

UC San Diego

UC San Diego Previously Published Works

Title

A quasi-conforming embedded reproducing kernel particle method for heterogeneous materials

Permalink

<https://escholarship.org/uc/item/8h31w13j>

Authors

Schlinkman, Ryan T
Baek, Jonghyuk
Beckwith, Frank N
et al.

Publication Date

2023-11-01

DOI

10.1016/j.cma.2023.116363

Copyright Information

This work is made available under the terms of a Creative Commons Attribution License, available at <https://creativecommons.org/licenses/by/4.0/>

Peer reviewed



A quasi-conforming embedded reproducing kernel particle method for heterogeneous materials

Ryan T. Schlinkman^a, Jonghyuk Baek^a, Frank N. Beckwith^b, Stacy M. Nelson^b, J.S. Chen^{a,*}

^a *University of California San Diego, La Jolla, CA, United States of America*

^b *Sandia National Laboratories, Albuquerque, NM, United States of America*

Received 23 March 2023; received in revised form 30 June 2023; accepted 8 August 2023

Available online xxxx

Abstract

We present a quasi-conforming embedded reproducing kernel particle method (QCE-RKPM) for modeling heterogeneous materials that makes use of techniques not available to mesh-based methods such as the finite element method (FEM) and avoids many of the drawbacks in current embedded and immersed formulations which are based on meshed methods. The different material domains are discretized independently thus avoiding time-consuming, conformal meshing. In this approach, the superposition of foreground (inclusion) and background (matrix) domain integration smoothing cells are corrected by a quasi-conforming quadtree subdivision on the background integration smoothing cells. Due to the non-conforming nature of the background integration smoothing cells near the material interfaces, a variationally consistent (VC) correction for domain integration is introduced to restore integration constraints and thus optimal convergence rates at a minor computational cost. Additional interface integration smoothing cells with area (volume) correction, while non-conforming, can be easily introduced to further enhance the accuracy and stability of the Galerkin solution using VC integration on non-conforming cells. To properly approximate the weak discontinuity across the material interface by a penalty-free Nitsche's method with enhanced coercivity, the interface nodes on the surface of the foreground discretization are also shared with the background discretization. As such, there are no tunable parameters, such as those involved in the penalty type method, to enforce interface compatibility in this approach. The advantage of this meshfree formulation is that it avoids many of the instabilities in mesh-based immersed and embedded methods. The effectiveness of QCE-RKPM is illustrated with several examples.

© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

Keywords: Meshfree modeling; Reproducing kernel particle method; Heterogeneous materials; Embedded discretization; Quasi-conforming; Quadtree subdivision

1. Introduction

A vast array of engineering problems involving heterogeneous materials have been modeled by the finite element method (FEM) with a body-fitted mesh in which the discretizations of two different materials conform to their interface allowing for gradient jumps across the interface. Constructing such meshes, however, can be quite time-consuming and complicated, especially for complex, three-dimensional geometry. One way to reduce meshing

* Corresponding author.

E-mail address: js-chen@eng.ucsd.edu (J.S. Chen).

time for heterogeneous materials is to use embedded or immersed methods, both of which allow one to create independent, typically uniform discretizations of the different materials and which can sort out the communication between the meshes during the analysis. However, this reduction in manual meshing time comes at the cost of the algorithmic complexity in enforcing the constraint between the two materials while maintaining stability and accuracy.

“Immersed methods” refer to those methods in which the weak form equations assume overlapping meshes and, therefore, a fictitious background domain is placed underneath the foreground domain. These may be classified by the constraint type with 1) interface-constrained immersed methods (ICIMs) enforcing compatibility only on the interface and 2) volume-constrained immersed methods (VCIMs) constraining the background deformation in the entire overlapping region to the foreground motion. The first ICIM devised was the immersed boundary method (IBM) proposed by Peskin [1,2] (cf. [3–5] for FEM and hybrid versions) which distributes fluid–structure interaction (FSI) forces and interpolates velocities using a smoothed approximation of the Dirac delta function. However, IBM severely restricts the mesh size ratio between foreground and background to prevent fluid from leaking through the solid boundary [4]. Also, the smooth Dirac delta function smears the solution over several grid cells rather than a sharp discontinuity in the normal gradient at the interface thus limiting IBM to first-order accuracy unless more complicated procedures are added [6,7]. A more recent ICIM formulation introduced by Bazilevs et al. [8,9] uses a uniform background mesh discretized with IGA splines [10] to approximate the momentum equations for both the background and foreground meshes. One issue with this approach is the undesirable smooth background approximation for the discontinuous kinematics in fracture and fragmentation which results in damage zones whose size scales with the background mesh [11].

VCIMs began with Glowinsky et al.’s distributed Lagrange multiplier method [12] which used Lagrange multipliers (LM) requiring the approximation spaces to meet the LBB condition [13,14]. Developments of VCIMs in recent years include the immersed finite element method (IFEM) [15,16] and the modified IFEM (mIFEM) [17] which use the reproducing kernel (RK) approximation [18,19] to interpolate velocities between solid and fluid, and FSI forces are treated as volumetric body forces. However, like all VCIMs, the smooth approximation in the fluid smears the effect of the interface over several elements leading to low convergence rates [20]. The variational multiscale immersed method (VMIM) for heterogeneous materials [21] uses the reproducing kernel particle method (RKPM) with volumetric Lagrange multipliers and a variational multiscale decomposition [22,23] of the background approximation leading to a residual-based stabilization which also suppresses the leaking instability caused by large solid to fluid mesh size ratios [24]. But again, it too suffers from large interpolation errors near the interface resulting in low convergence rates [21]. While VCIMs are effective without a tedious contour integral along the interface, achieving optimal convergence rates requires a direct treatment of the gradient discontinuity at the interface [25]. Furthermore, VCIMs require some sort of stabilization [21,26].

“Embedded methods” refer to those methods in which there are no overlapping meshes, and the conformity of matrix and inclusion domains (which allows for discontinuous normal gradients) is solved using techniques such as cutting elements, aggregating elements, using surrogate domains, etc. Cut element approaches either add extra DOFs to elements intersecting the interface (e.g., CutFEM [27,28]) or divide the original elements into smaller “integration elements” (e.g., discontinuity-enriched FEM (DE-FEM) [29–31]). Nonetheless, cutting or dividing elements requires potentially complex computational geometry algorithms and, more importantly, creates what is called the small cut element problem in which an element can be cut in an arbitrary fashion leaving a very small or slender element on one side [32]. This results in an ill-conditioned stiffness matrix. Several methods have been proposed to mitigate the effect of the cut cell problem such as the ghost penalty [32–34], bubble functions [35,36], carefully weighting the gradients in the Nitsche integrals [37,38], or discarding the problematic DOFs altogether [31]. The Finite Cell Method (FCM) [39–42] utilizes an adaptive integration scheme which sub-divides elements into sub-cells (not sub-elements) to place quadrature points thus allowing one to better approximate the boundary/interface and avoiding the problem of small elements. However, one must divide elements many times to accurately approximate the domain being integrated which yields a computationally intensive domain integration [41].

The shifted boundary/interface method (SBM/SIM) [43–45] integrates over a surrogate domain (e.g., all elements wholly inside the interface) and shifts the interface conditions from the true interface to the surrogate interface using a Taylor series expansion involving the intersected elements’ gradients; this allows the method to maintain optimal rates of convergence without cutting elements (cf. [46,47] for a similar idea). However, the transfer algorithm involves a potentially complex closest point projection procedure, and the transfer of Neumann boundary conditions

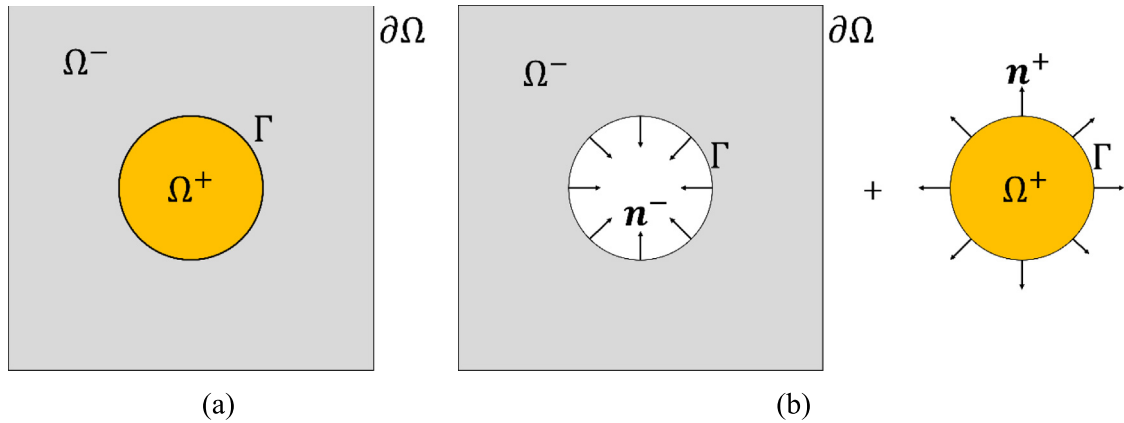


Fig. 1. (a) Heterogeneous material domain consisting of matrix Ω^- and inclusion Ω^+ materials with outer boundary $\partial\Omega$ and interface Γ between the two materials and (b) separate matrix and inclusion domains with their outward normal vectors on the interface.

requires either a higher-order approximation [48] or a stabilized equal-order mixed method [49] to prevent reduction in convergence rates. In the aggregated finite element method (AgFEM) [50–52], elements intersected by the interface are merged with elements inside the interface, and the problematic DOFs are constrained as a linear combination of non-problematic DOFs (cf. similar approaches in [53–56]). This avoids small cut elements and achieves optimal convergence rates but can possibly introduce error in the aggregation and constraint process.

In this paper, we propose a formulation that fits into the category of embedded methods but is unique in that it makes use of techniques available exclusively to meshfree methods and avoids many of the pitfalls of the formulations outlined above due to their reliance on mesh-based methods. First, the background (matrix) and foreground (inclusions) are discretized independently with superposition. Then what follows is a quasi-conforming quadtree subdivision on the background integration smoothing cells where needed (which mesh-based methods are incapable of doing). Due to the loss of integration smoothing cell conformity, a variationally consistent (VC) correction [57] of domain integration is employed at a minimal computational cost to restore the integration constraint for Galerkin exactness, enhance accuracy, and achieve optimal convergence rates. The interface nodes are shared by both domains which provides enough coercivity to make Nitsche’s penalty parameter unnecessary. Thus, unlike most immersed and embedded methods, this formulation does not have any tunable parameters present to enforce interface compatibility. Our meshfree method of choice is the reproducing kernel particle method (RKPM), and so, we call this formulation a quasi-conforming embedded RKPM (QCE-RKPM).

The remaining parts of this paper are outlined as follows. In Section 2, we detail the equations for heterogeneous materials. Section 3 reviews the reproducing kernel approximation and discusses domain integration in meshfree methods. Section 4 details the discretization process of the background domain from an immersed state to a quasi-conforming embedded domain. Section 5 then presents numerical examples to demonstrate the effectiveness of our method, and Section 6 summarizes our findings.

2. Governing equations

2.1. Strong form

Consider a heterogeneous solid domain Ω comprised of a matrix domain Ω^- and an inclusion domain Ω^+ , where $\overline{\Omega} = \overline{\Omega^+} \cup \overline{\Omega^-}$ and $\overline{\cdot}$ represents the closure of the domain. The interface between the two materials is denoted by Γ , where $\Gamma = \overline{\Omega^+} \cap \overline{\Omega^-}$. The matrix material possesses an outer boundary $\partial\Omega$ which is comprised of Neumann $\partial\Omega_N$ and Dirichlet $\partial\Omega_D$ components such that $\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$ and $\partial\Omega_N \cap \partial\Omega_D = \emptyset$, see Fig. 1(a) for an illustration.

Our governing equation is the equation of equilibrium for a linear elastic material:

$$\nabla \cdot \boldsymbol{\sigma}(\mathbf{x}) + \mathbf{b}(\mathbf{x}) = \mathbf{0}, \quad \mathbf{x} \in \Omega, \tag{1}$$

$$\boldsymbol{\sigma}(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = \mathbf{t}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega_N \tag{2}$$

$$\mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), \quad \mathbf{x} \in \partial\Omega_D, \tag{3}$$

$$\boldsymbol{\sigma}(\mathbf{x}) = \mathbf{C}(\mathbf{x}) : \boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x})), \quad \mathbf{x} \in \Omega, \tag{4}$$

where:

$$\mathbf{b}(\mathbf{x}) = \begin{cases} \mathbf{b}^-, & \mathbf{x} \in \Omega^- \\ \mathbf{b}^+, & \mathbf{x} \in \Omega^+ \end{cases} \tag{5}$$

$$\mathbf{u}(\mathbf{x}) = \begin{cases} \mathbf{u}^-, & \mathbf{x} \in \Omega^- \\ \mathbf{u}^+, & \mathbf{x} \in \Omega^+ \end{cases} \tag{6}$$

$$\mathbf{C}(\mathbf{x}) = \begin{cases} \mathbf{C}^-, & \mathbf{x} \in \Omega^- \\ \mathbf{C}^+, & \mathbf{x} \in \Omega^+ \end{cases} \tag{7}$$

hence:

$$\boldsymbol{\sigma}(\mathbf{x}) = \begin{cases} \boldsymbol{\sigma}^-(\mathbf{u}^-) = \mathbf{C}^- : \boldsymbol{\varepsilon}(\mathbf{u}^-), & \mathbf{x} \in \Omega^- \\ \boldsymbol{\sigma}^+(\mathbf{u}^+) = \mathbf{C}^+ : \boldsymbol{\varepsilon}(\mathbf{u}^+), & \mathbf{x} \in \Omega^+ \end{cases} \tag{8}$$

where $\boldsymbol{\sigma}(\mathbf{x})$ is the Cauchy stress tensor, $\mathbf{b}(\mathbf{x})$ is the body force vector, $\mathbf{n}(\mathbf{x})$ is the outward normal vector on $\partial\Omega_N$, $\mathbf{t}(\mathbf{x})$ is the prescribed traction vector on $\partial\Omega_N$, $\mathbf{u}(\mathbf{x})$ is the displacement vector, $\mathbf{g}(\mathbf{x})$ is the prescribed displacement vector on $\partial\Omega_D$, $\mathbf{C}(\mathbf{x})$ is the elastic constitutive tensor, and $\boldsymbol{\varepsilon}(\mathbf{u}(\mathbf{x})) = \frac{1}{2}(\nabla \otimes \mathbf{u}(\mathbf{x}) + \mathbf{u}(\mathbf{x}) \otimes \nabla)$ is the small strain tensor. Additionally, continuity and equilibrium conditions on the interface Γ are:

$$\mathbf{u}^+ - \mathbf{u}^- = \mathbf{0}, \quad \forall \mathbf{x} \in \Gamma, \tag{9}$$

$$\boldsymbol{\sigma}^+(\mathbf{u}^+) \cdot \mathbf{n}^+ + \boldsymbol{\sigma}^-(\mathbf{u}^-) \cdot \mathbf{n}^- = \mathbf{0}, \quad \forall \mathbf{x} \in \Gamma, \tag{10}$$

where \mathbf{n}^+ and \mathbf{n}^- are the outward normal vectors on Γ from the inclusion and matrix sides, respectively, as illustrated in Fig. 1(b).

2.2. Weak form

To begin deriving the weak form of Eqs. (1) through (4), we multiply Eq. (1) by a test function $\delta \mathbf{u}(\mathbf{x})$ and integrate over the domain Ω :

$$(\delta \mathbf{u}(\mathbf{x}), \nabla \cdot \boldsymbol{\sigma}(\mathbf{x}))_{\Omega} + (\delta \mathbf{u}(\mathbf{x}), \mathbf{b}(\mathbf{x}))_{\Omega} = \mathbf{0}, \quad \mathbf{x} \in \Omega, \tag{11}$$

where $(\cdot, \cdot)_{\Omega}$ is the inner product in Ω , and

$$\delta \mathbf{u}(\mathbf{x}) = \begin{cases} \delta \mathbf{u}^-, & \mathbf{x} \in \Omega^- \\ \delta \mathbf{u}^+, & \mathbf{x} \in \Omega^+ \end{cases} \tag{12}$$

Given the definitions in Eqs. (5) through (8) and (12), Eq. (11) may be split by domain:

$$(\delta \mathbf{u}^-, \nabla \cdot \boldsymbol{\sigma}^-(\mathbf{u}^-))_{\Omega^-} + (\delta \mathbf{u}^-, \mathbf{b}^-)_{\Omega^-} = \mathbf{0}, \tag{13}$$

$$(\delta \mathbf{u}^+, \nabla \cdot \boldsymbol{\sigma}^+(\mathbf{u}^+))_{\Omega^+} + (\delta \mathbf{u}^+, \mathbf{b}^+)_{\Omega^+} = \mathbf{0}. \tag{14}$$

Performing integration by parts on the terms containing the stress divergences and using Gauss's theorem, we have:

$$(\boldsymbol{\varepsilon}(\delta \mathbf{u}^-), \boldsymbol{\sigma}^-(\mathbf{u}^-))_{\Omega^-} - (\delta \mathbf{u}^-, \mathbf{n}^- \cdot \boldsymbol{\sigma}^-(\mathbf{u}^-))_{\Gamma} = (\delta \mathbf{u}^-, \mathbf{b}^-)_{\Omega^-} + (\delta \mathbf{u}^-, \mathbf{t})_{\partial\Omega_N}, \tag{15}$$

$$(\boldsymbol{\varepsilon}(\delta \mathbf{u}^+), \boldsymbol{\sigma}^+(\mathbf{u}^+))_{\Omega^+} - (\delta \mathbf{u}^+, \mathbf{n}^+ \cdot \boldsymbol{\sigma}^+(\mathbf{u}^+))_{\Gamma} = (\delta \mathbf{u}^+, \mathbf{b}^+)_{\Omega^+}. \tag{16}$$

Note that by substituting $\mathbf{n}^- = -\mathbf{n}^+$ into interface equilibrium (10), we have:

$$\boldsymbol{\sigma}^+(\mathbf{u}^+) \cdot \mathbf{n}^+ = \boldsymbol{\sigma}^-(\mathbf{u}^-) \cdot \mathbf{n}^+. \tag{17}$$

This implies that $\sigma^+(u^+)$, $\sigma^-(u^-)$, or a linear combination of the two may be used interchangeably in the interface traction terms. Define a linear combination of stresses on the interface as:

$$\bar{\sigma}(u) \equiv \alpha \sigma^+(u^+) + (1 - \alpha) \sigma^-(u^-) \tag{18}$$

where $\alpha \in [0, 1]$. We can rearrange (18) as $\bar{\sigma}(u) = \sigma^-(u^-) + \alpha(\sigma^+(u^+) - \sigma^-(u^-))$, and using (17), we have:

$$\bar{\sigma}(u) \cdot n^+ = \sigma^+(u^+) \cdot n^+ = -\sigma^-(u^-) \cdot n^-. \tag{19}$$

Substituting (19) into (15) and (16), we have:

$$(\epsilon(\delta u^-), \sigma^-(u^-))_{\Omega^-} + (\delta u^-, n^+ \cdot \bar{\sigma}(u))_{\Gamma} = (\delta u^-, b^-)_{\Omega^-} + (\delta u^-, t)_{\partial \Omega_N}, \tag{20}$$

$$(\epsilon(\delta u^+), \sigma^+(u^+))_{\Omega^+} - (\delta u^+, n^+ \cdot \bar{\sigma}(u))_{\Gamma} = (\delta u^+, b^+)_{\Omega^+}, \tag{21}$$

or:

$$\begin{aligned} & (\epsilon(\delta u^-), \sigma^-(u^-))_{\Omega^-} + \alpha(\delta u^-, n^+ \cdot \sigma^+(u^+))_{\Gamma} + (1 - \alpha)(\delta u^-, n^+ \cdot \sigma^-(u^-))_{\Gamma} \\ & = (\delta u^-, b^-)_{\Omega^-} + (\delta u^-, t)_{\partial \Omega_N}, \end{aligned} \tag{22}$$

$$\begin{aligned} & (\epsilon(\delta u^+), \sigma^+(u^+))_{\Omega^+} - \alpha(\delta u^+, n^+ \cdot \sigma^+(u^+))_{\Gamma} - (1 - \alpha)(\delta u^+, n^+ \cdot \sigma^-(u^-))_{\Gamma} \\ & = (\delta u^+, b^+)_{\Omega^+}, \end{aligned} \tag{23}$$

Since meshfree shape functions do not possess the Kronecker delta property, we use Nitsche’s method to enforce Dirichlet boundary conditions. Our weak form reads: Find $(u^+, u^-) \in H^1(\Omega^+) \times H^1(\Omega^-)$ such that $\forall (\delta u^+, \delta u^-) \in H^1(\Omega^+) \times H^1(\Omega^-)$

$$\begin{aligned} & (\epsilon(\delta u^-), \sigma^-(u^-))_{\Omega^-} + \alpha(\delta u^-, n^+ \cdot \sigma^+(u^+))_{\Gamma} + (1 - \alpha)(\delta u^-, n^+ \cdot \sigma^-(u^-))_{\Gamma} \\ & - (\delta u^-, n^- \cdot \sigma^-(u^-))_{\partial \Omega_D} - (\sigma^-(\delta u^-) \cdot n^-, u^-)_{\partial \Omega_D} + \beta(\delta u^-, u^-)_{\partial \Omega_D} \end{aligned} \tag{24}$$

$$\begin{aligned} & = (\delta u^-, b^-)_{\Omega^-} + (\delta u^-, t)_{\partial \Omega_N} - (\sigma^-(\delta u^-) \cdot n^-, g)_{\partial \Omega_D} + \beta(\delta u^-, g)_{\partial \Omega_D}, \\ & (\epsilon(\delta u^+), \sigma^+(u^+))_{\Omega^+} - \alpha(\delta u^+, n^+ \cdot \sigma^+(u^+))_{\Gamma} - (1 - \alpha)(\delta u^+, n^+ \cdot \sigma^-(u^-))_{\Gamma} \\ & = (\delta u^+, b^+)_{\Omega^+}, \end{aligned} \tag{25}$$

where β is the penalty parameter.

Remark 2.1. Alternatively, one could derive a symmetric weak form from the following potential:

$$\begin{aligned} \Pi & = \frac{1}{2}(\epsilon(u^-), C^- : \epsilon(u^-))_{\Omega^-} + \frac{1}{2}(\epsilon(u^+), C^+ : \epsilon(u^+))_{\Omega^+} - (u^-, b^-)_{\Omega^-} \\ & - (u^-, t)_{\partial \Omega_N} - (u^+, b^+)_{\Omega^+} - (\bar{\sigma}(u) \cdot n^+, u^+ - u^-)_{\Gamma} \\ & - (\sigma^-(u^-) \cdot n^-, u^- - g)_{\partial \Omega_D} + \frac{\beta}{2}(u^- - g, u^- - g)_{\partial \Omega_D}. \end{aligned} \tag{26}$$

Taking the variation of the potential, setting it equal to zero, splitting the equation in two by test function, and rearranging, we have:

$$\begin{aligned} & (\epsilon(\delta u^-), \sigma^-(u^-))_{\Omega^-} + \alpha(\delta u^-, n^+ \cdot \sigma^+(u^+))_{\Gamma} + (1 - \alpha)(\delta u^-, n^+ \cdot \sigma^-(u^-))_{\Gamma} \\ & + (1 - \alpha)(\sigma^-(\delta u^-) \cdot n^+, u^- - u^+)_{\Gamma} - (\delta u^-, n^- \cdot \sigma^-(u^-))_{\partial \Omega_D} \\ & - (\sigma^-(\delta u^-) \cdot n^-, u^-)_{\partial \Omega_D} + \beta(\delta u^-, u^-)_{\partial \Omega_D} \end{aligned} \tag{27}$$

$$\begin{aligned} & = (\delta u^-, b^-)_{\Omega^-} + (\delta u^-, t)_{\partial \Omega_N} - (\sigma^-(\delta u^-) \cdot n^-, g)_{\partial \Omega_D} + \beta(\delta u^-, g)_{\partial \Omega_D}, \\ & (\epsilon(\delta u^+), \sigma^+(u^+))_{\Omega^+} - \alpha(\delta u^+, n^+ \cdot \sigma^+(u^+))_{\Gamma} - (1 - \alpha)(\delta u^+, n^+ \cdot \sigma^-(u^-))_{\Gamma} \\ & + \alpha(\sigma^+(\delta u^+) \cdot n^+, u^- - u^+)_{\Gamma} = (\delta u^+, b^+)_{\Omega^+}. \end{aligned} \tag{28}$$

The only difference between this weak form and the one given in Eqs. (24) and (25) is the addition of the interface term $-(\bar{\sigma}(u) \cdot n^+, u^+ - u^-)_{\Gamma}$. This weak form is essentially Nitsche’s method without penalty for enforcing interface compatibility. It has been shown in [58] that an interface penalty term is not needed if the interface nodes are shared by the foreground and background discretizations for enhanced coercivity.

We have found that any value of α in Eqs. (24) and (25) using the proposed method to be discussed in Section 4 will provide an equally good approximation. This is in contrast with FEM cut-element techniques which must carefully balance the interface gradients to avoid matrix ill-conditioning [37,38]. For simplicity, we have chosen to use the inclusion interface stress exclusively for the interface terms (i.e., $\alpha = 1$). We thus arrive at the weak form used for our numerical examples:

$$\begin{aligned} & (\boldsymbol{\varepsilon}(\delta \mathbf{u}^-), \boldsymbol{\sigma}^-(\mathbf{u}^-))_{\Omega^-} + (\delta \mathbf{u}^-, \mathbf{n}^+ \cdot \boldsymbol{\sigma}^+(\mathbf{u}^+))_{\Gamma} - (\delta \mathbf{u}^-, \mathbf{n}^- \cdot \boldsymbol{\sigma}^-(\mathbf{u}^-))_{\partial \Omega_D} \\ & - (\boldsymbol{\sigma}^-(\delta \mathbf{u}^-) \cdot \mathbf{n}^-, \mathbf{u}^-)_{\partial \Omega_D} + \beta (\delta \mathbf{u}^-, \mathbf{u}^-)_{\partial \Omega_D} \end{aligned} \quad (29)$$

$$\begin{aligned} & = (\delta \mathbf{u}^-, \mathbf{b}^-)_{\Omega^-} + (\delta \mathbf{u}^-, \mathbf{t})_{\partial \Omega_N} - (\boldsymbol{\sigma}^-(\delta \mathbf{u}^-) \cdot \mathbf{n}^-, \mathbf{g})_{\partial \Omega_D} + \beta (\delta \mathbf{u}^-, \mathbf{g})_{\partial \Omega_D}, \\ & (\boldsymbol{\varepsilon}(\delta \mathbf{u}^+), \boldsymbol{\sigma}^+(\mathbf{u}^+))_{\Omega^+} - (\delta \mathbf{u}^+, \mathbf{n}^+ \cdot \boldsymbol{\sigma}^+(\mathbf{u}^+))_{\Gamma} = (\delta \mathbf{u}^+, \mathbf{b}^+)_{\Omega^+}. \end{aligned} \quad (30)$$

Remark 2.2. The foreground and background domains use separate approximations and are tied together using the shared interface nodes with the penalty-free Nitsche’s method. This allows the approximation to capture strain discontinuities across the material interface.

Remark 2.3. The use of Nitsche’s method for essential boundary condition enforcement comes at little additional cost when used in conjunction with SCNI (detailed in Section 3.2.1) since the evaluation of shape functions and gradients on the boundary will have already been calculated during the gradient smoothing calculations in domain integration and stored in memory. Thus, no new evaluations are needed for Nitsche’s method since one can simply reuse the shape functions and gradients calculated during the gradient smoothing process.

3. Variationally consistent reproducing kernel particle method

In this section, we detail the reproducing kernel approximation and discuss the stabilized nodal integration suitable for the proposed embedded method. The matrix form may be found in the appendix.

3.1. Reproducing kernel approximation

The reproducing kernel (RK) approximation [18,19] is constructed based on a set of NP scattered nodes that discretize the domain Ω as shown in Fig. 2.

Let $u^h(\mathbf{x})$ be the approximation of $u(\mathbf{x})$; the RK approximation is given as follows [19]:

$$u^h(\mathbf{x}) = \sum_{I=1}^{NP} \Psi_I(\mathbf{x}) u_I, \quad (31)$$

where u_I is the generalized coefficient associated with node I , and $\Psi_I(\mathbf{x})$ is the RK approximation function associated with node I expressed as:

$$\Psi_I(\mathbf{x}) = \mathbf{H}^T(\mathbf{0}) \mathbf{M}^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \phi_a(\mathbf{x} - \mathbf{x}_I), \quad (32)$$

where $\phi_a(\mathbf{x} - \mathbf{x}_I)$ is the kernel function that defines the locality and smoothness (i.e., the order of continuity) of the approximation. A common kernel function used in RK approximations is the cubic B-spline kernel which is C^2 continuous:

$$\phi_a(z) = \begin{cases} \frac{2}{3} - 4z_I^2 + 4z_I^3, & 0 \leq z_I \leq \frac{1}{2} \\ \frac{4}{3} (1 - z_I)^3, & \frac{1}{2} < z_I \leq 1, \\ 0, & z_I > 1 \end{cases}, \quad z_I = \frac{\|\mathbf{x} - \mathbf{x}_I\|}{a_I}, \quad (33)$$

where a_I is the support size of node I and $\mathbf{M}(\mathbf{x})$ is the moment matrix:

$$\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{NP} \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) \phi_a(\mathbf{x} - \mathbf{x}_I), \quad (34)$$

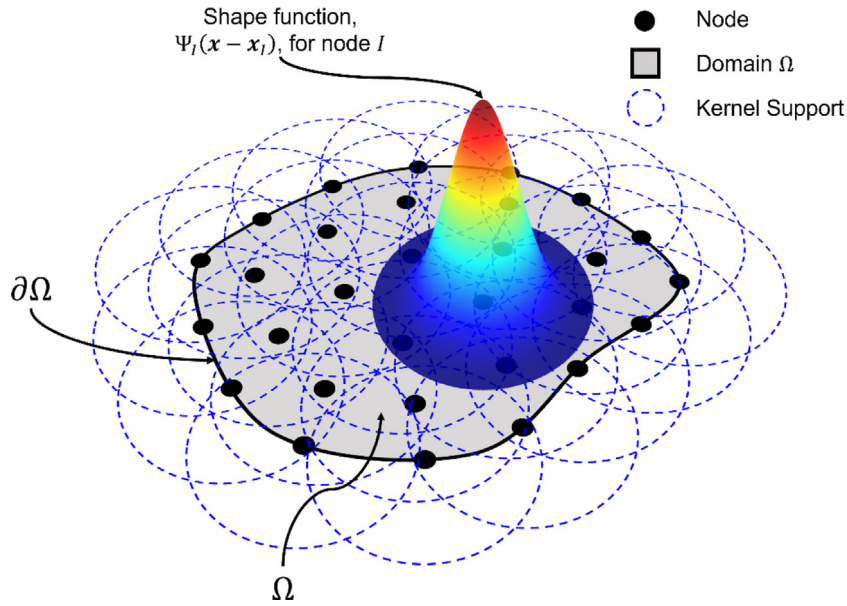


Fig. 2. A domain Ω with boundary $\partial\Omega$ is discretized by a set of nodes. Each node possesses a circular kernel support where the shape function Ψ_I is defined.

where $\mathbf{H}(\mathbf{x} - \mathbf{x}_I)$ is the basis vector of monomials:

$$\mathbf{H}^T(\mathbf{x} - \mathbf{x}_I) = [1, x_1 - x_{1I}, x_2 - x_{2I}, x_3 - x_{3I}, (x_1 - x_{1I})^2, \dots, (x_3 - x_{3I})^n]. \quad (35)$$

Here $\mathbf{H}(\mathbf{x} - \mathbf{x}_I)$ of basis order n enforces the n th order of completeness in the approximation:

$$\sum_{I=1}^{NP} \Psi_I(\mathbf{x}) x_{1I}^i x_{2I}^j x_{3I}^k = x_1^i x_2^j x_3^k, \quad 0 \leq i + j + k \leq n. \quad (36)$$

For the moment matrix to be invertible, the point being evaluated \mathbf{x} must be covered by at least the number of non-collinear (in 2D) or non-coplanar (in 3D) kernel supports equal to the dimension of the basis vector [59].

Remark 3.1.

1. When using meshfree methods in the presence of concave features, which commonly exist in microstructures, it is necessary to use a line-of-sight algorithm to determine a node’s influence on an evaluation point [60,61]. The fact that a node’s kernel support covers an evaluation point is not sufficient to determine influence; the node also needs to have a line of sight within the material body to the evaluation point. For example, consider Fig. 3. Node 1’s kernel support covers nodes 2, 3, and 4 along with other nodes in the domain. Node 1 has line-of-sight within the domain to node 2, and since the boundary of the domain connects nodes 1 and 3 with a straight line, node 1 also has line-of-sight to node 3. While node 4 is within node 1’s kernel support, a straight line drawn from node 1 to node 4 would pass outside the domain, and thus node 1’s kernel will have a value of zero when evaluated at node 4.
2. After doing a runtime study, the line-of-sight algorithm takes only about 10% of the total runtime when run in serial. Parallelization of this subroutine would lower that figure considerably.
3. The line-of-sight algorithm creates a discontinuous kernel function, which leads to an undesired discontinuity in the approximation. Another approach to handle concave features while maintaining continuity in the approximation is to use a diffracted kernel support [61–63] which bends around the corner, e.g., at node 3 in Fig. 3. In this work, we consider the line-of-sight approach for simplicity.

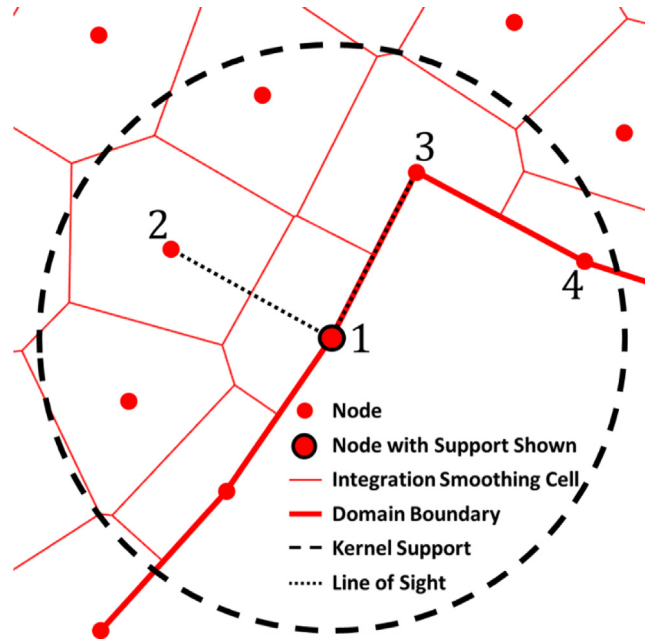


Fig. 3. Example of line-of-sight.

3.2. Domain integration

3.2.1. Stabilized conforming/non-conforming nodal integration

The Gauss quadrature rules commonly used in the finite element method are not suitable for meshfree methods due to the absence of a mesh in a meshfree discretization. While “background” meshes, independent of the discretization points, have been used to define Gauss quadratures, it is almost impossible to match the mesh with the supports of the RK kernel functions as shown in Fig. 3, leading to integration errors and hence reduced convergence rates in meshfree methods [57,59,64,65]. Higher-order quadratures have also been considered. For example, use of the cubic B-spline kernel in (33) in combination with a linear basis would require at least a 5th order Gauss quadrature rule for sufficient accuracy in common practice, which is computationally intractable [57,59,66,67]. Alternatively, direct nodal integration (DNI) which locates an integration point at the node is natural to meshfree methods, but it under-integrates the strain energy and yields spurious zero energy modes [65].

These problems with conventional integration methods led to the development of stabilized conforming nodal integration (SCNI) [65], where a smoothed gradient is calculated over a conforming nodal “integration smoothing cell” rather than taking a direct derivative at the nodal integration points, see Fig. 4. The following smoothed gradient has been introduced [65]:

$$\tilde{\Psi}_{I,i}(\mathbf{x}_L) = \frac{1}{V_L} \int_{\Omega_L} \Psi_{I,i}(\mathbf{x}) d\Omega = \frac{1}{V_L} \int_{\partial\Omega_L} \Psi_I(\mathbf{x}) n_i(\mathbf{x}) d\Gamma, \tag{37}$$

where $\tilde{\Psi}_{I,i}(\mathbf{x}_L)$ is the smoothed shape function gradient in the i th direction of node I which is constant over the cell of node L located at \mathbf{x}_L , V_L is the nodal cell volume of the integration smoothing cell Ω_L associated with node L , and n_i is the i th component of the outward normal vector on the integration smoothing cell boundary $\partial\Omega_L$; see Fig. 4. It has been shown that SCNI fulfills the integration constraint and meets linear exactness in the Galerkin approximation [57,65]. The conforming integration smoothing cells may be constructed in several ways including Voronoi diagrams, Delaunay triangulations, or from a finite element mesh with the requirement that the cells’ boundaries conform to each other.

To model extreme deformation and fragmentation problems, a non-conforming counterpart to SCNI was developed, the stabilized non-conforming nodal integration (SNNI) [68,69]. In SNNI, the integration smoothing cells are constructed with simple geometries, such as cubes or spheres centered on the nodes, and whose volume is

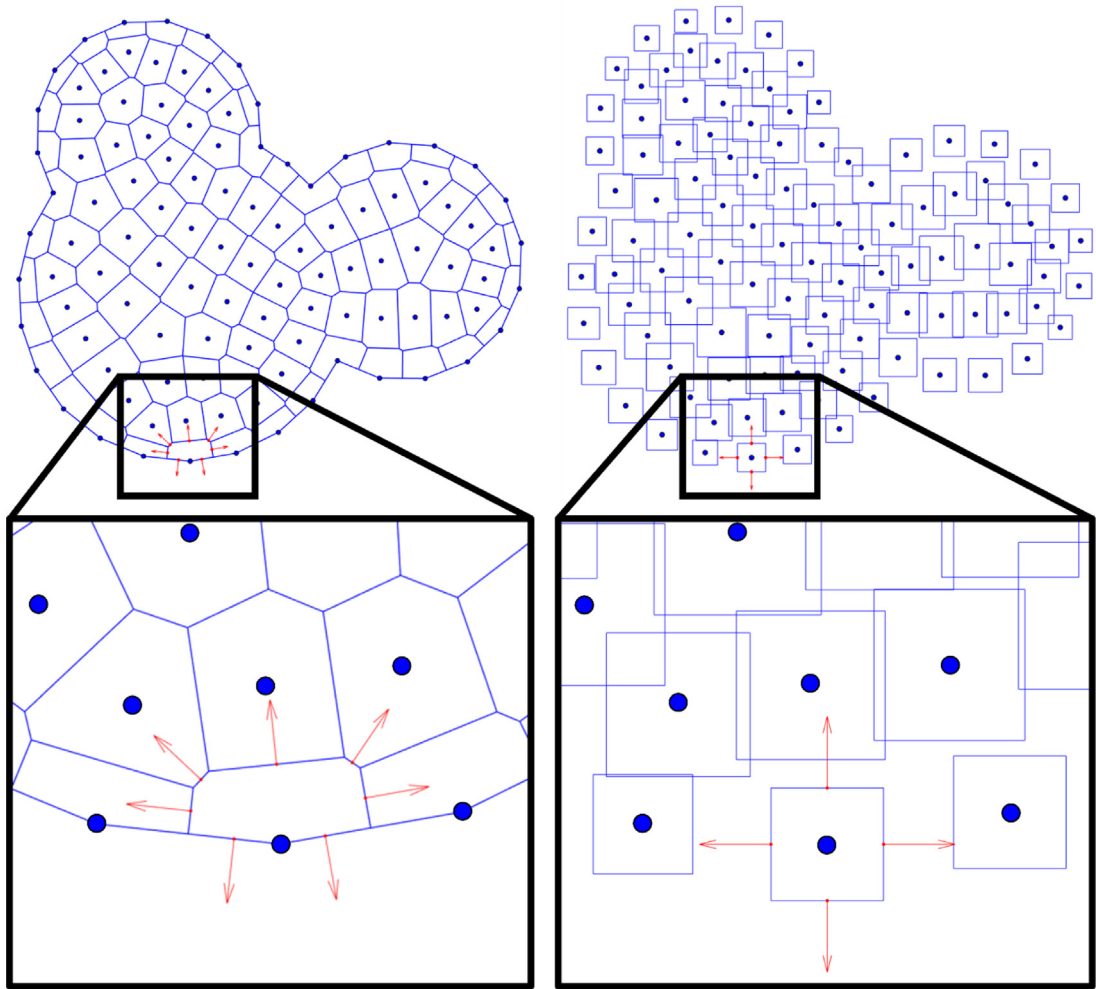


Fig. 4. Smoothing integration cells; SCNI using a Voronoi diagram to achieve conformity (left) and SNNI using square non-conforming cells (right); red arrows indicate the outward normal vectors of the cells at the cell boundary evaluation points.

equal to that of their conforming counterparts. Two-dimensional examples of SCNI and SNNI cells are shown in Fig. 4.

3.2.2. Variationally consistent correction for non-conforming nodal integration

For the proposed quasi-conforming embedded RKPM formulation, SNNI will be employed which is essential to the effectiveness and reliability of the proposed method. Since the integration smoothing cells in SNNI no longer conform to each other, SNNI loses first order variational consistency and, consequently, optimal convergence rates. To restore variational consistency, a variationally consistent correction of integration methods that are not variationally consistent, including SNNI, was developed and termed variationally consistent SNNI (VC-SNNI) [66]. The correction allows a nodal integration with conforming or non-conforming integration smoothing cells to achieve variational consistency for any basis order.

Variational consistency is the ability of a Galerkin method to achieve n th order Galerkin exactness for optimal convergence. Variational consistency of the n th order requires n th order completeness, which can be achieved by RK with n th order bases, and an n th order integration constraint of the following form [57]:

$$\int_{\Omega} \Psi_{1,j} \mathbf{P} d\Omega = \int_{\partial\Omega} \Psi_{1n_j} \mathbf{P} d\Gamma - \int_{\Omega} \Psi_1 \mathbf{P}_{,j} d\Omega, \tag{38}$$

where Ψ_I is the approximation function for the test function, “ $\hat{\int}$ ” denotes numerical integration, and \mathbf{P} is the $(n-1)$ th order monomial vector:

$$\mathbf{P} = [1 \quad x \quad y \quad x^2 \quad xy \quad y^2 \quad \dots \quad x^\alpha y^\beta \quad \dots \quad y^{(n-1)}]^\top. \tag{39}$$

Eq. (38) is called the integration constraint in [57]. For any Galerkin approximation to achieve a desired level of variational consistency, it must (1) have a trial function that can reproduce the desired basis order exactly in (36) and (2) satisfy the integration constraint in (38). For linear variational consistency, $\mathbf{P} = 1$, and Eq. (38) reduces to:

$$\hat{\int}_\Omega \Psi_{I,j} d\Omega = \hat{\int}_{\partial\Omega} \Psi_I n_j d\Gamma, \tag{40}$$

which can be achieved by Eq. (37) with conforming integration smoothing cells. This indicates that SCNI with a linear basis automatically meets first-order variational consistency and does not need VC correction. If we want a higher-order variational consistency in (38) for integration methods that are not variationally consistent, a modification to the test function gradient can be considered [57]:

$$\bar{\Psi}_{I,j} = \Psi_{I,j} + \Theta_I \mathbf{P}^\top \boldsymbol{\zeta}_{Ij}, \tag{41}$$

where $\bar{\Psi}_{I,j}$ is the modified test function gradient, Θ_I is:

$$\Theta_I = \begin{cases} 1, & \text{supp}(\Psi_I) \neq 0 \\ 0, & \text{else} \end{cases}, \tag{42}$$

and $\boldsymbol{\zeta}_{Ij}$ is a column vector of unknown coefficients for each node I and direction j . Substituting the modified test function gradient in (41) in place of the standard test function gradient of (38) and rearranging:

$$\hat{\int}_\Omega \Theta_I \mathbf{P} \mathbf{P}^\top d\Omega \boldsymbol{\zeta}_{Ij} = \hat{\int}_{\partial\Omega} \Psi_I n_j \mathbf{P} d\Gamma - \hat{\int}_\Omega (\Psi_I \mathbf{P}_{,j} + \Psi_{I,j} \mathbf{P}) d\Omega, \tag{43}$$

or:

$$\mathbf{M}_I \boldsymbol{\zeta}_{Ij} = \mathbf{r}_I^j, \tag{44}$$

$$\mathbf{M}_I = \hat{\int}_\Omega \Theta_I \mathbf{P} \mathbf{P}^\top d\Omega, \tag{45}$$

$$\mathbf{r}_I^j = \hat{\int}_{\partial\Omega} \Psi_I n_j \mathbf{P} d\Gamma - \hat{\int}_\Omega (\Psi_I \mathbf{P}_{,j} + \Psi_{I,j} \mathbf{P}) d\Omega, \tag{46}$$

where \mathbf{r}_I^j is the residual of the formulation without VC correction (38). Once (43) is solved for $\boldsymbol{\zeta}_{Ij}$, the modified test function gradients (41) can be computed. Using the modified test function gradients, the Galerkin approximation of elasticity can be expressed as:

$$\hat{\int}_\Omega \bar{\mathbf{B}}_I^\top \boldsymbol{\sigma} d\Omega = \hat{\int}_{\partial\Omega} \boldsymbol{\Psi}_I^\top \boldsymbol{\eta}^\top \boldsymbol{\sigma} d\Gamma + \hat{\int}_\Omega \boldsymbol{\Psi}_I^\top \mathbf{b} d\Omega, \tag{47}$$

where $\bar{\mathbf{B}}_I$ is the modified strain–displacement matrix for node I and is comprised of the modified test function gradients calculated from Eq. (41). Substituting $\boldsymbol{\sigma} = \mathbf{C} \mathbf{B} \mathbf{d}$ into the LHS of (47), the stiffness matrix is:

$$\mathbf{K}_{IJ} = \hat{\int}_\Omega \bar{\mathbf{B}}_I^\top \mathbf{C} \mathbf{B}_J d\Omega, \tag{48}$$

which is a Petrov–Galerkin formulation. It was shown in [57] that such a form is stable if the coefficients in $\boldsymbol{\zeta}_{Ij}$ are sufficiently small.

For the examples in this paper, we use a linear basis and seek to achieve first order variational consistency. For a linear basis, $\mathbf{P} = 1$, and Eqs. (44) to (46) become:

$$M_I \zeta_{Ij} = r_I^j, \tag{49}$$

$$M_I = \hat{\int}_\Omega \Theta_I d\Omega, \tag{50}$$

$$r_l^j = \int_{\partial\Omega} \Psi_{ln_j} d\Gamma - \int_{\Omega} \Psi_{l,j} d\Omega. \tag{51}$$

3.2.3. Naturally stabilized nodal integration

Although SCNI and VC-SNNI with linear bases achieve first-order variational consistency, they are still nodal integration methods and, consequently, reduced-order integration methods. As such, these integration methods can still trigger non-physical, low-energy modes which necessitate the use of stabilization methods as a remedy [66,70,71]. The most computationally efficient stabilization method that avoids tunable parameters is the naturally stabilized nodal integration (NSNI) [66]. NSNI expands the strain–displacement matrix in a Taylor series [72–74]:

$$\hat{\mathbf{B}}_I(\mathbf{x}_L) = \tilde{\mathbf{B}}_I(\mathbf{x}_L) + (x - x_L) \cdot \tilde{\mathbf{B}}_{Ix}^\nabla(\mathbf{x}_L) + (y - y_L) \cdot \tilde{\mathbf{B}}_{Iy}^\nabla(\mathbf{x}_L) + (z - z_L) \cdot \tilde{\mathbf{B}}_{Iz}^\nabla(\mathbf{x}_L) \tag{52}$$

where $\hat{\mathbf{B}}_I$ is the smoothed and expanded strain–displacement matrix, $\tilde{\mathbf{B}}_I$ is the smoothed shape function gradient matrix from SCNI or VC-SNNI following (37) and $\tilde{\mathbf{B}}_{Ix}^\nabla$, $\tilde{\mathbf{B}}_{Iy}^\nabla$, and $\tilde{\mathbf{B}}_{Iz}^\nabla$ are the smoothed gradients of the implicit gradients [66,74] in the x -, y -, and z -directions, respectively:

$$\tilde{\mathbf{B}}_{Ix}^\nabla = \begin{bmatrix} \tilde{\Psi}_{I,xx}^\nabla & 0 & 0 \\ 0 & \tilde{\Psi}_{I,yx}^\nabla & 0 \\ 0 & 0 & \tilde{\Psi}_{I,zx}^\nabla \\ \tilde{\Psi}_{I,yx}^\nabla & \tilde{\Psi}_{I,xx}^\nabla & 0 \\ \tilde{\Psi}_{I,zx}^\nabla & 0 & \tilde{\Psi}_{I,xx}^\nabla \\ 0 & \tilde{\Psi}_{I,zx}^\nabla & \tilde{\Psi}_{I,yx}^\nabla \end{bmatrix}, \tag{53}$$

$$\tilde{\mathbf{B}}_{Iy}^\nabla = \begin{bmatrix} \tilde{\Psi}_{I,xy}^\nabla & 0 & 0 \\ 0 & \tilde{\Psi}_{I,yy}^\nabla & 0 \\ 0 & 0 & \tilde{\Psi}_{I,zy}^\nabla \\ \tilde{\Psi}_{I,yy}^\nabla & \tilde{\Psi}_{I,xy}^\nabla & 0 \\ \tilde{\Psi}_{I,zy}^\nabla & 0 & \tilde{\Psi}_{I,xy}^\nabla \\ 0 & \tilde{\Psi}_{I,zy}^\nabla & \tilde{\Psi}_{I,yy}^\nabla \end{bmatrix}, \tag{54}$$

$$\tilde{\mathbf{B}}_{Iz}^\nabla = \begin{bmatrix} \tilde{\Psi}_{I,xz}^\nabla & 0 & 0 \\ 0 & \tilde{\Psi}_{I,yz}^\nabla & 0 \\ 0 & 0 & \tilde{\Psi}_{I,zz}^\nabla \\ \tilde{\Psi}_{I,yz}^\nabla & \tilde{\Psi}_{I,xz}^\nabla & 0 \\ \tilde{\Psi}_{I,zz}^\nabla & 0 & \tilde{\Psi}_{I,xz}^\nabla \\ 0 & \tilde{\Psi}_{I,zz}^\nabla & \tilde{\Psi}_{I,yz}^\nabla \end{bmatrix}, \tag{55}$$

where $\tilde{\Psi}_{I,ij}^\nabla$ denotes the smoothed gradient in the i th-direction using the implicit gradient shape function in the j th-direction. The implicit gradient shape function in the j th-direction is given as:

$$\Psi_{I,j}^\nabla(\mathbf{x}) = \mathbf{H}_j^T \mathbf{M}^{-1}(\mathbf{x}) \mathbf{H}(\mathbf{x} - \mathbf{x}_I) \phi_a(\mathbf{x} - \mathbf{x}_I) \tag{56}$$

where \mathbf{H}_j for a linear basis has the following values:

$$\mathbf{H}_x^T = [0, -1, 0, 0] \tag{57}$$

$$\mathbf{H}_y^T = [0, 0, -1, 0] \tag{58}$$

$$\mathbf{H}_z^T = [0, 0, 0, -1] \tag{59}$$

Note that (56) has the same form as the standard RK shape function Eq. (32) with the only difference being the substitution of \mathbf{H}_j^T in place of $\mathbf{H}^T(\mathbf{0})$. As a result, the NSNI stabilization requires only a minimal computational effort [66].

Remark 3.2. Complexity analyses of SCNI/SNNI and NSNI have been reported in [66], and complexity analyses of the direct RK gradient and implicit gradient have been studied in [75] and [76], respectively.

With NSNI in (52), the stiffness matrix with VC correction becomes:

$$\int_{\Omega} \hat{\mathbf{B}}_I^T \mathbf{C} \hat{\mathbf{B}}_J d\Omega = \sum_{L=1}^{NP} \left(\hat{\mathbf{B}}_I^T \mathbf{C} \tilde{\mathbf{B}}_J V_L + \tilde{\mathbf{B}}_{Ix}^{\nabla T} \mathbf{C} \tilde{\mathbf{B}}_{Jx}^{\nabla} M_{Lx} + \tilde{\mathbf{B}}_{Iy}^{\nabla T} \mathbf{C} \tilde{\mathbf{B}}_{Jy}^{\nabla} M_{Ly} + \tilde{\mathbf{B}}_{Iz}^{\nabla T} \mathbf{C} \tilde{\mathbf{B}}_{Jz}^{\nabla} M_{Lz} \right), \quad (60)$$

where $\hat{\mathbf{B}}_I$ is the VC-corrected, smoothed, and expanded strain–displacement matrix, $\tilde{\mathbf{B}}_I$ is the VC-corrected, smoothed strain–displacement matrix from SCNI or SNNI following (41) and (53) through (59), and M_{Lx} , M_{Ly} , and M_{Lz} are the second moments of inertia about node L . In (60), we have assumed that the first moments of inertia and products of inertia are zero when nodes are located at the centroids of their nodal cells. While boundary nodes are not located at their cell centers, it has been observed that the second moments of inertia are sufficient to stabilize these nodes [66].

As we will show in Section 4, the background matrix domain will be comprised of conforming smoothing integration cells except near the interface where the cells are non-conforming, while the foreground inclusion domain(s) will be comprised exclusively of conforming cells. Thus, VC-SNNI and SCNI, both stabilized with NSNI, will be used for the background and foreground domains, respectively. Direct nodal integration will be used for all other volume integrals such as the body force terms.

4. Quasi-conforming embedded discretization

The proposed quasi-conforming embedded (QCE) discretization is a three-stage process, and this section details the reasoning behind the geometric manipulations at each stage. The advantages of this meshfree strategy over standard immersed and embedded techniques rooted in mesh-based methods such as FEM will be noted throughout.

4.1. Initial immersed state

To avoid time-consuming, conformal discretization, the background and foreground domains are discretized independently, and the foreground discretization is initially immersed in the background discretization as shown in Fig. 5. The region shared by both background and foreground domains is called the *overlapping region*, and the background discretization in the overlapping region is called the *fictitious discretization*.

Remark 4.1.

1. In this work, we have chosen to use Voronoi cells as our conforming smoothing cells. Voronoi cell construction is quite fast as there is no concern for mesh connectivity or element quality. The use of the smoothed gradient (as detailed in Section 3.2.1) in combination with Voronoi cells avoids the pitfall of getting poor results due to the generation of distorted finite elements. Thus, the QCE algorithm is much faster than the manual meshing necessary to create a quality finite element mesh free of distorted elements.
2. The background discretization is comprised of simple, square smoothing cells which have only four evaluation points needed to integrate a cell’s area, and the typical foreground smoothing cell has only a few more. Thus, the computational burden for integrating the domain is relatively small.
3. It is possible to use non-conforming cells for both foreground and background discretizations which would minimize the complexity and computational cost of discretizing the domains. While utilizing this option will lower the accuracy of the approximation as compared to the use of conforming cells, the use of the variational correction detailed in Section 3.2.2 would still allow the approximation to achieve optimal convergence rates [57].

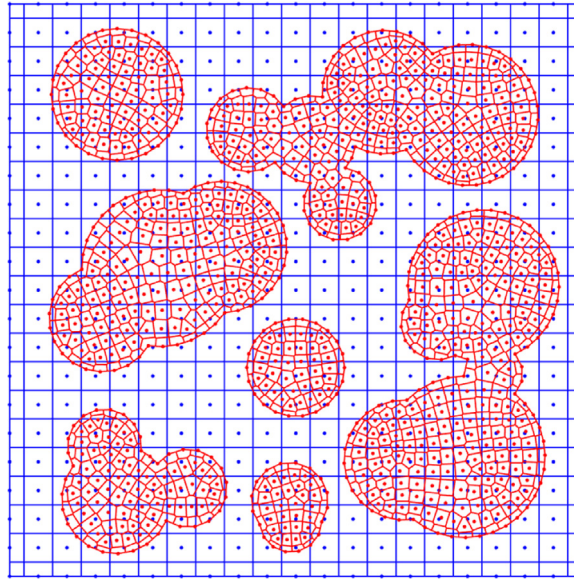


Fig. 5. Background and foreground smoothing integration cells at the initial immersed state.

4.2. Non-conforming quadtree subdivision

One problem with immersed methods (especially VCIMs) is the “smearing” of the background solution caused by that approximation’s continuity across the interface. Capturing the weak discontinuity at the interface is essential when modeling, for example, the debonding of inclusions in composite materials. The second problem with immersed methods is that the discretization ratio (between background and foreground domains) must stay in a certain range to prevent the ill-conditioning of the stiffness matrix. In our proposed framework, we remove the fictitious domain nodes and cells thereby preventing the smearing of the background approximation and the ill-conditioning of the stiffness matrix. For RKPM with RK approximation, these discretization cells are used as a “reference” for computing nodal areas (volumes) and as the integration smoothing cells for SCNI or SNNI strain smoothing.

For the matrix integration points to fulfill the first order integration constraint in a non-conforming manner, the gap resulting from the complete removal of the fictitious smoothing integration cells that intersect the interface can be recovered easily by correcting the area (volume) of these cells. In the case where the background to foreground nodal spacing ratio is too large (and vice versa), node and smoothing integration cell insertions can be done systematically, see Fig. 6.

A simple strategy to determine the level of sub-division of the background smoothing integration cells, denoted as n_R , is given as follows:

$$n_R = \begin{cases} \lfloor n_{log} \rfloor, & n_{log} > 0 \\ 0, & n_{log} \leq 0 \end{cases}, \tag{61}$$

$$n_{log} = \log_2(\mathcal{R}'), \tag{62}$$

$$\mathcal{R}' = \begin{cases} \lceil \mathcal{R} \rceil, & \mathcal{R} \bmod 1 \geq k, \mathcal{R} > 1 \\ \lfloor \mathcal{R} \rfloor, & \mathcal{R} \bmod 1 < k, \mathcal{R} > 1, \\ 1, & \mathcal{R} \leq 1 \end{cases}, \tag{63}$$

$$\mathcal{R} = \frac{h^-}{h_{\text{Inf}}^+}, \tag{64}$$

where n_R is the number of times background smoothing integration cells intersecting the interface are subdivided, $\lfloor \cdot \rfloor$ is the floor function which rounds a fraction down to the nearest integer, $\lceil \cdot \rceil$ is the ceiling function which

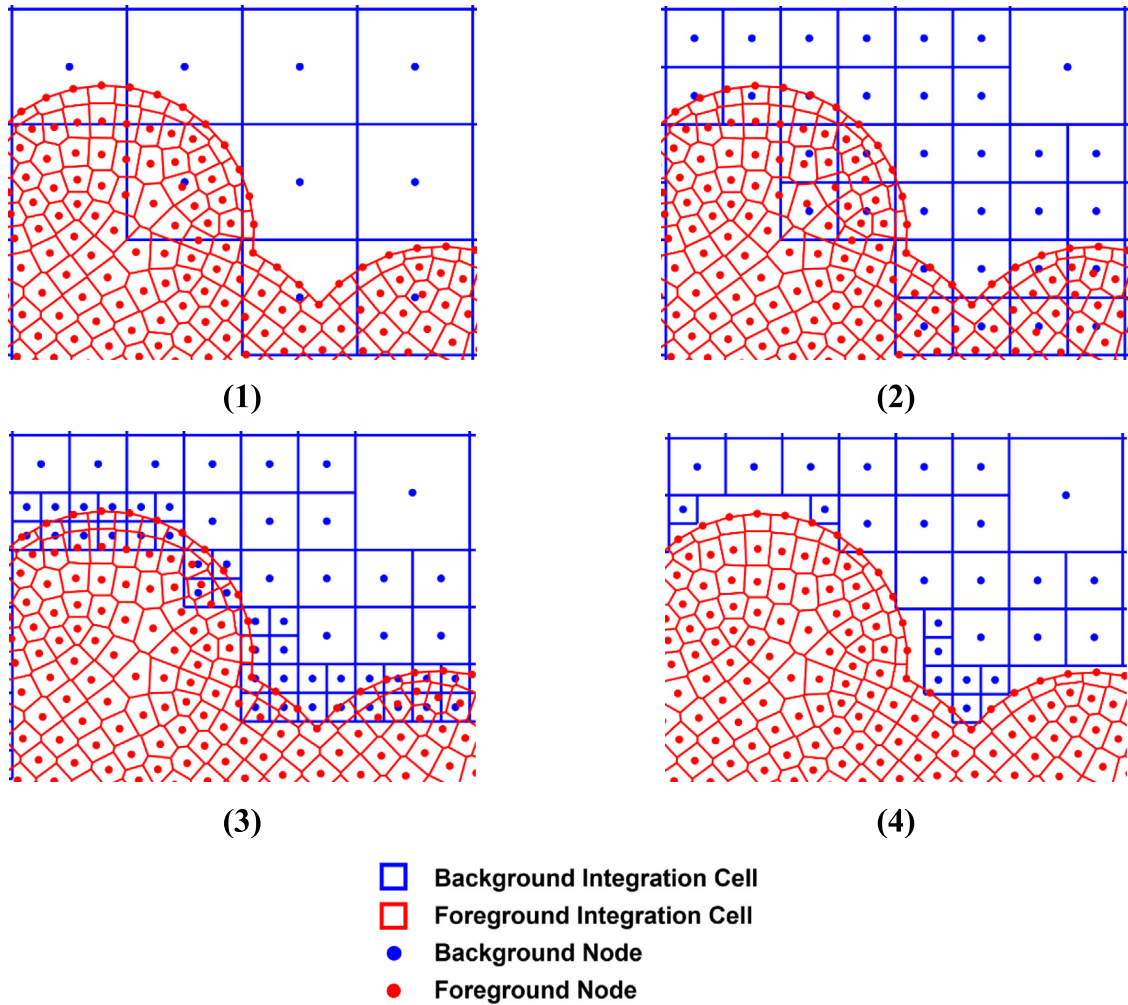


Fig. 6. Quadtree subdivision process: (1) Remove all nodes and smoothing integration cells that are completely in the overlapping region. (2) If $n_R \geq 1$, refine remaining smoothing integration cells within a specified distance or that intersect the interface. (3) If $n_R \geq 2$, repeat steps (1) and (2) above $n_R - 1$ more times for cells that straddle the interface. (4) Remove any new smoothing integration cells with cell boundary evaluation points in the overlapping region.

rounds a fraction up to the nearest integer, k is a real number between 0 and 1, \mathcal{R} is the nodal spacing ratio, h^- is the uniform background nodal spacing before quadtree subdivision, and h_{Inf}^+ is the averaged nodal spacing of the inclusion nodes on the interface. In our examples, as demonstrated in Fig. 6, we have used $k = 0.5$ which simply makes (63) round \mathcal{R} to the nearest integer unless $\mathcal{R} \leq 1$. This quadtree subdivision provides a smooth discretization density transition near the material interfaces. A similar strategy can be defined in the case where the background discretization density is higher than the foreground discretization density. Fig. 7(a) and Fig. 7(b) show one-level and two-level non-conforming quadtree subdivision intermediate states, respectively. This non-conforming quadtree subdivision strategy with variable subdivision levels can also be considered, as shown in Fig. 7(c).

4.3. Integration constraint correction of non-conforming nodal integration

In this section, we investigate the influence of the missing area (volume) due to the removal of overlapping smoothing cells and quadtree subdivisions. It is worth noting that the missing area (volume) can be corrected by the residual r_l^j in the VC correction of non-conforming nodal integration in (44) and (46). As mentioned in

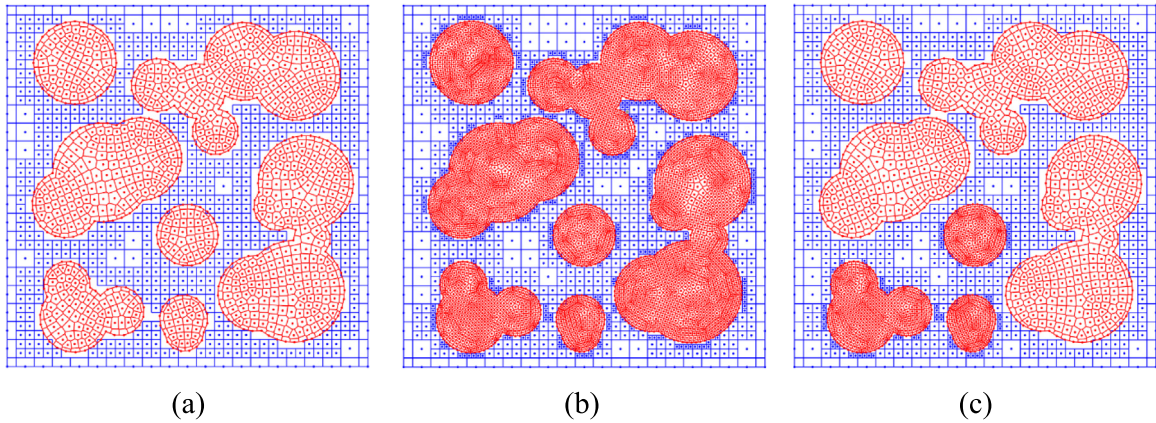


Fig. 7. Non-conforming quadtree subdivisions: (a) one-level subdivision, (b) two-level subdivision, and (c) variable-level quadtree subdivisions.

Section 3.2.2, the variationally consistent (VC) correction leads to a Petrov–Galerkin formulation, the stability of which depends on the magnitude of the residual r_I^j [57]. Thus, although it is not required that the volume of all smoothing cells equals the volume of the computational domain for the VC correction in (41) to work, the correction residual r_I^j is reduced by meeting the area (volume) conservation for better stability. The influence of the area (volume) conservation on the correction coefficients can be shown by replacing the RK shape functions in the first order integration constraint in (41) and (51) with a linear function. Let $\Psi_I = \Psi = x + y + z$ whose support covers the entirety of an arbitrary three-dimensional domain Ω . Then from (51), we have:

$$r^j = \int_{\partial\Omega} (x + y + z) n_j d\Gamma - \int_{\Omega} d\Omega, \tag{65}$$

Using Gauss’s theorem:

$$\int_{\partial\Omega} x_i n_j d\Gamma = \int_{\Omega} x_{i,j} d\Omega = \int_{\Omega} \delta_{ij} d\Omega = V \delta_{ij}. \tag{66}$$

where V is the volume of the computational domain. Using (66) in (65), we have:

$$r^j = V - \hat{V}, \tag{67}$$

where \hat{V} is the total volume of all integration smoothing cells:

$$\int_{\Omega} d\Omega = \sum_{I=1}^{NC} V_I \equiv \hat{V}, \tag{68}$$

in which NC is the number of smoothing cells and V_I is the volume of each smoothing cell. Since the support of $\Psi_I = \Psi$ extends over the entire computational domain, then $\theta_I = \theta = 1$ for $\forall \mathbf{x} \in \Omega$, and since the integral in (50) is over all smoothing cells, (50) becomes:

$$M_I = M = \int_{\Omega} \theta d\Omega = \int_{\Omega} d\Omega = \sum_{I=1}^{NC} V_I \equiv \hat{V}, \tag{69}$$

Therefore, (49) becomes:

$$\hat{V} \zeta_j = V - \hat{V}, \tag{70}$$

or:

$$\zeta_j = \frac{V - \hat{V}}{\hat{V}} = \frac{V}{\hat{V}} - 1, \tag{71}$$

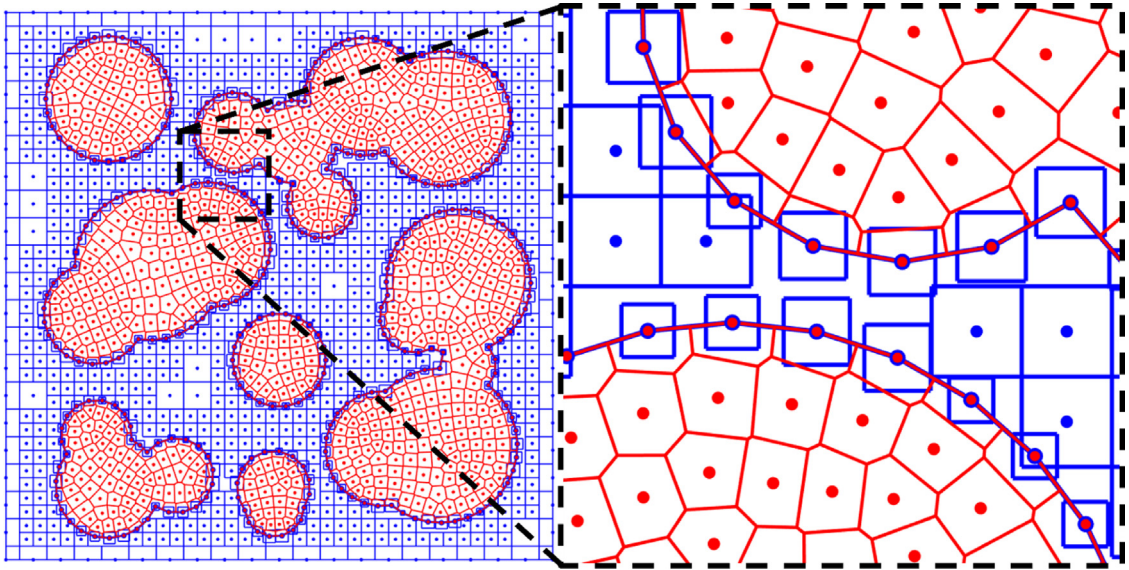


Fig. 8. Addition of square, non-conforming integration smoothing cells centered on interface nodes.

Using (71) and noting that $\mathbf{P}^T = \mathbf{P} = \mathbf{1}$ for a linear basis, (41) becomes:

$$\bar{\Psi}_{I,j} = \bar{\Psi}_{,j} = 1 + \frac{V}{\hat{V}} - 1 = \frac{V}{\hat{V}}. \quad (72)$$

Eqs. (71) and (72) show that the VC correction, in addition to meeting variational consistency, also acts as a volume correction, where the size of the VC correction is partially driven by the deviation of the total integration smoothing cell area (volume) from the computational domain area (volume).

Remark 4.2. Enforcing area (volume) conservation in the integration smoothing cells, though non-conforming, allows the VC correction to contribute mainly to the correction of integration smoothing cell non-conformity, resulting in a more accurate Galerkin solution. Additionally, area (volume) conservation reduces the size of the VC coefficients resulting in better stability [57]. Integration smoothing cell volume conservation can be easily achieved by adding non-conforming “volume-recovery” smoothing cells centered on the interface nodes, as shown in Fig. 8.

Remark 4.3. The interface nodes can be shared by both foreground and background domain approximations, which has the following advantages: (1) as discussed in Section 2.2, the enhanced coercivity allows for a penalty-free Nitsche’s method, and (2) the partition of unity in the RK approximation can be maintained in each of the sub-domains.

Remark 4.4. All the procedures detailed in Sections 4.2 and 4.3 to create the QCE discretization are automatic and require no manual user input. After doing a runtime study, these procedures take only about 10% of the total runtime when run in serial. Parallelization of this subroutine would lower that figure considerably.

5. Numerical examples

In this section, the effectiveness of the proposed QCE-RKPM is demonstrated through several numerical examples with comparisons to body-fitted FEM and analytical solutions. For all examples, the RK approximation uses a linear basis and a cubic B-spline kernel function with a normalized support size of $c = 2.0$, and the line-of-sight algorithm (see Section 3.1 and the modification in Section 4.3) is used to determine the influence of nodes. Additionally, the interface nodes are shared by both foreground and background domain approximations. SCNI and VC-SNNI are utilized for the domain integration of the foreground and background domains, respectively,



Fig. 9. Composite bar with left end fixed.

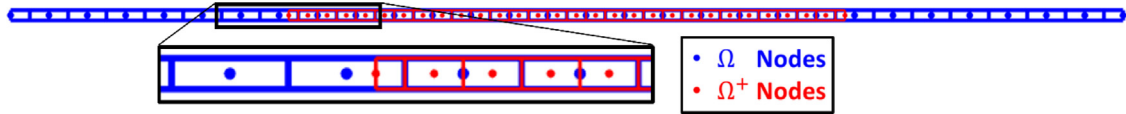


Fig. 10. Pre-QCE immersed discretization.

and NSNI is employed for the stabilization of both. Nitsche’s method with a penalty parameter of $\beta = 100 \frac{E^-}{h^-}$ is utilized for enforcing Dirichlet boundary conditions where h^- is the original nodal spacing of the background domain prior to quadtree subdivision. The quadtree subdivision of the background integration smoothing cells near the interface follows the strategies given in Section 4.2. For post-processing the QCE-RKPM results, the nodal smoothed gradients are used to calculate strain values at the nodes for those nodes with integration smoothing cells, while the direct gradient evaluated at the node is used for those nodes without integration smoothing cells. The RK shape functions are then utilized to interpolate those values to the Gauss points which have the dual purpose of both error calculation and contour plotting. To make a fair comparison, the FEM strains are calculated using the standard shape function gradients evaluated at the nodes for each element, and the nodal strains are averaged over all elements to which a node belongs. The element shape functions are then used to interpolate averaged nodal strain values to the Gauss points. To calculate error norms, 10th order Gauss quadratures are used for sufficient accuracy.

5.1. One-dimensional composite bar

This example is used to examine how the “non-conforming” integration smoothing cells in the proposed QCE-RKPM affect the solution accuracy and convergence rates. As shown in Fig. 9, a fixed-end composite bar is subjected to different loading and boundary conditions in the following studies. The linear elastic material properties are set to be $E^+ = 100,000$ and $E^- = 1,000$.

5.1.1. A piece-wise linear patch test with weak discontinuities

In this sub-section, a displacement of $u = g = 0.3$ is prescribed at the right end of the composite bar. As shown in Fig. 10, the foreground domain Ω^+ is discretized with 30 nodes and is initially immersed in the entire domain $\bar{\Omega} = \bar{\Omega}^- \cup \bar{\Omega}^+$ which is also discretized by 30 nodes. This creates the situation where the foreground nodal spacing is half that of the background nodal spacing and such that their corresponding integration smoothing cells do not match at the interface as shown in Fig. 10.

As outlined in Section 4.2, the background nodes and integration smoothing cells completely inside the foreground discretization are deleted, and background cells within $1.5h^+$ of the interface are divided in two. Any background cells still intersecting the interface are removed. Two cases were created to test the effect of the volume-recovery integration smoothing cells discussed in Section 4.3, one with these cells present (Fig. 11(a)) and one without (Fig. 11(b)). Lastly, the foreground interface nodes are shared with the background discretization.

As expected, both QCE-RKPM cases exactly reproduce the analytical solution at the nodes to nearly machine precision and give L_2 error norms of 1.6×10^{-13} for the case with volume-recovery cells and 1.8×10^{-13} for the case without these cells. The ability of QCE-RKPM to obtain an exact solution in this manufactured problem is due to the VC-SNNI that meets first order variational consistency *even with non-conforming integration smoothing cells*. Fig. 12 shows the displacement, strain, and stress plots.

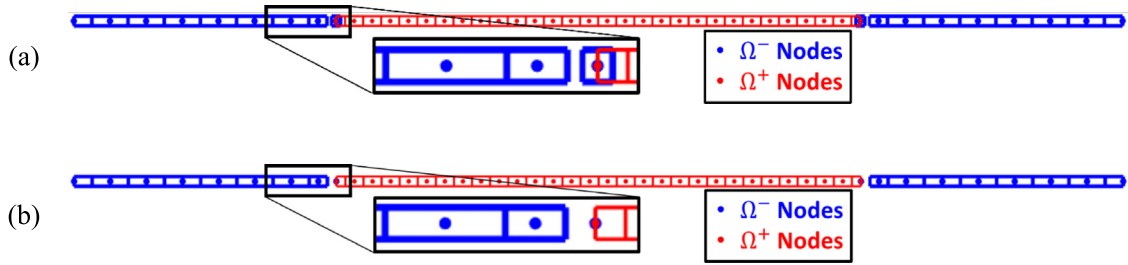


Fig. 11. QCE-RKPM discretization for the 1D bar problem: (a) volume-recovery cells added around interface nodes and (b) no volume-recovery cells.

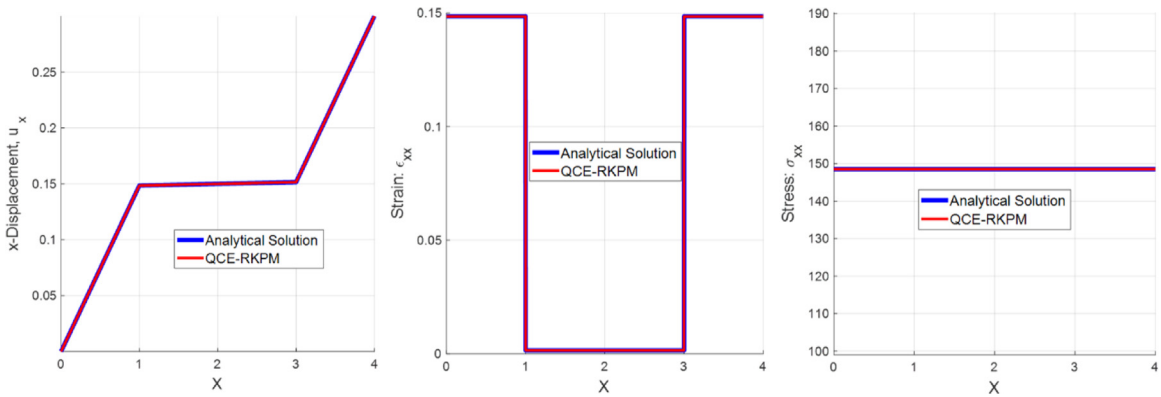


Fig. 12. Displacement (left), strain (center), and stress (right) plots for both QCE-RKPM cases.

5.1.2. Piecewise higher-order solution with weak discontinuities

This problem exhibits a piecewise higher-order solution and is designed to examine QCE-RKPM’s convergence properties when linear basis functions are used. The composite bar is subjected to piecewise sinusoidal body force loadings as follows:

$$b^- = 10 \sin\left(\frac{\pi x}{L_1}\right), \quad 0 \leq x \leq x_1, \tag{73}$$

$$b^+ = 50 \sin\left(\frac{\pi(x - x_1)}{L_2}\right), \quad x_1 \leq x \leq x_2, \tag{74}$$

$$b^- = 10 \sin\left(\frac{\pi(x - x_2)}{L_3}\right), \quad x_2 \leq x \leq x_3. \tag{75}$$

The body force terms (see Eqs. (85) and (91) in the Appendix) are integrated with direct nodal integration.

Fig. 13 shows the QCE-RKPM displacement, strain, and stress results for the discretizations shown in Fig. 11 along with the analytical solution. The results for the QCE-RKPM case with volume recovery cells (Fig. 13(a)) are shown to be highly accurate compared with the exact solution, and while the QCE-RKPM case with no volume-recovery cells (Fig. 13(b)) does show the expected inaccuracy near the material interfaces, the results are still very good.

Fig. 14 shows the convergence plots in the L_2 norm and H_1 semi-norm for both QCE-RKPM cases as well as body-fitted FEM. The numbers in the legends are the convergence rates. As shown in Fig. 14, both QCE-RKPM cases and FEM converge at an optimal rate in the L_2 norm and possess super-convergence in the H_1 error semi-norm due to the re-interpolation of the smoothed and averaged nodal strain values, respectively. While the QCE-RKPM overall error is higher in the L_2 norm compared to FEM, the QCE-RKPM with volume-recovery cells is shown to have roughly the same error level as body-fitted FEM in the H_1 error semi-norm.

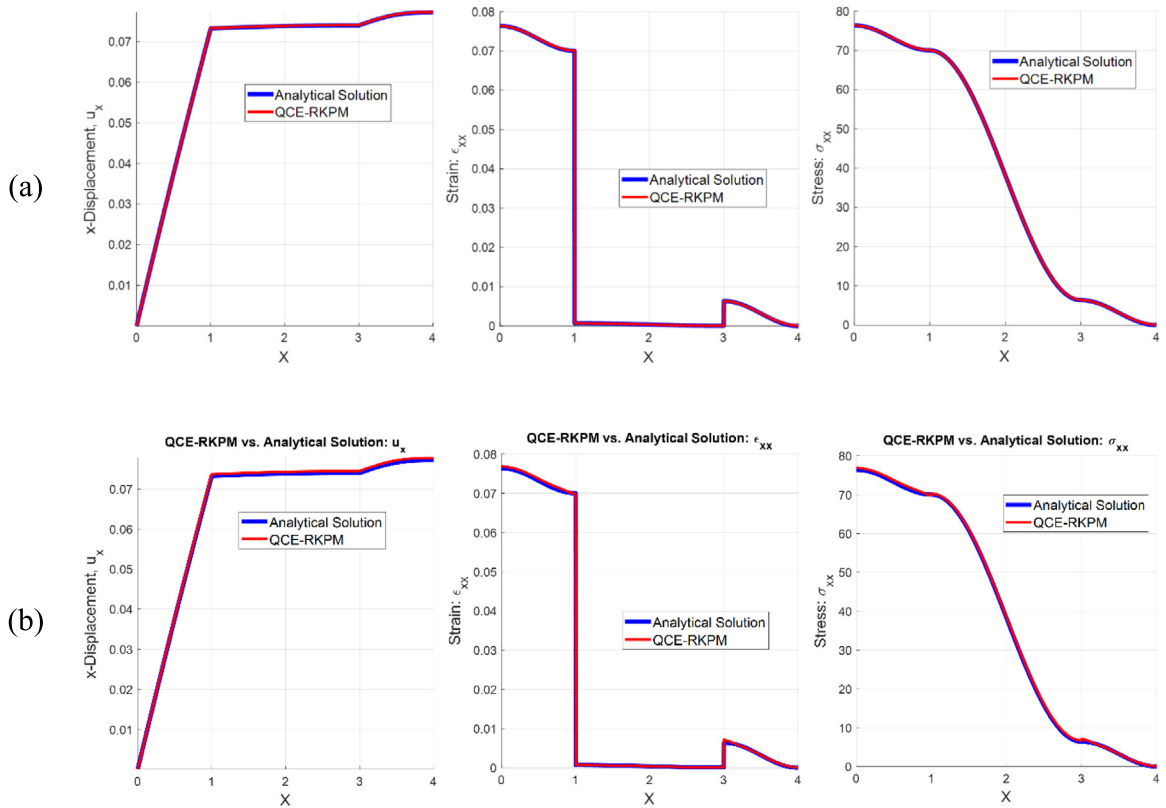


Fig. 13. Displacement (left), strain (center), and stress (right) plots: (a) QCE-RKPM case with volume-recovery cells and (b) QCE-RKPM case without these cells.

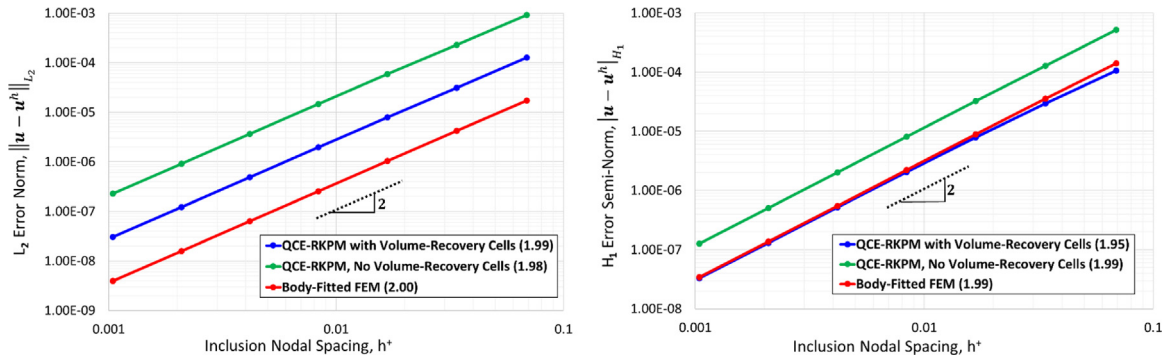


Fig. 14. Convergence plots for the L_2 error norm (left) and H_1 error semi-norm (right).

5.2. Circular inclusion in an infinite plate subjected to far-field traction

In this example, an infinite plate with a circular inclusion of diameter $D = 2$ is subjected to a far-field traction $F = 100$ as shown in Fig. 15.

The computational domain is a 4×4 square with outer boundary $\partial\Omega$ and an interface Γ between the background matrix domain and foreground inclusion domain as shown in Fig. 16. The elastic moduli used for this example are $E^+ = 100,000$ and $E^- = 1,000$, and the Poisson's ratios are $\nu^+ = \nu^- = 0.3$. The problem is modeled as plane stress. This problem possesses an analytical solution which is applied as a Dirichlet boundary condition to the entire background outer boundary $\partial\Omega$. The analytical solution to this problem can be found in [77]. Three discretization

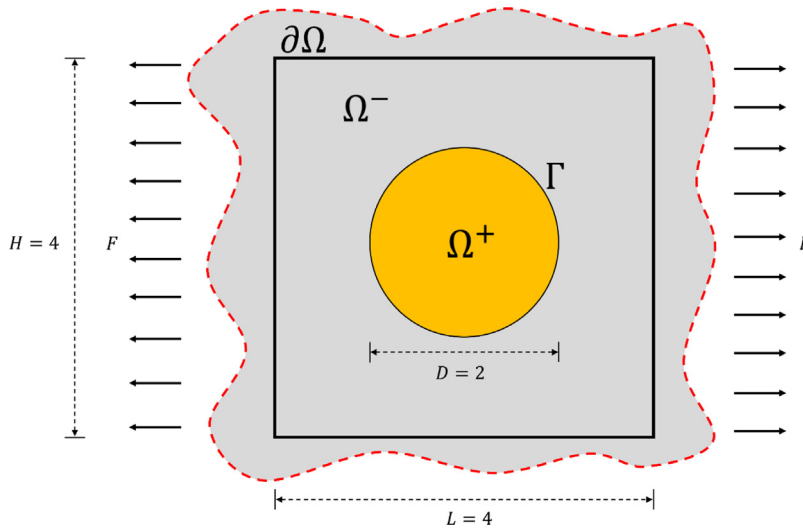


Fig. 15. Infinite plate with circular inclusion subjected to far-field traction.

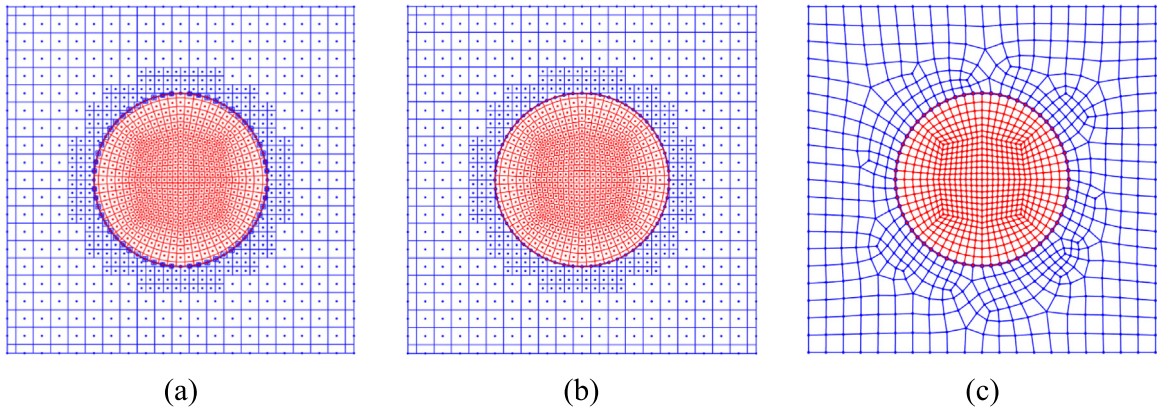


Fig. 16. $h^- = 0.2$, $h_{\text{Intf}}^+ \approx 0.1$: (a) QCE-RKPM with volume-recovery integration smoothing cells centered on interface nodes, (b) QCE-RKPM without the volume-recovery smoothing cells, (c) and body-fitted FEM with gradual refinement.

cases were run for the convergence study, two QCE-RKPM cases and one body-fitted FEM case. The two QCE-RKPM cases use an initial background nodal spacing which is approximately twice the spacing of the foreground interface nodes with quadtree subdivision around the interface. The first QCE-RKPM case uses square, volume-recovery non-conforming integration smoothing cells centered on the interface nodes (Fig. 16(a) and Fig. 17(a)), while the second QCE-RKPM case does not employ these volume-recovery integration smoothing cells (Fig. 16(b) and Fig. 17(b)). Note that, for both QCE-RKPM cases shown in Fig. 17, the foreground interface nodes are shared with the background domain for enhanced coercivity in a penalty-free Nitsche’s method as well as ensuring the partition of unity in both domains, as discussed in Sections 2.2 and 4.3. The corresponding body-fitted FEM case has a gradual element refinement analogous to the quadtree refinement as shown in Fig. 16(c) for comparison.

Fig. 18 shows the strain fields and the σ_{yy} field for the discretizations in Fig. 16 as well as the analytical solution. The contour plots for the QCE-RKPM case with volume-recovery integration smoothing cells show very little difference between them and the analytical solution, demonstrating the effectiveness of the proposed QCE-RKPM in achieving an accurate solution without the tedious effort in constructing a body-fitted discretization. The contour plots for the QCE-RKPM case without volume-recovery cells, while showing good results in the strain fields, do show a deviation in the stress from the analytical solution inside the inclusion region. This shows that conserving area (volume) with volume-recovery cells allows for more integration points and for the VC correction to contribute

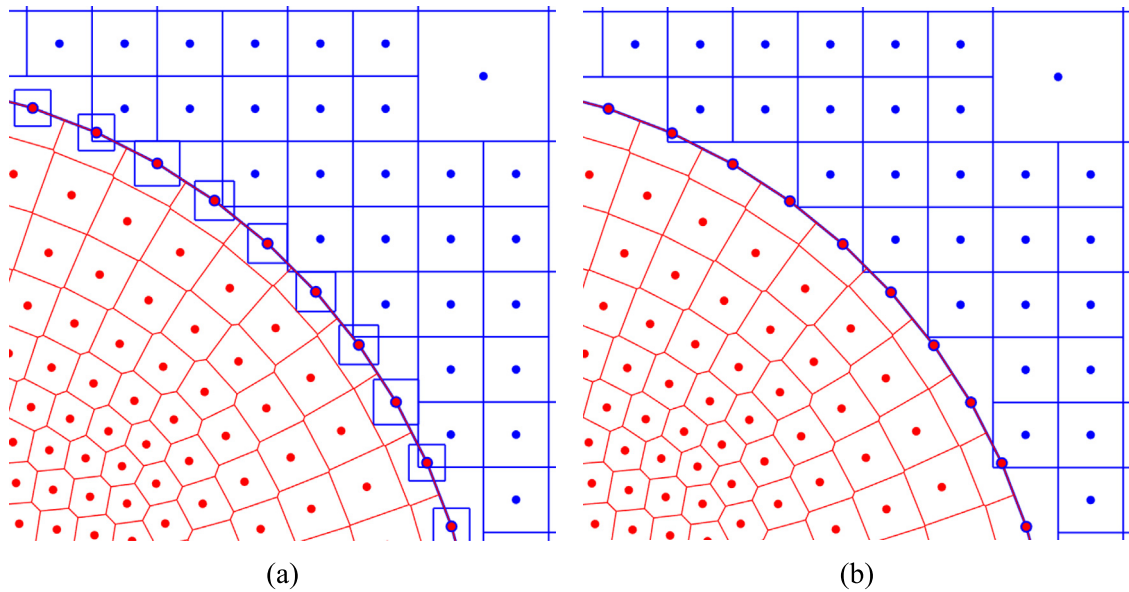


Fig. 17. Magnified views of (a) Fig. 16(a) and (b) Fig. 16(b).

mainly to the lack of variational consistency thereby increasing the accuracy of the approximation. The body-fitted FEM solution, while superior to the QCE-RKPM case without volume-recovery cells, appears rough and asymmetric in places and is inferior in most respects to the QCE-RKPM case with volume-recovery integration smoothing cells.

Fig. 19 plots cross-sections of the normal strains and stresses along $y = 0$ for the discretizations in Fig. 16. Again, these cross-sections show that the QCE-RKPM formulation using volume-recovery cells, though non-conforming, provides better accuracy than body-fitted FEM without the tedium of creating a body-fitted mesh. Cross-sections of the normal strains and stresses for the most refined case ($h_{\text{Intf}}^+ \approx h^- = 0.0125$) are plotted in Fig. 20 and show that both QCE-RKPM cases converge to the analytical solution with refinement. The convergence studies are shown in the plots of Fig. 21. These plots demonstrate that QCE-RKPM with volume-recovery cells, while non-conforming, can achieve optimal convergence properties and similar overall error levels in the L_2 error norm as those of body-fitted FEM, and both QCE-RKPM cases achieve a super-convergent rate and superior overall error levels in the H_1 error semi-norm compared to body-fitted FEM.

5.3. Multiple inclusion microstructure subjected to combined tension and shear loading

Our last example is of a heterogeneous microstructure with multiple, arbitrarily shaped interfaces, some of which contain concave features. Fig. 22 shows a heterogeneous material consisting of randomly sized circular inclusions which is subjected to a combined tension and shear displacement of $\mathbf{g} = [0.02, 0.02]$. We assign the material properties to be $E^- = 1,000$, $\nu^- = 0.3$, $E^+ = 100,000$, $\nu^+ = 0.3$.

In this study, QCE-RKPM results obtained from a coarse smoothing-cell discretization and two finer smoothing-cell discretizations with and without quadtree subdivisions (Fig. 23) are compared with the reference solution obtained from a highly-refined, body-fitted FEM model (Fig. 25). As shown in Fig. 24, all QCE-RKPM cases possess background integration smoothing cells that are non-conforming near the interface. The QCE-RKPM discretizations in Fig. 23(a), (b), and (c) are with 709, 1,639, and 1,839 nodes, respectively, and the FEM discretization in Fig. 25 is with 424,063 nodes. All QCE-RKPM discretizations utilize the volume-recovery integration smoothing cells for enhanced accuracy. The FEM reference solution was run in Abaqus CAE [78] and post-processed in MATLAB. The results displayed in Fig. 26 show that QCE-RKPM can achieve similar results to a highly refined, body-fitted FEM solution without the effort of constructing a body-fitted discretization.

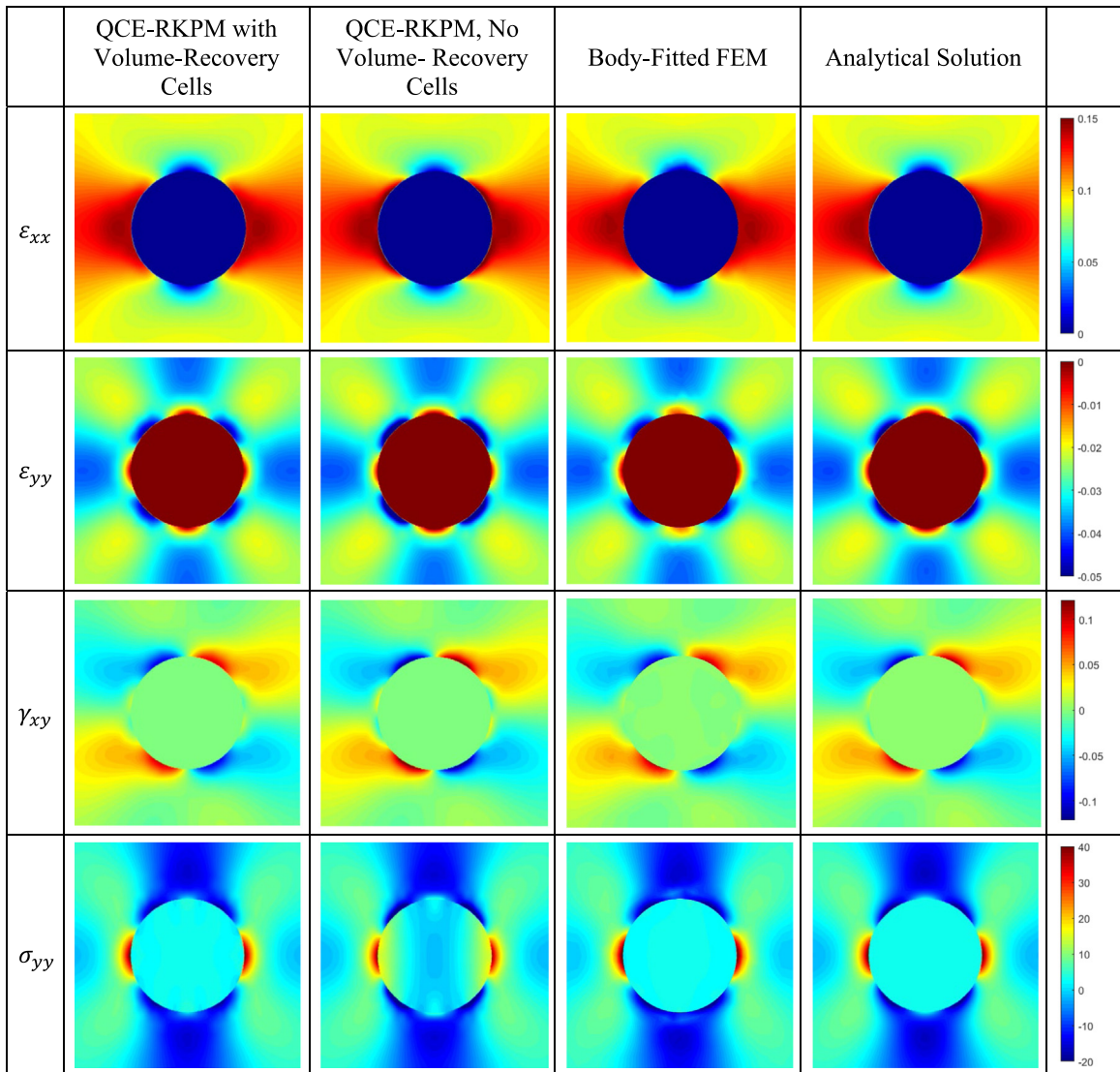


Fig. 18. ϵ_{xx} , ϵ_{yy} , γ_{xy} , and σ_{yy} fields for the discretizations in Fig. 16 and the analytical solution.

6. Summary

In this work, we proposed a quasi-conforming embedded reproducing kernel particle method (QCE-RKPM) for problems involving heterogeneous materials. The proposed formulation has demonstrated many advantages over the standard mesh-based immersed and embedded methods. The inclusion and matrix domains are discretized independently which avoids time-consuming conformal meshing. After immersing the inclusion discretization in a uniform matrix discretization, the fictitious domain, the source of stability and interpolation error problems in conventional immersed methods, is removed. Due to the non-conforming discretization, the VC correction is needed to meet first order exactness in the Galerkin approximation for optimal convergence. Additional interface smoothing cells with volume correction, while non-conforming, can be introduced in a straightforward way to further enhance solution accuracy. This ability to add nodes and smoothing cells where needed without concerns for mesh connectivity and conformity is a major advantage over mesh-based methods which require cell-cutting, cell-aggregation, or surrogate domain techniques. Finally, the inclusion nodes on the interface are shared with the matrix discretization to provide additional coercivity for a penalty-free Nitsche’s method as well as ensuring the

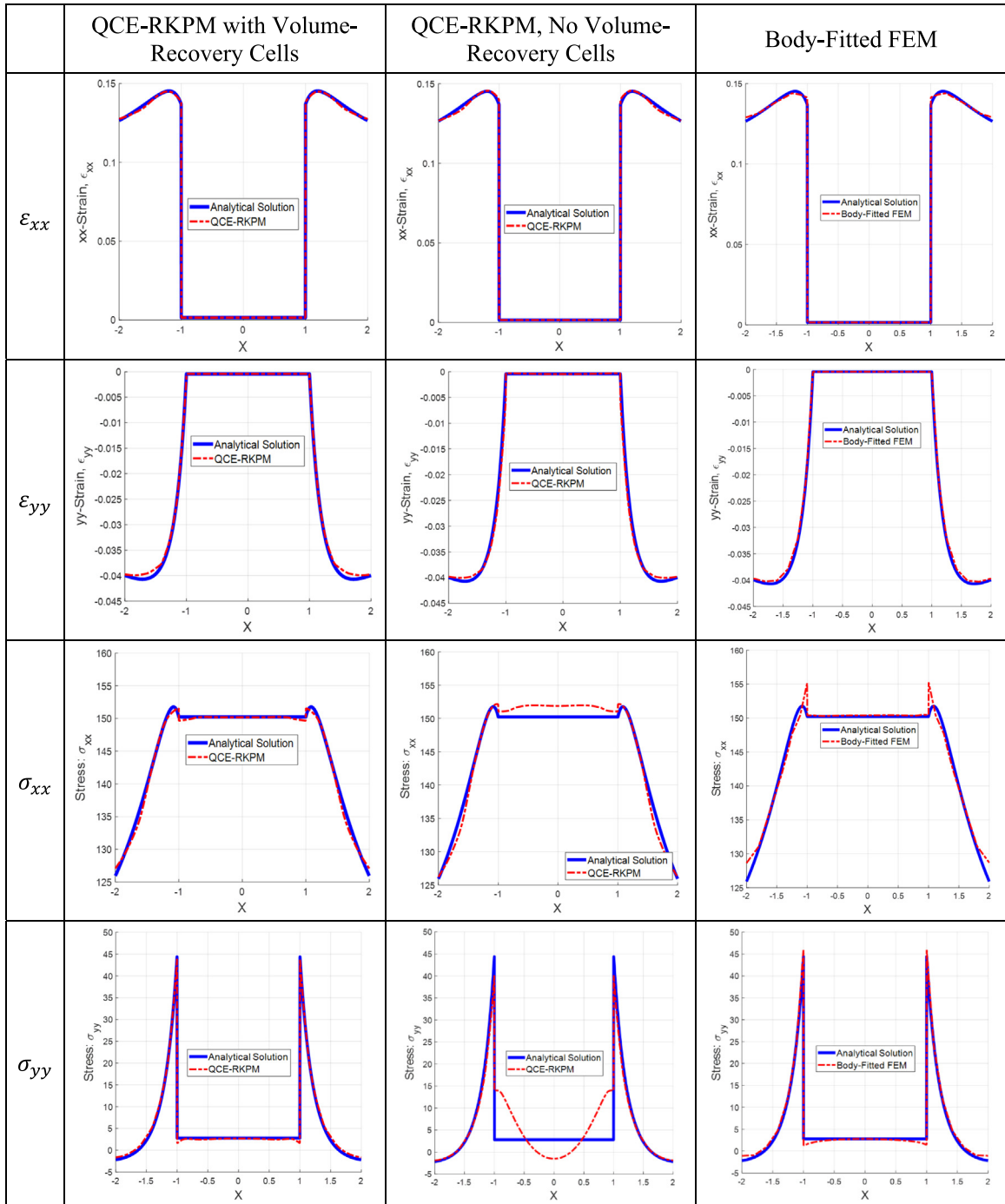


Fig. 19. Cross-sections of the normal strains and stresses along $y = 0$ obtained by QCE-RKPM and Body-fitted FEM with discretizations in Fig. 16.

partition of unity in the approximation in each of the sub-domains. Thus, unlike most immersed and embedded methods, our formulation does not have any tunable parameters present to enforce interface compatibility.

QCE-RKPM was implemented with stabilized conforming/non-conforming nodal integration, variationally consistent correction, NSNI stabilization, and a line-of-sight algorithm. Several numerical examples were given which

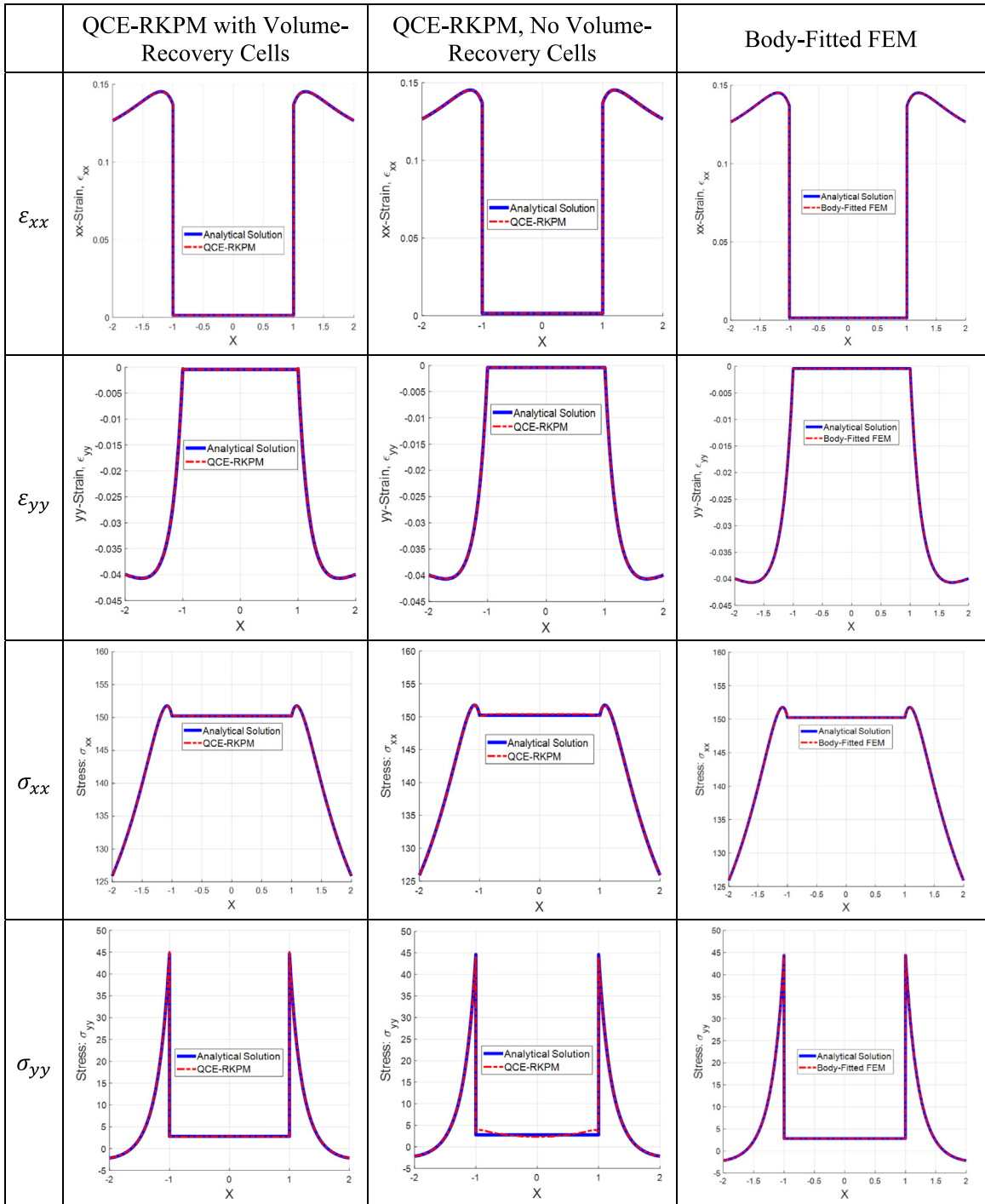


Fig. 20. Cross-sections of the normal strains and stresses along $y = 0$ for the most refined discretization ($h_{\text{Inf}}^+ \approx h^- = 0.0125$).

show that QCE-RKPM is a viable embedded method for modeling weak discontinuities in heterogeneous materials. The first numerical example shows that the proposed QCE-RKPM passes the piecewise linear patch test with weak discontinuities, and in problems with piecewise higher order solutions, it exhibits optimal convergence in the L_2 norm and superconvergence in the H_1 semi-norm. In a two-dimensional example of an infinite plate with a circular

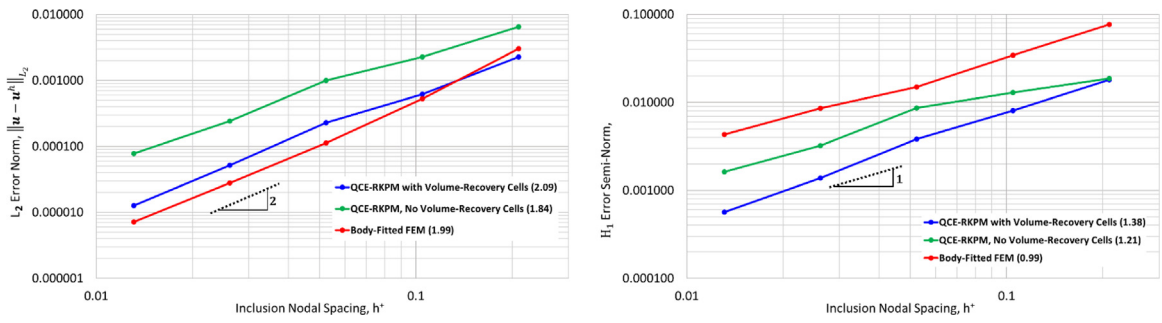


Fig. 21. Convergence plots for the L_2 error norm and the H_1 error semi-norm.

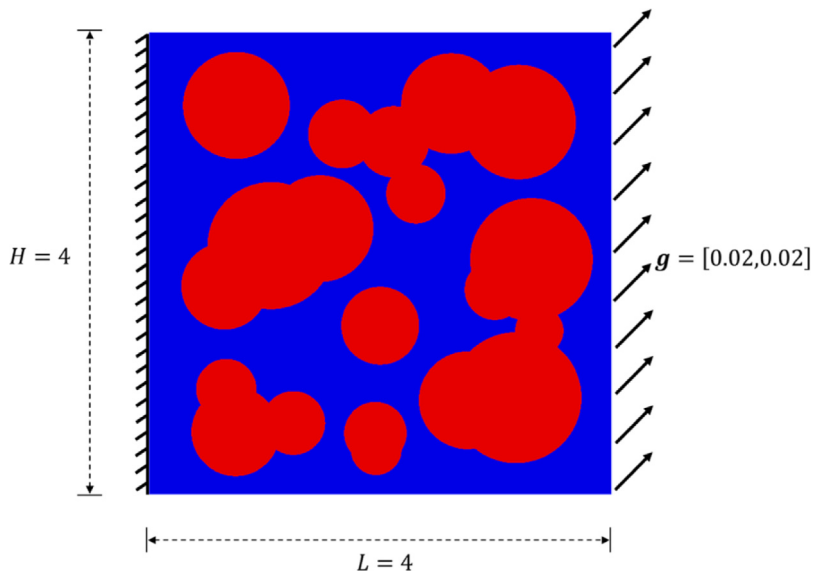


Fig. 22. Combined tensile and shear loading of a heterogeneous microstructure.

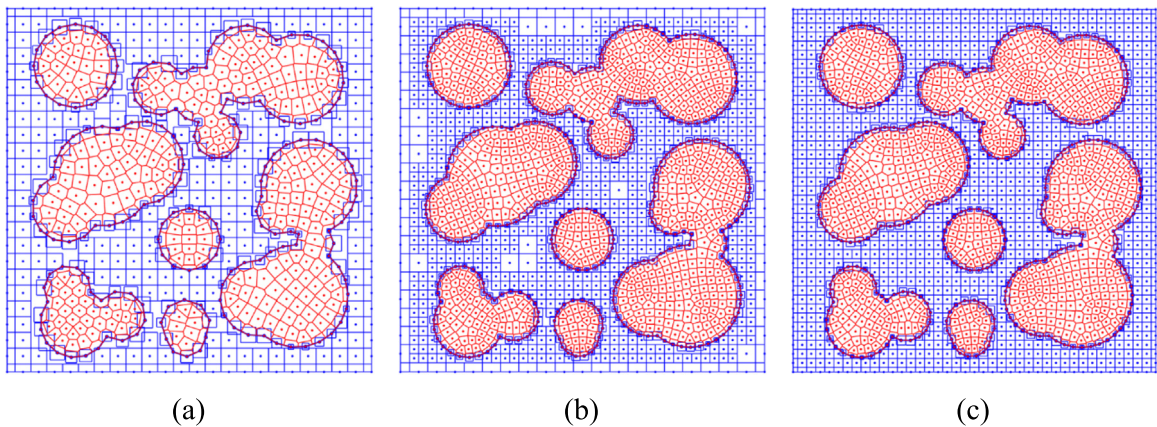


Fig. 23. (a) $h^- \approx h^+ \approx 0.175$ with 709 nodes; (b) $h^- \approx 0.2, h^+ \approx 0.1$ with 1,639 nodes; (c) $h^- \approx h^+ \approx 0.1$ with 1,839 nodes.

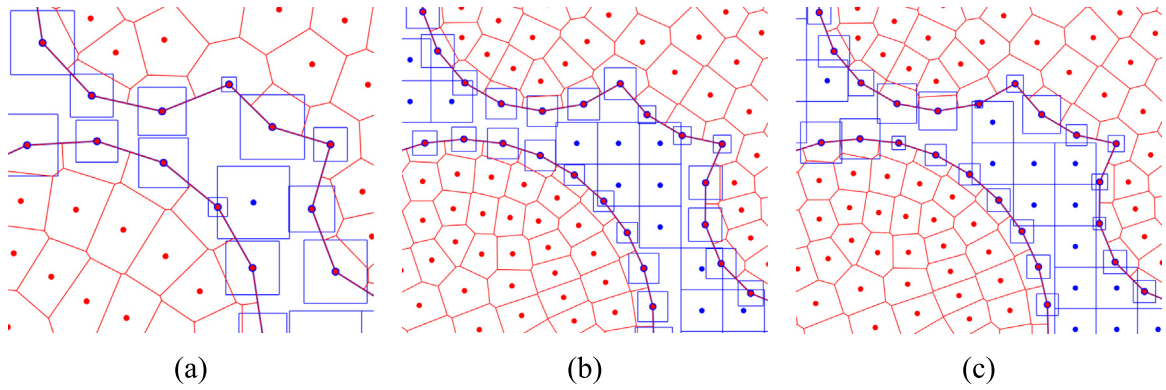


Fig. 24. Magnified views of (a) Fig. 23(a), (b) Fig. 23(b), and (c) Fig. 23(c).

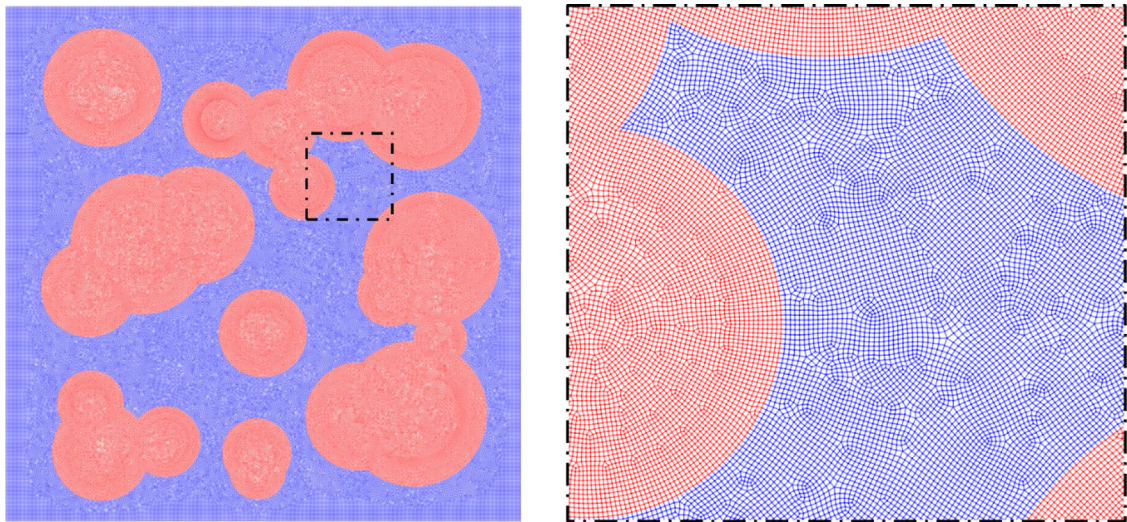


Fig. 25. Conforming FEM discretization with $h^- = h^+ \approx 0.00625$ with 424,063 nodes used as the reference solution.

inclusion, QCE-RKPM with volume-recovery cells achieves accuracy comparable to body-fitted FEM and yields optimal convergence rates in the L_2 error norm. It also shows that QCE-RKPM with or without volume-recovery cells outperforms body-fitted FEM in the H_1 error semi-norm in both overall error and convergence rate. Finally, a two-dimensional example of a multi-inclusion microstructure was simulated using QCE-RKPM and was compared with a highly refined, body-fitted FEM mesh. The QCE-RKPM with much coarser discretizations provided similar results to the highly-refined, body-fitted FEM reference solution.

In summary, our method is distinct from standard, mesh-based immersed and embedded formulations. Unlike immersed methods in general, our formulation does not involve a fictitious domain and thus does not have a discretization ratio limitation. Unlike VCIMs, it presents a sharp discontinuity at the interface which captures the gradient jump. Unlike standard embedded methods, our formulation does not suffer from the small cut element problem, use surrogate domains, or aggregate elements. While we have shown the QCE discretization process in two dimensions and present two-dimensional examples, the extension to three dimensions is straightforward. The quadtree refinement extended to 3D would be to subdivide a cubic background cell into eight smaller cells (i.e., octree refinement). The inclusion of foreground interface nodes in the background approximation is the same, and if one uses the “volume-recovery” cells mentioned in Remark 4.2 above, these cells would be simple cubes centered on the interface nodes.

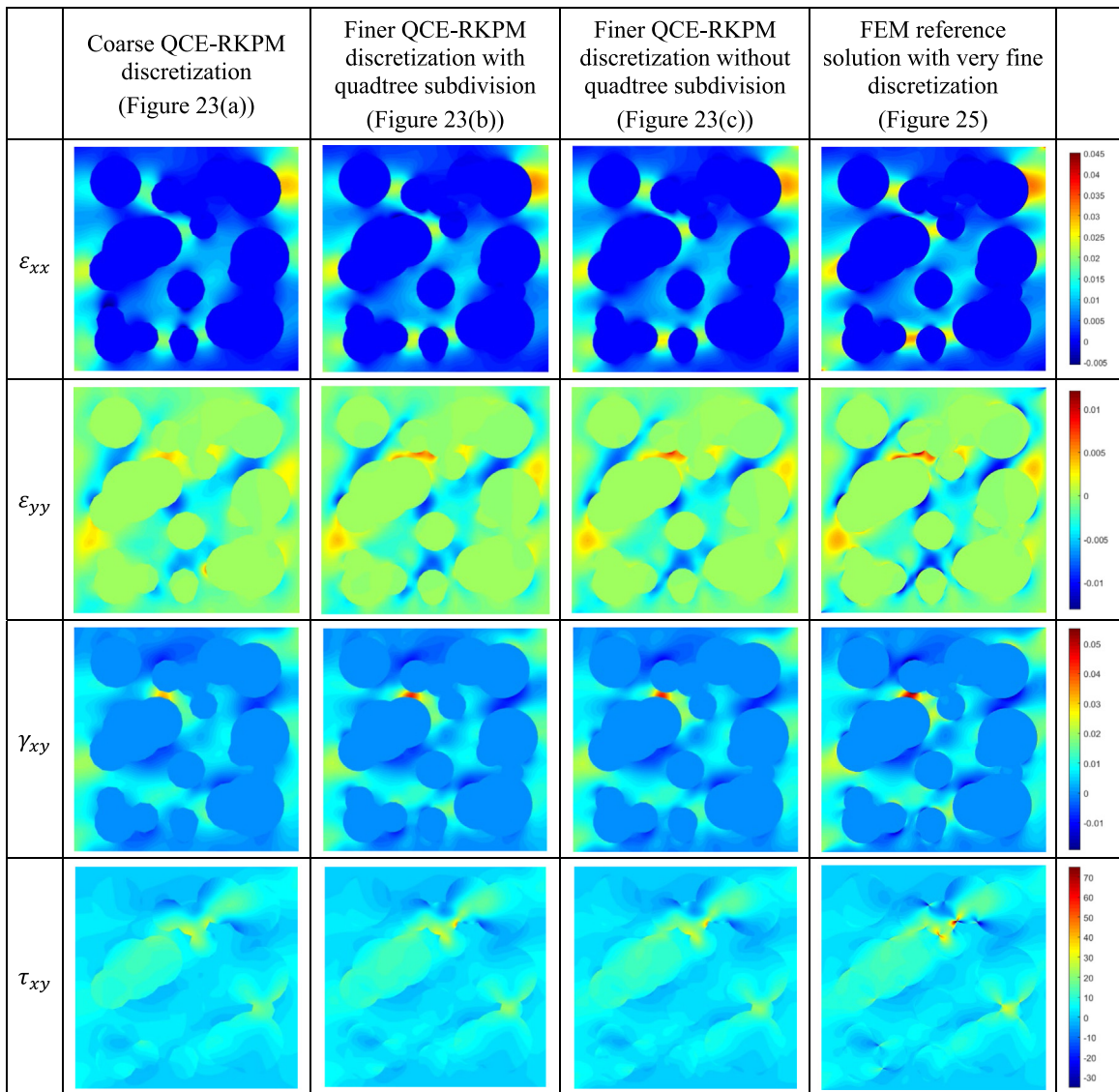


Fig. 26. Strain and τ_{xy} fields for the QCE-RKPM discretizations shown in Fig. 23 and the body-fitted FEM reference discretization shown in Fig. 25.

Declaration of competing interest

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

Data availability

No data was used for the research described in the article.

Acknowledgments

The support of this work by the Sandia National Laboratories to UC San Diego under Contract Agreement 1655264 is greatly acknowledged. Sandia National Laboratories is a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell

International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. The support from National Science Foundation under Award Number CMMI-1826221 to University of California, San Diego is greatly appreciated.

Appendix. Galerkin and matrix forms

Let \mathbf{u}^{h-} and \mathbf{u}^{h+} be the RK approximations of \mathbf{u}^- and \mathbf{u}^+ in Ω^- and Ω^+ , respectively. Substituting these approximations into the weak form of (29) and (30) yields the Galerkin form: Find $(\mathbf{u}^{h+}, \mathbf{u}^{h-}) \in H^1(\Omega^+) \times H^1(\Omega^-)$ such that $\forall (\delta \mathbf{u}^{h+}, \delta \mathbf{u}^{h-}) \in H^1(\Omega^+) \times H^1(\Omega^-)$

$$\begin{aligned} & (\boldsymbol{\varepsilon}(\delta \mathbf{u}^{h-}), \boldsymbol{\sigma}^-(\mathbf{u}^{h-}))_{\Omega^-} + (\delta \mathbf{u}^{h-}, \mathbf{n}^+ \cdot \boldsymbol{\sigma}^+(\mathbf{u}^{h+}))_{\Gamma} - (\delta \mathbf{u}^{h-}, \mathbf{n}^- \cdot \boldsymbol{\sigma}^-(\mathbf{u}^{h-}))_{\partial \Omega_D} \\ & - (\boldsymbol{\sigma}^-(\delta \mathbf{u}^{h-}) \cdot \mathbf{n}^-, \mathbf{u}^{h-})_{\partial \Omega_D} + \beta (\delta \mathbf{u}^{h-}, \mathbf{u}^{h-})_{\partial \Omega_D} \end{aligned} \tag{76}$$

$$\begin{aligned} & = (\delta \mathbf{u}^{h-}, \mathbf{b}^-)_{\Omega^-} + (\delta \mathbf{u}^{h-}, \mathbf{t})_{\partial \Omega_N} - (\boldsymbol{\sigma}^-(\delta \mathbf{u}^{h-}) \cdot \mathbf{n}^-, \mathbf{g})_{\partial \Omega_D} \\ & + \beta (\delta \mathbf{u}^{h-}, \mathbf{g})_{\partial \Omega_D}, \end{aligned} \tag{77}$$

$$(\boldsymbol{\varepsilon}(\delta \mathbf{u}^{h+}), \boldsymbol{\sigma}^+(\mathbf{u}^{h+}))_{\Omega^+} - (\delta \mathbf{u}^{h+}, \mathbf{n}^+ \cdot \boldsymbol{\sigma}^+(\mathbf{u}^{h+}))_{\Gamma} = (\delta \mathbf{u}^{h+}, \mathbf{b}^+)_{\Omega^+}, \tag{77}$$

Using VC-SNNI and SCNI both with NSNI stabilization for the background matrix and foreground inclusion domains, respectively, we arrive at our matrix form:

$$\begin{aligned} & \left[\int_{\Omega^-} \hat{\mathbf{B}}_I^{-T} \mathbf{C}^- \hat{\mathbf{B}}_J^- d\Omega - \int_{\partial \Omega_D} \boldsymbol{\Psi}_I^{-T} \boldsymbol{\eta}^{-T} \mathbf{C}^- \tilde{\mathbf{B}}_J^- d\Gamma - \int_{\partial \Omega_D} \tilde{\mathbf{B}}_I^{-T} \mathbf{C}^- \boldsymbol{\eta}^- \boldsymbol{\Psi}_J^- d\Gamma \right. \\ & \left. + \beta \int_{\partial \Omega_D} \boldsymbol{\Psi}_I^{-T} \boldsymbol{\Psi}_J^- d\Gamma \right] \mathbf{d}_J^- + \int_{\Gamma} \boldsymbol{\Psi}_I^{-T} \boldsymbol{\eta}^{+T} \mathbf{C}^+ \tilde{\mathbf{B}}_J^+ d\Gamma \mathbf{d}_J^+ \end{aligned} \tag{78}$$

$$\begin{aligned} & = \int_{\Omega^-} \boldsymbol{\Psi}_I^{-T} \mathbf{b}^- d\Omega + \int_{\partial \Omega_N} \boldsymbol{\Psi}_I^{-T} \mathbf{t} d\Gamma - \int_{\partial \Omega_D} \tilde{\mathbf{B}}_I^{-T} \mathbf{C}^- \boldsymbol{\eta}^- \mathbf{g} d\Gamma + \beta \int_{\partial \Omega_D} \boldsymbol{\Psi}_I^{-T} \mathbf{g} d\Gamma, \\ & \left[\int_{\Omega^+} \hat{\mathbf{B}}_I^{+T} \mathbf{C}^+ \hat{\mathbf{B}}_J^+ d\Omega - \int_{\Gamma} \boldsymbol{\Psi}_I^{+T} \boldsymbol{\eta}^{+T} \mathbf{C}^+ \tilde{\mathbf{B}}_J^+ d\Gamma \right] \mathbf{d}_J^+ = \int_{\Omega^+} \boldsymbol{\Psi}_I^{+T} \mathbf{b}^+ d\Omega, \end{aligned} \tag{79}$$

or:

$$\begin{bmatrix} \hat{\mathbf{K}}_{IJ}^{--} - \check{\mathbf{K}}_{IJ}^{--} - \check{\mathbf{K}}_{IJ}^{--T} + \mathbf{K}_{IJ}^{\beta--} & \mathbf{K}_{IJ}^{\Gamma-+} \\ \mathbf{0} & \hat{\mathbf{K}}_{IJ}^{++} - \mathbf{K}_{IJ}^{\Gamma++} \end{bmatrix} \begin{bmatrix} \mathbf{d}_J^- \\ \mathbf{d}_J^+ \end{bmatrix} = \begin{bmatrix} \mathbf{f}_I^{b-} + \mathbf{f}_I^{t-} - \check{\mathbf{f}}_I^{g-} + \mathbf{f}_I^{\beta-} \\ \mathbf{f}_I^{b+} \end{bmatrix}, \tag{80}$$

where:

$$\hat{\mathbf{K}}_{IJ}^{--} = \int_{\Omega^-} \hat{\mathbf{B}}_I^{-T} \mathbf{C}^- \hat{\mathbf{B}}_J^- d\Omega, \tag{81}$$

$$\check{\mathbf{K}}_{IJ}^{--} = \int_{\partial \Omega_D} \boldsymbol{\Psi}_I^{-T} \boldsymbol{\eta}^{-T} \mathbf{C}^- \tilde{\mathbf{B}}_J^- d\Gamma, \tag{82}$$

$$\mathbf{K}_{IJ}^{\beta--} = \beta \int_{\partial \Omega_D} \boldsymbol{\Psi}_I^{-T} \boldsymbol{\Psi}_J^- d\Gamma, \tag{83}$$

$$\mathbf{K}_{IJ}^{\Gamma-+} = \int_{\Gamma} \boldsymbol{\Psi}_I^{-T} \boldsymbol{\eta}^{+T} \mathbf{C}^+ \tilde{\mathbf{B}}_J^+ d\Gamma, \tag{84}$$

$$\mathbf{f}_I^{b-} = \int_{\Omega^-} \boldsymbol{\Psi}_I^{-T} \mathbf{b}^- d\Omega, \tag{85}$$

$$\mathbf{f}_I^{t-} = \int_{\partial \Omega_N} \boldsymbol{\Psi}_I^{-T} \mathbf{t} d\Gamma, \tag{86}$$

$$\check{\mathbf{f}}_I^{g-} = \int_{\partial \Omega_D} \tilde{\mathbf{B}}_I^{-T} \mathbf{C}^- \boldsymbol{\eta}^- \mathbf{g} d\Gamma, \tag{87}$$

$$\mathbf{f}_I^{\beta-} = \beta \int_{\partial\Omega_D} \boldsymbol{\Psi}_I^{-T} \mathbf{g} d\Gamma, \quad (88)$$

$$\hat{\mathbf{K}}_{IJ}^{++} = \int_{\Omega^+} \hat{\mathbf{B}}_I^{+T} \mathbf{C}^+ \hat{\mathbf{B}}_J^+ d\Omega, \quad (89)$$

$$\mathbf{K}_{IJ}^{\Gamma++} = \int_{\Gamma} \boldsymbol{\Psi}_I^{+T} \boldsymbol{\eta}^{+T} \mathbf{C}^+ \tilde{\mathbf{B}}_J^+ d\Gamma, \quad (90)$$

$$\mathbf{f}_I^{b+} = \int_{\Omega^+} \boldsymbol{\Psi}_I^{+T} \mathbf{b}^+ d\Omega. \quad (91)$$

and where \mathbf{d}_J^- and \mathbf{d}_J^+ are the background and foreground generalized nodal displacements, respectively. Again, $\hat{\mathbf{B}}_I$ is a VC-corrected and NSNI stabilized smoothed strain–displacement matrix, $\tilde{\mathbf{B}}_I$ is a NSNI stabilized smoothed strain–displacement matrix, \mathbf{B}_I is a smoothed strain–displacement matrix, $\boldsymbol{\Psi}_I = \boldsymbol{\Psi}_I \mathbf{I}$ is the multi-dimensional shape function matrix, \mathbf{I} is the identity matrix, and $\boldsymbol{\eta}$ is the matrix of outward normal vector components on either the interface Γ or the outer boundary $\partial\Omega$.

Remark A.1. Note that the smoothed strain–displacement matrices used for domain integration have been reused in the contour integrals for computational efficiency. From numerical experiments, we have determined that using the smoothed gradient in the contour integrals works just as well as using the direct gradient of the RK shape function.

References

- [1] C.S. Peskin, Flow patterns around heart valves: A numerical method, *J. Comput. Phys.* 10 (2) (1972) 252–271.
- [2] C.S. Peskin, The immersed boundary method, *Acta Numer.* 11 (2002) 479–517.
- [3] D. Boffi, L. Gastaldi, A finite element approach for the immersed boundary method, *Comput. Struct.* 81 (8–11) (2003) 491–501.
- [4] D. Boffi, L. Gastaldi, L. Heltai, Numerical stability of the finite element immersed boundary method, *Math. Models Methods Appl. Sci.* 17 (10) (2007) 1479–1505.
- [5] B.E. Griffith, X. Luo, Hybrid finite difference/finite element immersed boundary method, *Int. J. Numer. Methods Biomed. Eng.* 33 (12) (2017) e2888.
- [6] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid–structure interaction, *Prog. Aerosp. Sci.* 65 (2014) 1–21.
- [7] W. Kim, H. Choi, Immersed boundary methods for fluid–structure interaction: A review, *Int. J. Heat Fluid Flow* 75 (2019) 301–309.
- [8] Y. Bazilevs, K. Kamran, G. Moutsanidis, D. Benson, E. Oñate, A new formulation for air-blast fluid–structure interaction using an immersed approach. Part I: basic methodology and FEM-based simulations, *Comput. Mech.* 60 (2017) 83–100.
- [9] Y. Bazilevs, G. Moutsanidis, J. Bueno, K. Kamran, D. Kamensky, M.C. Hillman, H. Gomez, J.-S. Chen, A new formulation for air-blast fluid–structure interaction using an immersed approach: Part II—coupling of IGA and meshfree discretizations, *Comput. Mech.* 60 (1) (2017) 101–116.
- [10] J.A. Cottrell, T.J. Hughes, Y. Bazilevs, *Isogeometric Analysis: Toward Integration of CAD and FEA*, John Wiley & Sons, New York, 2009.
- [11] M. Behzadinasab, G. Moutsanidis, N. Trask, J.T. Foster, Y. Bazilevs, Coupling of IGA and peridynamics for air-blast fluid–structure interaction using an immersed approach, *Forces Mech.* 4 (2021) 100045.
- [12] R. Glowinsky, T.-W. Pan, T.I. Hesla, D.D. Joseph, J. Periaux, A distributed Lagrange multiplier/fictitious domain method for the simulation of flow around moving rigid bodies: application to particulate flow, *Comput. Methods Appl. Mech. Engrg.* 184 (2–4) (2000) 241–267.
- [13] I. Babuška, The finite element method with Lagrangian multipliers, *Numer. Math.* 20 (3) (1973) 179–192.
- [14] D. Boffi, F. Brezzi, M. Fortin, *Mixed Finite Element Methods and Applications*, Vol. 44, Springer, Heidelberg, 2013.
- [15] L. Zhang, A. Gerstenberger, X. Wang, W.K. Liu, Immersed finite element method, *Comput. Methods Appl. Mech. Engrg.* 193 (2051) (2004) 21–22–2067.
- [16] L.T. Zhang, M. Gay, Immersed finite element method for fluid–structure interactions, *J. Fluids Struct.* 23 (6) (2007) 839–857.
- [17] X. Wang, L.T. Zhang, Modified immersed finite element method for fully-coupled fluid–structure interactions, *Comput. Methods Appl. Mech. Engrg.* 267 (2013) 150–169.
- [18] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, *Int. J. Numer. Methods Fluids* 20 (1081) (1995) 8–9–1106.
- [19] J.-S. Chen, C. Pan, C. Wu, W.K. Liu, Reproducing kernel particle methods for large deformation analysis of nonlinear structures, *Comput. Methods Appl. Mech. Engrg.* 139 (1–4) (1996) 195–227.
- [20] X. Wang, L.T. Zhang, Interpolation functions in the immersed boundary and finite element methods, *Comput. Mech.* 45 (4) (2010) 321–334.
- [21] T.-H. Huang, J.-S. Chen, M.R. Tupek, F.N. Beckwith, J.J. Koester, H.E. Fang, A variational multiscale immersed meshfree method for heterogeneous materials, *Comput. Mech.* 67 (4) (2021) 1059–1097.
- [22] T.J. Hughes, L.P. Franca, M. Balestra, A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuška-Brezzi condition: A stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations, *Comput. Methods Appl. Mech. Engrg.* 59 (1) (1986) 85–99.

- [23] T.J. Hughes, G.R. Feijóo, L. Mazzei, J.-B. Quincy, The variational multiscale method—A paradigm for computational mechanics, *Comput. Methods Appl. Mech. Engrg.* 166 (1–2) (1998) 3–24.
- [24] T.-H. Huang, J.-S. Chen, M.R. Tupek, F.N. Beckwith, H.E. Fang, A variational multiscale immersed meshfree method for fluid structure interactive systems involving shock waves, *Comput. Methods Appl. Mech. Engrg.* 389 (2022) 114396.
- [25] B.E. Griffith, N.A. Patankar, Immersed methods for fluid–structure interaction, *Annu. Rev. Fluid Mech.* 52 (2020) 421–448.
- [26] J. Wang, G. Zhou, M. Hillman, A. Madra, Y. Bazilevs, J. Du, K. Su, Consistent immersed volumetric Nitsche methods for composite analysis, *Comput. Methods Appl. Mech. Engrg.* 385 (2021) 114042.
- [27] A. Hansbo, P. Hansbo, An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems, *Comput. Methods Appl. Mech. Eng.* 191 (5537) (2002) 47–48–5552.
- [28] E. Burman, S. Claus, P. Hansbo, M.G. Larson, A. Massing, CutFEM: Discretizing geometry and partial differential equations, *Internat. J. Numer. Methods Engrg.* 104 (7) (2015) 472–501.
- [29] S. Soghrati, A.M. Aragón, C.A. Duarte, P.H. Geubelle, An interface-enriched generalized FEM for problems with discontinuous gradient fields, *Internat. J. Numer. Methods Engrg.* 89 (8) (2012) 991–1008.
- [30] S. Soghrati, Hierarchical interface-enriched finite element method: An automated technique for mesh-independent simulations, *J. Comput. Phys.* 275 (2014) 41–52.
- [31] A.M. Aragón, A. Simone, The discontinuity-enriched finite element method, *Internat. J. Numer. Methods Engrg.* 112 (11) (2017) 1589–1613.
- [32] E. Burman, P. Hansbo, Fictitious domain finite element methods using cut elements: II. A stabilized Nitsche method, *Appl. Numer. Math.* 62 (4) (2012) 328–341.
- [33] P. Hansbo, M.G. Larson, S. Zahedi, A cut finite element method for a Stokes interface problem, *Appl. Numer. Math.* 85 (2014) 90–114.
- [34] E. Burman, P. Hansbo, Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes’ problem, *ESAIM: Math. Model. Numer. Anal.* 48 (3) (2014) 859–874.
- [35] H.M. Mourad, J. Dolbow, I. Harari, A bubble-stabilized finite element method for Dirichlet constraints on embedded interfaces, *Internat. J. Numer. Methods Engrg.* 69 (4) (2007) 772–793.
- [36] J.E. Dolbow, L.P. Franca, Residual-free bubbles for embedded Dirichlet problems, *Comput. Methods Appl. Mech. Engrg.* 197 (3751) (2008) 45–48–3759.
- [37] C. Annavarapu, M. Hautefeuille, J.E. Dolbow, A robust Nitsche’s formulation for interface problems, *Comput. Methods Appl. Mech. Engrg.* 225–228 (2012) 44–54.
- [38] W. Jiang, C. Annavarapu, J.E. Dolbow, I. Harari, A robust Nitsche’s formulation for interface problems with spline-based finite elements, *Internat. J. Numer. Methods Engrg.* 104 (7) (2015) 676–696.
- [39] J. Parvizian, A. Düster, E. Rank, Finite cell method: h- and p-extension for embedded domain problems in solid mechanics, *Comput. Mech.* 41 (1) (2007) 121–133.
- [40] A. Düster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, *Comput. Methods Appl. Mech. Engrg.* 197 (2008) 3768–3782.
- [41] D. Schillinger, M. Ruess, The finite cell method: A review in the context of higher-order structural analysis of CAD and image-based geometric models, *Arch. Comput. Methods Eng.* 22 (3) (2015) 391–455.
- [42] D. Kamensky, M.-C. Hsu, D. Schillinger, J.A. Evans, A. Aggarwal, Y. Bazilevs, M.S. Sacks, T.J. Hughes, An immersogeometric variational framework for fluid–structure interaction: Application to bioprosthetic heart valves, *Comput. Methods Appl. Mech. Engrg.* 284 (2015) 1005–1053.
- [43] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part I: poisson and stokes problems, *J. Comput. Phys.* 372 (2018) 972–995.
- [44] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part II: linear advection–diffusion and incompressible Navier–Stokes equations, *J. Comput. Phys.* 372 (2018) 996–1026.
- [45] K. Li, N.M. Atallah, G.A. Main, G. Scovazzi, The shifted interface method: A flexible approach to embedded interface computations, *Internat. J. Numer. Methods Engrg.* 121 (3) (2020) 492–518.
- [46] J. Cheung, M. Perego, P. Bochev, M. Gunzburger, Optimally accurate higher-order finite element methods for polytopial approximations of domains with smooth boundaries, *Math. Comp.* 88 (319) (2019) 2187–2219.
- [47] J. Cheung, M. Gunzburger, P. Bochev, M. Perego, An optimally convergent higher-order finite element coupling method for interface and domain decomposition problems, *Results Appl. Math.* 6 (2020) 100094.
- [48] N.M. Atallah, C. Canuto, G. Scovazzi, The high-order shifted boundary method and its analysis, *Comput. Methods Appl. Mech. Engrg.* 394 (2022) 114885.
- [49] L. Nogueira, M. Ricchiuto, G. Scovazzi, High-order gradients with the shifted boundary method: An embedded enriched mixed formulation for elliptic PDEs, *J. Comput. Phys.* 398 (2019) 108898.
- [50] S. Badia, F. Verdugo, A.F. Martín, The aggregated unfitted finite element method for elliptic problems, *Comput. Methods Appl. Mech. Engrg.* 336 (2018) 533–553.
- [51] S. Badia, A.F. Martín, F. Verdugo, Mixed aggregated finite element methods for the unfitted discretization of the Stokes problem, *SIAM J. Sci. Comput.* 40 (6) (2018) B1541–B1576.
- [52] S. Badia, E. Neiva, F. Verdugo, Linking ghost penalty and aggregated unfitted methods, *Comput. Methods Appl. Mech. Engrg.* 388 (2022) 114232.
- [53] A. Johansson, M.G. Larson, A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary, *Numer. Math.* 123 (4) (2013) 607–628.
- [54] W. Qiu, M. Solano, P. Vega, A high order HDG method for curved-interface problems via approximations from straight triangulations, *J. Sci. Comput.* 69 (3) (2016) 1384–1407.

- [55] P. Huang, H. Wu, Y. Xiao, An unfitted interface penalty finite element method for elliptic interface problems, *Comput. Methods Appl. Mech. Engrg.* 323 (2017) 439–460.
- [56] E. Burman, P. Hansbo, M.G. Larson, Cutfem based on extended finite element spaces, *Numer. Math.* 152 (2) (2022) 331–369.
- [57] J.-S. Chen, M. Hillman, M. Ruter, An arbitrary order variationally consistent integration for Galerkin meshfree methods, *Internat. J. Numer. Methods Engrg.* 95 (5) (2013) 387–418.
- [58] D. Wang, Y. Sun, L. Li, A discontinuous Galerkin meshfree modeling of material interface, *CMES Comput. Model. Eng. Sci.* 45 (1) (2009) 57–82.
- [59] T. Belytschko, J.-S. Chen, M. Hillman, *Meshfree and Particle Methods*, Publishing Company, Hoboken, NJ: Wiley, forthcoming.
- [60] T. Belytschko, Y.Y. Lu, L. Gu, Element-free Galerkin methods, *Internat. J. Numer. Methods Engrg.* 37 (2) (1994) 229–256.
- [61] P. Krysl, T. Belytschko, Element-free Galerkin method: Convergence of the continuous and discontinuous shape functions, *Comput. Methods Appl. Mech. Engrg.* 148 (3–4) (1997) 257–277.
- [62] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless methods: An overview and recent developments, *Comput. Methods Appl. Mech. Engrg.* 139 (1–4) (1996) 3–47.
- [63] D. Organ, M. Fleming, T. Terry, T. Belytschko, Continuous meshless approximations for nonconvex bodies by diffraction and transparency, *Comput. Mech.* 18 (1996) 225–235.
- [64] J. Dolbow, T. Belytschko, Numerical integration of the Galerkin weak form in meshfree methods, *Comput. Mech.* 23 (3) (1999) 219–230.
- [65] J.-S. Chen, C.-T. Wu, S. Yoon, Y. You, A stabilized conforming nodal integration for Galerkin meshfree methods, *Internat. J. Numer. Methods Engrg.* 50 (2) (2001) 435–466.
- [66] M. Hillman, J.-S. Chen, An accelerated, convergent, and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics, *Internat. J. Numer. Methods Engrg.* 107 (7) (2016) 603–630.
- [67] J.-S. Chen, M. Hillman, S.-W. Chi, Meshfree methods: Progress made after 20 years, *J. Eng. Mech.* 143 (4) (2017) 04017001.
- [68] J.-S. Chen, Y. Wu, Stability in Lagrangian and semi-Lagrangian reproducing kernel discretizations using nodal integration in nonlinear solid mechanics, in: *Computational Methods in Applied Sciences*, Dordrecht, 2007.
- [69] P.-C. Guan, J.-S. Chen, Y. Wu, H. Teng, J. Gaidos, K. Hofstetter, M. Alsaleh, Semi-Lagrangian reproducing kernel formulation and application to modeling earth moving operations, *Mech. Mater.* 41 (6) (2009) 670–683.
- [70] J.-S. Chen, W. Hu, M.A. Puso, Y. Wu, X. Zhang, Strain smoothing for stabilization and regularization of Galerkin meshfree method, in: *Meshfree Methods for Partial Differential Equations III*, Vol. 57, Springer, Berlin, 2007, pp. 57–76.
- [71] M.A. Puso, J.-S. Chen, E. Zywickz, W. Elmer, Meshfree and finite element nodal integration methods, *Internat. J. Numer. Methods Engrg.* 74 (3) (2008) 416–446.
- [72] S. Li, W.K. Liu, Reproducing kernel hierarchical partition of unity, Part I—formulation and theory, *Internat. J. Numer. Methods Engrg.* 45 (3) (1999) 251–288.
- [73] S. Li, W.K. Liu, Reproducing kernel hierarchical partition of unity, Part II—applications, *Internat. J. Numer. Methods Engrg.* 45 (3) (1999) 289–317.
- [74] J.-S. Chen, X. Zhang, T. Belytschko, An implicit gradient model by a reproducing kernel strain regularization in strain localization problems, *Comput. Methods Appl. Mech. Engrg.* 193 (2827) (2004) 27–29–2844.
- [75] H.-Y. Hu, C.-K. Lai, J.-S. Chen, A study on convergence and complexity of reproducing kernel collocation method, *Interact. Multiscale Mech.* 2 (3) (2009) 295–319.
- [76] S.-W. Chi, J.-S. Chen, H.-Y. Hu, J.P. Yang, A gradient reproducing kernel collocation method for boundary value problems, *Internat. J. Numer. Methods Engrg.* 93 (13) (2013) 1381–1402.
- [77] M.L. Kachanov, B. Shafiro, I. Tsukrov, *Handbook of Elasticity Solutions*, Springer Science & Business Media, Dordrecht, 2003.
- [78] Simulia Corp, *Abaqus/CAE User's Guide*, Simulia Corp., 2016, [Online]. Available: Accessed 10 2020.