerkin approximation:

$$
\begin{aligned}
\tilde{\nabla}\Psi_I(\boldsymbol{x}_L) &= \frac{1}{W_L}\int_{\Omega_L}\nabla\Psi_I(\boldsymbol{x})\,\mathrm{d}\Omega \\
&= \frac{1}{W_L}\int_{\partial\Omega_L}\Psi_I(\boldsymbol{x})\,\boldsymbol{n}\,\mathrm{d}\Gamma, \quad W_L = \int_{\Omega_L} d\Omega
\end{aligned}
\tag{14}
$$

where Ω_L denotes the nodal representative conforming smoothing cells, and $\boldsymbol{n}$ represents the unit outward normal of the smoothing cell boundaries. A convenient way of generating conforming smoothing cells is to create the Voronoi diagram according to the domain boundaries and nodal coordinates, as illustrated in Fig. 4, where the boundary integral in the smoothed gradient in Eq. (14) is carried out by the cell boundary quadrature points.

The associated gradient matrix $\widetilde{\boldsymbol{B}}(\boldsymbol{x}_L)$ of RK approximation evaluated at nodal integration point $\boldsymbol{x}_L$ is now expressed

in terms of smoothed gradient as:

$$
\tilde{\boldsymbol{B}}_I(\boldsymbol{x}_L) = \begin{bmatrix} \tilde{b}_{I1}(\boldsymbol{x}_L) & 0 \\ 0 & \tilde{b}_{I2}(\boldsymbol{x}_L) \\ \tilde{b}_{I2}(\boldsymbol{x}_L) & \tilde{b}_{I1}(\boldsymbol{x}_L) \end{bmatrix}
$$

$$
\tilde{b}_{Ii}(\boldsymbol{x}_L) = \frac{1}{W_L}\int_{\Gamma_L}\Psi_I(\boldsymbol{x})n_i(\boldsymbol{x})d\Gamma
\tag{15}
$$

The stiffness matrix is then integrated by nodal integration with the smoothed gradient as:

$$
\boldsymbol{K} = \sum_{L=1}^{NP}\widetilde{\boldsymbol{B}}^T(\boldsymbol{x}_L)\boldsymbol{C}(\boldsymbol{x}_L)\widetilde{\boldsymbol{B}}(\boldsymbol{x}_L)W_L
\tag{16}
$$

It is noted that Voronoi cells conformed to the material interfaces without confining to the existing pixel points can be constructed, as demonstrated in Fig. 4. In such construction, the centers of those Voronoi cells can be viewed as the integration points that are not coincided with the image pixel points. A naturally stabilized nodal integration [56] can be added to SCNI for additional stabilization. Details of constructing those Voronoi cells near the material interface are given in Appendix A.

3 Support vector machine (SVM) classification of micro-CT images and model discretization

3.1 Support vector machine (SVM) classification algorithm

The Support Vector Machine (SVM) is a class of supervised machine learning algorithms that assigns labels to objects through training [61]. Let a labeled classification dataset containing l sets of data be denote as $\mathbf{D} = \{(\boldsymbol{x}_i,\ y_i)\}_{i=1}^l$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the d-dimensional data points, and y_i is the label

corresponding to the *i*th data point. Since the primary focus of this work is bi-material classification, y_i is assumed to be either -1 or $+1$, representing the negative (matrix) and positive (inclusion) classes, respectively. If the given dataset is perfectly linearly separable, the SVM classification process can be described as to find a separating hyperplane in the form of a linear discriminant function in d-dimension:

$$h(\boldsymbol{x}; \{\boldsymbol{w}, b\}) = \boldsymbol{w}^T \boldsymbol{x} + b \tag{17}$$

where $\boldsymbol{w}$ denotes a d-dimensional *weight vector* and b denotes a scalar *bias*. Additionally, $h(\boldsymbol{x}; \{\boldsymbol{w}, b\})$ serves as a linear classifier for class prediction following the decision rule:

$$y = \begin{cases} +1 & if \ h(\boldsymbol{x}; \{\boldsymbol{w}, b\}) > 0 \\ -1 & if \ h(\boldsymbol{x}; \{\boldsymbol{w}, b\}) < 0 \end{cases} \tag{18}$$

Therefore, the weight vector $\boldsymbol{w}$ is orthogonal to the defined hyperplane, and for each $\boldsymbol{x}_i \in \mathbf{D}$, the relative distance in terms of $\boldsymbol{w}$ to the defined hyperplane can be expressed as:

$$\xi_i = \frac{y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|} \tag{19}$$

The *margin* of the linear classifier is identified by selecting a collection of the data points that achieve a minimum distance to $h(\boldsymbol{x}; \{\boldsymbol{w}, b\})$, which are called the *support vectors* $\{\boldsymbol{x}_i^{SV}\}_{i=1}^{N_{SV}}$, and are defined as:

$$\{\boldsymbol{x}_i^{SV}\}_{i=1}^{N_{SV}} = \underset{\boldsymbol{x}_i \in \boldsymbol{D}}{argmin} \left\{ \frac{y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b)}{\|\boldsymbol{w}\|} \right\} \tag{20}$$

Note that if the distances of all support vectors to the hyperplane are normalized to be 1, the margin can be defined as $\xi^* = \frac{1}{\|\boldsymbol{w}\|}$. Thus, the goal of training the SVM with a linear classifier as Eq. (17) can be described as to find the optimal hyperplane $h^*(\boldsymbol{x})$ as follows:

$$h^*(\boldsymbol{x}) = h(\boldsymbol{x}; \{\boldsymbol{w}^*, b^*\}),$$
$$\boldsymbol{w}^*, b^* = \underset{\boldsymbol{w}, b}{argmax} \left\{ \frac{1}{\|\boldsymbol{w}\|} \right\},$$
$$subject \ to: \ y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \quad \forall \boldsymbol{x}_i \in \mathbf{D} \tag{21}$$

The constrained optimization problem described in Eq. (21) can be formulated as an equivalent convex constrained minimization problem:

$$\underset{\boldsymbol{w}, b}{\min} \ J(\boldsymbol{w}) = \frac{1}{2} \|\boldsymbol{w}\|^2,$$
$$subject \ to: \ y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) \geq 1, \quad \forall \boldsymbol{x}_i \in \mathbf{D} \tag{22}$$

which is called the *primal* formulation of SVM with linear classifier [18]. Instead of directly solving the primal convex minimization problem, it is computationally more efficient to solve the dual problem, formulated using the *Lagrange multipliers*. To construct the dual problem, a Lagrange multiplier λ_i is introduced for each linear constraint based on the Karush–Kuhn–Tucker (KKT) conditions [62]:

$$\lambda_i \left(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 \right) = 0, \ \lambda_i \geq 0 \tag{23}$$

Then the objective of the dual problem can be formulated as:

$$\underset{\boldsymbol{w}, b}{\min} L = \frac{1}{2} \|\boldsymbol{w}\|^2 - \sum_{i=1}^{l} \lambda_i \left(y_i(\boldsymbol{w}^T \boldsymbol{x}_i + b) - 1 \right) \tag{24}$$

By finding the stationary point of the Lagrangian L with respect to $\boldsymbol{w}$, the optimal weight vector $\boldsymbol{w}$ can be expressed in terms of a linear combination of the data points, data label, and Lagrange multipliers:

$$\boldsymbol{w} = \sum_{i=1}^{l} \lambda_i y_i \boldsymbol{x}_i \tag{25}$$

Additionally, a new constrain arises when minimizing L with respect to the bias, which indicates that the sum of the labeled Lagrange multipliers must be equal to zero. Therefore, by substituting Eq. (25) into Eq. (24), the dual problem's training objective can be formulated as:

$$\underset{\boldsymbol{\lambda}}{\max} \ L_{dual} = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j,$$

$$subject \ to: \lambda_i \geq 0, \sum_{i=1}^{l} \lambda_i y_i = 0, \quad for \ i = 1, 2, 3, \dots, l \tag{26}$$

where Eq. (26) forms a well-known convex quadratic programming problem (QPP).

Nevertheless, it is possible that no such hyperplane can be found through Eq. (26) as the real-world data sets are rarely perfectly separable. To deal with cases with overlapping classes, a non-negative slack variable $\nabla \epsilon_i$ is introduced to each data point $\boldsymbol{x}_i \in \mathbf{D}$, such that the linear constraint in Eqs. (21) and (22) is modified as:

$$y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) \geq 1 - \nabla \epsilon_i,$$
$$\forall \boldsymbol{x}_i \in \mathbf{D}, \nabla \epsilon_i \geq 0, \forall i = 1, 2, 3, \dots, l \tag{27}$$

It is worth noting that the magnitude of $\nabla \epsilon_i$ affects the correctness of the classification of its corresponding data point $\boldsymbol{x}_i$: when $\nabla \epsilon_i \geq 1$, the data point will be misclassified as

it appears on the wrong side of the hyperplane. As a result, for non-separable data sets, SVM introduces a "soft margin" concept into the training process, and the new training objective function can be described as:

$$\min_{\boldsymbol{w}, b, \{\nabla \epsilon_i\}} \left\{ \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{l} \nabla \epsilon_i \right\},$$

$$\text{subject to: } y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) \geq 1 - \nabla \epsilon_i, \quad \forall \boldsymbol{x}_i \in \mathbf{D},$$

$$\nabla \epsilon_i \geq 0, \quad \forall i = 1, 2, 3, \ldots, l \tag{28}$$

where C is a weight parameter that penalizes the cost of misclassification and $\sum_{i=1}^{l} \nabla \epsilon_i$ gives the loss due to the deviation from the separable cases with the introduction of slack variables. Moreover, the penalty weight parameter C controls the trade-off between maximizing the hyperplane's margin and minimizing the misclassification's loss. Therefore, the selection of C depends on the nature of problems and datasets at hand. Figure 5 presents the hard- and soft-margined linear SVM classifiers trained on a binary dataset. For the hard-margined SVM classifier, the support vectors are located exactly on the maximum lower and upper margins, while the soft-margined SVM classifier relaxes the linear separability constraints, which allows some support vectors to cross over the decision boundary. Note that although the dataset in Fig. 5 is linearly separable, if the linear separability is strictly enforced, as for the hard-margined case, the resulting margin is significantly smaller than the one for the soft-margined case.

By introducing Lagrange multipliers λ_i and β_i, corresponding to each of the constraints in Eq. (28), a Lagrangian of Eq. (28) can be formulated as:

$$L = \frac{1}{2} \|\boldsymbol{w}\|^2 + C \sum_{i=1}^{l} \nabla \epsilon_i$$

$$- \sum_{i=1}^{l} \lambda_i \left(y_i \left(\boldsymbol{w}^T \boldsymbol{x}_i + b \right) - 1 + \nabla \epsilon_i \right) - \sum_{i=1}^{l} \beta_i \nabla \epsilon_i \tag{29}$$

By minimizing the L with respect to $\boldsymbol{w}$, b, and $\nabla \epsilon_i$, respectively, one additional relationship connecting the penalty weight parameter C to λ_i and β_i is obtained, in addition to the ones acquired in the linearly separable cases:

$$C = \lambda_i + \beta_i \tag{30}$$

Therefore, the dual objective can be described as:

$$\max_{\boldsymbol{\lambda}} L_{dual} = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j,$$

$$\text{subject to: } 0 \leq \lambda_i \leq C, \sum_{i=1}^{l} \lambda_i y_i = 0, \quad \text{for } i = 1, 2, 3, \ldots, l \tag{31}$$

Note that the objective function achieved for the inseparable cases is the same as the one obtained from the linearly separable cases in Eq. (26), except one additional constraint on the Lagrange multiplier λ_i.

For complicated data sets, the linear classifier is often found inadequate. One strategy is to introduce non-linear transformation function ϕ, which maps the data points $\boldsymbol{x}$ to a higher dimension so that the projected data points $\phi(\boldsymbol{x})$ are approximately linearly separable in the higher dimensional feature space. However, upscaling the dimensionality usually leads to high and impractical computational costs. Since the Lagrange dual formulation in Eq. (31) only depends on the dot product between two vectors in the feature space, the SVM can utilize the "kernel trick" to include high-degree polynomial features. The idea of the kernel trick is to represent l data point $\boldsymbol{x}$ by a l by l kernel matrix $\boldsymbol{K}$ that contains elements $k_{i,j} = \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$, which performs pairwise similarity comparisons between the original low dimensional data points without an explicit definition of the transformation function ϕ for mapping data to high dimensions. More detailed introduction to the requirement and existence of kernel function can be found in [19]. Therefore, the dual formulation of the training objective for non-linear SVM can be described by replacing $\boldsymbol{x}_i^T \boldsymbol{x}_j$ in Eq. (31) by a kernel function $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$:

$$\max_{\boldsymbol{\lambda}} L_{dual} = \sum_{i=1}^{l} \lambda_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \lambda_i \lambda_j y_i y_j \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j),$$

$$\text{subject to: } 0 \leq \lambda_i \leq C, \sum_{i=1}^{l} \lambda_i y_i = 0, \quad \text{for } i = 1, 2, 3, \ldots, l \tag{32}$$

Note that Eq. (32) can be viewed as a generalized Lagrange dual formulation of SVM since for linear cases, the kernel function can be expressed as the data point dot product as:

$$\text{Linear: } \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \boldsymbol{x}_i^T \boldsymbol{x}_j \tag{33}$$

Other widely used kernel functions are polynomial and Gaussian radial basis kernel functions, which are illustrated in Eq. (34) and (35), respectively.

$$\text{Polynomial: } \boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(1 + \boldsymbol{x}_i^T \boldsymbol{x}_j\right)^q, q = 2, 3, \ldots, \tag{34}$$
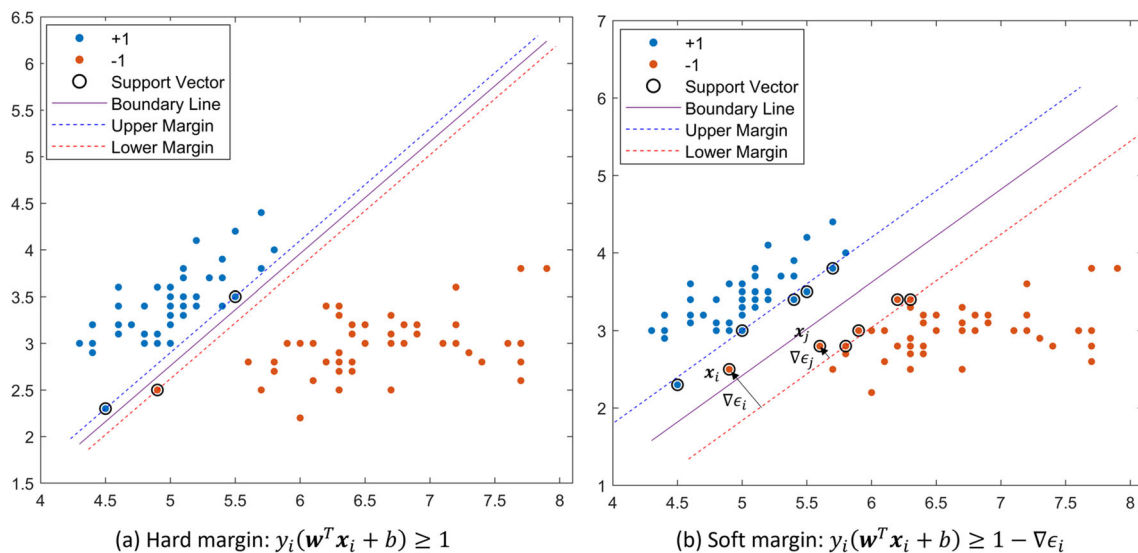
Fig. 5 Hard- and soft-margined SVM linear classifiers

(a) Hard margin: $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1$

(b) Soft margin: $y_i(\boldsymbol{w}^T\boldsymbol{x}_i + b) \geq 1 - \nabla\epsilon_i$

Gaussian radial basis : $\boldsymbol{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}, \gamma > 0$

$$(35)$$

Figure 6 demonstrates the transformation of the 2-dimensional training data to 3-dimension using a Gaussian kernel. It is clear that the 3-dimensional data points become linearly separable by a 2-dimensional hyperplane, and the resulting separating hyperplane can be projected back to the 2-dimensional space, which becomes a nonlinear decision boundary.

Moreover, the SVM produces a classification score by predicting new datasets that provide information about the material class and reveal the location of material interfaces. The classification score is a signed distance measure for an observation point $\boldsymbol{x}$ to its nearest decision boundary, with a score of zero denoting $\boldsymbol{x}$ is precisely on the decision boundary. Therefore, the classification score acts as a guide in identifying material boundaries in the image, facilitating more accurate numerical model generation. The classification score for predictions at $\boldsymbol{x}$ to the positive class is defined as:

$$S(\boldsymbol{x}) = \sum_{j=1}^{N_{SV}} \lambda_j y_j \boldsymbol{K}(\boldsymbol{x}_j, \boldsymbol{x}) + b \tag{36}$$

where N_{SV} is the total number of support vectors and $(\lambda_1, \lambda_2, \ldots, \lambda_{N_{SV}}, b)$ are the trained SVM parameters.

In summary, the SVM algorithms have shown promising performances in many application fields [63]. During the hyperplane selection process, SVMs utilize different kernel functions to transform the low-dimensional, non-linear, and possibly non-separable training data to higher-dimensional feature spaces, which allows the data to be linearly separated.

In addition, the selected high-dimensional hyperplane can be projected back down to the original space where the training data belong, providing non-linear decision boundaries between the separated classes [64]. As a result, SVMs not only aid in classifying different material pixels from micro-CT images of heterogeneous materials but also inherently identify material interfaces. Here, we use SVMs to guide numerical model discretization for the proposed Interface-modified Reproducing Kernel Particle Method, which will be introduced in the later sections.

3.2 From micro-CT images to numerical models

In this work, the sample images are taken from micro-CT, which is an imaging technique that generates three-dimensional images of an object's microstructure with (sub)micron resolution using an X-ray tube with cone-beam geometry as a source and a rotating sample holder [65]. The overall processes of generating Support Vector Machine Guided Reproducing Kernel (SVM-RK) numerical model from micro-CT images are summarized in Fig. 7.

3.2.1 Training data preparation and training the SVM

The training data points are located at the centroid of each pixel cell in the sample image, and the physical coordinates of those data points are assigned as the training data for the SVM. To supervise the SVM's training, response labels y, are created by segmenting the sample image using Otsu's method [66]. Otsu's method selects a global threshold that maximizes inter-class intensity variance from the zeroth- and the first-order cumulative moments of the sample image's
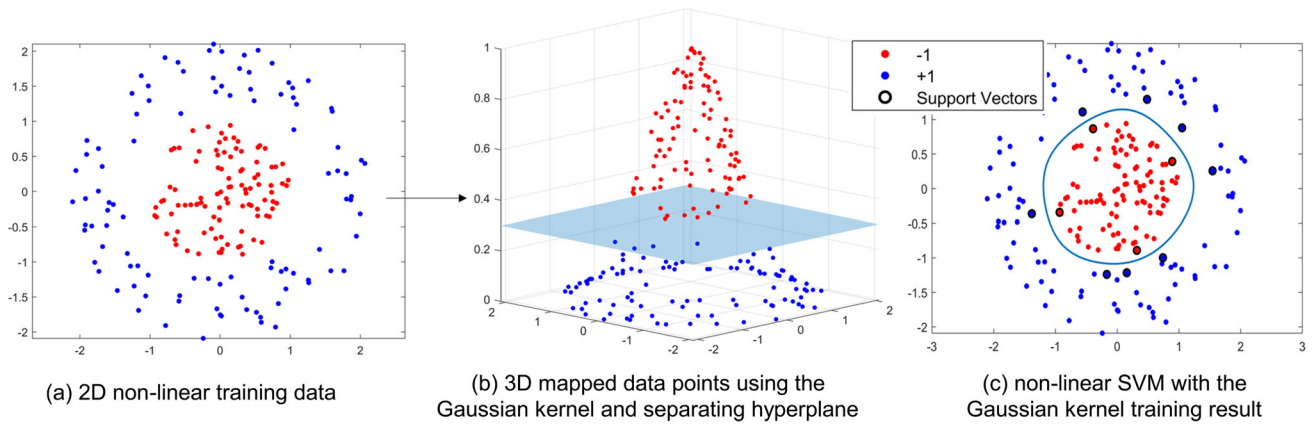
(a) 2D non-linear training data

(b) 3D mapped data points using the Gaussian kernel and separating hyperplane

(c) non-linear SVM with the Gaussian kernel training result

Fig. 6 An example of training the non-linear SVM with the Gaussian kernel (The circled data points are the support vectors)

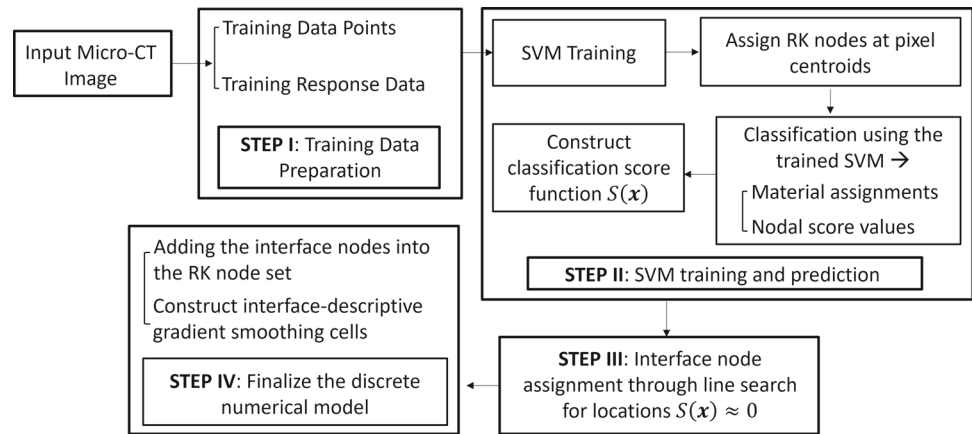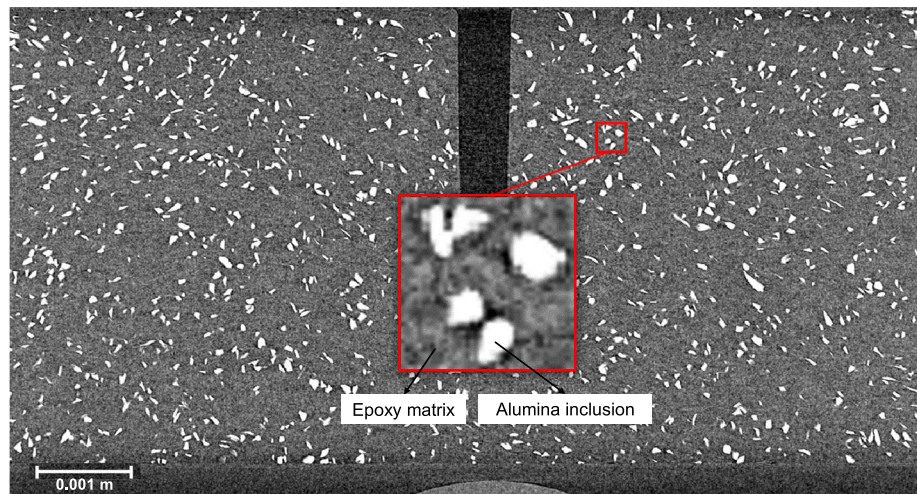**Fig. 7** Flowchart of SVM-RK numerical model generation

Fig. 8 Sample alumina-epoxy image

intensity-level histogram. Figure 8 illustrates an alumina-epoxy composite micro-CT image slice, where the white areas in the sample image indicate the alumina inclusion material and the grey areas represent the epoxy material in the matrix. A ROI of 30×30 pixels (area in the red box) containing 4 irregularly shaped alumina particles is selected to demonstrate the SVM training and numerical model generation processes. Note that only the pixel centroid material class assignments and physical coordinates are provided as labeled training data, and the goal of training is to identify material class at arbitrary locations within the image domain for numerical integration purposes.

Table 1 SVM hyperparameters selected for training the sample image

SVM training hyperparameters	Values
Kernel function	Gaussian radial basis function
Kernel scale (γ in Eq. (35))	0.25
Penalty weight parameter (C in Eq. (28))	500

Specific hyperparameters of the training must be determined beforehand to facilitate SVM classification, which are summarized in Table 1. The kernel scale determines the extent to which each data point affects the shape of the decision boundaries, which is selected as 0.25. Furthermore, a penalty weight parameter C of 500 is chosen to ensure that the resulting separation hyperplane resembles the material interface. The Gaussian radial basis function is selected as the kernel function based on the geometries of the inclusions, as they are distinctive small particles. The selections of the kernel scale and penalty weight parameter are optimized utilizing an iterative Bayesian optimization process, and the objective function for the Bayesian optimization process is to minimize the fivefold cross validation classification loss. A total of 30 iterations are performed, and once the iterations are completed, the hyperparameter configuration associated with the smallest validation classification loss is selected as the optimal set of hyperparameters for the SVM model. Additionally, a standardization in which the training data points are normalized to have a zero mean and a standard deviation of unity is performed. The standardization of the training data is critical because SVM training is based on the relative distances between the training data points, and without standardization, larger-scale training data may dominate in distance determination, leading to a biased model.

3.2.2 RK interface nodes

In the Interface-modified RK approximation to be proposed in Sect. 4, a set of interface nodes are included to introduce proper weak discontinuity across material interfaces. In this section, we present an approach to generate interface nodes, utilizing the score $S(x)$ (Eq. (36)) that is produced during the SVM classification and can be interpreted as a scaled signed distance function.

Let $\mathbb{S}^0 \equiv \{x_I\}_{I=1}^{NP_0}$ be the set of training data points located at the image pixel centroids in the image domain $\overline{\Omega}$, and define $\mathbb{S}^+ \equiv \{x \in \mathbb{S}^0 | S(x) \geq 0\}$ and $\mathbb{S}^- \equiv \{x \in \mathbb{S}^0 | S(x) < 0\}$. Consequently, defining a set of interface-searching node pairs $\{x_K^+, x_K^-\}_{K=1}^{NP_{IF}}$ in which $x_K^\pm \in \mathbb{S}^\pm$ is a near-interface master/slave node (see Remark

3.1 for details). The search of an interface node $x_K^* \equiv x_K^+ + d_K^* R_K$ can be defined as follows:

Find $d_K^* \in \mathbb{R}$ such that:

$$S(x_K^+ + d_K^* R_K) = 0, \ \forall K = 1 \cdots NP^* \tag{37}$$

where $R_K = (x_K^- - x_K^+)/\|x_K^- - x_K^+\|$ is the line search direction. The resulting RK node set is then $\mathbb{S}^{RK} \equiv \mathbb{S}^0 \cup \mathbb{S}^{IF}$ with $\mathbb{S}^{IF} \equiv \{x_K^*\}_{K=1}^{NP^*}$, which serves as the SVM-RK discretized model.

Remark 3.1 The interface-searching node pairs $\{x_K^+, x_K^-\}_{K=1}^{NP_{IF}}$ can be determined in various ways. In this work, the following approach is taken: given the set of support vectors $\{x_L^{SV}\}_{L=1}^{N_{SV}}$, define the master candidate node set $\mathbb{S}^{C+} \equiv \{x_I \in \mathbb{S}^+ | \|x_I - x_L^{SV}\| \leq \xi \ell, \ \forall L = 1 \cdots N_{SV}\}$, in which ℓ and ξ denote the image voxel size and a scaling factor, respectively. In this work $\xi = 1.5$ is used. The corresponding nearest slave nodes x_K^- are found such that, for $x_K^+ \in \mathbb{S}^{C+}$,

$$x_K^- = \underset{x_I \in S^{C-}}{argmin} \|x_I - x_K^+\| \tag{38}$$

with the slave candidate node set $\mathbb{S}^{C-} \equiv \{x_I \in \mathbb{S}^- | \|x_I - x_L^{SV}\| \leq \xi \ell, \ \forall L = 1 \cdots N_{SV}\}$. Note that Eq. (38) can result in multiple x_K^- for one x_K^+ and lead to the master–slave pairs. Figure 9a illustrates an example of the master and slave candidate nodes plotted along with the support vectors. The corresponding candidate node pairs are shown in Fig. 9b.

Remark 3.2 The solution of d_K^* in Eq. (37) can be determined iteratively by the Newton–Raphson method. For the $(\nu + 1)^{th}$ iteration, the increment $\Delta d^{\nu+1}$ for $d_K^{*\nu+1} = \Delta d_K^{*\nu+1} + d_K^{*\nu}$ is obtained as follows:

$$\Delta d_K^{*\nu+1} = -S(x_K^{*\nu}) / \left(\frac{\partial S(x_K^{*\nu})}{\partial x_K^{*\nu}} \cdot R_K \right) \tag{39}$$

Remark 3.3. One may consider interpolating the score function with SVM predicted nodal score values without constructing Eq. (36) as follows:

$$\widetilde{S}(x) = \sum_{I=1}^{NP_0} \Psi_I^0(x) s_I \tag{40}$$

where $\Psi_I^0(x)$ and s_I are the RK shape function constructed on $\mathbb{S}^0$ and SVM score value (signed distance) for node x_I, respectively. The RK shape function can serve as a filter for potentially noisy predicted score values to provide smoothing on the zig-zag material interface determined directly from
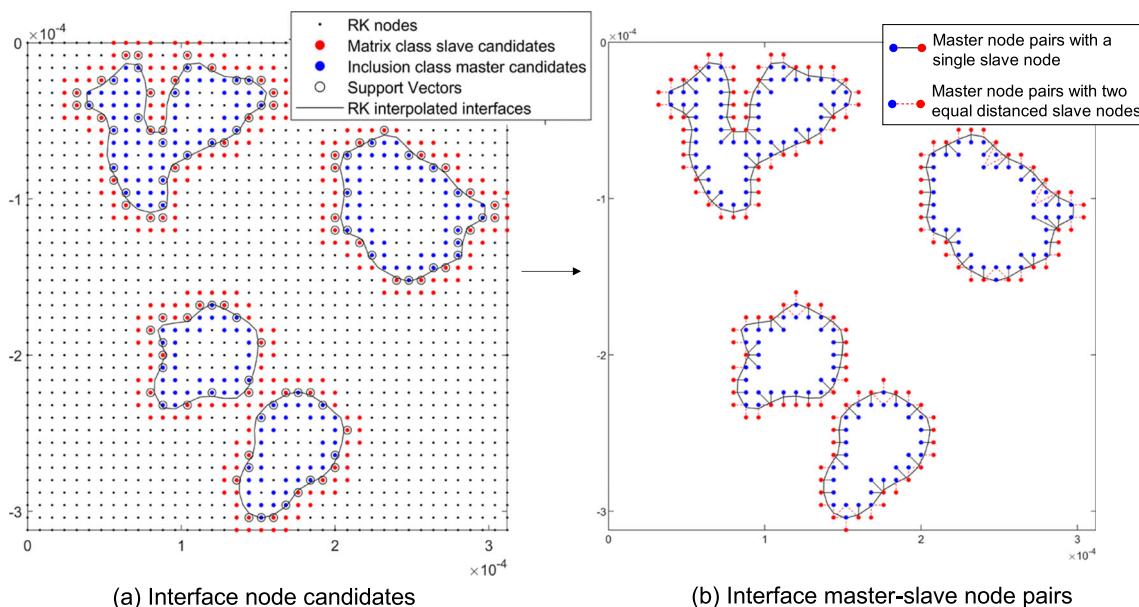
(a) Interface node candidates

(b) Interface master-slave node pairs

Fig. 9 Interface node candidates and resulting master–slave interface node search pairs

image pixels. Equation (40) is used for numerical implementation in the current work.

Remark 3.4. To ensure relative even distribution of nodes around interfaces, a MATLAB built-in function "uniqetol" with a relative tolerance 0.01 is applied to the interface node set $\mathbb{S}^{IF}$. In addition, $x_I \in \mathbb{S}^0$ is removed if $\|x_I - x_K^*\| < \zeta \ell$ for all $x_K^* \in \mathbb{S}^{IF}$, and $\zeta = 1/3$ is selected in this work. An example of rearranged RK nodes is illustrated in Fig. 10. The material interfaces are represented by a simple line connection in Fig. 10 for visualization purposes; the interface-modified RK approximation to be discussed next requires only the interface point locations and the signed distance of each discrete point obtained from SVM. Note that

discretization far from the material interfaces can be made coarser to improve computational efficiency.

3.3 Image-based SVM-RK model validation

3.3.1 Validation with a synthetic image

A synthetic two-phase image containing the known locations of inclusions is generated, as illustrated in Fig. 11. The synthetic image has a dimension of 10 mm × 10 mm and a resolution of 244 × 244 pixels. To account for uncertainties in the imaging process, Gaussian noise is added to the original image, and the manufactured testing image is scaled down to

Table 2 Results of interface node search algorithm with various kernel support sizes

Support size	Number of constructed interface nodes	Mean iteration number	Mean score values	Edge detection mean squared error
1.10	134	3.72	1.63E−12	0.0065
1.50	134	4.61	5.91E−12	0.0065
2.00	134	4.60	4.40E−12	0.0065
2.50	133	4.62	2.99E−12	0.0066
3.00	131	4.97	4.15E−12	0.0067
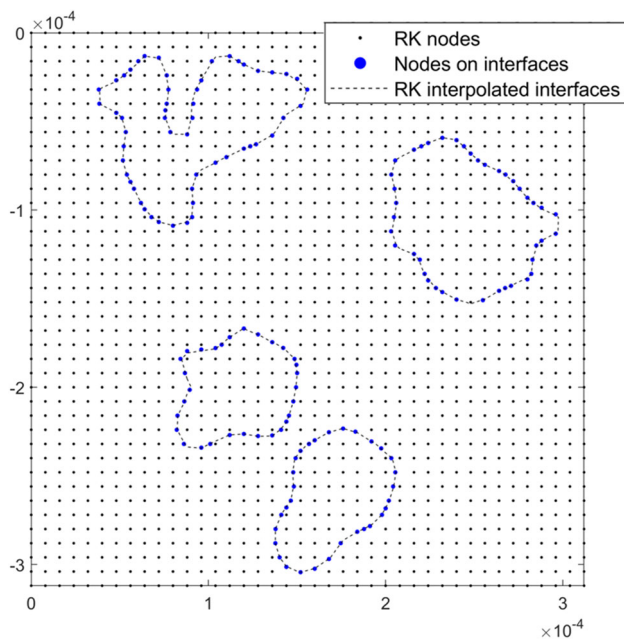
Fig. 10 RK numerical model for the test image
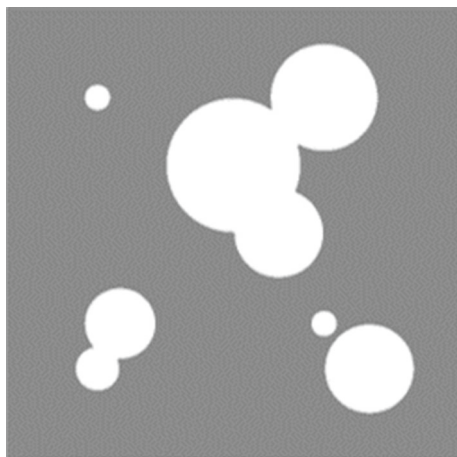
Fig. 11 Synthetic testing image for validating the SVM-RK interface node generation

Fig. 12 Manufactured noisy testing image for validating the SVM-RK interface node generation
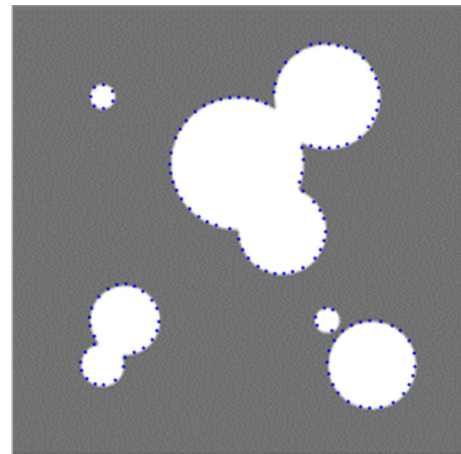
Fig. 13 Interface nodes overlapped with the manufactured test image

100×100 pixels to lower the resolution, especially around the material interfaces. Figure 12 demonstrates the manufactured noisy testing image, which will serve as the input image for the image-based SVM-RK model generation. The accuracy of the obtained interface nodes is determined by a normalized mean square error of the discretized material interfaces as:

$$MSE = \frac{1}{NC \cdot L} \left(\sum_{j=1}^{NC} \sum_{K=1}^{NP^*} \left(\|\boldsymbol{x}_K^* - \boldsymbol{c}_j\| - R_j \right)^2 \right)^{\frac{1}{2}} \qquad (41)$$

where $\boldsymbol{c}_j$ and R_j represent the center coordinates and radius of the inclusion to which the interface node $\boldsymbol{x}_K^* \in \mathbb{S}^{IF}$ belongs, NP^* is the total number of generated interface nodes in the set $\mathbb{S}^{IF}$, and NC and L denote the total number of inclusions in the synthetic image and the x-dimension of the synthetic image, respectively.

As previously discussed, for implementation the score function is interpolated using the RK shape function in Eq. (40), and the locality and smoothness of the RK shape function may differ depending on the size and order of continuity of the kernel function chosen for its construction. Therefore, various RK kernel support sizes and kernel functions with different orders of continuity are employed to study their effects on the accuracy of the SVM-RK interface node generation algorithm. Figure 13 illustrates the obtained interface nodes overlaid with the synthetic image using a cubic B spline RK kernel (B3, C^2 continuity) with a normalized support size of 2 and linear bases. In addition, results of

Fig. 14 Micro-CT input image selection for quantitative RK discretization model validation
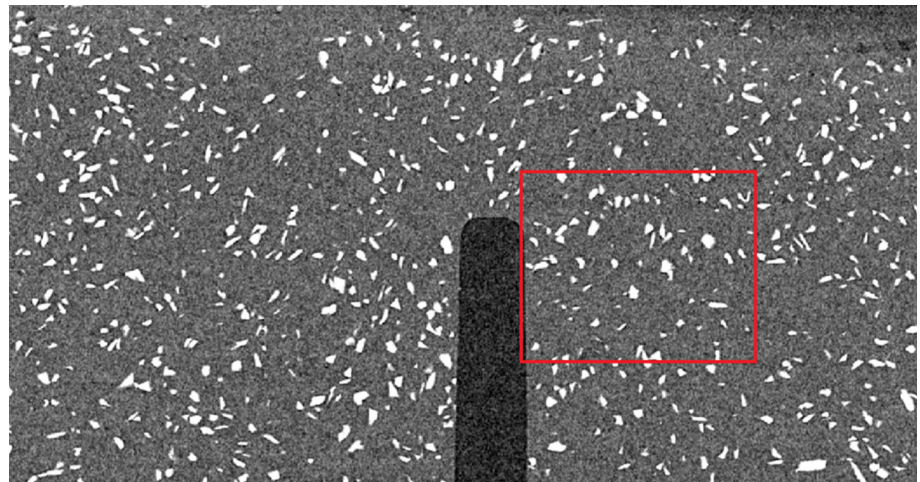
Table 3 Results of interface node search algorithm with various RK kernel functions

Kernel Function	Number of constructed interface nodes	Mean iteration number	Mean score values	Edge detection mean squared error
C^0(Tent)	133	3.62	5.92E−13	0.0065
C^1(B2)	132	4.64	4.78E−12	0.0066
C^2(B3)	134	4.60	4.40E−12	0.0065

the interface node generated with various support sizes and kernel function continuities can be found in Tables 2 and 3, respectively. Upon comparison of results in Tables 2 and 3, it can be observed that the proposed SVM-RK interface node search algorithm converges less than an average of 5 iterations for all instances, and the resulting interface nodes achieve average scores (Eq. (40)) to the order of 10^{-12}. Additionally, the normalized mean square error of the proposed image-based RK discretization model generation process is approximated 0.65% for all scenarios. Moreover, the results show that the generated interface nodes are not sensitive to the choices of the kernel support size and kernel continuity (tent function with C^0 continuity, quadratic B spline function (B2) with C^1 continuity, and B3 with C^2 continuity) used in constructing the score function (Eq. (40)).

3.3.2 Validation of the image-based SVM-RK discrete model with Scanning Electron Microscopy (SEM) images

To analyze the quality of the proposed image-based RK discretization model generation procedure, a comparison is made between the constructed digital surface model from micro-CT and a surface image obtained from Scanning Electron Microscopy (SEM) with a spatial resolution of 1.5 μm for the same specimen as a comparison reference. SEM uses an electron beam to scan the surface of a material, producing a high-resolution image that reveals details such as surface topography, crystalline structure, chemical composition, and
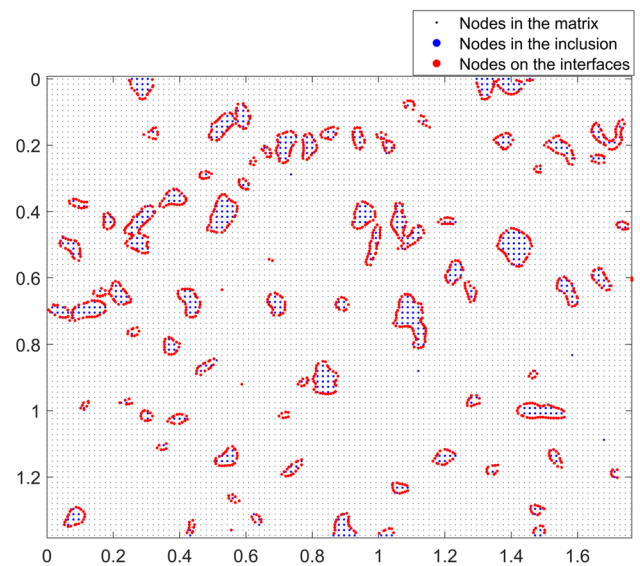
Fig. 15 Constructed RK discretization model for the quantitative validation

electrical behavior of the top 1 μm portion of a specimen [67]. The inclusion materials in SEM are identified based on the Energy Dispersive X-ray Spectroscopy (EDS), a chemical analysis technique that detects X-rays emitted by the material in response to the electron beam to form an elemental mapping of the SEM-scanned specimen surface [68]. The micro-CT input image for constructing the numerical model
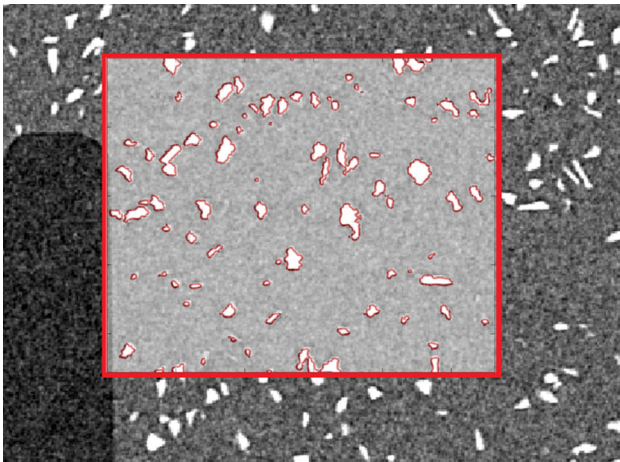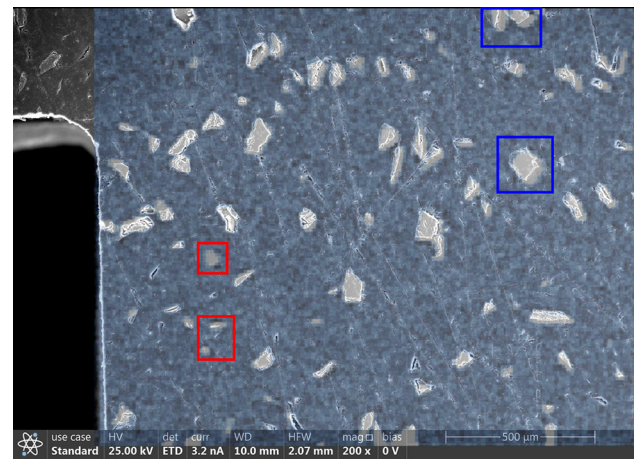
Fig. 16 Constructed SVM-RK discretization model superimposed with the micro-CT input image (only showing the interface nodes)

Fig. 18 Comparison between the SEM surface image and micro-CT image

is selected accordingly near the surface of the same specimen, and a ROI around 2.82 mm by 2.37 mm is chosen to match the SEM scanned area, which is highlighted in the red box in Fig. 14.
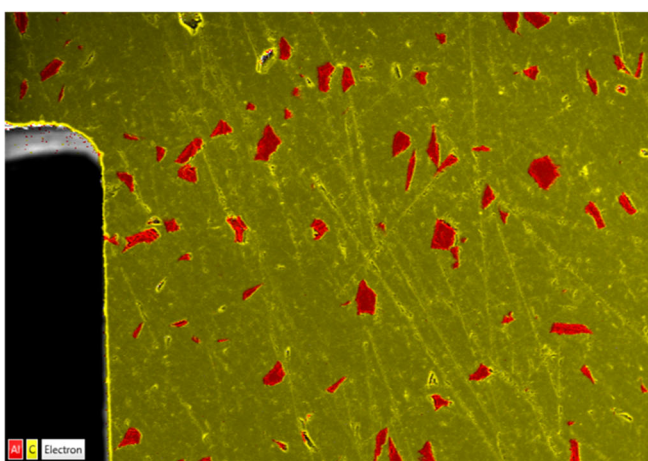
An SVM-RK discretization model is created from the input 2D slice of the micro-CT image using the proposed method, as illustrated in Fig. 15, containing discretized nodes in the epoxy matrix, alumina inclusions, and on the identified material interfaces. The constructed SVM-RK discretization model is superimposed over the original micro-CT image, and the result is shown in Fig. 16. Furthermore, the alumina inclusions enclosed by the identified interface nodes in the constructed SVM-RK discretization model are highlighted and overlaid onto the SEM surface image, which is contrasted with the EDS-layered SEM image shown in Fig. 17. As can

be seen, the obtained image-based RK discretization model agrees well with the input micro-CT image in detail.

Remark 3.5 It is worth noting that the surface of the specimen was polished to enhance imaging quality for SEM, which may cause slight alterations to the distribution of surface particles. Figure 18 illustrates the minor discrepancies between the SEM and micro-CT images. As the blue boxes indicate, misalignments can be observed for certain inclusion particles. Additionally, smaller particles were not captured in the SEM scan, as highlighted in the red boxes. Consequently, the inclusion particles identified by the SVM-RK discretization model exhibit slight variances compared to the EDS elemental mapping result, as illustrated in Fig. 17. Nevertheless, they are mostly consistent, particularly for the larger and more distinctive inclusion particles.

(a) SEM image overlaid with EDS mapping
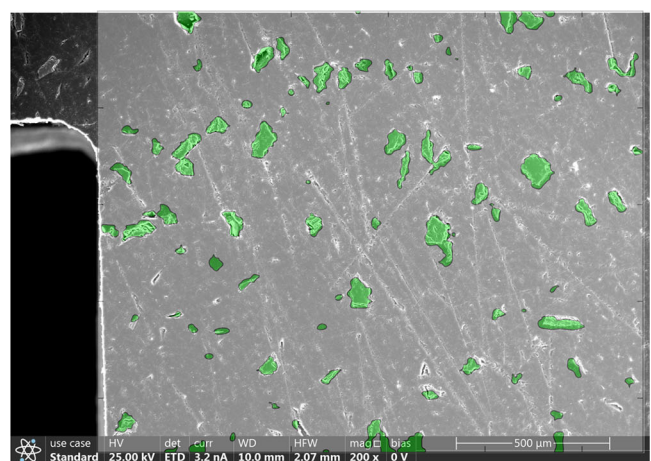
(b) SEM image overlaid with SVM-RK discretization model

Fig. 17 Comparison of EDS overlaid and SVM-RK discretization model overlaid SEM surface images

(a) All cubic B spline kernels

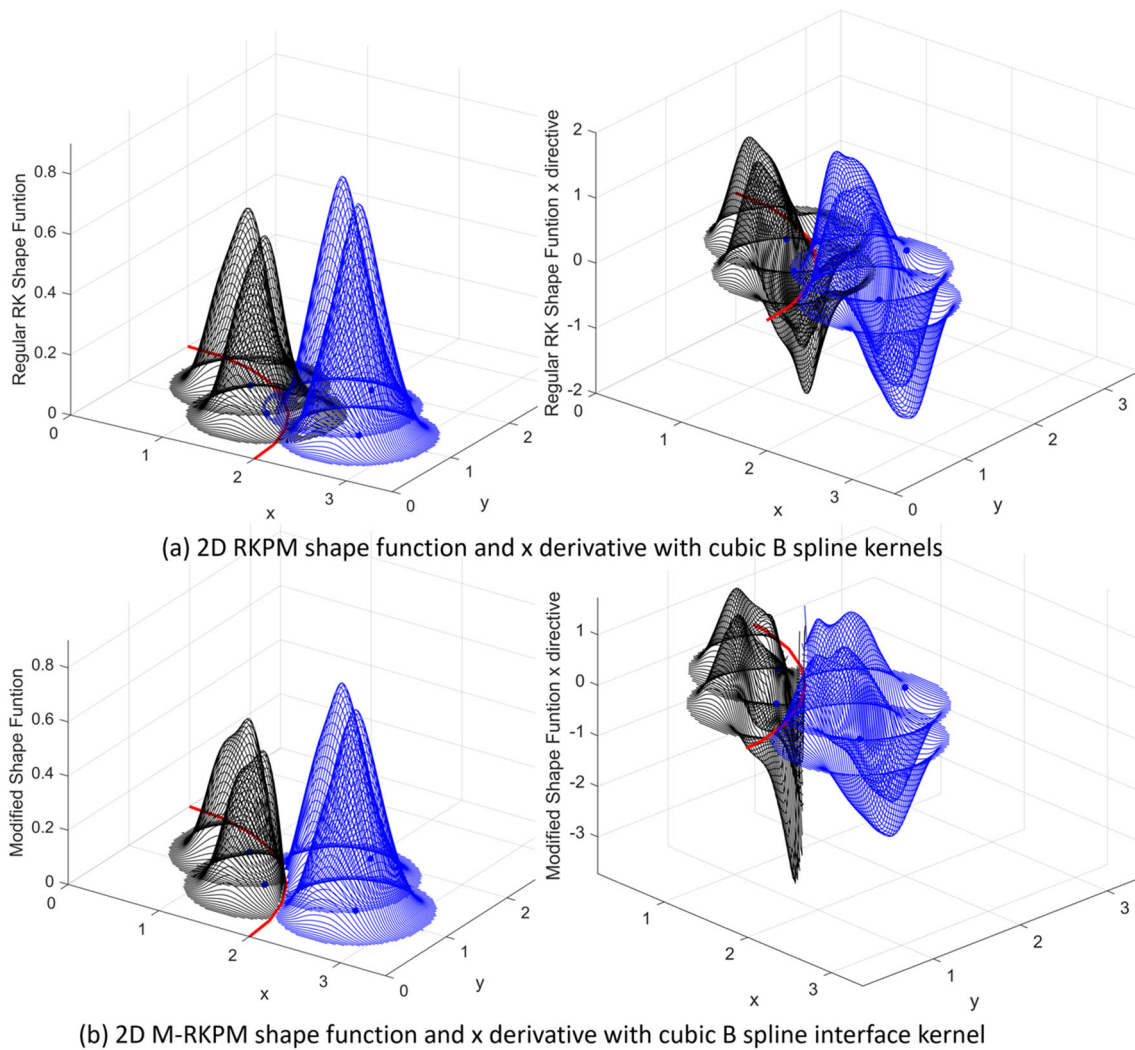(b) Order 4 power function interface kernel, cubic B spline kernels for the rest
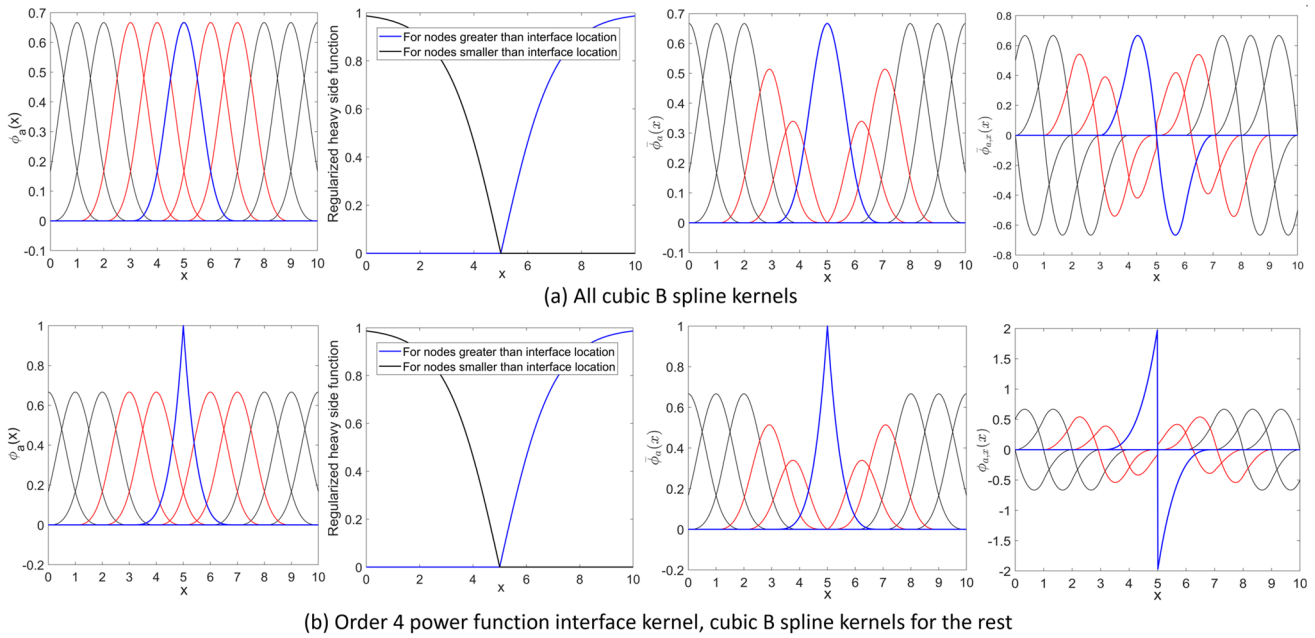
Fig. 19 Plots of 1D interface-modified kernels with different interface kernels (Left to right: original kernel function, regularized Heaviside scaling function, modified kernel function, modified kernel function's derivative)

Fig. 20 1D traditional RK and IM-RK shape functions and derivatives

(a) RKPM shape function and derivative with cubic B spline kernels

(b) IM-RKPM shape function and derivative with cubic B spline interface kernel

(a) 2D RKPM shape function and x derivative with cubic B spline kernels

(b) 2D M-RKPM shape function and x derivative with cubic B spline interface kernel
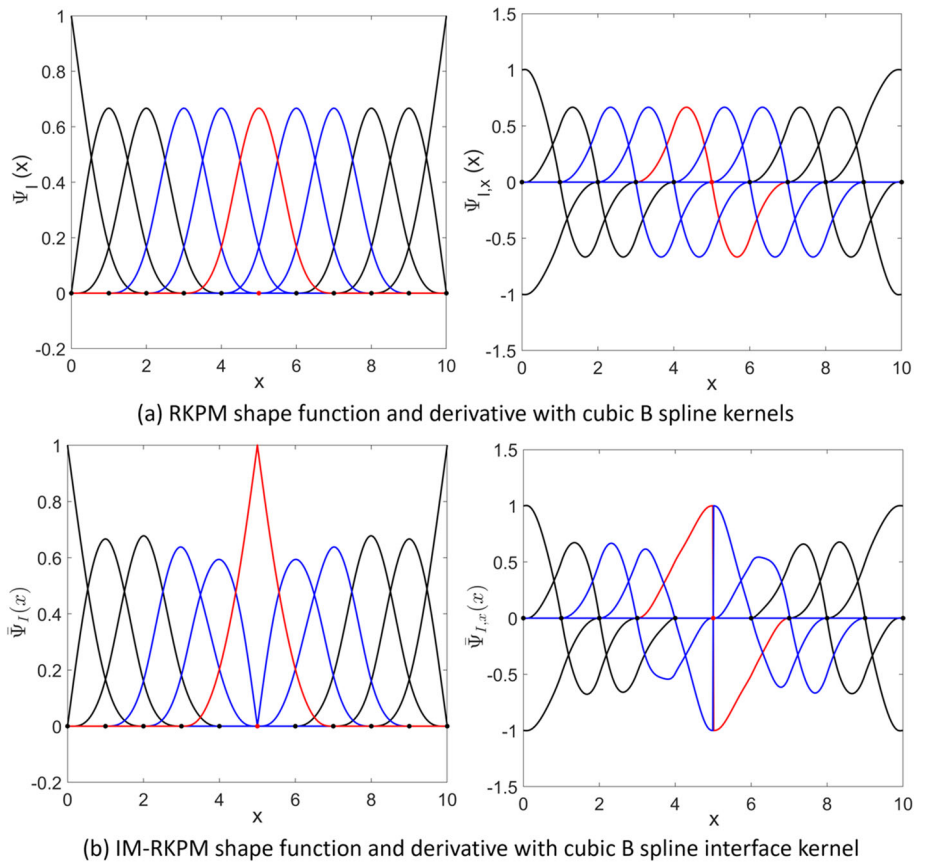
Fig. 21 Comparisons between 2D traditional RKPM and IM-RKPM shape functions and their derivatives (the blue and the black functions represent shape functions on the opposite sides of the material interface)

4 Interface-modified reproducing kernel approximation guided by support vector machine

4.1 Interface-modified kernel functions

With the material interface segmented by the SVM, the weak discontinuities across the material interfaces are to be introduced by modifying the regular RK kernel function with a regularized Heaviside function $\widetilde{H}$ as follows:

$$\overline{\phi}_a(\boldsymbol{x} - \boldsymbol{x}_I) = \phi_a(\boldsymbol{x} - \boldsymbol{x}_I)\widetilde{H}\left(\overline{\xi}_I(\boldsymbol{x})\right) \tag{42}$$

where $\overline{\phi}_a(\boldsymbol{x} - \boldsymbol{x}_I)$ is a modified kernel function, and $\widetilde{H}(\cdot)$ and $\overline{\xi}_I(\boldsymbol{x})$ in Eq. (42) are defined as:

$$\widetilde{H}(\cdot) = \max(0, \tanh(\cdot)) \tag{43}$$

and

$$\overline{\xi}_I(\boldsymbol{x}) = \begin{cases} -\frac{S(\boldsymbol{x})}{c}, & S(\boldsymbol{x}_I) < 0 \\ +\frac{S(\boldsymbol{x})}{c}, & S(\boldsymbol{x}_I) > 0 \end{cases} \tag{44}$$

where $S(\boldsymbol{x})$ is the score function, and c denotes a scaling factor that has a length of the order of nodal spacing. Note that $S(\boldsymbol{x})$ is a signed distance of an evaluation point to its nearby interface, which is given from the output of the SVM-RK image segmentation and is readily available for evaluation of regularized Heaviside function $\widetilde{H}$. This normalized distance measure $\overline{\xi}(\boldsymbol{x})$ is applicable to general n-dimensional image data. The kernel functions associated with nodes away from the interfaces have been scaled to zero at the material interfaces by the regularized Heaviside function, and the kernel functions associated with the interface nodes are not scaled. As will be discussed in the next section, the "reproduced" RK

(a) RK node distribution (red arc indicates the interface)

(b) IM-RK function interpolation with B3 kernel: interpolated function (left) and x-derivative (right)

(c) Standard RK function interpolation with B3 kernel: interpolated function (left) and x-directional derivative (right)
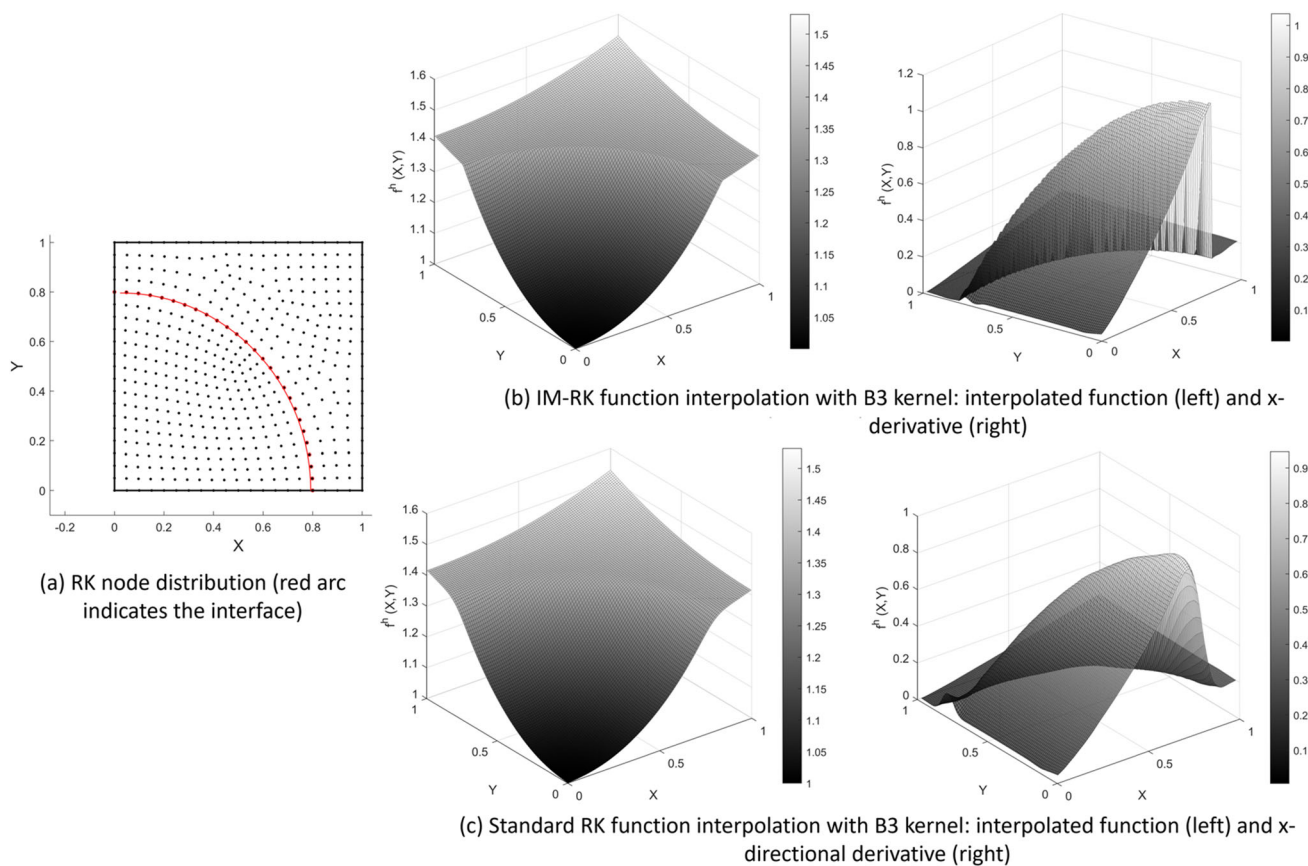
Fig. 22 IM-RK interpolation of a 2D function

shape functions via the reproducing conditions given in Eqs. (9)–(10) reveals a weak (C^0) continuity of the approximated function at the material interface due to the Heaviside scaling in Eq. (42) regardless of the continuity of kernel function associated with nodes at the material interfaces. Hence, cubic B spline (B3) kernel functions with C^2 continuity or power kernel (PK) function with C^0 continuity [69] can be considered for the kernel function associated with the interface nodes:

$$\phi_a^{B3}(z) = \begin{cases} \frac{2}{3} - 4|z|^2 + 4|z|^3 & for\ 0 \leq |z| \leq \frac{1}{2} \\ \frac{4}{3} - 4|z| + 4|z|^2 - \frac{4}{3}|z|^3 & for\ \frac{1}{2} \leq |z| \leq 1 \\ 0 & otherwise \end{cases} \quad z = \frac{x - x_I}{a}$$

(45)

$$\phi_a^{PK}(z) = \begin{cases} (1-z)^\alpha & for\ 0 \leq z \leq 1 \\ 0 & otherwise \end{cases} \quad z = \frac{x - x_I}{a}$$

(46)

Figure 19 shows the un-modified kernel functions $\phi_a(x - x_I)$, regularized Heaviside function $\widetilde{H}(\overline{\xi}_I(x))$, the interface-modified kernel functions $\overline{\phi}_a(x - x_I)$ and their derivatives $\overline{\phi}_{a,x}(x - x_I)$ in 1D with different choices of interface kernels. The blue, red, and black kernel functions

are associated with the interface node, nodes within the support of interface nodes, and nodes away from the interface, respectively.

Remarks 4.1. 1. One can observe that after the interface modification, the influence domains of all nodes, except the interface node, terminate at the interface location, which naturally introduces a weak discontinuity to interface-modified kernel functions, even for the case of the smooth B-spline kernel at the interface.

2. The added computational cost to perform the proposed kernel modifications is marginal because only a scaling is applied to the original kernel functions to construct kernel modifications, and this can be done effortlessly for arbitrary spatial dimensions.

4.2 Interface-modified RK (IM-RK) approximation

Let us consider the RK discretization node set $\mathbb{S}^{RK}$ in the SVM-RK discretized numerical model. Recall that interface nodes in $\mathbb{S}^{RK}$ are contained in the set $\mathbb{S}^{IF}$. Let $\mathbb{S}^{RK} \backslash \mathbb{S}^{IF}$ denotes the set of all RK discrete points excluding those on the interfaces, then the RK shape function can be written as
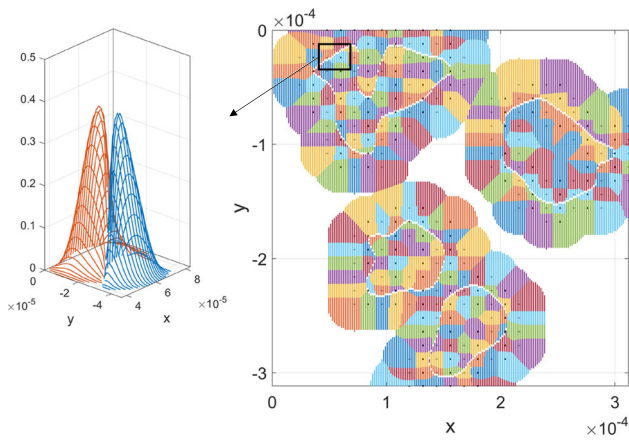
Fig. 23 IM-RK shape functions for nodes around the interfaces: top view (right) and the zoom-in plot of two shape functions in the black box (left)
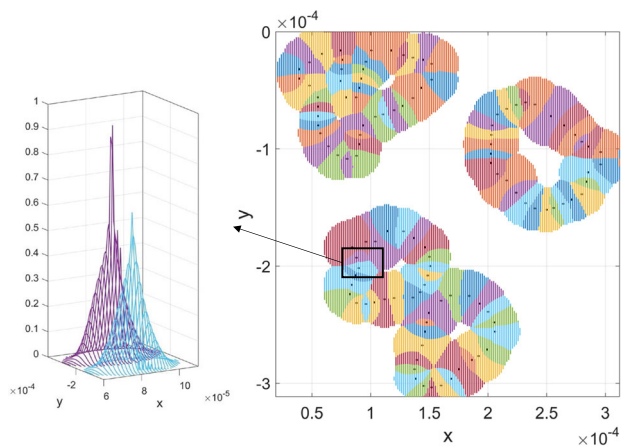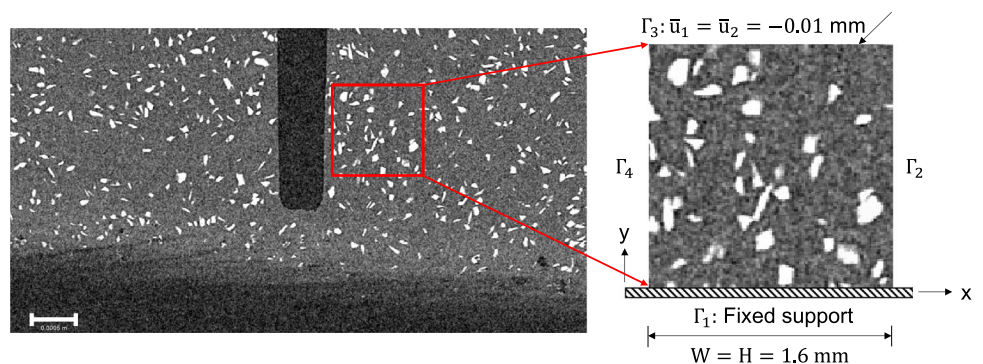

Fig. 24 IM-RK shape function for interface nodes: top view (right) and the zoom-in plot of two shape functions in the black box (left)

follows:

$$\begin{cases} \Psi_I(x) = C(x; x - x_I)\phi_a(x - x_I) \\ \quad = (H^T(x - x_I)b(x))\phi_a(x - x_I), \forall I \in \mathbb{S}^{IF} \\ \Psi_I(x) = C(x; x - x_I)\overline{\phi}_a(x - x_I) \\ \quad = (H^T(x - x_I)b(x))\overline{\phi}_a(x - x_I), \forall I \in \mathbb{S}^{RK}\backslash\mathbb{S}^{IF} \end{cases}$$

$$(47)$$

where $\phi_a(x - x_I)$ is the regular kernel functions without interface modification and $\overline{\phi}_a(x - x_I)$ is the interface-modified kernel function defined in Eq. (42). The unknown coefficient vector $b(x)$ is obtained by imposing the nth order reproducing conditions, as shown in Eq. (10). Substituting Eq. (47) into Eq. (10) yields:

$$b(x) = \overline{M}^{-1}(x)H(0) \qquad (48)$$

where $\overline{M}(x)$ is the modified moment matrix:

$$\overline{M}(x) = \sum_{I \in \mathbb{S}^{IF}} H^T(x - x_I) H(x - x_I)\phi_a(x - x_I)$$
$$+ \sum_{I \in \mathbb{S}^{RK}\backslash\mathbb{S}^{IF}} H^T(x - x_I) H(x - x_I)\overline{\phi}_a(x - x_I)$$

$$(49)$$

By substituting Eq. (48) into Eq. (47), the interface modified reproducing kernel shape function is obtained as:

$$\overline{\Psi}_I(x) = \begin{cases} H^T(0)\overline{M}^{-1}(x)H(x - x_I)\phi_a(x - x_I), \forall I \in \mathbb{S}^{IF} \\ H^T(0)\overline{M}^{-1}(x)H(x - x_I)\overline{\phi}_a(x - x_I), \forall I \in \mathbb{S}^{RK}\backslash\mathbb{S}^{IF} \end{cases}$$

$$(50)$$

Finally, the IM-RK approximation of the displacement field $u^h(x)$ is expressed as:

$$u_i^h(x) = \sum_{I=1}^{NP} \overline{\Psi}_I(x)d_{iI} \qquad (51)$$

As shown in Eq. (51), no duplicated degrees of freedom associated with the interface nodes are added (such as interface enrichments) when using the IM-RK to approximate the displacement field. Figure 20 compares the 1D traditional RK and IM-RK shape functions and their derivatives in a domain [0, 10] with a material interface at $x = 5$. The red-colored node in Fig. 20 is the interface node, and the shape functions colored black, blue, and red are associated with nodes outside the support of the interface node, nodes within the influence of the interface node, and the interface node, respectively.

Fig. 25 Schematic of compression-shear test on a polymer-ceramic composite
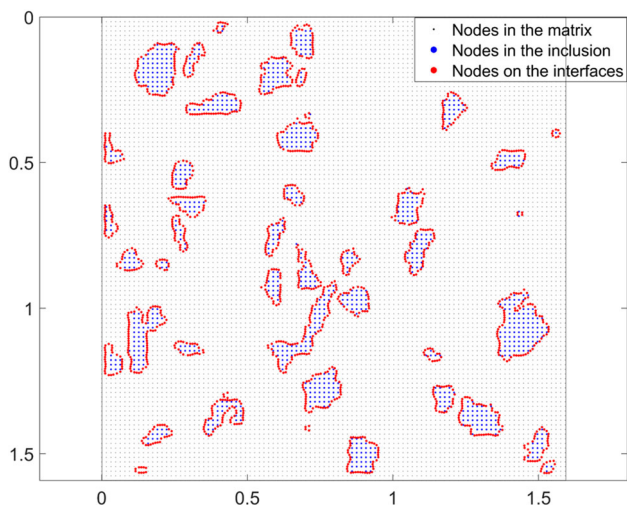
Fig. 26 Discretized RK numerical model for IM-RKPM simulation (unit: mm)
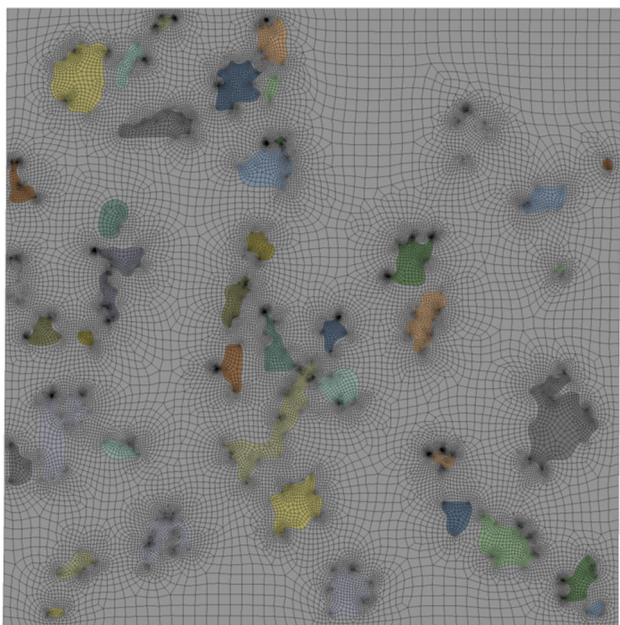
Fig. 27 FEM body-fitted mesh

Remarks 4.2. 1. Owing to the regularization function $\widetilde{H}(\bar{\bar{\xi}}_I(\boldsymbol{x}))$ introduced in Eqs. (43)–(44), nodes on different sides of the interface lose communication, leading to weak discontinuities in the IM-RK shape functions. This is true even when a smooth cubic B-spline kernel is used for all nodes, including the interface nodes, as shown in Fig. 20. Same weak discontinuity properties exist in high dimensions, as shown in Fig. 21, regardless of the smoothness of interface

kernel functions. Figure 22 illustrates the IM-RK interpolation of a function:

$$f(\boldsymbol{x}) = \begin{cases} e^{c_1 \|\boldsymbol{x}\|^2}, \ if \|\boldsymbol{x}\| \leq R \\ e^{c_2 \|\boldsymbol{x}\|^2} + (e^{c_1 R^2} - e^{c_2 R^2}), \ if \|\boldsymbol{x}\| > R \end{cases}$$

where $c_1 = 0.5$, $c_2 = 0.1$, and the interface is a circular arc with $R = 0.8$. IM-RK shape functions with smooth cubic B-spline kernels can effectively capture weak discontinuities along the interface in the interpolated function and represent its discontinuous x-directional derivative field, while both the interpolated function and its x-directional derivative fields are smooth across the interface when using the standard RK shape functions with B3 kernels.

2. Since all kernel functions vanish at the interface, except for the kernel functions defined on the interface nodes, the resulting IM-RK approximation functions possess weak Kronecker delta properties, as have been discussed in [46]. These weak Kronecker delta properties, however, do not exist in high dimensions because the supports of interface nodes overlap except for interface nodes located on domain boundaries.

In this work, the meshfree method using the above IM-RK approximation as the approximation function for the test and trial functions under the Galerkin framework is named the Interface-Modified Reproducing Kernel Particle Method (IM-RKPM). Two numerical examples are presented in Appendix B to validate the accuracy and convergence of proposed IM-RKPM.

4.3 IM-RK shape functions for the image-based numerical model

Figures 23 and 24 respectively show the IM-RK shape functions of non-interface nodes near the interfaces and the IM-RK shape functions of the interface nodes, constructed on the image-based SVM-RK discretized model shown in Fig. 10.

In Figs. 23 and 24, the non-zero IM-RK shape functions are color-coded by different color blocks, and the maximum shape function is shown at each plotting point in the top view. By observing the results in Fig. 23, the shape functions are truncated across arbitrarily shaped interfaces. The interface nodes' shape functions, however, provide support coverage to the nodes located on both sides of the interface with C^0 continuity along the interfaces' normal direction for embedding weak discontinuities normal to the interface.

Fig. 28 IM-RKPM and FEM approximated displacement solution in both x- and y-directions (unit: mm)

(a) Approximated displacement solutions using IM-RKPM

(b) Approximated displacement solutions using FEM
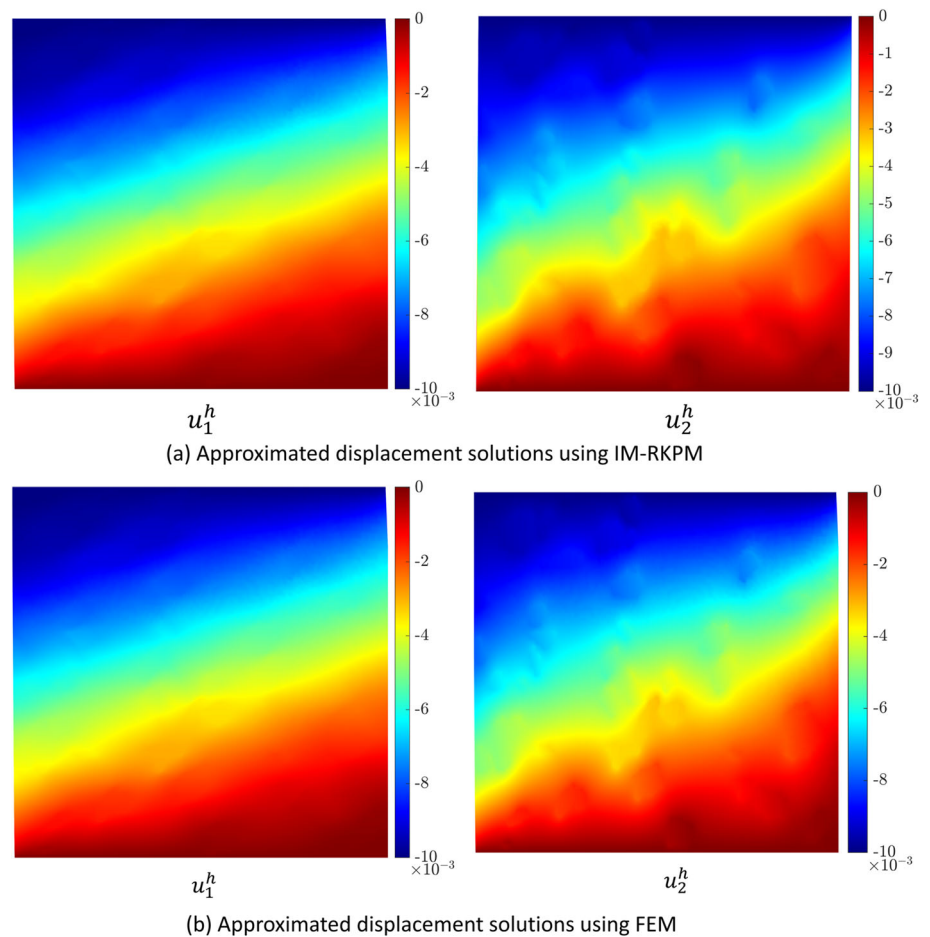
5 Image-based numerical results

5.1 Compression-shear test on 2D composite microstructure

In this numerical example, a compression-shear test is conducted on a composite constructed based on the image shown in Fig. 25. The image consists of 200×200 pixels with a pixel size of 8 μm. The physical dimensions of the specimen are 1.6 mm in width and height. The bottom edge of the specimen is fixed in both x- and y- directions, while the top edge is prescribed with a total displacement of -0.01 mm in both x- and y- directions. In addition, two vertical edges of the specimen are assigned as traction-free. The material constants for the alumina inclusion materials are: $E_1 = 320$ GPa, $\nu_1 = 0.23$, while the epoxy material is with material constants: $E_1 = 3.66$ GPa, $\nu_1 = 0.358$.

The problem at hand is examined using the proposed IM-RKPM and compared with the results produced using ANSYS [72], a commercially available FEM software with a much refined body-fitted mesh. All numerical analyses are performed under the 2D plane strain condition. The proposed IM-RKPM uses Nitsche's method [70] to apply Dirichlet

boundary conditions. The numerical model of the test image is constructed following Sect. 3.2, as shown in Figs. 26, and the IM-RKPM approximation functions presented in Sect. 4 using cubic B-spline kernel with normalized support size of 2 and linear bases. The model employed for FEM analysis is manually traced from the inclusion geometries of the test image, resulting in a slight variation between the FEM and IM-RKPM discretization near interfaces. The FEM approximation involves a fine body-fitted mesh that comprises of 37,454 elements and 112,538 nodes, as illustrated in Fig. 27. On the other hand, the IM-RKPM approximation uses only 11,316 nodes to discretize the image domain, which is approximately one-tenth of the number of nodes used in the FEM model. Figure 28 demonstrates the IM-RKPM and FEM approximated displacement solutions, respectively, and it is observed that both IM-RKPM and FEM predict similar displacements.

Figure 29 shows the strains predicted by IM-RKPM and FEM, respectively. As shown in Fig. 29, the strains of IM-RKPM display sharp transitions across the material interfaces and concentrated strains around the corners of the material interfaces, comparable to the results obtained
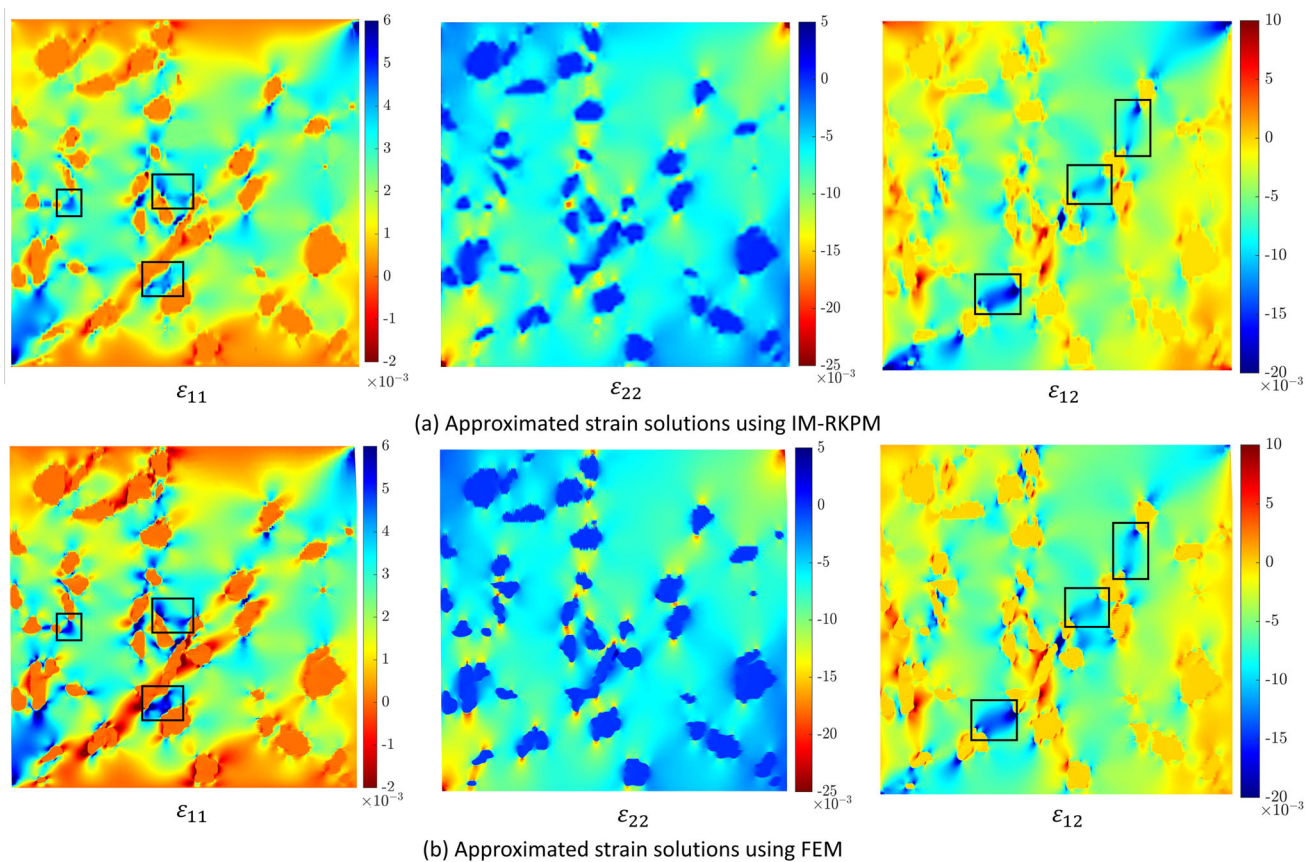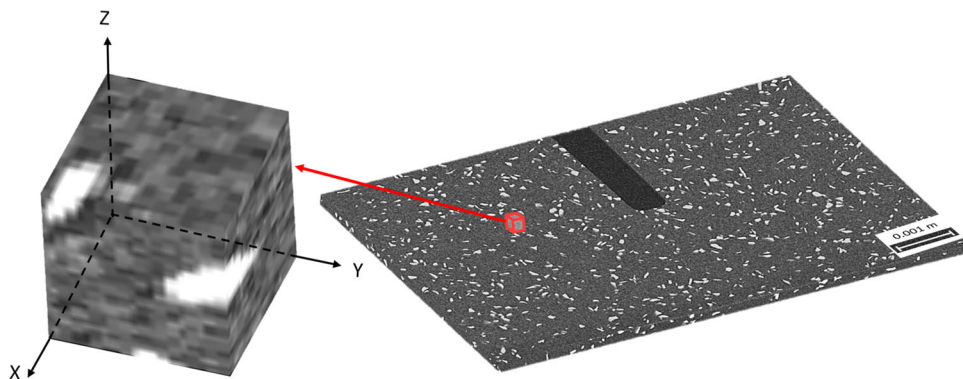
(a) Approximated strain solutions using IM-RKPM

(b) Approximated strain solutions using FEM

Fig. 29 IM-RKPM and FEM approximated strain solutions in plane view

**Fig. 30** 3D view of the input image volumetric data matrix

using the FEM approximation. Furthermore, both the IM-RKPM and FEM approximations of strain solutions show some coalescence of the strain concentration around some closely positioned inclusions, as indicated by the boxed areas in Fig. 29. The results show that the proposed IM-RKPM, accompanied by the SVM-based RK discretization with simple interface modified RK approximation functions, is capable of modeling composite materials with complicated microstructure and arbitrarily shaped inclusions with accuracy comparable to that obtained from a much refined and laborious FEM model.

5.2 Uniaxial tensile test on 3D composite microstructure

In this example, a three-dimensional image-based SVM-RK model is constructed, and a uniaxial tensile test is conducted on the specimen's numerical model with the same material properties and essential boundary treatment by Nitsche's method [70] as those used in Example 5.1. The input of imaged-based 3D numerical model generation is performed by stacking 30 slices of ROI of 30 by 30 pixels extracted from reconstructed micro-CT 2D images of a specimen's internal
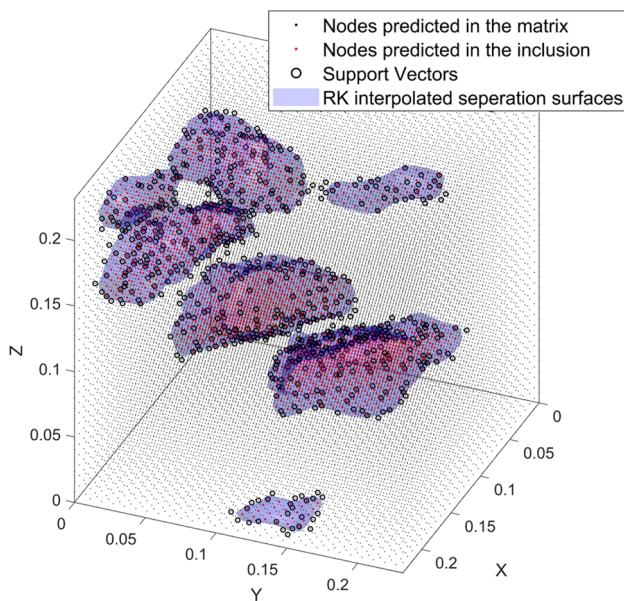
Fig. 31 SVM training and prediction results and RK interpolated decision boundaries (unit: mm)

microstructure along the z-direction into a volumetric data matrix, as illustrated in Fig. 30. The size of the test volume is 0.24 mm × 0.24 mm × 0.24 mm, corresponding to an input image voxel size of 8 µm. The uniaxial tension is applied to the two surfaces with surface normal in the z-direction under prescribed displacements in the z-direction while without constraints in the x–y displacements.

Although the training data points are now in $\mathbb{R}^3$, the discrete model generation procedures remain the same as those for 2D, which corresponds to the physical coordinates of voxel centroids, as detailed in Sect. 3.2. The training response labels are obtained by stacking the segmented ROIs using Otsu's method into a binary volumetric data label matrix. It is important to note that both matrices that contain the training data points and training response labels are concatenated before being fed into the SVM, which means that the combined training data set can be represented as $\mathbf{D} = \{\boldsymbol{x}_i, \ y_i\}_{i=1}^{l}$, where $\boldsymbol{x}_i \in \mathbb{R}^3$, $y_i \in \{-1, 1\}$, and $l = 27,000$ for the present case. For the SVM training, the same hyperparameters specified in Sect. 3.2.1 are utilized. Figure 31 demonstrates the SVM material classification results, the support vectors resulting from training, and the RK interpolated decision boundaries. It is worth mentioning that the resulting RK interpolated separating hyperplane, which is the material interface determined by the SVM training and RK interpolation, exhibits a smoother appearance in contrast to the inclusion geometries represented by binary label data volumetric matrix. This outcome is not surprising, given that SVM considers all three dimensions of the training data points to identify an appropriate separating hyperplane,

which is more realistic and resembles a specimen image stack with higher resolution (i.e., with a smaller voxel size).

Figure 32 illustrates the results of identified interface nodes and the 3D RK discrete model of the test volume, where the black (small) nodes, blue nodes, and red nodes represent the epoxy material points, alumina material points, and points on the material interfaces, respectively. The 3D SVM-RK discretization model contains in total 17,648 discretized nodes, among which 2330 nodes are on the material interfaces.

The produced 3D SVM-RK discrete model is utilized for a uniaxial tensile test in the z-direction. The model's bottom surface is fixed in all three directions, while a z-directional displacement of 0.01 mm is prescribed at the top surface of the model. The other surfaces of the specimen are assigned traction-free boundary conditions. The proposed IM-RKPM is employed for the numerical solution, and Fig. 33 depicts the displacement solution in all three directions. The predicted normal strains are plotted in Fig. 34, to which a transparency filter is applied such that strain with a large magnitude is not visible. As shown in Fig. 34, the regions with the relatively small magnitude of strains are consistent with the shapes of the alumina inclusions, which is expected because the alumina inclusions are significantly stiffer than the surrounding epoxy matrix. In Figs. 35, 36, and 37, the normal strains are plotted on multiple slices and are compared with the slices of the inclusion contours. The results show that distinctive strain transitions in all three dimensions can be observed across interfaces. In addition, some strain concentrations are observed between two nearby inclusions and around the corner of the inclusions. Overall, this example demonstrates the capability of the proposed SVM-RK image-based model and IM-RKPM in modeling composite material with arbitrarily shaped inclusions in three dimensions.

6 Conclusion

A Support Vector Machine (SVM) guided model discretization and reproducing kernel approximation, utilizing micro-CT images of heterogeneous materials as input, is introduced in this work. The trained SVM-RK model generates classification scores for RKPM model discretization from the image, enabling their use as inputs for (1) interface node generation and (2) the interface kernel modification to construct a modified RK approximation of weak discontinuities. The SVM classification scores, representing the signed distances, enables identification of material phase, interface discretization, and interface surface normals, allowing automatic construction of RK approximation with weak discontinuities and interface-conforming gradient smoothing cells for SCNI based domain integration. The proposed image-based SVM-RK model generation process was validated through a
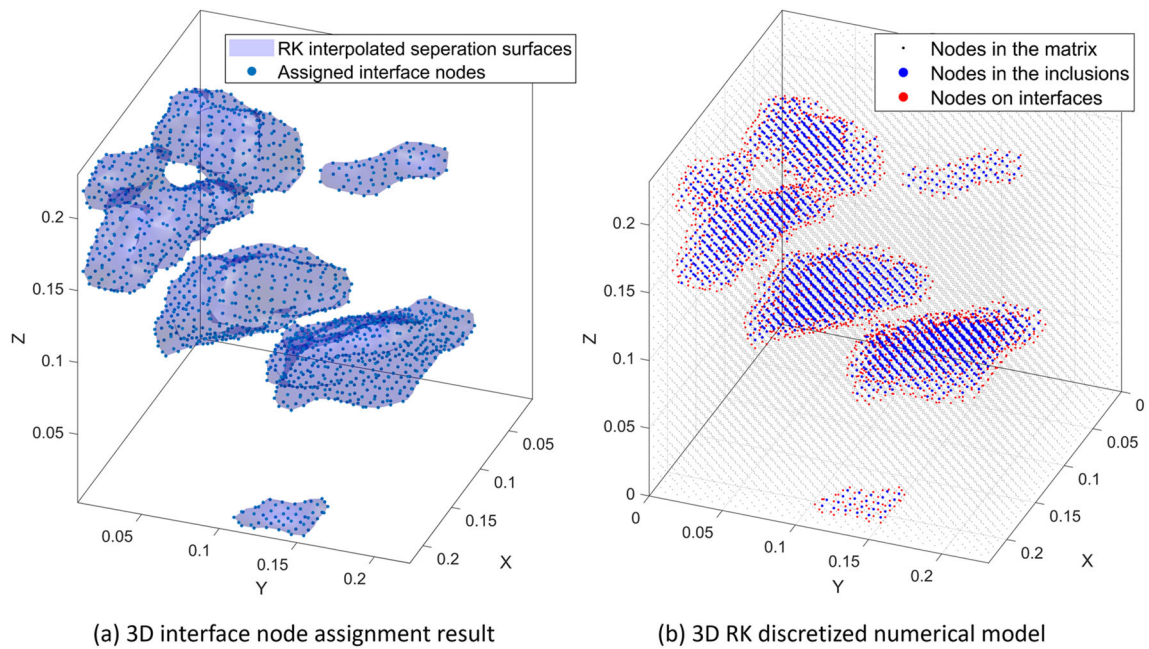
(a) 3D interface node assignment result

(b) 3D RK discretized numerical model

Fig. 32 3D interface node assignment result and RK discretized numerical model for the test volume (unit: mm)

(a) u_1^h

(b) u_2^h
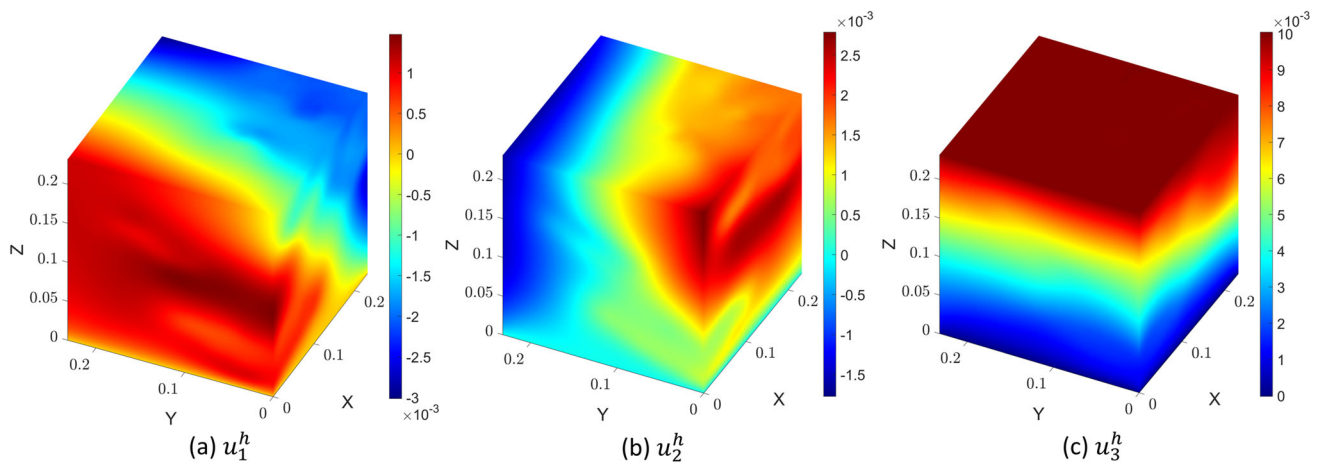
(c) u_3^h

Fig. 33 IM-RKPM approximated displacement solutions (unit: mm)

(a) ε_{11}

(b) ε_{22}

(c) ε_{33}
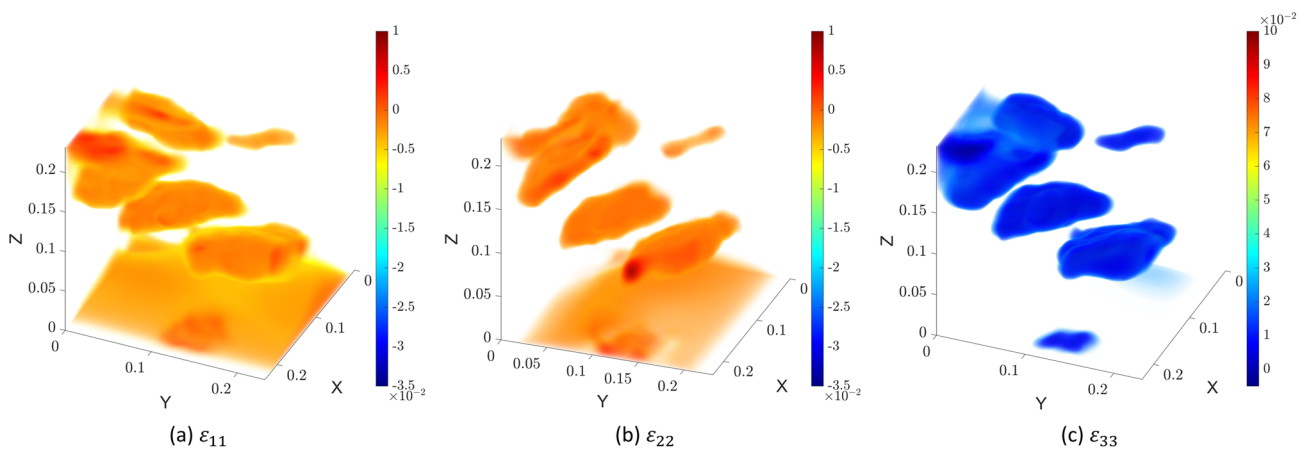
Fig. 34 IM-RKPM approximated normal strain solutions (unit: mm)

(a) At Z = 0.032 mm

(b) At Z = 0.064 mm

(c) At Z = 0.112 mm

(d) At Z = 0.176 mm
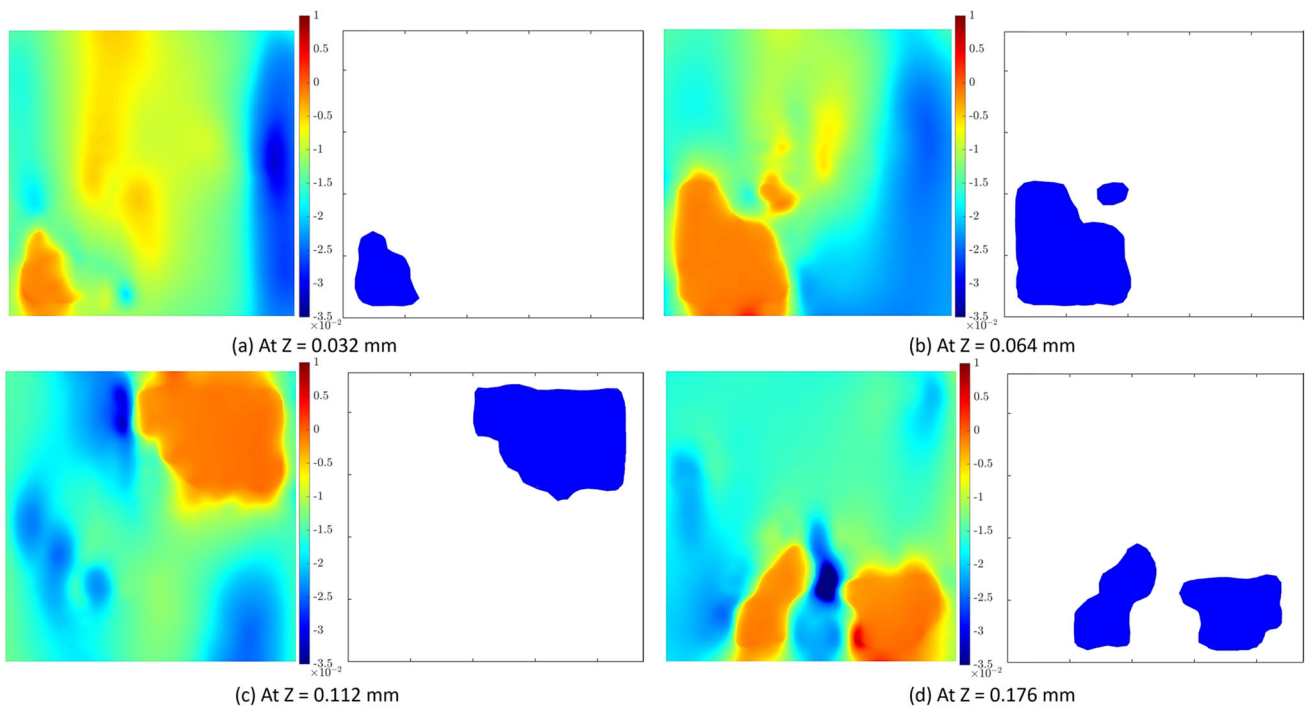
Fig. 35 Slices of x-directional normal strain results compared to the interface contours

(a) At Z = 0.048 mm

(b) At Z = 0.088 mm

(c) At Z = 0.152 mm

(d) At Z = 0.192 mm
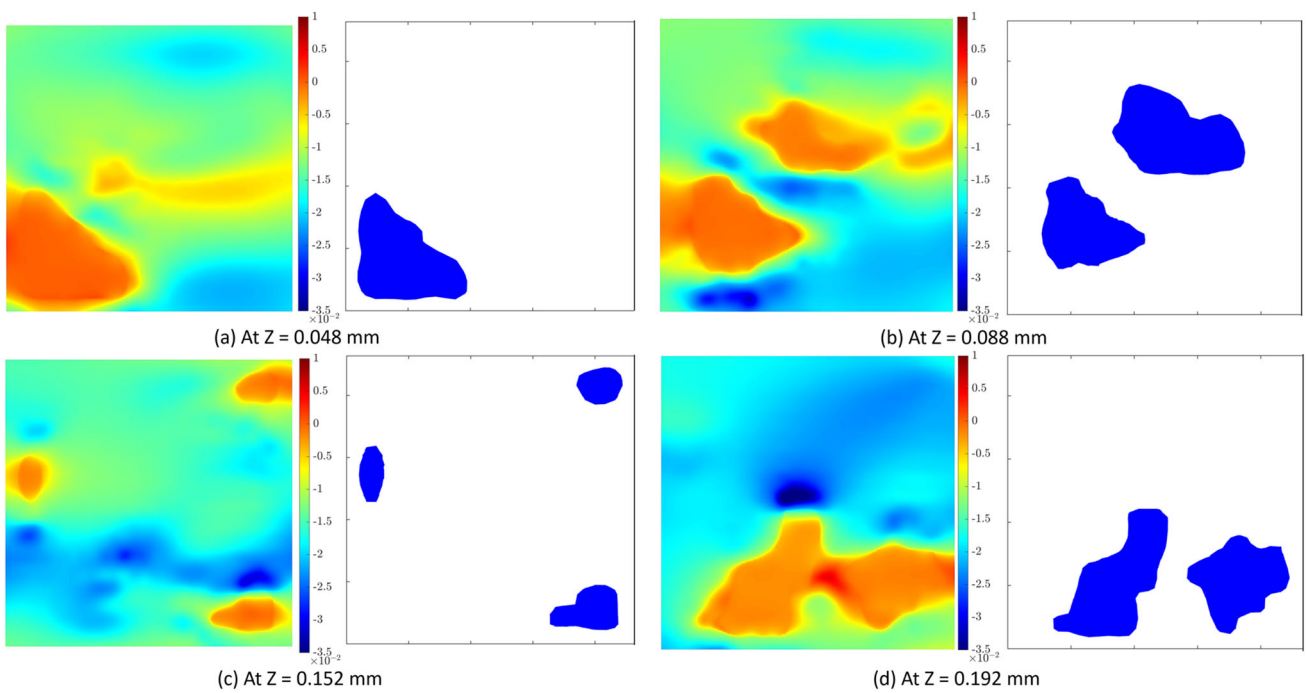
Fig. 36 Slices of y-directional normal strain results compared to the interface contours
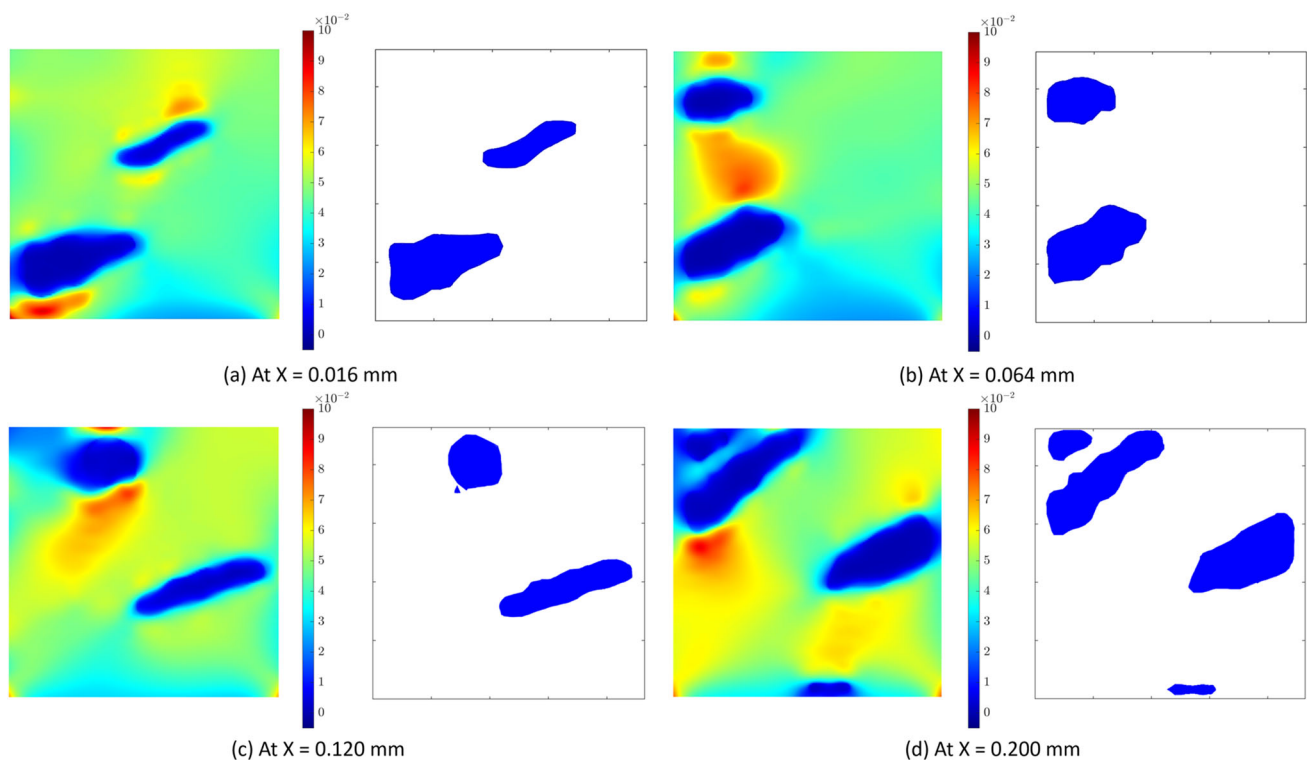
Fig. 37 Slices of z-directional normal strain results compared to the interface contours

synthetic image and a high-resolution surface image obtained from the SEM.

The resulting Interface-Modified Reproducing Kernel Particle Method (IM-RKPM) effectively remedy Gibb's oscillations commonly seen in the conventional RKPM for modeling problems with weak discontinuities. The proposed method incorporates a regularized Heaviside function defined on the SVM classification score to achieve RK approximation with interface weak discontinuity while avoiding Gibb-type oscillations. These procedures involved in the proposed SVM guided RK approximation with interface weak discontinuities are fully automatic in 3-dimensions and without the need of using duplicated degrees of freedom on the interface nodes common in other interface-enriched meshfree methods [33, 36]. In addition, this IM-RKPM with interface weak discontinuities can be constructed by kernel functions with arbitrary smoothness/roughness while achieving optimal convergence as demonstrated in the numerical examples using both Gauss integration and SCNI.

Finally, the effectiveness of the proposed automated SVM-RK model generation process in conjunction with the IM-RKPM method is demonstrated through numerical examples based on test micro-CT images in both 2- and 3-dimensions. Notably, the 3D example shows that the proposed approach is applicable for 3D simulations where the SVM-RK model precisely represents the geometry of the inclusion particles, trained based on stacked image slices.

It is worth mentioning that while the present work utilizes the standard binary SVM library for two-phase materials, it is possible to explore multi-class SVM algorithms, or more advanced machine learning algorithms, such as the convolutional neural network, for multi-phase materials segmentation.
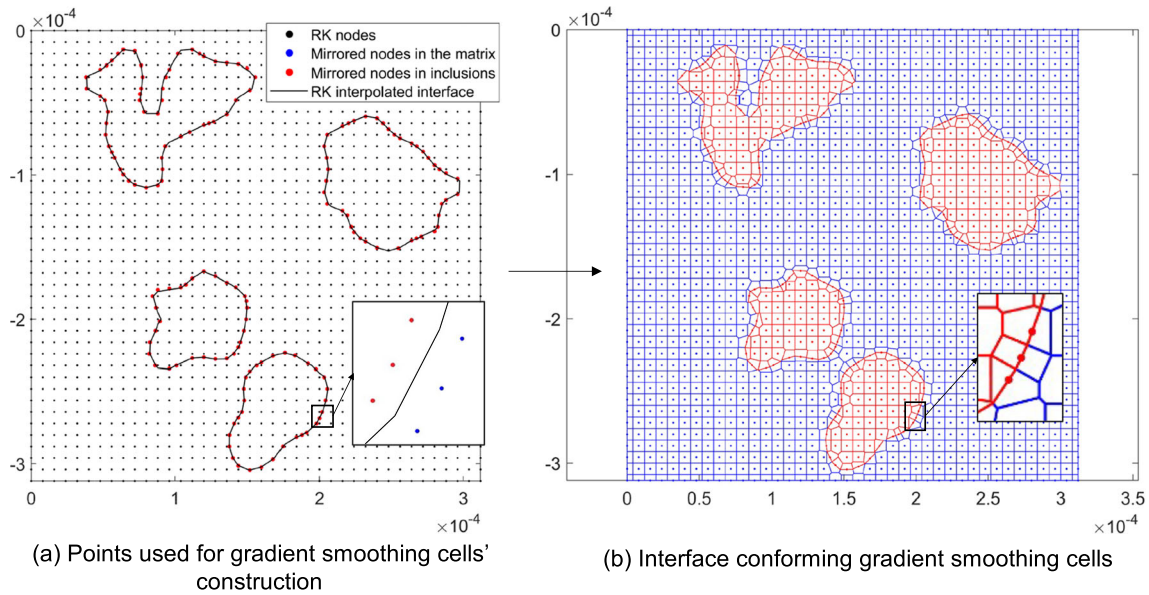
Declarations

Conflict of interest The authors have no relevant financial or non-financial interests to disclose.

(a) Points used for gradient smoothing cells' construction

(b) Interface conforming gradient smoothing cells

Fig. 38 Interface conforming gradient smoothing cells' construction (Perturbation distances are magnified 10^2 times for mirrored nodes in (a) for demonstration purpose)

Appendix A: Construction of interface-conforming gradient smoothing cells

Since the interface locations are determined by the RK interpolated score function in Eq. (40), the outward unit normal of the interfaces at an interface point x_K^* can be calculated as follows:

$$n(x_K^*) = \frac{\nabla \widetilde{S}(x)}{\|\nabla \widetilde{S}(x)\|}\bigg|_{x=x_K^*} \tag{52}$$

To construct interface-conforming gradient smoothing cells for SCNI domain integration, a mirroring technique is utilized. For all interface nodes $x_K^* \in \mathbb{S}^{IF}$, the mirrored node pair $\{x_K^{*+}, x_K^{*-}\}$ is obtained as follows:

$$x_K^{*\pm} = x_K^* \pm \epsilon \, \overrightarrow{n}\left(x_K^*\right) \tag{53}$$

where ϵ is a small perturbation number, and the interface normal $\overrightarrow{n}\left(x_K^*\right)$ is defined in Eq. (52). In this work, $\epsilon = 10^{-3}\ell$ is chosen, where ℓ is the image voxel size. As an illustration example, mirrored nodes are shown in the black box in Fig. 38a, and the resulting gradient smoothing cells are shown in Fig. 38b. Note that these mirrored node pairs are only used to generate the "interface conforming" Voronoi cells; they are not the RK nodes, and they don't carry degrees of freedom.

This approach allows for the use of conventional techniques, such as Voronoi tessellation, which will result in the two smoothing cells adjacent to either side of the interface
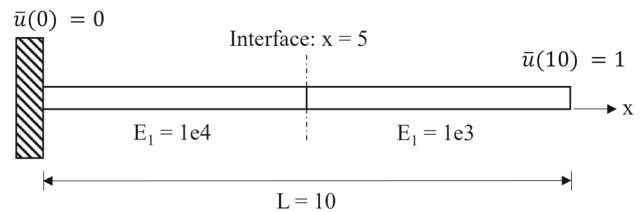
Fig. 39 Schematic of the 1D bi-material rod problem

having a common boundary along the interface location as shown in Fig. 38b. In addition, the material class for the smoothing cells can be assigned according to the centered mirrored nodes' material classes. As a result, smoothing cells away from the material interfaces are uniformly arranged, and the material interfaces are well represented by the adjacent two layers of smoothing cells.

Appendix B: Verification of IM-RKPM

### Numerical example 1: one-dimensional composite rod problem

A one-dimensional composite rod with a centered material interface is fixed at the left and is subjected to a displacement of 1 on the right end, as demonstrated in Fig. 39. The rod is also subjected to a polynomial body force up to the third order $b(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$. The Young's modulus of the two sections of the rod are set as $E_1 = 10000$, for $x \in [0, 5]$ and $E_2 = 1000$, for $x \in]5, 10]$. The exact solution to this problem is provided in [33].

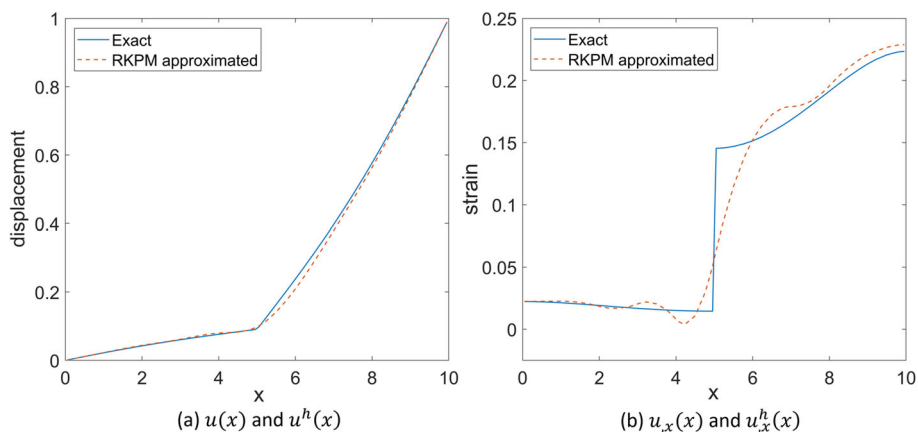**Fig. 40** Case (1) RKPM approximated solutions compared to the analytical solutions

(a) $u(x)$ and $u^h(x)$

(b) $u_{,x}(x)$ and $u^h_{,x}(x)$
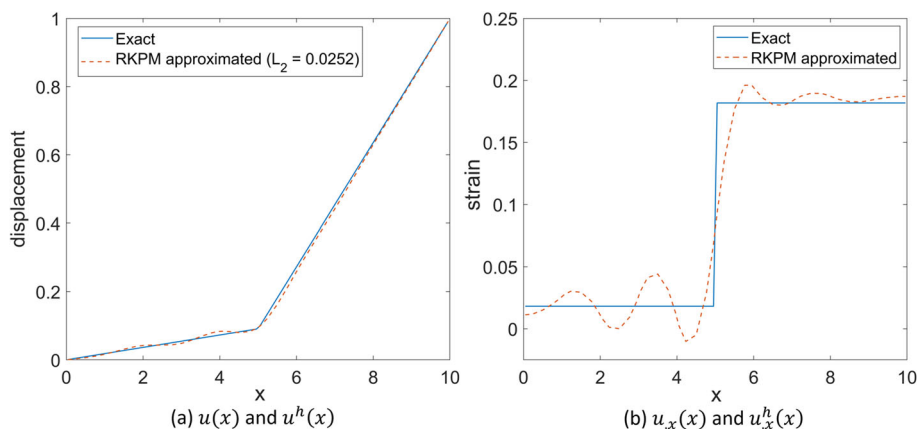
Fig. 41 Case (1) IM-RKPM approximated solutions compared to the analytical solutions


(a) $u(x)$ and $u^h(x)$

(b) $u_{,x}(x)$ and $u^h_x(x)$
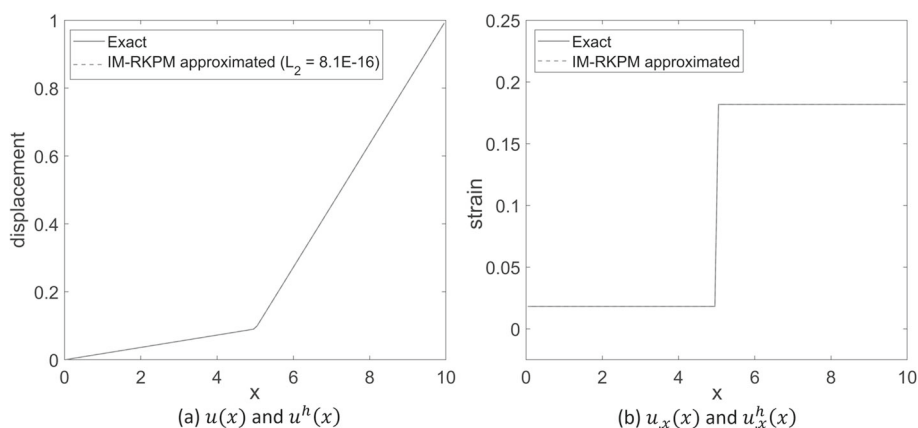
The example is analyzed with two body force cases: (1)$b = 0$; (2)$b(x) = 25x - 7.5x^2 + 0.5x^3$. The 1D problem domain is discretized with 11 uniformly distributed nodes, and the problem is approximated using a linear basis in both standard RKPM and IM-RKPM with a constant normalized nodal support size of 2. SCNI and 5-point Gauss integration are selected as numerical integration methods for case (1) and case (2), respectively. Figures 40 and 41 demonstrate the approximated displacement and strain solutions using RKPM and IM-RKPM for case (1), respectively. The results

show that RKPM strain solution exhibits Gibb's-like oscillations and fails to reproduce the exact weak discontinuity at the material interface. On the other hand, IM-RKPM with SCNI can precisely capture the displacement and strain field to the machine's precision. Similar behaviors are observed for case (2), as illustrated in Figs. 42 and 43, IM-RKPM significantly reduces the oscillations of the strain solution and can accurately capture the weak discontinuity across the material interface.

Fig. 42 Case (2) RKPM approximated solutions compared to the analytical solutions

(a) $u(x)$ and $u^h(x)$

(b) $u_{,x}(x)$ and $u^h_x(x)$

Fig. 43 Case (2) IM-RKPM approximated solutions compared to the analytical solutions

(a) $u(x)$ and $u^h(x)$

(b) $u_{,x}(x)$ and $u_{,x}^h(x)$
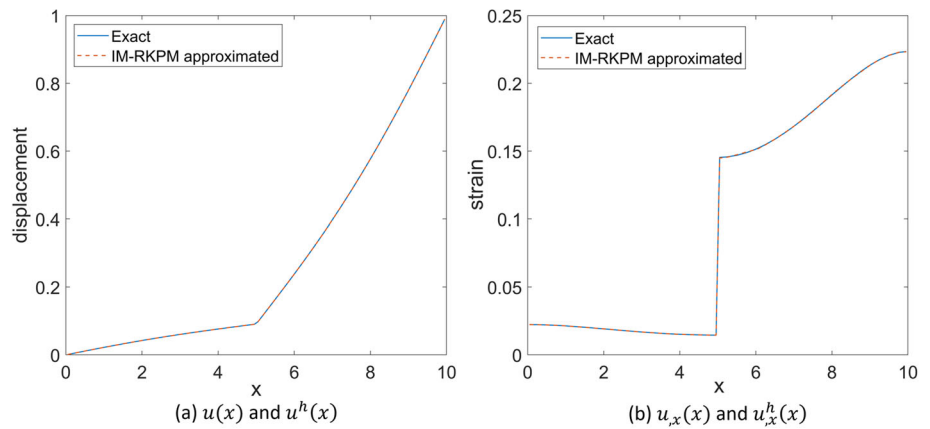
Fig. 44 Accuracy of RKPM and IM-RKPM with different interface kernels (R: rate of convergence)

(a) Normalized displacement norms

(b) Normalized energy norms
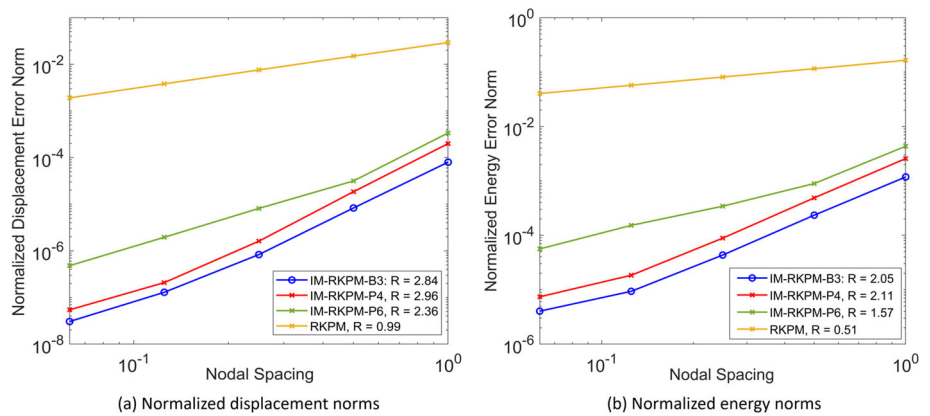
In addition, the convergence behaviors of IM-RKPM with the cubic B spline and the power interface kernels and standard RKPM are investigated in terms of the normalized displacement and energy error norms as follows with high-order Gauss quadrature rule:

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_0 = \sqrt{\frac{\int_\Omega \left(\boldsymbol{u}^{\mathrm{exact}}(\boldsymbol{x}) - \boldsymbol{u}^h(\boldsymbol{x})\right)^2 d\Omega}{\int_\Omega (\boldsymbol{u}^{\mathrm{exact}}(\boldsymbol{x}))^2 d\Omega}} \tag{54}$$

$$\|\boldsymbol{u} - \boldsymbol{u}^h\|_E = \sqrt{\frac{\int_\Omega \left(\boldsymbol{\varepsilon}^{\mathrm{exact}}(\boldsymbol{x}) - \boldsymbol{\varepsilon}^h(\boldsymbol{x})\right) \cdot \left(\boldsymbol{\sigma}^{\mathrm{exact}}(\boldsymbol{x}) - \boldsymbol{\sigma}^h(\boldsymbol{x})\right) d\Omega}{\int_\Omega \boldsymbol{\varepsilon}^{\mathrm{exact}}(\boldsymbol{x}) \cdot \boldsymbol{\sigma}^{\mathrm{exact}}(\boldsymbol{x}) d\Omega}} \tag{55}$$

As illustrated in Fig. 44, standard RKPM exhibits a suboptimal convergence rate of 1 for the displacement norms and 0.5 for the energy norm, while the accuracy and convergence rates are substantially improved in IM-RKPM, restoring the optimal convergence rates of 2 and 1, independent to the continuity of the interface kernel functions.

Numerical example 2: 2D circular inclusion in an infinite plate

An infinite plate with a circular inclusion subjected to a constant dilatational eigenstrain $\varepsilon^* = 0.01$, as shown in Fig. 45,
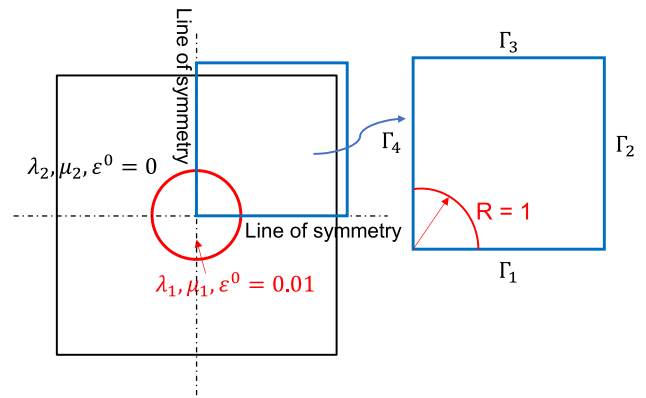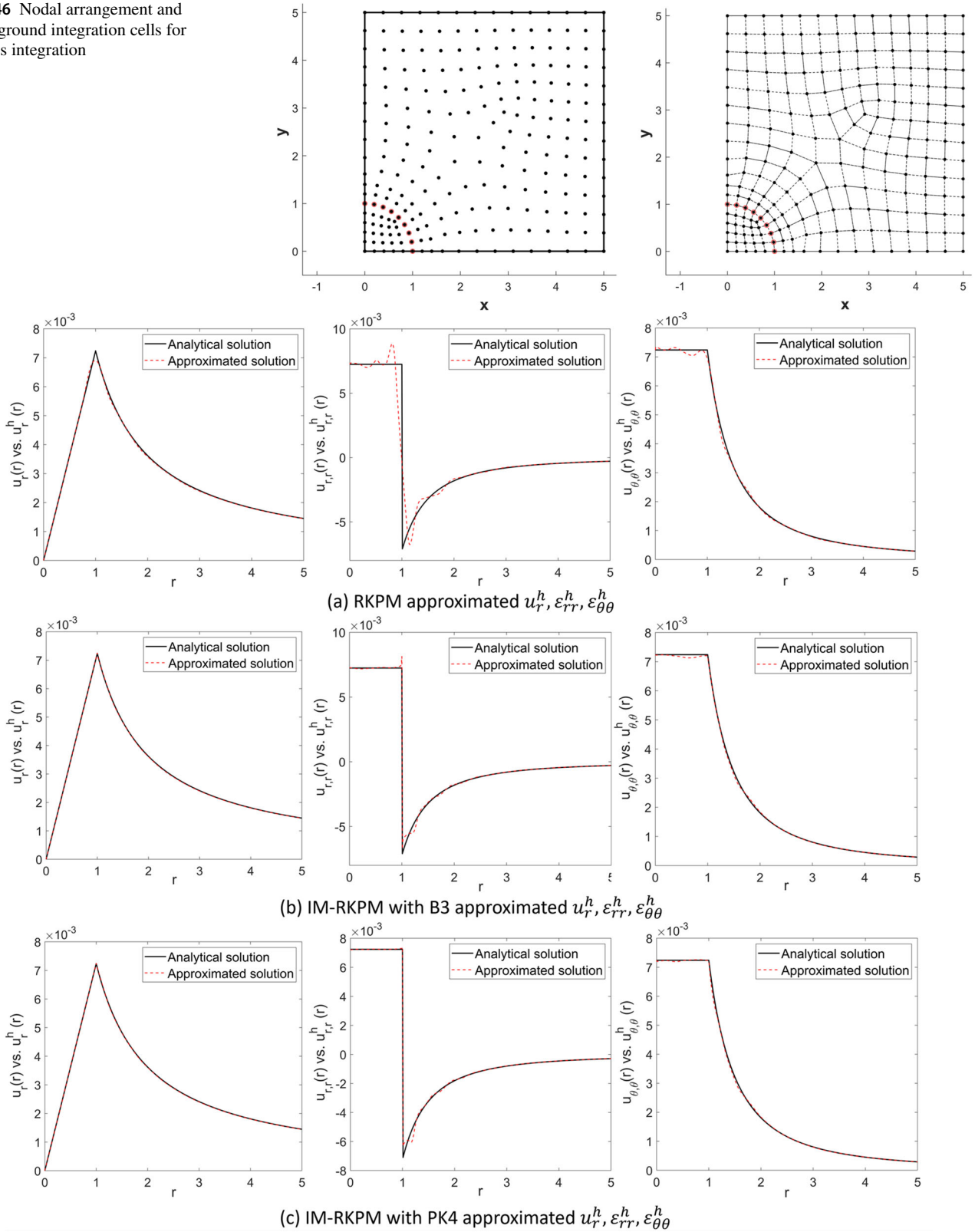

**Fig. 45** Schematic of the 2D infinite plate with circular inclusion problem

is analyzed.

The material constants selected for the inclusion material are: $\lambda_1 = 497.16$, $\mu_1 = 390.63$, and matrix material are: $\lambda_2 = 656.79$, $\mu_2 = 338.35$, where λ and μ are Lamé parameters. Due to the symmetry of the domain and loading conditions, only the upper right quadrant of the domain is modeled. The length of each side of the finite quarter domain is 5, the radius of the circular inclusion is $R = 1$, and an analytical displacement field is prescribed on the boundaries.

Fig. 46 Nodal arrangement and background integration cells for Gauss integration

(a) RKPM approximated $u_r^h, \varepsilon_{rr}^h, \varepsilon_{\theta\theta}^h$

(b) IM-RKPM with B3 approximated $u_r^h, \varepsilon_{rr}^h, \varepsilon_{\theta\theta}^h$

(c) IM-RKPM with PK4 approximated $u_r^h, \varepsilon_{rr}^h, \varepsilon_{\theta\theta}^h$

Fig. 47 $u_r^h, \varepsilon_{rr}^h, \varepsilon_{\theta\theta}^h$ approximated using RKPM and IM-RKPM with Gauss integration

Fig. 48 Accuracy of RKPM and IM-RKPM with different interface kernels with Gauss integration (R: rate of convergence)

(a) Normalized displacement norms

(b) Normalized energy norms
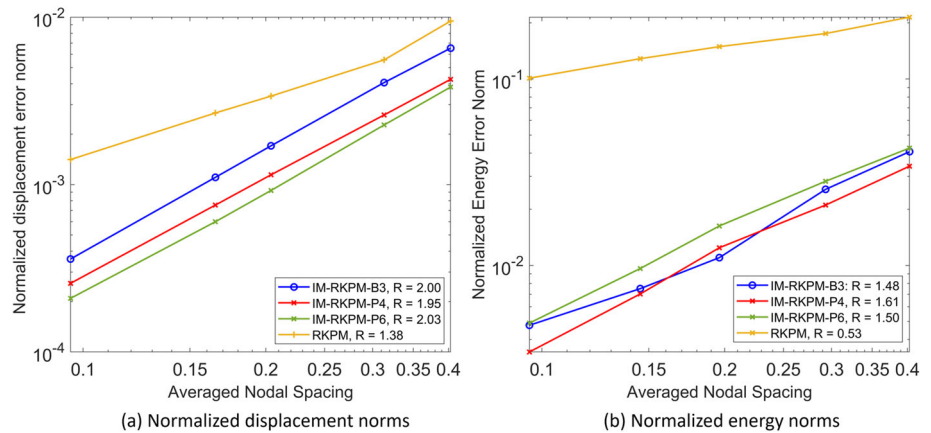
Fig. 49 Nodal arrangement and conforming strain smoothing cells for SCNI

(a) Nodal arrangement

(b) Conforming gradient smoothing cells
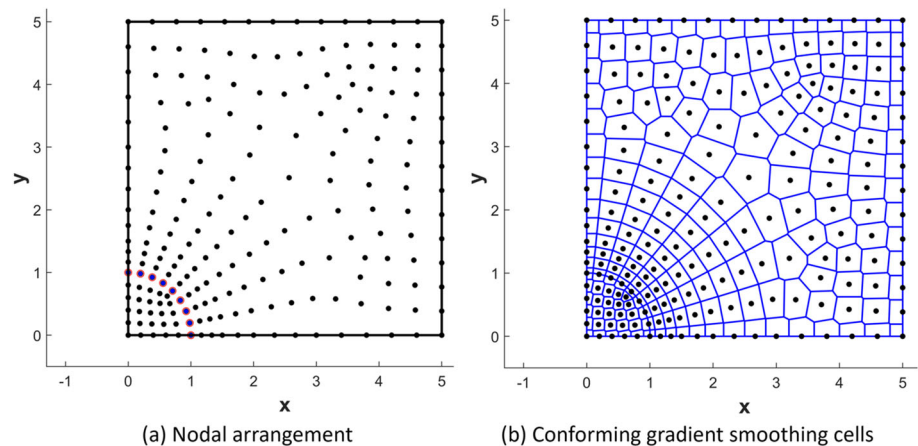
The analytical solutions in cylindrical coordinates can be found in [71].

The example is modeled as a plane strain axisymmetric problem. Both 5×5 Gauss integration and SCNI method are employed as the numerical integration schemes, and RK approximation with linear basis and a normalized support size of 2 are utilized throughout the numerical analysis. Figure 46 demonstrates an example of domain discretization and background integration cell arrangement for the Gauss integration. The approximated radial displacement, radial strain, and hoop strain solutions using RKPM, IM-RKPM with cubic B spline interface kernels, and IM-RKPM with fourth-order power interface kernels, accompanied with 224 non-uniform nodes and 5×5 Gauss integration, are plotted along the line $y = x$ in Fig. 47. Like the 1D composite rod example, the RKPM solution of the radial strain and hoop strain are both oscillatory near the interface. IM-RKPM, on the other hand, effectively alleviates the oscillations in the strain solutions. In addition, a convergence study is performed for both RKPM and IM-RKPM with different interface kernels, and the results are shown in Fig. 48. The IM-RKPM recovers the optimal convergence rates with the Gauss domain integration for both smooth and C^0 interface kernels.

Next, the same problem is solved using the computationally efficient SCNI. Figure 49 demonstrates an arrangement of 211 non-uniformly distributed nodes and conforming strain smoothing cells for SCNI. The numerical solutions obtained by RKPM and IM-RKPM with different interface kernels are plotted in Fig. 50. Similar to the Gauss integration, the standard RKPM with SCNI again experiences strain oscillations near the interface. However, solutions obtained by the IM-RKPM with different interface kernel functions are consistent with the ones obtained with Gauss integration in Fig. 47. By observing results in the convergence plots shown in Fig. 51, IM-RKPM with SCNI has displacement and strain solutions to converge optimally. These results show that the proposed IM-RKPM performs well with different selections of numerical domain integration techniques.
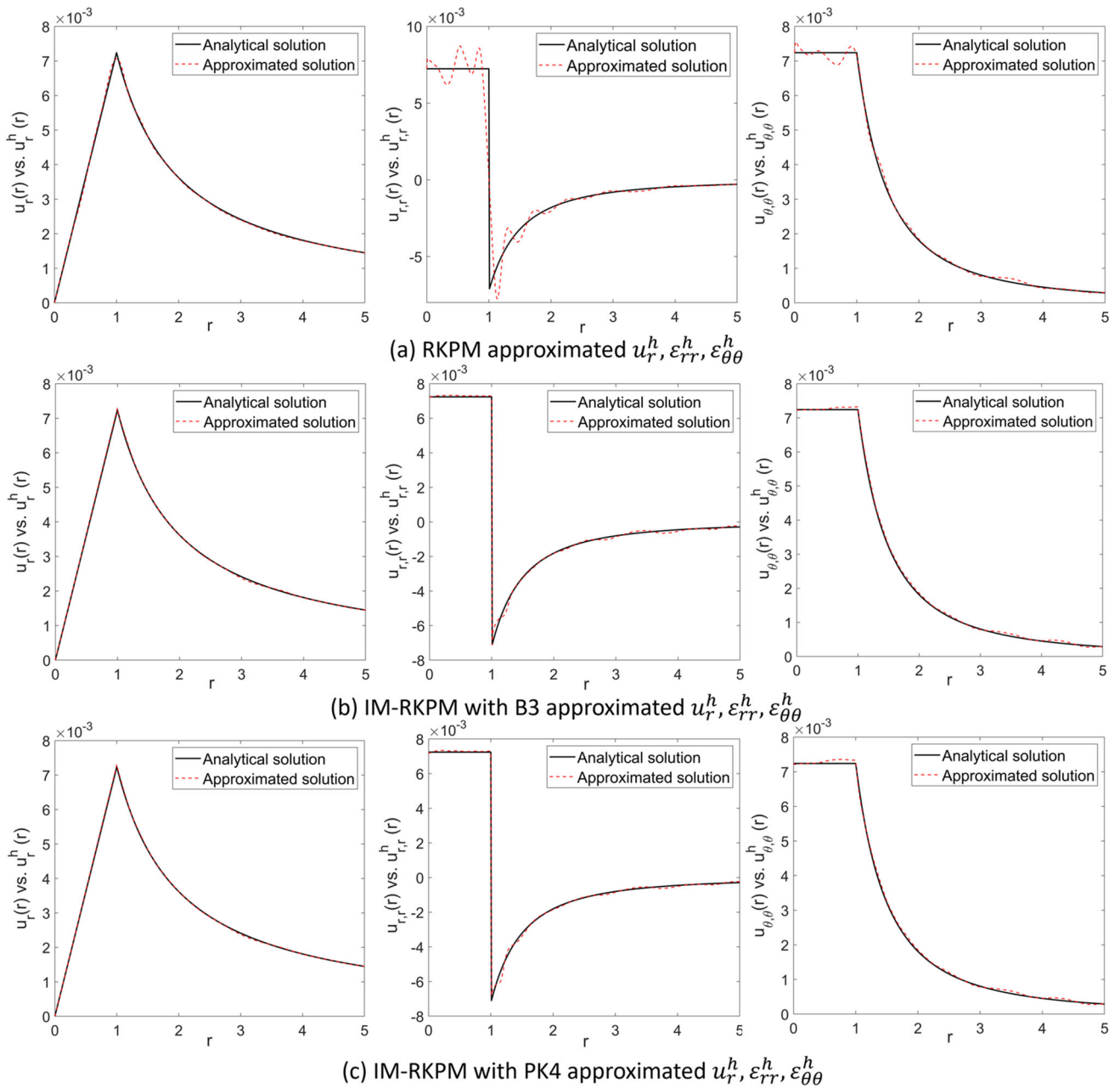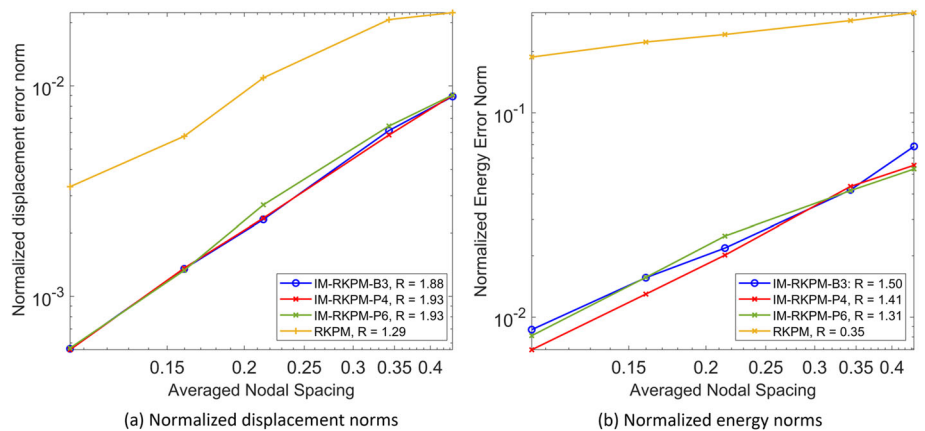
(a) RKPM approximated u_r^h, ε_{rr}^h, $\varepsilon_{\theta\theta}^h$

(b) IM-RKPM with B3 approximated u_r^h, ε_{rr}^h, $\varepsilon_{\theta\theta}^h$

(c) IM-RKPM with PK4 approximated u_r^h, ε_{rr}^h, $\varepsilon_{\theta\theta}^h$

Fig. 50 u_r^h, ε_{rr}^h, $\varepsilon_{\theta\theta}^h$ approximated using RKPM and IM-RKPM with SCNI

Fig. 51 Accuracy of RKPM and IM-RKPM with different interface kernels with SCNI (R: rate of convergence)

(a) Normalized displacement norms

(b) Normalized energy norms

References

1. Schilling PJ, Karedla BPR, Tatiparthi AK, Verges MA, Herrington PD (2005) X-ray computed microtomography of internal damage in fiber reinforced polymer matrix composites. Compos Sci Technol 65(14):2071–2078. https://doi.org/10.1016/J.COMPSCITECH.2005.05.014

2. Croom B, Wang W-M, Li J, Li X (2016) Unveiling 3D deformations in polymer composites by coupled micro X-ray computed tomography and volumetric digital image correlation. Exp Mech 56:999–1016. https://doi.org/10.1007/s11340-016-0140-7

3. Tang Y, Su K, Man R, Hillman MC, Du J (2021) Investigation of internal cracks in epoxy-alumina using in situ mechanical testing coupled with micro-CT. JOM 73:2452. https://doi.org/10.1007/s11837-021-04714-x

4. Liu G, Tang Y, Hattar K, Wang Y, Winders W, Haque A, Du J (2023) An investigation of fracture behaviors of NBG-18 nuclear graphite using in situ mechanical testing coupled with micro-CT. J Mater Res. https://doi.org/10.1557/s43578-023-00929-7

5. Norouzi A, Rahim MSM, Altameem A, Saba T, Rad AE, Rehman A, Uddin M (2014) Medical image segmentation methods, algorithms, and applications. IETE Tech Rev 31(3):199–213. https://doi.org/10.1080/02564602.2014.906861

6. Sahoo PK, Soltani S, Wong AKC (1988) A survey of thresholding techniques. Comput Vis Gr Image Process 41(2):233–260. https://doi.org/10.1016/0734-189X(88)90022-9

7. Haralick RM, Shapiro LG (1985) Image segmentation techniques. Comput Vis Gr Image Process 29(1):100–132. https://doi.org/10.1016/S0734-189X(85)90153-7

8. Saxena A, Prasad M, Gupta A, Bharill N, Patel OP, Tiwari A, Er MJ, Ding W, Lin CT (2017) A review of clustering techniques and developments. Neurocomputing 267:664–681. https://doi.org/10.1016/J.NEUCOM.2017.06.053

9. Guo G, Wang H, Bell D, Bi Y, Greer K (2003) KNN model-based approach in classification. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinf) 2888:986–996. https://doi.org/10.1007/978-3-540-39964-3_62

10. Bishop CM, Nasrabadi NM (2006) Pattern recognition and machine learning, Vol. 4. Springer, New York, p 738

11. Minaee S, Boykov Y, Porikli F, Plaza A, Kehtarnavaz N, Terzopoulos D (2022) Image segmentation using deep learning: a survey. IEEE Trans Pattern Anal Mach Intell 44:1. https://doi.org/10.1109/TPAMI.2021.3059968

12. Shen D, Wu G, Suk H-I (2017) Deep Learning in Medical Image Analysis. Annu Rev Biomed Eng 1:221–248. https://doi.org/10.1146/annurev-bioeng-071516-044442

13. Osher S, Sethian JA (1988) Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. J Comput Phys 79:12–49. https://doi.org/10.1016/0021-9991(88)90002-2

14. Chen JS, Basava RR, Zhang Y, Csapo R, Malis V, Sinha U, Hodgson J, Sinha S (2016) Pixel-based meshfree modelling of skeletal muscles. Comput Methods Biomech Biomed Eng Imaging Vis 4(2):73–85. https://doi.org/10.1080/21681163.2015.1049712

15. Srinivasa Reddy B, Chatterji BN (1996) An FFT-based technique for translation, rotation, and scale-invariant image registration. IEEE Trans Image Process 5(8):1266–1271. https://doi.org/10.1109/83.506761

16. Wettimuny R, Penumadu D (2004) Application of fourier analysis to digital imaging for particle shape analysis. J Comput Civ Eng 18(1):2–9. https://doi.org/10.1061/(ASCE)0887-3801(2004)18:1(2)

17. Cortes C, Vapnik V, Saitta L (1995) Support-vector networks. Mach Learning 20:273–297

18. Vapnik V (1995) The nature of statistical learning theory, 2nd ed. Springer, New York

19. Cervantes J, Garcia-Lamont F, Rodríguez-Mazahua L, Lopez A (2020) A comprehensive survey on support vector machine classification: applications, challenges and trends. Neurocomputing 408:189–215. https://doi.org/10.1016/J.NEUCOM.2019.10.118

20. Osuna E, Freund R, Girosit F (1997) Training support vector machines: an application to face detection. In: Proceedings of IEEE computer society conference on computer vision and pattern recognition, pp 130–136. https://doi.org/10.1109/CVPR.1997.609310.

21. Lee Y-J, Mangasarian OL (2001) RSVM: reduced support vector machines. In: Proceedings of the 2001 SIAM international conference on data mining (SDM), pp 1–17. https://doi.org/10.1137/1.9781611972719.13.

22. Suykens JAK, Vandewalle J (1999) Least squares support vector machine classifiers. Neural Process Lett 9:293–300. https://doi.org/10.1023/A:1018628609742

23. Mangasarian OL, Musicant DR (2001) Lagrangian support vector machines. J Mach Learn Res 1:161–177

24. Hsu C-W, Lin C-J (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Networks 13(2):415. https://doi.org/10.1109/72.991427

25. Babuska I (1970) The finite element method for elliptic equations with discontinuous coefficients. Computing 5:207–218. https://doi.org/10.1007/BF02248021

26. Parvizian J, Düster A, Rank E (2007) Finite cell method : h-and p-extension for embedded domain problems in solid mechanics. Comput Mech 41(1):121–133. https://doi.org/10.1007/s00466-007-0173-y

27. Korshunova N, Jomo J, Lékó G, Reznik D, Balázs P, Kollmannsberger S (2020) Image-based material characterization of complex microarchitectured additively manufactured structures. Comput Math with Appl 80(11):2462–2480. https://doi.org/10.1016/J.CAMWA.2020.07.018

28. Korshunova N, Alaimo G, Hosseini SB, Carraturo M, Reali A, Niiranen J, Auricchio F, Rank E, Kollmannsberger S (2021) Image-based numerical characterization and experimental validation of tensile behavior of octet-truss lattice structures. Addit Manuf 41:101949. https://doi.org/10.1016/j.addma.2021.101949

29. Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. Int J Numer Methods Eng 37(2):229–256. https://doi.org/10.1002/nme.1620370205

30. Liu WK, Jun S, Zhang YF (1995) Reproducing kernel particle methods. Int J Numer Methods Fluids 20(8–9):1081–1106. https://doi.org/10.1002/FLD.1650200824

31. Liu WK, Jun S, Li S, Adee J, Belytschko T (1995) Reproducing kernel particle methods for structural dynamics. Int J Numer Methods Eng 38(10):1655–1679. https://doi.org/10.1002/NME.1620381005

32. Chen JS, Pan C, Wu CT, Liu WK (1996) Reproducing kernel particle methods for large deformation analysis of non-linear structures. Comput Methods Appl Mech Eng 139(1–4):195–227. https://doi.org/10.1016/S0045-7825(96)01083-3

33. Krongauz Y, Belytschko T (1998) EFG approximation with discontinuous derivatives. Int J Numer Methods Eng 41(7):1215–1233. https://doi.org/10.1002/(sici)1097-0207(19980415)41:7%3c1215::aid-nme330%3e3.0.co;2-%23

34. Fries TP, Belytschko T (2010) The extended/generalized finite element method: an overview of the method and its applications. Int J Numer Methods Eng 84(3):253–304. https://doi.org/10.1002/NME.2914

35. Jirásek M (2000) Comparative study on finite elements with embedded discontinuities. Comput Methods Appl Mech Eng 188(1):307–330. https://doi.org/10.1016/S0045-7825(99)00154-1

36. Chen JS, Kotta V, Lu H, Wang D, Moldovan D, Wolf D (2004) A variational formulation and a double-grid method for meso-scale modeling of stressed grain growth in polycrystalline materials. Comput Methods Appl Mech Eng 193(12–14):1277–1303. https://doi.org/10.1016/j.cma.2003.12.020

37. Masuda S, Noguchi H (2006) Analysis of structure with material interface by Meshfree method. Comput Model Eng Sci 11(3):131–144. https://doi.org/10.3970/CMES.2006.011.131

38. Cordes LW, Moran B (1996) Treatment of material discontinuity in the Element-Free Galerkin method. Comput Methods Appl Mech Eng 139(1–4):75–89. https://doi.org/10.1016/S0045-7825(96)01080-8

39. Cockburn B, Karniadakis GE, Shu CW (2000) The development of discontinuous Galerkin methods. In: Cockburn B, Karniadakis GE, Shu CW (eds) Discontinuous Galerkin methods. Lecture notes in computational science and engineering, vol 11. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-59721-3_1

40. Zienkiewicz OC, Taylor RL, Sherwin SJ, Peiró J (2003) On discontinuous Galerkin methods. Int J Numer Methods Eng 58(8):1119–1148. https://doi.org/10.1002/NME.884

41. Wang D, Sun Y, Li L (2009) A discontinuous Galerkin Meshfree modeling of material interface. Comput Model Eng Sci 45(1):57–82

42. Huang T-H, Chen J-S, Tupek MR, Beckwith FN, Koester JJ, Eliot Fang H (2021) A variational multiscale immersed meshfree method for heterogeneous materials. Comput Mech 67:1059–1097. https://doi.org/10.1007/s00466-020-01968-1

43. Wang J, Zhou G, Hillman M, Madra A, Bazilevs Y, Du J, Su K (2021) Consistent immersed volumetric Nitsche methods for composite analysis. Comput Methods Appl Mech Eng 385:1142. https://doi.org/10.1016/j.cma.2021.114042

44. Chen J-S, Hillman M, Chi W (2017) Meshfree methods: progress made after 20 years. Am Soc Civ Eng 143(4):1. https://doi.org/10.1061/(ASCE)EM.1943-7889.0001176

45. Chen JS, Pan C, Roque CMOL, Wang HP (1998) A Lagrangian reproducing kernel particle method for metal forming analysis. Comput Mech 22(3):289–307. https://doi.org/10.1007/S004660050361

46. Chen J-S, Han W, You Y, Meng X (2003) A reproducing kernel method with nodal interpolation property. Int J Numer Methods Eng 56:935–960. https://doi.org/10.1002/nme.592

47. Chen J, Liu W, Hillman M, Chi S, Lian Y, Bessa M (2017) Reproducing kernel approximation and discretization. In: Encyclopedia of computational mechanics, 2nd ed. Wiley

48. Joulaian M, Hubrich S, Düster A (2016) Numerical integration of discontinuities on arbitrary domains based on moment fitting. Comput Mech 57(6):979–999. https://doi.org/10.1007/s00466-016-1273-3

49. Hubrich S, Stolfo PD, Kudela L, Kollmannsberger S, Rank E, Schröder A, Düster A (2017) Numerical integration of discontinuous functions: moment fitting and smart octree. Comput Mech 60(5):863–881. https://doi.org/10.1007/s00466-017-1441-0

50. Kudela L, Zander N, Kollmannsberger S, Rank E (2016) Smart octrees: accurately integrating discontinuous functions in 3D. Comput Methods Appl Mech Eng 306:406–426. https://doi.org/10.1016/J.CMA.2016.04.006

51. Chen JS, Wu CT, Yoon S, You Y (2001) Stabilized conforming nodal integration for Galerkin mesh-free methods. Int J Numer Methods Eng 50(2):435–466. https://doi.org/10.1002/1097-0207(20010120)50:2%3c435::AID-NME32%3e3.0.CO;2-A

52. Yoo JW, Moran B, Chen JS (2004) Stabilized conforming nodal integration in the natural-element method. Int J Numer Methods Eng 60(5):861–890. https://doi.org/10.1002/NME.972

53. Chen JS, Yoon S, Wu CT (2002) Non-linear version of stabilized conforming nodal integration for Galerkin mesh-free methods. Int J Numer Methods Eng 53(12):2587–2615. https://doi.org/10.1002/NME.338

54. Chen JS, Hu W, Puso M, Wu Y, Zhang X (2007) Strain Smoothing for Stabilization and Regularization of Galerkin Meshfree Methods. Lecture Notes in Computational Science and Engineering. Springer, Berlin, pp 57–75. https://doi.org/10.1007/978-3-540-46222-4_4

55. Guan PC, Chi SW, Chen JS, Slawson TR, Roth MJ (2011) Semi-Lagrangian reproducing kernel particle method for fragment-impact problems. Int J Impact Eng 38(12):1033–1047. https://doi.org/10.1016/j.ijimpeng.2011.08.001

56. Hillman M, Chen J-S (2016) An accelerated, convergent, and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics. Int J Numer Methods Eng 107:603–630. https://doi.org/10.1002/nme.5183

57. Chen J-S, Hillman M, Rüter M (2013) An arbitrary order variationally consistent integration for Galerkin meshfree methods. Int J Numer Methods Eng 95:387–418. https://doi.org/10.1002/nme.4512

58. Beissel S, Belytschko T (1996) Nodal integration of the element-free Galerkin method. Comput Methods Appl Mech Eng 139(1–4):49–74. https://doi.org/10.1016/S0045-7825(96)01079-1

59. Liu GR, Zhang GY, Wang YY, Zhong ZH, Li GY, Han X (2007) A nodal integration technique for meshfree radial point interpolation method (NI-RPIM). Int J Solids Struct 44(11–12):3840–3860. https://doi.org/10.1016/J.IJSOLSTR.2006.10.025

60. Puso MA, Chen JS, Zywicz E, Elmer W (2008) Meshfree and finite element nodal integration methods. Int J Numer Methods Eng 74(3):416–446. https://doi.org/10.1002/NME.2181

61. Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Proceedings of the fifth annual workshop on computational learning theory, pp 144–152. https://doi.org/10.1145/130385.130401

62. Kuhn HW, Tucker AW (2013) Nonlinear Programming. Traces Emerg Nonlinear Program, pp 247–258. https://doi.org/10.1007/978-3-0348-0439-4_11.

63. Burges CJC (1998) A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 2:121–167. https://doi.org/10.1023/A:1009715923555

64. Noble WS (2006) What is a support vector machine? Nat Biotechnol 24:1565–1567. https://doi.org/10.1038/nbt1206-1565

65. Vásárhelyi L, Kónya Z, Kukovecz RV (2020) Microcomputed tomography–based characterization of advanced materials: a review. Mater Today Adv 8:1. https://doi.org/10.1016/J.MTADV.2020.100084

66. Otsu N (1979) A threshhold selection method from gray level histograms. IEEE Trans Syst Man Cybern C(1):62–66

67. Vernon-Parry KD (2000) Scanning electron microscopy: an introduction. III-Vs Rev 13(4):40–44. https://doi.org/10.1016/S0961-1290(00)80006-X

68. Shindo D, Oikawa T (2002) Energy dispersive X-ray spectroscopy. In: Analytical electron microscopy for materials science. Springer, Tokyo. https://doi.org/10.1007/978-4-431-66988-3_4

69. Roth MJ, Chen J-S, Danielson KT, Slawson TR (2016) Hydrodynamic meshfree method for high-rate solid dynamics using a Rankine-Hugoniot enhancement in a Riemann-SCNI framework. Int J Numer Methods Eng 108(12):1525–1549. https://doi.org/10.1002/nme.5266

70. Fernández-Méndez S, Huerta A (2004) Imposing essential boundary conditions in mesh-free methods. Comput Methods Appl Mech Eng 193(12–14):1257–1275. https://doi.org/10.1016/J.CMA.2003.12.019

71. Mura T (2013) Micromechanics of Defects in Solids. Springer, Berlin
72. Ansys® Workbench, Release 2023 R1, Help System, Workbench User's Guide, ANSYS, Inc.