Coarsened C^1-continuous shape functions using B-spline patches for simulating nonlinear heterogeneous structures

Wei Chen[a], Ming Li[a,*]

[a]*State Key Laboratory of CAD&CG, Zhejiang University, Hangzhou, China*

Abstract

Numerical coarsening is an approach that constructs coarsened shape (or basis) functions for simulating heterogeneous structures on a coarse mesh, and seeks the fine mesh solution on a finite dimensional space spanned by these shape functions. It avoids the challenging issue of conforming meshing, reduces computational costs while maintaining high simulation accuracy, and has huge potential in the simulation of composites or even complex CAD models in fictitious domain methods. The C^1-continuity of the coarsened shape functions is of vital importance in determining the simulation quality, such as the continuity of stress solution, but till now has not been developed. In addressing the issue, the study proposes to construct coarsened shape functions using B-spline patches (piecewise surfaces in 2D or volumes in 3D). By carefully devising an intermediate interpolation patch that covers the coarse element boundaries, important properties of C^1-continuity and partition of unity are achieved, being able to produce high simulation stability and accuracy. By being explicitly formulated as two sequent displacement mappings from coarse nodes to the boundary fine nodes and then to the interior fine nodes, the shape functions are cheaply derived with small efforts in solving a small-size linear system for each coarse element. Performance of the coarsened shape functions, implemented within the material point method (MPM), is demonstrated through various numerical examples in comparisons with benchmark results in terms of stability, simulation accuracy, and computational efficiency.

Keywords: Numerical coarsening; Coarsened shape functions; C^1-continuity; B-spline patch; Coarsened material point method (cMPM);

1. Introduction

Heterogeneous structures, such as alloys or reinforced composites, have found tremendous applications in automotive industry, aerospace and mechanical engineering [1, 2, 3, 4, 5, 6]. Additionally, heterogeneous structures can also be used to describe complex CAD models or cellular models in their physical simulation in fictitious domain methods that avoid tedious or unstable process of defeaturing or conformal mesh generation [7, 8, 9, 10, 11]. An efficient and reliable simulation of the physical behavior of a heterogeneous structure is thus a fundamental issue in applied mechanics. However, the fine-scale material heterogeneities and/or tiny geometric features of the structure make an outstanding numerical method pretty challenging, particularly in cases of nonlinear large deformations. Classical finite element method (FEM) [12] or even the recent finite cell method (FCM) [11, 13] requires a mesh fine enough to resolve the fine-scale heterogeneities so as to capture the proper behavior of a heterogeneous structure for the overall solution computation or element matrix integration, leading to prohibitive computational loads. Developing an appropriate approach that balances computational accuracy and efficiency is pressingly required.

Numerical coarsening, or extended multiscale finite element method (EMsFEM), has been promised as a powerful approach to address the challenges of heterogeneous structure simulation [14, 15, 16, 17]. Different from the widely studied numerical homogenization [18, 19], which usually assumes scale separation and periodic distribution, numerical coarsening tries to construct coarsened shape (or basis) functions to represent the solution on a coarse mesh [14, 15, 16]. The shape functions are constructed for each coarse element, and the fine mesh solution is sought on a finite dimensional space spanned by these shape functions [20, 21]. Continuity of the coarsened shape functions, or more specifically its continuity across the element boundaries, is of vital importance in determining the quality of the simulation solution. For example, it is crucial in producing a continuous stress/strain field with no post-process required, achieving better accuracy and higher convergence rate [22], or a key factor in avoiding cell-crossing artifact in the material point method (MPM)

*Corresponding author: liming@cad.zju.edu.cn

for nonlinear large deformations [23, 24]. Besides, it's also required for several numerical analysis problems involving second derivatives of the unknown function in the potential energy, such as the Kirchhoff-Love shell theory [25] and strain gradient elasticity [26, 27]. However, although a variety of coarsened shape functions have been proposed [20, 21, 17, 28] and the importance of C^1-continuity has been frequently addressed, no coarsened shape function of C^1-continuity has been developed as far as we know. The topic is to be addressed in this study for the challenging case of the nonlinear deformation of heterogeneous structures.

In this study, novel coarsened shape functions of C^1-continuity are constructed using B-spline surfaces/volumes in 2D/3D (or B-spline patches for short) for simulations of nonlinear heterogeneous structures. They are used together within a material point method (MPM) framework, called *cMPM*, to demonstrate their performances, considering the MPM's robustness in avoiding element distortions in nonlinear large deformations. The study has the following contributions:

- Coarsened shape functions of C^1-continuity are constructed for the first time as free-form B-spline patches, being able to capture the fine-scale heterogeneities within the coarse element that allows high simulation accuracy and continuous stress/strain solution fields.
- The coarse element possesses additional DOFs along its boundaries, besides those on its vertices, whose number can be set to balance different requirements of simulation accuracy or efficiency.
- The coarsened shape functions are matrix-valued to account for the anisotropy of the coarse element. They satisfy the important property of *partition of unity*, which avoids some un-physical simulation results.
- The coarsened shape functions are derived for each coarse element as two sequent mappings from coarse nodes to the boundary fine nodes, and then to the interior fine nodes. The process is very cheaply achieved with only small efforts in solving a small-size linear system.
- Using the coarsened shape functions, a cMPM for simulating nonlinear heterogeneous structures is devised. It has an efficiency improvement with an order of magnitude and eliminates the cell-crossing artifact inherent in conventional MPM caused by the discontinuous derivatives of shape functions [23, 24].
- Performance of the coarsened shape functions is demonstrated through various numerical examples in comparisons with benchmark results in terms of stability, simulation accuracy, and computational efficiency.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. The problem and preliminaries on coarsened B-spline patch shape functions are described in Section 3. Their construction is then outlined in Section 4, and detailed in Section 5. Properties of the shape functions are discussed in Section 6, and their extension to 3D, combination with the MPM are presented in Section 7. After demonstrating extensive numerical examples in Section 8, we conclude the study in Section 9.

2. Related work

2.1. Simulation of heterogeneous structures

Simulation of heterogeneous structures can be achieved in several different ways to reduce the computational loads while capturing proper physical behavior. The multigrid/multilevel approaches use the residual equation on a coarse mesh to relax the error [29, 30]. Nevertheless, most multigrid methods simply consider geometric prolongation/restriction operators and may have low simulation accuracy when dealing with high-contrast materials. Substructuring applies static condensation for each coarse element (or superelement) to eliminate the internal degrees of freedom [31, 32]. It achieves higher simulation accuracy for heterogeneous structures but produces a denser global stiffness matrix and is only applicable to linear elastic models.

The numerical homogenization methods replace each coarse element with an averaged effective property by solving a local characteristic simulation problem [18, 19]. The multilevel FEM (FE^2) conducts the analysis through an iterative transition between fields (stress and strain) in the macroscale and microscale until reaching convergence [33, 34]. However, both methods are based on an assumption of scale separation and periodicity, which does not apply to general heterogeneous structures as studied here. Several strategies for removing this assumption are proposed, e.g. high-order computational homogenization approaches [35, 36] or direct FE^2 [37]. Among these approaches, a second-order two-scale computational homogenization procedure [38] with C^1-continuous solution is devised by taking the C^1 2D triangular finite elements as the macroscale discretization and the C^0 quadrilateral finite elements as the discretization of microscale RVE (representative volume element).

Numerical coarsening, tries to construct on a coarse mesh material-aware shape functions for each coarse element that closely capture the deformation of its interior heterogeneity [14, 15, 16], instead of using the classical polynomial

shape functions. They have demonstrated high simulation accuracy for heterogeneous structures of non-separated scales. The coarsened shape functions are usually represented in a piecewise linear form [15, 20] and derived via a spectral expansion [14, 16] or a constrained optimization problem [20]. EMsFEM [39, 40] constructs the numerical shape functions by solving the static equilibrium equation for each coarse element, which has been successfully applied to small strain analysis [41], dynamic analysis [42, 43], and geometric nonlinear analysis [44, 45] of heterogeneous materials, and linear, periodic, oversampling, and oversampling periodic boundary displacement constraints [46] are proposed for the local static equilibrium.Most of these previous approaches generally have no geometric continuity across the element boundaries, e.g. discontinuous Galerkin methods [47], or only have C^0-continuity via value blending along the element boundary [15], or being softly imposed as optimization constraints [20]. Recently, global continuous deformation or C^0-continuity is achieved by interpolating common boundary of adjacent coarse elements from curved bridge nodes (CBN) [21]. The idea of boundary interpolation is to be extended in this study to devise coarsened shape functions of C^1-continuity. Note that high-order multiscale shape functions are also proposed based on p-FEM [48, 49], but they only have C^0-continuity across the coarse element boundaries. As a consequence, these approaches are prone to cause separations/overlaps between coarse elements, or to produce discontinuous stress/strain fields, or to cause cell-crossing artifact in MPM [24].

In addition, the basic geometric properties of shape functions, such as the partition of unity, are seldom explored in previous studies, and thus they may inevitably produce un-physical results [50]. To better exploit the anisotropic behavior of the complex heterogeneous objects, matrix-valued shape functions are introduced to account for interactions among quantities along different axes [20]. A similar matrix form is also adopted in this study.

2.2. Cell-crossing artifact in MPM

The material point method (MPM) [23, 51] bypasses the mesh distortion issue due to the absence of a conforming mesh, and consequently outperforms the conventional mesh-based approaches (e.g. FEM) when dealing with large deformations. However, the conventional MPM [23, 51] undergoes the cell-crossing artifact [24, 52] when particles cross the cell boundaries, and this artifact arises from the discontinuous derivatives of C^0 linear shape functions in conventional MPM [23]. Various approaches have been proposed to deal with the cell-crossing artifact.

The generalized interpolation material point (GIMP) method [24] resolves the cell-crossing artifact by representing each particle as a characteristic function instead of an infinitesimal point. Based on the former, C^1-continuous GIMP weighting functions are constructed. The convected particle domain interpolation (CPDI) [53] further uses fully tracked particle domains that deform with the deformation gradients, and takes them as parallelograms, quadrilaterals, triangles, or polygons [54] in 2D, and tetrahedrons or polyhedrons [55] in 3D, to reduce the gaps and overlaps between particles. However, these methods require a large computation load for mesh-generation or remeshing (as the particle domains are distorted), which is not consistent with the spirit of meshfree methods.

The total Lagrangian MPM (TLMPM) [56] efficiently overcomes the artifact by not resetting the background mesh. However, it is susceptible to mesh distortions since the mesh "moves" with particles, similar to the FEM. The dual domain MPM (DDMPM) [57] is proposed to develop C^1-continuous shape functions for unstructured grids by first mapping the particle stresses to the grid nodes and then interpolating them to obtain a continuous stress field. The B-spline MPM (BSMPM) approach [58, 59], which directly adopts high-order B-spline basis functions, eliminats the cell-crossing artifact. It is straightforward to implement and has a higher convergence rate than that of the conventional MPM. Similar to BSMPM, smooth B-spline patch shape functions are proposed in this study to address the cell-crossing artifact while greatly improving the computational efficiency due to its employed coarse mesh.

3. Problem and preliminaries

3.1. Dynamic simulation of heterogeneous structures

The problem formulation of the dynamic simulation of a heterogeneous nonlinear elastic structure is first described. Note that a 2D description is used throughout this study.

Given a continuum body $\Omega \in \mathbb{R}^2$ of heterogeneous nonlinear elastic material (Neo-Hookean in this study), its governing equations in an updated Lagrangian description are given by:

$$\begin{aligned} &\frac{D\rho}{Dt} + \rho\nabla\cdot\mathbf{v} = 0, && \text{in } \Omega \\ &\rho\mathbf{a} = \nabla\cdot\boldsymbol{\sigma} + \rho\mathbf{b}, && \text{in } \Omega \\ &\mathbf{u} = \bar{\mathbf{u}}, && \text{on } \Gamma_D \\ &\mathbf{t} = \bar{\mathbf{t}}, && \text{on } \Gamma_N \end{aligned} \tag{1}$$

Table 1
Summary of important notations used in this study

D^h	:	A fine background mesh
D^e	:	A fine element in D^h
$\mathcal{X}^e$	:	The connectivity of D^e
D^H	:	A coarse background mesh
D^α	:	A coarse element in D^H
$\mathcal{X}^\alpha$	:	The connectivity of D^α
$\mathcal{X}^\alpha_f$	:	Fine nodes of D^α
$\mathcal{X}^\alpha_b$	:	Boundary fine nodes of D^α
$\mathcal{X}^\alpha_i$	:	Interior fine nodes of D^α
$\mathbf{Q}^\alpha$	:	Displacement vector of coarse nodes in $\mathcal{X}^\alpha$
$\mathbf{q}^\alpha$	:	Displacement vector of fine nodes in $\mathcal{X}^\alpha_f$
$\mathbf{\Phi}^I(\mathbf{x})$	:	The coarsened shape function at coarse node I
$\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$	:	The representation of $\mathbf{\Phi}^I(\mathbf{x})$ on D^α
$\Phi^{\alpha,I}_w(\mathbf{x})$	:	The components of the matrix-valued $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$
$\mathbf{\Phi}^\alpha(\mathbf{x})$	:	The coarsened element shape function of D^α
$\mathbf{p}^\alpha$	:	The control matrix for $\mathbf{\Phi}^\alpha(\mathbf{x})$

where $\frac{D\rho}{Dt}$ denotes the material time derivative of the mass density ρ, ∇ denotes the gradient operator with respect to the current configuration, $\mathbf{b}$ is the body force vector, $\mathbf{v}$, $\mathbf{a}$, and $\boldsymbol{\sigma}$ represent the velocity vector, acceleration vector, and Cauchy stress tensor [60] that all depend on displacement vector $\mathbf{u}$. In Eq. (1), the first two equations represent the conservation equations of mass and momentum, and the other two equations represent the Dirichlet boundary Γ_D of prescribed displacements $\bar{\mathbf{u}}$ and the Neumann boundary Γ_N under exerted tractions $\bar{\mathbf{t}}$.

Following the principle of virtual work and representing the displacement $\mathbf{u}$ through a discrete vector $\mathbf{Q}$, Eq. (1) is computed by solving the following discrete equilibrium equation incrementally at time $t + \Delta t$ using the previous results at time t,

$$\boldsymbol{f}\left(\hat{\mathbf{Q}}^{t+\Delta t}\right) = \mathbf{0}, \tag{2}$$

in terms of the solution vector $\hat{\mathbf{Q}}^{t+\Delta t}$ at time $t + \Delta t$, the vector of out-of-balance forces $\boldsymbol{f}\left(\hat{\mathbf{Q}}^{t+\Delta t}\right)$ is defined as,

$$\boldsymbol{f}\left(\hat{\mathbf{Q}}^{t+\Delta t}\right) := \mathbf{f}^{\text{ext},t+\Delta t} + \mathbf{f}^{\text{int},t+\Delta t} - \mathbf{M}^t\,\mathbf{a}^{t+\Delta t}, \tag{3}$$

where $\mathbf{f}^{\text{ext},t+\Delta t}$ and $\mathbf{f}^{\text{int},t+\Delta t}$ are the vectors of nodal external and internal forces at time $t + \Delta t$, $\mathbf{a}^{t+\Delta t}$ is the vector of nodal accelerations at time $t+\Delta t$, and $\mathbf{M}^t$ is the mass matrix at time t. Note that $\mathbf{f}^{\text{int},t+\Delta t}$ and $\mathbf{a}^{t+\Delta t}$ depend on $\hat{\mathbf{Q}}^{t+\Delta t}$, while $\mathbf{f}^{\text{ext},t+\Delta t}$ is assumed deformation-independent for simplicity, and thus it does not depend on $\hat{\mathbf{Q}}^{t+\Delta t}$.

Due to the material nonlinearity, Eq. (2) is solved using the Newton-Raphson iteration method by solving the following linear equation in the k-th Newton-Raphson iteration,

$${}^{(k-1)}\overline{\mathbf{K}}\ {}^{(k)}\Delta\mathbf{Q} = {}^{(k-1)}\boldsymbol{f}, \tag{4}$$

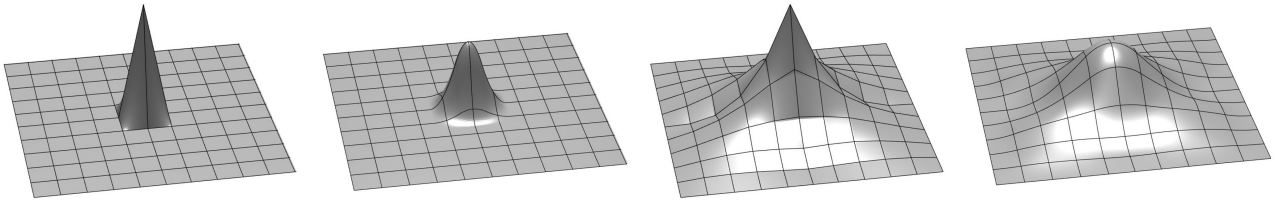
where ${}^{(k)}\Delta\mathbf{Q} = \hat{\mathbf{Q}}^{t+\Delta t} - {}^{(k-1)}\mathbf{Q}^{t+\Delta t}$ is the vector of incremental displacements to be computed, ${}^{(k-1)}\mathbf{Q}^{t+\Delta t}$ is the solution vector computed in the $(k-1)$-th iteration, ${}^{(k-1)}\overline{\mathbf{K}}$ and ${}^{(k-1)}\boldsymbol{f}$ are respectively the dynamic stiffness matrix and the out-of-balance force vector evaluated at ${}^{(k-1)}\mathbf{Q}^{t+\Delta t}$ and computed based on the backward Euler time integration,

$${}^{(k-1)}\overline{\mathbf{K}} = {}^{(k-1)}\mathbf{K}^{t+\Delta t} + \frac{\mathbf{M}^t}{\Delta t^2}, \tag{5}$$

$${}^{(k-1)}\boldsymbol{f} = \mathbf{f}^{\text{ext},t+\Delta t} + {}^{(k-1)}\mathbf{f}^{\text{int},t+\Delta t} - \mathbf{M}^t\left(\frac{\left({}^{(k-1)}\mathbf{Q}^{t+\Delta t} - \mathbf{Q}^t\right)}{\Delta t^2} - \frac{\mathbf{v}^t}{\Delta t}\right), \tag{6}$$

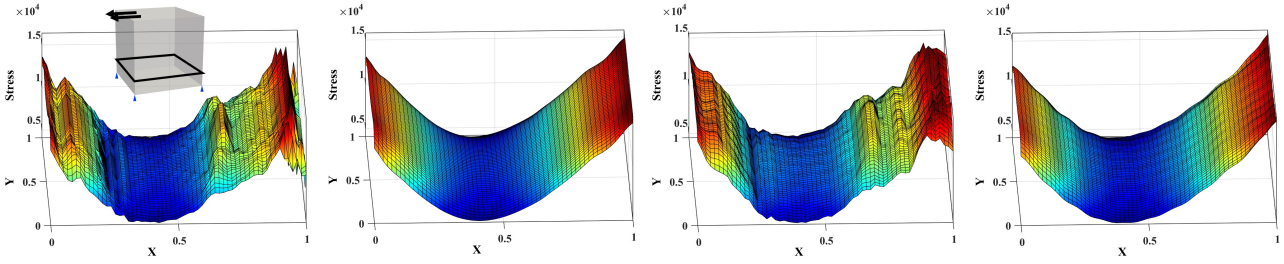
where ${}^{(k-1)}\mathbf{f}^{int,t+\Delta t}$ is the internal force vector, ${}^{(k-1)}\mathbf{K}^{t+\Delta t}$ is the tangent stiffness matrix, and $\mathbf{M}^t$ is the lumped mass matrix. More details can be found in [61, 62].

The solution to the above elasticity simulation problem is usually computed following the classical Galerikin method [25], expressing the solution as a linear combination of the shape functions. Different types of shape functions have been developed, for example, the linear or B-spline ones on a *fine mesh* in Figs. 1(a),(b), the coarsened as piecewise linear patches on a *coarse mesh* in Fig. 1(c) [21], or the proposed B-spline patches on a *coarse mesh* in Fig. 1(d). Utilizing



(a) Linear shape function (b) B-spline shape function (c) CBN shape function (d) Proposed coarsened shape function

Fig. 1. Four different types of shape functions: linear, quadratic B-spline, CBN, and the proposed coarsened B-patch shape functions.



(a) Linear shape function (b) B-spline shape function (c) CBN shape function (d) Proposed coarsened shape function

Fig. 2. Von Mises stress fields obtained by the MPM using the linear, quadratic B-spline, CBN, and proposed coarsened shape functions, as shown in Fig. 1, on the black plane of a cube shearing example in (a). More details are referred to Section 8.1.1.

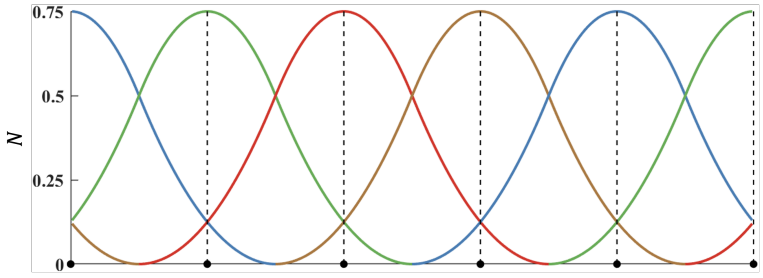


Fig. 3. A series of quadratic B-spline basis functions in 1D.

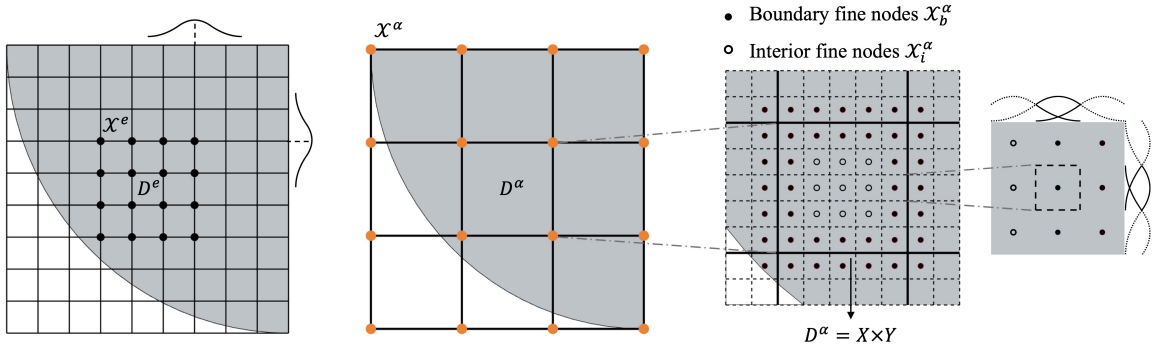
these shape functions for the simulations (within the MPM framework) may produce radically different stress distributions in Fig. 2; the result in (b) was the benchmark. Either the linear shape functions in (a) or their coarsened version in (c), of only C^0-continuity, produced a perturbed stress field and consequently incorrect results. In comparison, the B-spline types in (b) and (d), of C^1-continuity, were able to produce continuous stress fields and better results. Note the coarsened ones in (d) also much reduced the computational cost for large-scale problems.

Although numerical coarsening approaches using linear or even higher order shape functions [48, 49] have been developed, these shape functions are at most of C^0-continuity across element boundary. *Simply extending ideas on linear shape functions to higher order ones does not necessarily produce proper coarsened shape functions of C^1-continuity*. They are to be addressed in this study by carefully designing *B-spline patch* shape functions, as explained below.

3.2. Fine solution using B-spline basis functions

The B-spline basis functions are widely used in CAD with properties of global continuity and partition of unity [63, 64]. Fig. 3 plots a series of quadratic B-spline basis functions. They are adopted in expressing the coarsened shape functions.

Given a structure Ω in Eq. (1), we embed it in a fine regular background mesh $D^h = \{D^e, e = 1, 2, \ldots, m\}$ of elements D^e and nodes I (Fig. 4(a)). The coarse mesh, fine mesh, and the associated elements involved in this study are all shown in Fig. 4.



(a) $D^h = \{D^e, \mathrm{e} = 1,2,\ldots,m\}$ (b) $D^H = \{D^\alpha, \alpha = 1,2,\ldots,M\}$

Fig. 4. Fine and coarse meshes: (a) in the fine mesh D^h, the connectivity $\mathcal{X}^e$ of a fine element D^e contains 16 nodes (black); (b) in the coarse mesh D^H, the connectivity $\mathcal{X}^\alpha$ of a coarse element D^α contains 16 coarse nodes (orange). Note that on $D^\alpha = X \times Y$, a set of $\tilde{n} = n \times n$ fine nodes $\mathcal{X}_f^\alpha$ are classified into boundary fine nodes $\mathcal{X}_b^\alpha$ (filled circles) and interior fine nodes $\mathcal{X}_i^\alpha$ (hollow circles). There are 3×3 non-zero B-spline basis functions existing in each knot span. $n = 7$ in the figure.

A quadratic uniform B-spline basis function $B_I(\mathbf{x})$ is defined at a node I of coordinate (x_I, y_I) as follows,

$$B_I(\mathbf{x}) = B_I(x, y) = N(\frac{x - x_I}{h})\, N(\frac{y - y_I}{h}), \tag{7}$$

where h is the fine element size and $N(\cdot)$ is the 1D quadratic B-spline basis function,

$$N(\xi) = \begin{cases} \frac{1}{2}\xi^2 + \frac{3}{2}\xi + \frac{9}{8}, & -\frac{3}{2} \le \xi \le -\frac{1}{2}, \\ -\xi^2 + \frac{3}{4}, & -\frac{1}{2} \le \xi \le \frac{1}{2}, \\ \frac{1}{2}\xi^2 - \frac{3}{2}\xi + \frac{9}{8}, & \frac{1}{2} \le \xi \le \frac{3}{2}. \end{cases} \tag{8}$$

Some facts on the B-spline basis functions are first given below for their later usage in coarsened shape function construction.

Lemma 1. *In a 2D element $D^e \in D^h$,* 16 *non-zero quadratic B-spline basis functions exist, and each is associated with one of the* 16 *nodes on or adjacent to it. These basis functions are C^1-continuous across the element boundaries.*
The collection of these nodes is referred to as the connectivity *of element D^e, denoted by $\mathcal{X}^e$.*

Proof. The 1D quadratic B-spline basis function $N(\xi)$ has a local support covering 4 $1D$ elements adjacent to it, as shown in Fig. 3. Thus, the 2D quadratic B-spline basis function $B_I(\mathbf{x})$ in Eq. (7) has a local support covering $4 \times 4 = 16$ 2D elements D^e adjacent to it. It can then be deduced that 16 non-zero shape functions $B_I(\mathbf{x})$ exist in D^e, and each is associated with a node on or adjacent to it. Proof of the C^1-continuity is referred to [63]. □

Based on the B-spline basis functions in Eq. (7) and the classical Galerkin method [25], the displacement solution $\mathbf{u}(\mathbf{x})$ to Eq. (1) is expressed as a linear combination of the shape functions $B_I(\mathbf{x})$,

$$\underset{(2\times1)}{\mathbf{u}(\mathbf{x})} = \underset{(2\times32)}{\mathbf{B}^e(\mathbf{x})}\ \underset{(32\times1)}{\mathbf{q}^e}\ , \quad \mathbf{x} \in D^e, \tag{9}$$

where $\mathbf{q}^e$ of size (32×1) is the vector of the x- and y-displacements of the 16 nodes $\mathcal{X}^e$ of D^e, and $\mathbf{B}^e(\mathbf{x})$ of size (2×32) is the matrix of the B-spline basis functions $B_I(\mathbf{x})$ in D^e,

$$\underset{(2\times32)}{\mathbf{B}^e(\mathbf{x})} = \begin{bmatrix} B_1(\mathbf{x}) & 0 & B_2(\mathbf{x}) & 0 & \ldots & B_{16}(\mathbf{x}) & 0 \\ 0 & B_1(\mathbf{x}) & 0 & B_2(\mathbf{x}) & \ldots & 0 & B_{16}(\mathbf{x}) \end{bmatrix}. \tag{10}$$

A discrete nodal displacement $\mathbf{q}^e$ accordingly gives the overall solution to Eq. (1). For the studied heterogeneous structures, capturing the proper behavior of the fine-scale heterogeneities requires a highly resolved mesh D^h (or a small h), and consequently high computational loads. On the contrary, a coarse mesh (or a large h) ignores the details and much reduces the accuracy. To effectively balance the computational efficiency and accuracy, we are to construct for each coarse element a set of material-aware coarsened shape functions as free-form B-spline patches, as explained below.

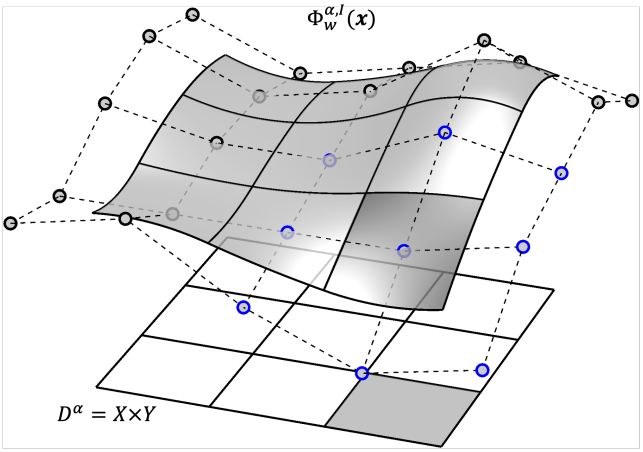


Fig. 5. Quadratic uniform B-spline surface $\Phi_w^{\alpha,I}(\mathbf{x})$ on coarse element D^α. The black lines on the surface denote the knot spans that divide the surface into sections. One knot span is defined by 3×3 control values (cf. knot span in dark gray and control values in blue). $n = 5$ in the figure.

3.3. Coarsened shape functions using matrix-valued B-spline patches

Notations are first defined. Given a heterogeneous structure Ω, we embed it in a coarse regular background mesh $D^H = \{D^\alpha, \alpha = 1, 2, \ldots, M\}$ of coarse elements D^α and coarse nodes I. For a coarse element D^α, let $\mathcal{X}^\alpha$ denote its *connectivity*. From Lemma 1, the number of coarse nodes in $\mathcal{X}^\alpha$ is 16, or $r = |\mathcal{X}^\alpha| = 16$; see also Fig. 4(b).

Without loss of generality, let $D^\alpha = [0, 1] \times [0, 1]$, or coarse element span $H = 1$. Within D^α a tensor product $X \times Y$ of two uniform knot vectors $X = \{x_1, x_2, \ldots, x_{n+p+1}\}$ and $Y = \{y_1, y_2, \ldots, y_{n+p+1}\}$ is introduced, where $p = 2$ is the polynomial degree.

The knot vectors X and Y define a set of $\tilde{n} = n \times n$ ($\tilde{n} > r$) quadratic uniform B-spline basis functions $B_k(\mathbf{x})$, $k = 1, 2, \ldots, \tilde{n}$ associated to each of the $\tilde{n}$ fine nodes $\mathcal{X}_f^\alpha$ at the centers of knot spans. There are 3×3 non-zero B-spline basis functions existing in each knot span. We divide fine nodes $\mathcal{X}_f^\alpha$ into $n_b = (n-4)^2$ *boundary fine nodes* $\mathcal{X}_b^\alpha$ and $n_i = \tilde{n} - n_b$ *interior fine nodes* $\mathcal{X}_i^\alpha$, for later usages, based on whether they are shared by knot spans from adjacent coarse elements or not.

We are now ready to introduce our matrix-valued coarsened shape function using B-spline patches, which has a local support covering 16 coarse elements adjacent to it. On a coarse element D^α, for each coarse node $I \in \mathcal{X}^\alpha$, the coarsened shape function $\mathbf{\Phi}^I(\mathbf{x})$ on D^α, denoted by $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$, is written as a 2×2 matrix,

$$\underset{(2\times2)}{\mathbf{\Phi}^{\alpha,I}(\mathbf{x})} = \begin{bmatrix} \Phi_1^{\alpha,I}(\mathbf{x}) & \Phi_2^{\alpha,I}(\mathbf{x}) \\ \Phi_3^{\alpha,I}(\mathbf{x}) & \Phi_4^{\alpha,I}(\mathbf{x}) \end{bmatrix}. \tag{11}$$

where each component $\Phi_w^{\alpha,I}(\mathbf{x})$, $w = 1, 2, 3, 4$ is a free-form quadratic B-spline surface on D^α, or a linear combination of B-spline basis functions $\{B_k(\mathbf{x})\}$ in the following matrix form,

$$\underset{(1\times1)}{\Phi_w^{\alpha,I}(\mathbf{x})} = \begin{bmatrix} B_1(\mathbf{x}) & B_2(\mathbf{x}) & \ldots & B_{\tilde{n}}(\mathbf{x}) \end{bmatrix} \underset{(\tilde{n}\times1)}{\mathbf{p}_w^{\alpha,I}}, \quad w = 1, 2, 3, 4, \tag{12}$$

where $\mathbf{p}_w^{\alpha,I}$ is a $(\tilde{n} \times 1)$ control vector to control the shape of $\Phi_w^{\alpha,I}(\mathbf{x})$, as plotted in Fig. 5.

Accordingly, the overall coarsened shape function $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ in Eq. (11) becomes,

$$\underset{(2\times2)}{\mathbf{\Phi}^{\alpha,I}(\mathbf{x})} = \underset{(2\times2\tilde{n})}{\mathbf{B}^\alpha(\mathbf{x})} \; \underset{(2\tilde{n}\times2)}{\mathbf{p}^{\alpha,I}}, \tag{13}$$

where $\mathbf{p}^{\alpha,I}$ is a $(2\tilde{n} \times 2)$ control matrix for $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ and $\mathbf{B}^\alpha(\mathbf{x})$ is the matrix form of the $\tilde{n}$ B-spline basis functions $B_k(\mathbf{x})$ in D^α,

$$\underset{(2\times2\tilde{n})}{\mathbf{B}^\alpha(\mathbf{x})} = \begin{bmatrix} B_1(\mathbf{x}) & 0 & B_2(\mathbf{x}) & 0 & \ldots & B_{\tilde{n}}(\mathbf{x}) & 0 \\ 0 & B_1(\mathbf{x}) & 0 & B_2(\mathbf{x}) & \ldots & 0 & B_{\tilde{n}}(\mathbf{x}) \end{bmatrix}. \tag{14}$$

By assembling the shape functions $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ for all the r coarse nodes $I \in \mathcal{X}^\alpha$, the coarsened element shape function $\mathbf{\Phi}^\alpha(\mathbf{x})$ of D^α, having a size of $(2 \times 2r)$, is represented as,

$$\underset{(2\times2r)}{\mathbf{\Phi}^\alpha(\mathbf{x})} = \begin{bmatrix} \mathbf{\Phi}^{\alpha,1}(\mathbf{x}) & \mathbf{\Phi}^{\alpha,2}(\mathbf{x}) & \ldots & \mathbf{\Phi}^{\alpha,r}(\mathbf{x}) \end{bmatrix} = \underset{(2\times2\tilde{n})}{\mathbf{B}^\alpha(\mathbf{x})} \; \underset{(2\tilde{n}\times2r)}{\mathbf{p}^\alpha}, \tag{15}$$

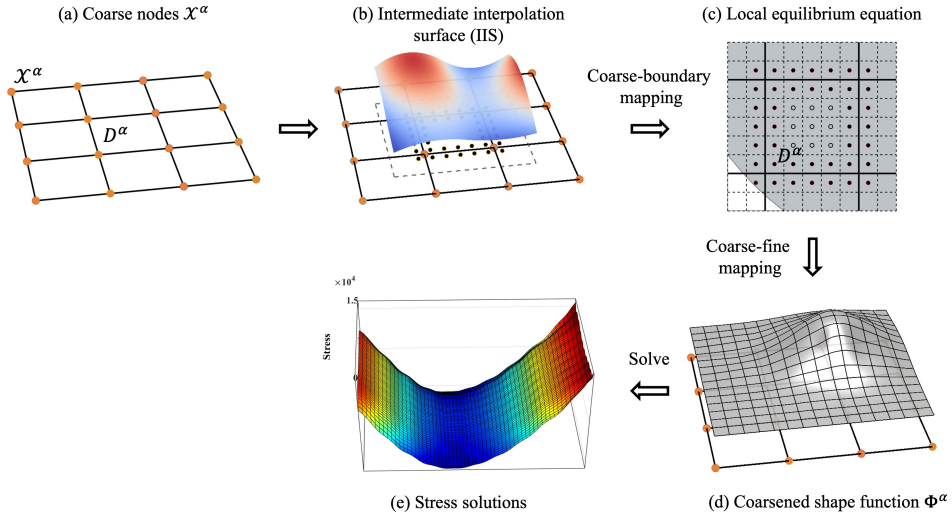


Fig. 6. Flowchart for constructing the coarsened shape functions: (a) employ a coarse regular background mesh with proper coarse nodes (in orange), (b) build an IIS that covers all the boundary fine nodes (in black) for each coarse element, (c) based on the obtained coarse–boundary mapping, construct the boundary–interior mapping by solving a local static equilibrium on the fine mesh, (d) construct the B-spline patch shape functions for each coarse element, (d) conduct simulation using the shape functions, which results in a continuous stress field.

where $\mathbf{p}^\alpha$ denotes a $(2\tilde{n} \times 2r)$ control matrix for the coarsened element shape function $\mathbf{\Phi}^\alpha(\mathbf{x})$. In addition, its derivatives are computed as,

$$\frac{\partial \mathbf{\Phi}^\alpha(\mathbf{x})}{\partial x} = \frac{\partial \mathbf{B}^\alpha(\mathbf{x})}{\partial x}\,\mathbf{p}^\alpha, \quad \frac{\partial \mathbf{\Phi}^\alpha(\mathbf{x})}{\partial y} = \frac{\partial \mathbf{B}^\alpha(\mathbf{x})}{\partial y}\,\mathbf{p}^\alpha. \tag{16}$$

Remark 1. *Instead of a diagonal matrix, a general* 2×2 *matrix form is used for* $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ *in Eq.* (11)*, so as to better couple different dimensions and handle the anisotropy [20, 21].*

Remark 2. *In contrast to the open knot vectors usually used in IGA [63, 64], there are no repeated knots in the knot vectors X and Y. This is particularly set to facilitate the* C^1*-continuity between multiple patches, as to be addressed in Section 5.1.*

Remark 3. *The B-spline basis functions are used in two scenarios in this study: (1) being defined at the fine nodes for constructing our B-spline patch shape functions in Eq.* (15)*; (2) being defined at the coarse nodes for constructing the intermediate interpolation surface in Section 5.1.*

4. Overview on coarsened B-spline patch shape function construction

A specific control matrix $\mathbf{p}^\alpha$ provides a concrete set of coarsened shape functions $\mathbf{\Phi}^\alpha(\mathbf{x})$ in Eq. (15). The value should fulfill the following critical requirements: (1) owning C^1-continuity across the coarse element boundaries; (2) meeting the basic geometric property of partition of unity to avoid un-physical results; (3) capturing the deformation behavior of the heterogeneous element D^α. Such properties have not been fully satisfied previously and are to be achieved in this study.

4.1. Basic idea

Our construction of the coarsened shape functions is based on the following observations, as also indicated in Fig. 6.

Firstly, for each heterogeneous coarse element, the shape functions essentially build a coarse–fine mapping from the coarse nodal displacements, which are DOFs in computing the overall solution to the problem in Eq. (1), to the fine nodal displacements, which describe the overall solution at any point. We see the mapping as two sequent mappings from the coarse nodes to the coarse element's boundary fine nodes and then to its interior fine nodes. The strategy not only allows for an easy and efficient implementation but also facilitates achieving the C^1-continuity. Additional coarse nodes can also be introduced, besides the element vertices, which can be set to balance different requirements of simulation accuracy or efficiency. Fig. 6(a) illustrates a typical coarse element set up in this study; more details are referred to Section 5.

Secondly, for the shape functions' C^1-continuity between different coarse elements across their common boundaries, we introduce for each coarse element an intermediate interpolation surface (*IIS* for short) that covers all the boundary fine nodes; see also Fig. 6(b). The IIS itself has a higher-order continuity, and identical values with those of its adjacent coarse elements around the boundaries, which ultimately ensures the shape functions' C^1-continuity. Sampling from the IIS gives the (displacement) values on the boundary fine nodes, whose matrix form builds the coarse–boundary mapping. In addition, the IIS has the property of partition of unity, which, together with that of B-spline basis functions, ultimately results in the property of our constructed coarsened shape functions.

Thirdly, given the values at boundary fine nodes obtained above, for the shape function's awareness of the interior heterogeneities within the coarse element, we are to construct the boundary–interior mapping by solving a local static equilibrium problem on the fine mesh; see also Fig. 6(c). The product of the mapping with the constructed coarse–boundary mapping above ultimately produces the desired coarse–fine mapping, and consequently the target shape functions on the coarse element with the desired properties; see also Fig. 6(d). It can then be used within a classical simulation framework (MPM here), resulting in a reliable structure deformation and particularly a continuous stress field; see also Fig. 6(e).

4.2. Mathematical formulations and overview

Let $\mathbf{Q}^\alpha$ be the coarse nodal displacements (or DOFs) to be computed at coarse nodes $\mathcal{X}^\alpha$ of a coarse element D^α. A linear combination of the coarsened shape function $\mathbf{\Phi}^\alpha(\mathbf{x})$ in Eq. (15), via the coarse node solution $\mathbf{Q}^\alpha$, gives the displacement $\mathbf{u}(\mathbf{x})$ at any point $\mathbf{x} \in D^\alpha$,

$$\mathbf{u}(\mathbf{x}) = \mathbf{\Phi}^\alpha(\mathbf{x})\, \mathbf{Q}^\alpha = \mathbf{B}^\alpha(\mathbf{x})\, \mathbf{p}^\alpha\, \mathbf{Q}^\alpha, \quad \mathbf{x} \in D^\alpha. \tag{17}$$

On the other hand, considering $D^\alpha = X \times Y$, together with its fine nodes $\mathcal{X}_f^\alpha$, as a fine mesh, the fine nodal displacements $\mathbf{q}^\alpha$ at fine nodes $\mathcal{X}_f^\alpha$ can be globally computed from the structure Ω or locally from D^α. A high-fidelity displacement $\hat{\mathbf{u}}(\mathbf{x})$ on any point $\mathbf{x} \in D^\alpha$ can also expressed as a linear combination of the B-spline basis functions $\mathbf{B}^\alpha(\mathbf{x})$ in Eq. (14), via the $\mathbf{q}^\alpha$, as follows,

$$\hat{\mathbf{u}}(\mathbf{x}) = \mathbf{B}^\alpha(\mathbf{x})\, \mathbf{q}^\alpha, \quad \mathbf{x} \in D^\alpha. \tag{18}$$

Taking into account Eqs. (17) and (18), the coarsened element shape function $\mathbf{\Phi}^\alpha(\mathbf{x})$, or the control matrix $\mathbf{p}^\alpha$, is best to satisfy the following equation,

$$\underset{(2\tilde{n}\times 1)}{\mathbf{q}^\alpha} = \underset{(2\tilde{n}\times 2r)}{\mathbf{p}^\alpha}\ \underset{(2r\times 1)}{\mathbf{Q}^\alpha}. \tag{19}$$

Equivalently, the control matrix $\mathbf{p}^\alpha$, which originally determines the coarsened shape functions in Eq. (15), maps the coarse nodal displacements $\mathbf{Q}^\alpha$ to the fine nodal displacements $\mathbf{q}^\alpha$.

Based on the idea explained above, $\mathbf{p}^\alpha$ is to be reformulated as two sequent mappings from the coarse nodes to the boundary fine nodes, and then to the interior fine nodes. Mathematically, $\mathbf{p}^\alpha$ is derived as a product of *coarse–boundary mapping* matrix $\mathbf{\Psi}^\alpha$ of size $(2n_b \times 2r)$ and *boundary–interior mapping* matrix $\mathbf{M}^\alpha$ of size $(2\tilde{n} \times 2n_b)$ for the coarse node number r, fine node number $\tilde{n}$ and boundary fine node number n_b, that is,

$$\underset{(2\tilde{n}\times 2r)}{\mathbf{p}^\alpha} = \underset{(2\tilde{n}\times 2n_b)}{\mathbf{M}^\alpha}\ \underset{(2n_b\times 2r)}{\mathbf{\Psi}^\alpha}, \tag{20}$$

where matrix $\mathbf{\Psi}^\alpha$, $\mathbf{M}^\alpha$ consequently maps the coarse nodal displacements $\mathbf{Q}^\alpha$ to the boundary fine nodal displacements $\mathbf{q}_b$ through an IIS construction, and then to the interior fine nodal displacements $\mathbf{q}_i$ via solving a local static equilibrium problem. This yields

$$\mathbf{q}^\alpha = \mathbf{M}^\alpha\, \mathbf{q}_b, \quad \mathbf{q}_b = \mathbf{\Psi}^\alpha\, \mathbf{Q}^\alpha. \tag{21}$$

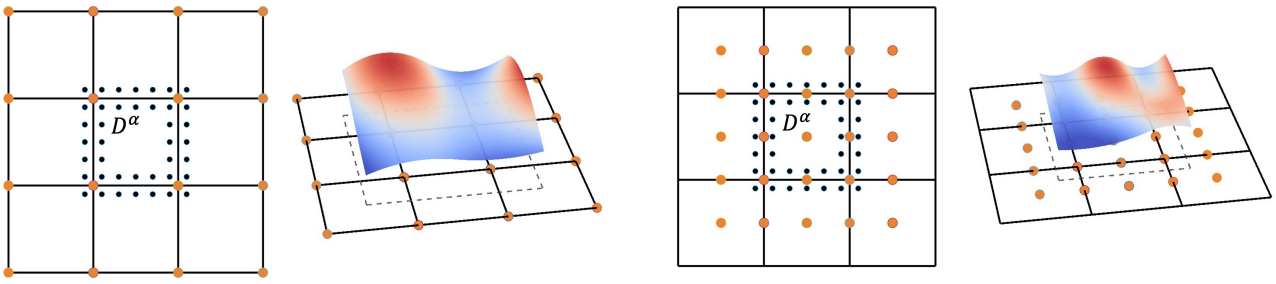
Constructions of the two mappings are detailed next.

5. Details on coarsened shape function constructions

5.1. Coarse–boundary mapping

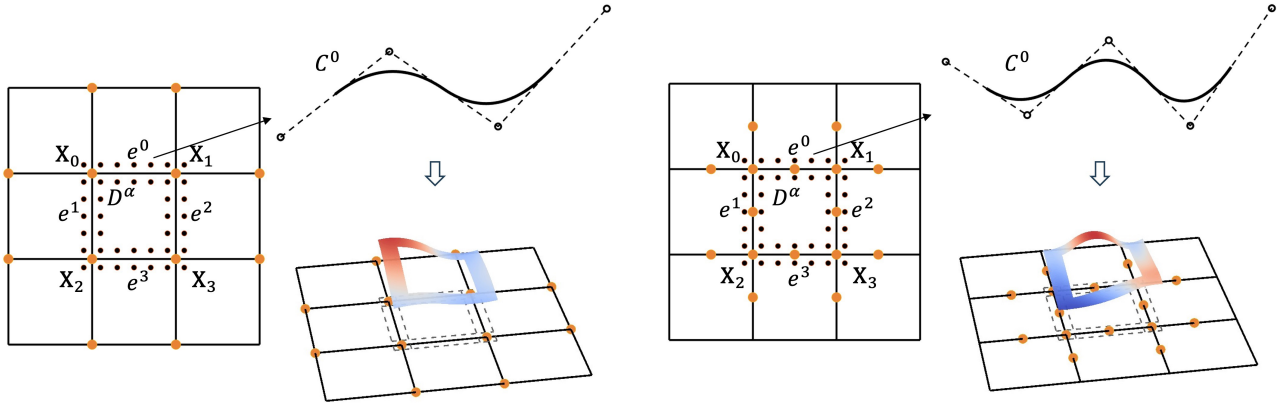
As explained in Section 4, the core to construct the coarse–boundary mapping $\mathbf{\Psi}^\alpha$ lies in the construction of an IIS that covers the coarse element's boundary fine nodes. We are to construct for a coarse element D^α the IIS, say $\mathcal{I}^\alpha(\mathbf{x})$, in a form as follows,

$$\mathcal{I}^\alpha(\mathbf{x}) = \sum_I^r \psi_I(\mathbf{x})\, Q_I, \tag{22}$$



(a) $L = H$ (b) $L = H/2$

Fig. 7. B-spline coarse element: (a) the IIS is set as quadratic B-spline basis functions on the 16 coarse nodes (in orange) around the coarse element D^α, and the connectivity $\mathcal{X}^\alpha$ contains $r = 16$ coarse nodes; (b) inserting new coarse nodes so that $L = H/2$ generates a coarse element containing $r = 25$ coarse nodes.



(a) $L = H$ (b) $L = H/2$

Fig. 8. Boundary B-spline coarse element: (a) we construct a B-spline curve for each element edge based on the four coarse nodes (in orange) along its direction, and extrude it along its adjacent curves to obtain an extruded IIS for each coarse element D^α; the connectivity $\mathcal{X}^\alpha$ contains $r = 12$ coarse nodes; (b) inserting new coarse nodes so that $L = H/2$ generates a coarse element containing $r = 16$ coarse nodes.

where Q_I denotes one displacement component in the coarse nodal displacement vector $\mathbf{Q}_I$, as introduced in Eq. (17), and $\psi_I(\mathbf{x})$ denotes a specific basis function. Collecting both x and y components, we have the matrix form,

$$\begin{aligned}\boldsymbol{\mathcal{I}}^\alpha(\mathbf{x}) &= \begin{bmatrix} \psi_1(\mathbf{x}) & 0 & \psi_2(\mathbf{x}) & 0 & \dots & \psi_r(\mathbf{x}) & 0 \\ 0 & \psi_1(\mathbf{x}) & 0 & \psi_2(\mathbf{x}) & \dots & 0 & \psi_r(\mathbf{x}) \end{bmatrix} \mathbf{Q}^\alpha \\ &= \boldsymbol{\psi}^\alpha(\mathbf{x})\, \mathbf{Q}^\alpha,\end{aligned} \tag{23}$$

where matrix $\boldsymbol{\psi}^\alpha(\mathbf{x})$, $\mathbf{Q}^\alpha$ are respectively of size $(2 \times 2r)$ and $(2r \times 1)$.

Sampling the matrix $\boldsymbol{\psi}^\alpha(\mathbf{x})$ at each boundary fine node $\mathbf{x}_k$, $k = 1, 2, \dots, n_b$, and arranging them row by row, we have the desired coarse–boundary matrix $\boldsymbol{\Psi}^\alpha$,

$$\boldsymbol{\Psi}^\alpha = \begin{bmatrix} \boldsymbol{\psi}^\alpha(\mathbf{x}_1) \\ \boldsymbol{\psi}^\alpha(\mathbf{x}_2) \\ \dots \\ \boldsymbol{\psi}^\alpha(\mathbf{x}_{n_b}) \end{bmatrix}, \tag{24}$$

which maps the coarse nodal displacements $\mathbf{Q}^\alpha$ to boundary fine nodal displacements $\mathbf{q}_b$ (in Eq. (21)).

Different placements (i.e. number and locations) of the coarse nodes $\mathcal{X}^\alpha$ determine different basis functions $\boldsymbol{\psi}^\alpha(\mathbf{x})$, and consequently, the IIS and matrix mapping $\boldsymbol{\Psi}^\alpha$. Two typical coarse elements are devised and explained below.

5.1.1. B-spline coarse element

In this strategy, the IIS is set as quadratic B-spline basis functions defined on the 16 coarse nodes around the coarse element D^α. Note the connectivity $\mathcal{X}^\alpha$ of D^α contains $r = 16$ coarse nodes according to Lemma 1.

More specifically, as illustrated in Fig. 7(a), we pick up the 16 coarse nodes (in orange), and each of them has a corresponding B-spline basis $B_I(\mathbf{x})$ in a form as Eq. (7). Replacing $\psi_I(\mathbf{x})$ in Eq. (22) with the newly constructed $B_I(\mathbf{x})$ gives the desired IIS.

More DOFs for a coarse element can be introduced for accuracy improvement, by inserting new coarse nodes using the h-refinement strategy in IGA [64], for example, by inserting coarse nodes at the centers of both the elements and edges, as illustrated in Fig. 7(b). In this case, the coarse node has a span $L = H/2$.

5.1.2. Boundary B-spline coarse element

In this strategy, we construct a B-spline curve for each coarse element edge and extrude it along directions of its adjacent curves, which produces an extruded IIS that locally covers boundary fine nodes $\mathcal{X}_b^\alpha$.

Suppose $C^i(\mathbf{x}),\ i = 0, 1, 2, 3$, is the quadratic B-spline curve associated with the i-th edge e^i. The four control points for curve $C^i(\mathbf{x})$ are taken as the displacements (DOFs) on the coarse nodes along its edge direction; see also Fig. 8(a). Note all together a coarse element D^α contains $r = 12$ coarse nodes, or the connectivity of $r = |\mathcal{X}^\alpha| = 12$, according to Lemma 1.

Accordingly, given a boundary fine node $\mathbf{x}_k$ with the edge e^0 closest to it, as illustrated in Fig. 9(a), the value of the extruded IIS at $\mathbf{x}_k$ is defined as,

$$\mathcal{I}^\alpha(\mathbf{x}_k) = C^0(\mathbf{x}^0) + (1-a)\,[C^1(\mathbf{x}^1) - C^1(\mathbf{X}_0)] + a\,[C^2(\mathbf{x}^2) - C^2(\mathbf{X}_1)], \tag{25}$$

where e^1 and e^2 are the two edges adjacent to edge e^0, $\mathbf{X}_0$ and $\mathbf{X}_1$ are the two endpoints of edge e^0, $\mathbf{x}^i,\ i = 0, 1, 2$ are the projection points of $\mathbf{x}_k$ onto edges e^i, and the weight a is the relative distance fraction given by,

$$a = \frac{\|\mathbf{x}_k - \mathbf{x}^1\|_2 - \frac{s}{2}}{H - s}, \tag{26}$$

where H is the coarse element span and s is the fine knot span.

We comment on the cases that the fine nodes are shared by different edges during the above IIS construction. First, for the corner vertices shared by two edges, for example, $\mathbf{X}_0$ shared by edges e^0 and e^1, we may have $C^0(\mathbf{X}_0) \neq C^1(\mathbf{X}_0)$ due to the non-interpolation property of a B-spline curve. This issue is resolved by taking their average value at $\mathbf{X}_0$.

Second, given a coarse element D^α, we have four boundary fine nodes around each corner vertex of it. During the extruded IIS construction, they are shared by two adjacent edges; see Fig. 9(b). Because of the setting of weight a in Eq. (26), Eq. (25) is guaranteed to give an identical IIS value no matter which edge the value is computed based on.

Similarly, as in the case of the B-spline coarse element, to improve the simulation accuracy, new coarse nodes can be inserted at the centers of the edges. In this case, the connectivity $\mathcal{X}^\alpha$ contains $r = 16$ coarse nodes; see Fig. 8(b). The coarse node has a span $L = H/2$.

When using either B-spline or boundary B-spline coarse element, the IISs for adjacent coarse elements have the same sampling values on their common boundary fine nodes, as summarized below. This is important in ensuring the C^1-continuity of the constructed shape functions, as well detailed in Section 6.

Lemma 2. *Given two adjacent coarse elements D^α, D^β prescribed above of the same type, we have*

$$\mathcal{I}^\alpha(\mathbf{x}_k) = \mathcal{I}^\beta(\mathbf{x}_k), \tag{27}$$

where $\mathcal{I}^\alpha(\mathbf{x})$ and $\mathcal{I}^\beta(\mathbf{x})$ are the associated IISs constructed above, and $\mathbf{x}_k$ are the common boundary fine nodes around the common edge. Accordingly, the two coarsened shape functions $\mathbf{\Phi}^\alpha(\mathbf{x})$ and $\mathbf{\Phi}^\beta(\mathbf{x})$ in Eq. (15) *share the same control values corresponding to the common boundary fine nodes.*

Proof: the result is clear noticing that the IISs for adjacent coarse elements share a common part around the common boundary fine nodes; See also Fig. 10. □

5.2. Boundary–interior mapping

The boundary–interior mapping matrix $\mathbf{M}^\alpha$ maps the boundary fine nodal displacements $\mathbf{q}_b$ to interior fine nodal displacements $\mathbf{q}_i$ by solving a local boundary-constrained static equilibrium within the coarse element D^α. It takes knot spans as the analysis mesh, B-spline basis functions at fine nodes as shape functions, and the mapped values $\mathbf{q}_b$ from $\mathbf{Q}^\alpha$ in Eq. (23) as Dirichlet boundary conditions.

Instead of solving this local problem using the full Newton-Raphson method for converged displacements of interior fine nodes, it is found that only applying the first iteration of the Newton-Raphson method greatly improves the efficiency while maintaining very close accuracy. The static equilibrium equation is thus built as,

$$\tilde{\mathbf{K}}^\alpha\ \mathbf{q}^\alpha = \tilde{\mathbf{f}}^{ext,\alpha} + \tilde{\mathbf{f}}^{int,\alpha}, \tag{28}$$

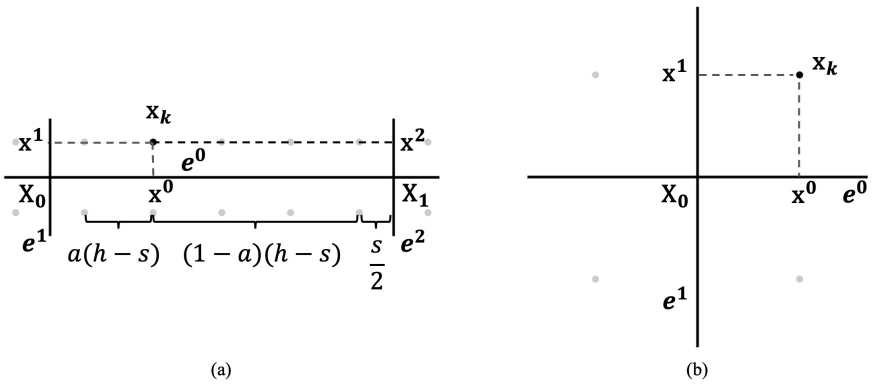


Fig. 9. The extruded IIS' samplings at (a) a regular boundary fine node $\mathbf{x}_k$ and (b) a special boundary fine node $\mathbf{x}_k$ shared by two adjacent edges.

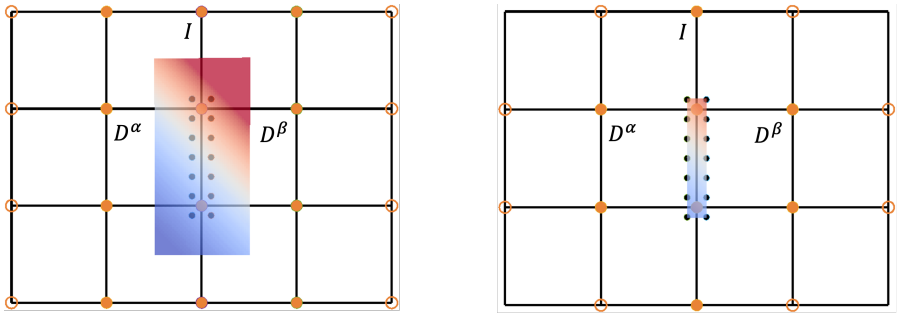


Fig. 10. The common part of the IISs for adjacent coarse elements D^α and D^β, covers the $2n$ common boundary fine nodes (black) when using (a) B-spline or (b) boundary B-spline coarse element. The 12 and 8 filled orange nodes in (a), and (b) are respectively the common coarse nodes shared by D^α and D^β.

where $\tilde{\mathbf{K}}^\alpha$ denotes the tangent stiffness matrix, $\tilde{\mathbf{f}}^{ext,\alpha}$ and $\tilde{\mathbf{f}}^{int,\alpha}$ are the external and internal force vectors, respectively.

Considering the boundary fine nodes $\mathcal{X}_b^\alpha$ and interior fine nodes $\mathcal{X}_i^\alpha$ as retained and truncated nodes respectively, Eq. (28) can be rewritten as

$$\begin{bmatrix} \mathbf{K}_b & \mathbf{K}_{bi} \\ \mathbf{K}_{ib} & \mathbf{K}_i \end{bmatrix} \begin{bmatrix} \mathbf{q}_b \\ \mathbf{q}_i \end{bmatrix} = \begin{bmatrix} \mathbf{f}_b \\ \mathbf{f}_i \end{bmatrix}, \tag{29}$$

where $\mathbf{K}_b$, $\mathbf{K}_i$, $\mathbf{K}_{bi}$, $\mathbf{K}_{ib}$ are the associated sub-matrices of $\tilde{\mathbf{K}}^\alpha$, $\mathbf{f}_b$ and $\mathbf{f}_i$ are the sub-vectors of out-of-balance force. By solving the equation in the second-row of Eq. (29) for $\mathbf{q}_i$, while assuming that $\mathbf{f}_i = \mathbf{0}$, the following is obtained,

$$\mathbf{q}_i = -\mathbf{K}_i^{-1}\, \mathbf{K}_{ib}\, \mathbf{q}_b. \tag{30}$$

Consequently,

$$\mathbf{q}^\alpha = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_i^{-1}\, \mathbf{K}_{ib} \end{bmatrix} \mathbf{q}_b = \mathbf{M}^\alpha\, \mathbf{q}_b, \quad \text{for} \quad \mathbf{M}^\alpha = \begin{bmatrix} \mathbf{I} \\ -\mathbf{K}_i^{-1}\, \mathbf{K}_{ib} \end{bmatrix}, \tag{31}$$

where $\mathbf{I}$ is the identity matrix of size $(2n_b \times 2n_b)$ and $\mathbf{M}^\alpha$ is the desired boundary–interior mapping matrix for D^α having a size of $(2\tilde{n} \times 2n_b)$.

Remark 4. *The knot spans in some coarse elements may not contain any particle, especially for those truncated by structure boundary. The special case may result in a singular matrix* $\mathbf{K}_i$ *in Eq.* (30)*. A small value* $\epsilon = 1e^{-5}$ *is added in the diagonal of* $\mathbf{K}_i$ *to handle the issue in the implementation.*

6. Properties of the coarsened B-spline patch shape functions

The proposed coarsened shape functions have important properties of C^1-continuity, partition of unity, and awareness of interior material distribution, as detailed below.

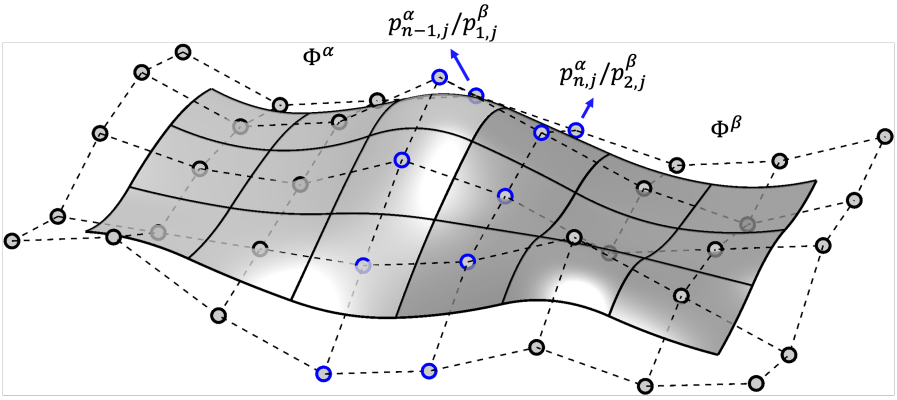


Fig. 11. A pair of C^1-continuous quadratic uniform B-spline patches Φ^{α} and Φ^{β} (light gray and dark gray), abbreviations of $\Phi^{\alpha,I}_w$ and $\Phi^{\beta,I}_w$ on coarse elements D^{α} and D^{β} of the coarsened shape function Φ^I_w (one component). Let $p^{\alpha}_{i,j}$ and $p^{\beta}_{i,j}$, $i, j = 1, 2, \ldots, n$ be their control values, respectively of the control vectors $\mathbf{p}^{\alpha,I}_w$ and $\mathbf{p}^{\beta,I}_w$ in Eq. (12). The control values corresponding to the common boundary fine nodes are shown as blue points (i.e., $p^{\alpha}_{n-1,j}/p^{\beta}_{1,j}$ and $p^{\alpha}_{n,j}/p^{\beta}_{2,j}$).

6.1. C^1-continuity

The C^1-continuity is mainly derived from the property of the IIS constructed in Section 5, as described below.

Theorem 3. *Given two adjacent coarse elements D^{α} and D^{β}, let $\Phi^{\alpha,I}_w(\mathbf{x})$ and $\Phi^{\beta,I}_w(\mathbf{x})$ be the two adjacent B-spline patches on D^{α} and D^{β} of the coarsened shape function Φ^I_w (one component) constructed above for a coarse node I. We have that $\Phi^{\alpha,I}_w(\mathbf{x})$ and $\Phi^{\beta,I}_w(\mathbf{x})$ are C^1-continuous, i.e. they share the same values and first derivatives along their common boundary, as depicted in Fig. 11. The coarsened shape function Φ^I_w (or $\mathbf{\Phi}^I$) is thus C^1-continuous across the coarse element boundaries.*

Proof: The C^1-continuity can be proved from the fact that the two adjacent coarse elements share the same IIS values on their common boundary fine nodes, together with the dependence of the B-spline surface boundary on surrounding control values. More details are explained in Appendix A. □

Fig. 12 shows the derivatives of the first component $\Phi^I_1(\mathbf{x})$ of $\mathbf{\Phi}^I(\mathbf{x})$ at a coarse node shared by four coarse elements. Continuous derivatives are observed, demonstrating its C^1-continuity.

6.2. Partition of unity

The proposed coarsened shape functions satisfy the partition of unity (PU) property, which enables the translation invariance, and conservation of mass and linear momentum [60]. An 1-vector of size a is denoted by $e_a = [1, \cdots, 1, \cdots, 1]^T$ below.

The PU of the coarsened shape functions entails the IIS' PU in Eq. (22). This is obvious for the B-spline surface in the case of the B-spline coarse element. The case of boundary B-spline coarse element is then explained. By making coarse nodal displacements equal to 1, i.e. $Q_I = 1$, then $C^i(\mathbf{x}) = 1$ holds due to the PU of the B-spline curve. The extruded IIS' PU is thus satisfied, as

$$\mathcal{I}^{\alpha}(\mathbf{x}_k) = 1 + a(1-1) + (1-a)(1-1) = 1. \tag{32}$$

The coarse–boundary mapping matrix $\mathbf{\Psi}^{\alpha}$ will map 1-vector $\mathbf{e}_{2r}$ to $\mathbf{e}_{2n_b}$,

$$\mathbf{e}_{2n_b} = \mathbf{\Psi}^{\alpha}\, \mathbf{e}_{2r}. \tag{33}$$

Consequently, the PU property of coarsened element shape function $\mathbf{\Phi}^{\alpha}(\mathbf{x})$ is derived from Eq. (33),

$$\begin{aligned} \mathbf{\Phi}^{\alpha}(\mathbf{x})\, \mathbf{e}_{2r} &= \mathbf{B}(\mathbf{x})\, \mathbf{M}^{\alpha}\, \mathbf{\Psi}^{\alpha}\, \mathbf{e}_{2r} \\ &= \mathbf{B}(\mathbf{x})\, \mathbf{M}^{\alpha}\, \mathbf{e}_{2n_b} \\ &= \mathbf{B}(\mathbf{x})\, \mathbf{e}_{2\tilde{n}} \\ &= \mathbf{e}_2, \end{aligned} \tag{34}$$

where equation $\mathbf{M}^{\alpha}\, \mathbf{e}_{2n_b} = \mathbf{e}_{2\tilde{n}}$ holds since any row (or column) of the tangent stiffness matrix in Eq. (29) represents an equilibrium force system, and equation $\mathbf{B}(\mathbf{x})\, \mathbf{e}_{2\tilde{n}} = \mathbf{e}_2$ comes from the PU of B-spline basis functions [64].

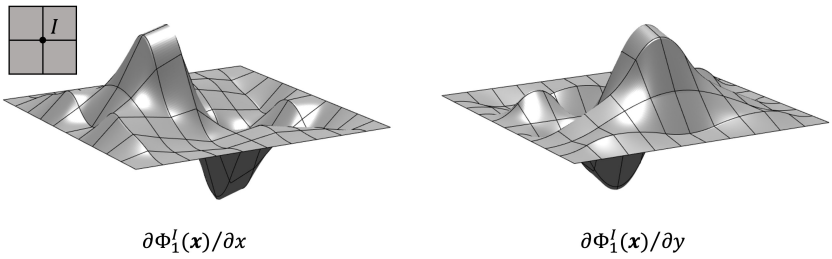


Fig. 12. Plots of the derivatives $\partial\Phi_1^I(\mathbf{x})/\partial x$ and $\partial\Phi_1^I(\mathbf{x})/\partial y$ of $\Phi_1^I(\mathbf{x})$ at a corner coarse node shared by four coarse elements. Their continuity demonstrates the C^1-continuity of our coarsened shape functions.

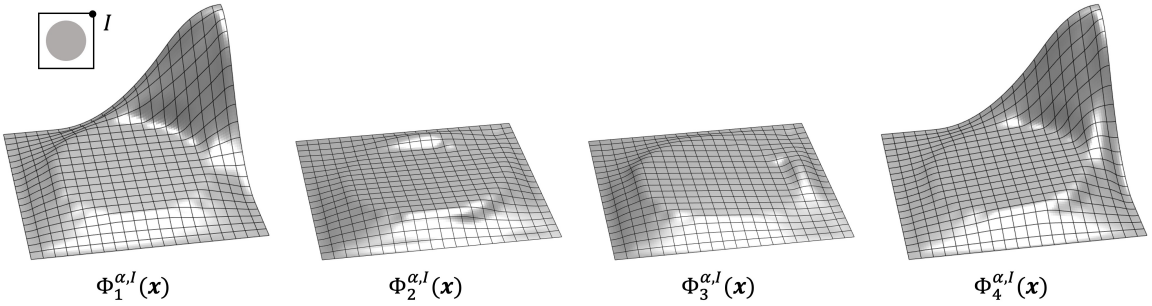


Fig. 13. The four components of the coarsened shape function $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ at the right top corner coarse node, with a circular hard filling in the coarse element.

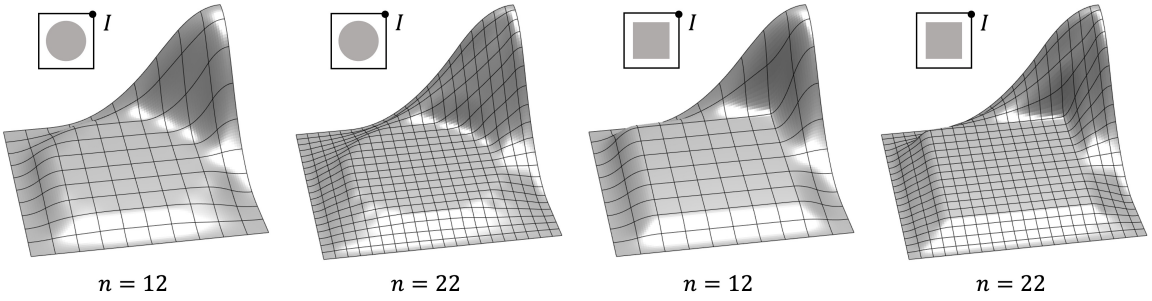


Fig. 14. The first components $\Phi_1^{\alpha,I}(\mathbf{x})$ of the coarsened shape functions $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ at a right top corner coarse node, with different material distributions (circular/square hard filling) in the coarse element or different fine node numbers n per direction.

Remark 5. *The proposed coarsened shape functions do not satisfy the Lagrange property (or the Kronecker delta property), since the employed B-spline surface is non-interpolating at coarse nodes. The absence of the Lagrange property makes the enforcement of the Dirichlet boundary conditions difficult. Plane Dirichlet boundaries are assumed for simplicity in the numerical examples in Section 8. More complex cases may be resolved through the weak enforcement approaches [65].*

6.3. Awareness of material heterogeniety

In contrast to the conventional linear/B-spline shape function, the coarsened shape function $\mathbf{\Phi}^I(\mathbf{x})$ for a node I has an asymmetric 2×2 matrix value, which better captures the anisotropy of the heterogeneous structure inside [20].

Fig. 13 shows the four components $\Phi_w^{\alpha,I}(\mathbf{x})$ of the coarsened shape function $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ on a coarse element D^α containing a circular hard filling within its interior, at the right top corner coarse node I. They are able to capture the interior material distributions. It can also be observed that the diagonal components $\Phi_1^{\alpha,I}(\mathbf{x})$ and $\Phi_4^{\alpha,I}(\mathbf{x})$ play major roles in the function values, while the off-diagonal components $\Phi_2^{\alpha,I}(\mathbf{x})$ and $\Phi_3^{\alpha,I}(\mathbf{x})$ regulate the interpolations by coupling different axes.

Fig. 14 also shows the first component $\Phi_1^{\alpha,I}(\mathbf{x})$ of the coarsened shape function on coarse elements with different fine node number n per direction. A larger n allows a finer exposure of the interior material.

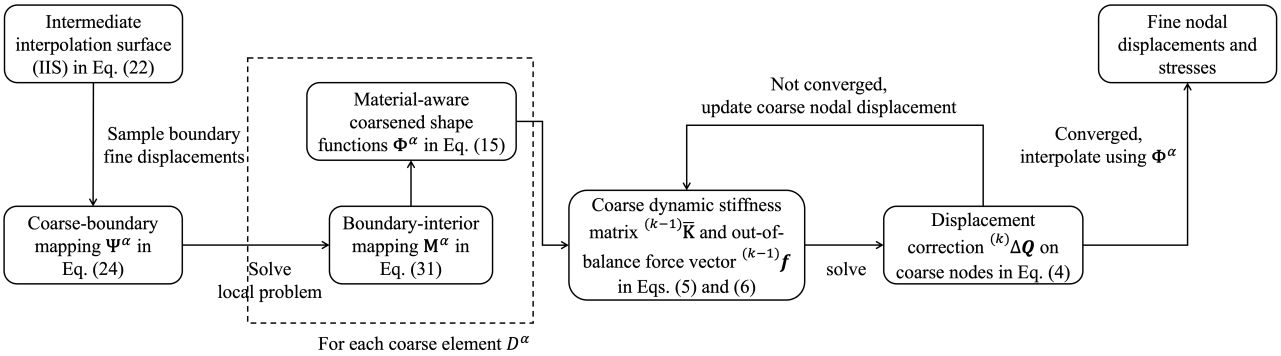


Fig. 15. Flowchart of the entire simulation process (per time step) for heterogeneous nonlinear structures using the proposed coarsened shape functions.

7. Extensions

We discuss below the extension of the coarsened shape functions to 3D cases and their application to the MPM framework for nonlinear deformations.

7.1. Extension to 3D cases

Derivation of the 3D coarsened shape functions follows the same procedure as that in 2D described above. Their main differences are summarized below.

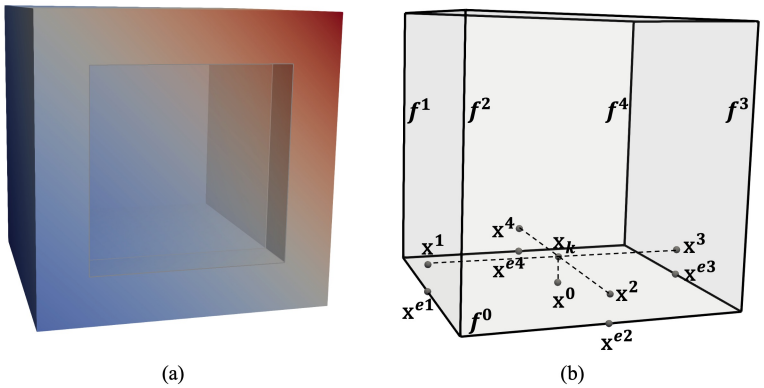
Firstly, the coarsened shape function $\mathbf{\Phi}^{\alpha,I}(\mathbf{x})$ at a coarse node I on a 3D coarse element D^α takes a form of 3×3 matrix,

$$\mathbf{\Phi}^{\alpha,I}(\mathbf{x}) = \begin{bmatrix} \Phi_1^{\alpha,I}(\mathbf{x}) & \Phi_2^{\alpha,I}(\mathbf{x}) & \Phi_3^{\alpha,I}(\mathbf{x}) \\ \Phi_4^{\alpha,I}(\mathbf{x}) & \Phi_5^{\alpha,I}(\mathbf{x}) & \Phi_6^{\alpha,I}(\mathbf{x}) \\ \Phi_7^{\alpha,I}(\mathbf{x}) & \Phi_8^{\alpha,I}(\mathbf{x}) & \Phi_9^{\alpha,I}(\mathbf{x}) \end{bmatrix}, \tag{35}$$

where each component $\Phi_w^{\alpha,I}(\mathbf{x})$, $w = 1,2,\ldots,9$ is a quadratic B-spline volume [64] taking its control values as the displacements on the $\tilde{n}$ fine nodes $\mathcal{X}_f^\alpha$ for $\tilde{n} = n^3$. We also mention here that the fine nodes $\mathcal{X}_f^\alpha$ are divided into $n_b = (n-4)^3$ boundary fine nodes around the 6 element faces and $n_i = \tilde{n} - n_b$ interior fine nodes.

Secondly, the 3D boundary–interior matrix $\mathbf{M}^\alpha$ in Eq. (20) is derived via solving a 3D static equilibrium on the fine mesh D^α.

Thirdly, the 3D coarse–boundary matrix $\mathbf{\Psi}^\alpha$ also has two types of coarse elements: B-spline and boundary B-spline coarse elements. The former is straightforward and we mainly explain the latter. Similarly, as in the 2D case, we construct a B-spline surface for each element face and extrude each along its adjacent surfaces to obtain a compact extruded intermediate interpolation volume (*IIV* for short) that covers boundary fine nodes $\mathcal{X}_b^\alpha$; see also Fig. 16(a).



(a) (b)

Fig. 16. (a) The extruded IIV of a boundary B-spline coarse element D^α. (b) The extruded IIV's sampling at a boundary fine node $\mathbf{x}_k$.

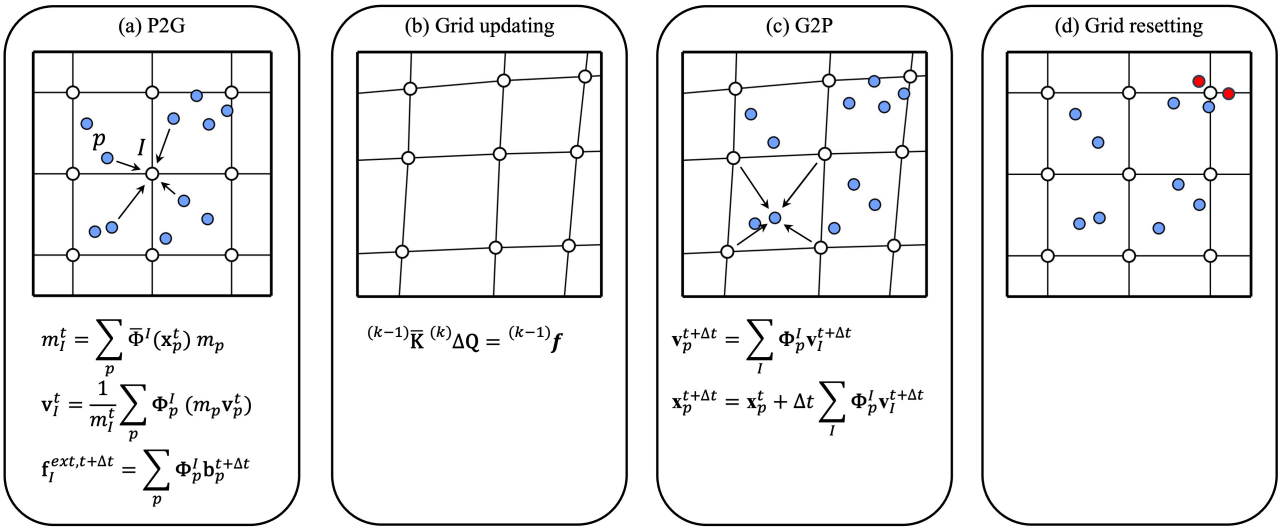


Fig. 17. Flow diagram per time step of our cMPM, where each particle carries the physical quantities, such as the position $\mathbf{x}_p^t$, mass m_p, volume V_p^t, velocity $\mathbf{v}_p^t$, deformation gradient $\mathbf{F}_p^t$, Cauchy stress $\boldsymbol{\sigma}_p^t$, and body force $\mathbf{b}_p^t$. The $\boldsymbol{\Phi}_p^I = \boldsymbol{\Phi}^I(\mathbf{x}_p^t)$ represents the value of the coarsened shape function $\boldsymbol{\Phi}^I(\mathbf{x})$ at particle p, and the $\bar{\Phi}^I(\mathbf{x}_p^t)$ denotes the averaged scalar shape functions of $\boldsymbol{\Phi}_p^I$ for mass extrapolation. In (d), the particles crossing cell boundaries after grid resetting are marked in red.

Suppose $S^i(\mathbf{x})\,, i = 0, 1, \ldots, 5$ is the quadratic B-spline surface associated to the i-th face f^i. Given a boundary fine node $\mathbf{x}_k$, let f^0 be the closest face to it, as illustrated in Fig. 16(b). The value of the extruded IIV at $\mathbf{x}_k$ is defined as

$$\mathcal{I}^\alpha(\mathbf{x}_k) = S^0(\mathbf{x}^0) + \frac{1-a}{2}\,[S^1(\mathbf{x}^1) - S^1(\mathbf{x}^{e1})] + \frac{1-b}{2}\,[S^2(\mathbf{x}^2) - S^2(\mathbf{x}^{e2})] + \frac{a}{2}\,[S^3(\mathbf{x}^3) - S^3(\mathbf{x}^{e3})] + \frac{b}{2}\,[S^4(\mathbf{x}^4) - S^4(\mathbf{x}^{e4})], \quad (36)$$

where f^i, $i = 1, 2, 3, 4$ are the four faces adjacent to face f^0, e^i, $i = 1, 2, 3, 4$ are the four edges of face f^0, $\mathbf{x}^i$, $i = 0, 1, 2, 3, 4$ are the projection points of $\mathbf{x}_k$ onto face f^i, $\mathbf{x}^{ei}$, $i = 1, 2, 3, 4$ are the projection points of point $\mathbf{x}^i$ onto edges e^i. The weights a and b are the relative distance fractions given by,

$$a = \frac{\|\mathbf{x}_k - \mathbf{x}^1\|_2 - \frac{s}{2}}{H - s}, \quad b = \frac{\|\mathbf{x}_k - \mathbf{x}^2\|_2 - \frac{s}{2}}{H - s}. \quad (37)$$

Note also that a coarse element D^α contains $r = 56$ or $r = 98$ coarse nodes with coarse node span $L = H$ or $L = H/2$ respectively, according to Lemma 1.

7.2. Usage in nonlinear deformation

The proposed coarsened shape functions are particularly conducted within the material point method (MPM) framework due to the latter's ability to handle large deformations, coined coarsened MPM (cMPM).

The MPM is a hybrid Eulerian/Lagrangian approach that uses a set of material points (or particles) to represent the structure Ω, and a background mesh (or grid) to represent a temporary computational domain. The MPM procedure primarily consists of four phases to solve the equilibrium in Eq. (2): particle to grid (P2G), grid updating, grid to particle (G2P), and grid resetting, as illustrated in Fig. 17.

Performance of the MPM heavily depends on the continuity of its shape functions [60]. The conventional MPM [23, 51] employs C^0 linear shape functions and consequently suffers from the cell-crossing artifact [24, 52] as the particles (the red ones in Fig. 17(d)) cross the cell boundaries, which may induce a perturbed stress field and even affect the convergence of an implicit solver. Many approaches have been proposed to remedy this artifact, and among them, B-spline MPM (BSMPM) [58, 59] resolves this by employing smooth B-spline basis functions. Similar to BSMPM, our proposed B-spline patch shape functions satisfy C^1-continuity as explained in Section 6.1. On the other hand, the coarsened shape functions are constructed based on a coarse mesh, which greatly reduces the analysis DOFs from a multiscale aspect. Building upon this, the present cMPM approach conducts the MPM simulation on a coarse background mesh and replaces the B-spline basis functions with our C^1-continuous coarsened shape functions in Eq. (15).

The overall calculation process in outlined in Fig. 15.Some comments are made below. First, the coarsened shape function $\boldsymbol{\Phi}^\alpha$, or the control matrix $\mathbf{p}^\alpha$, is computed per time step by updating the boundary–interior mapping matrix $\mathbf{M}^\alpha$, as the material distribution changes. Second, the coarse–boundary mapping matrix $\boldsymbol{\Psi}^\alpha$ is the same for all coarse elements and thus only computed once at the beginning of the algorithm.

Table 2
Simulation statistics of each example of nonlinear material, including the number of particles, material parameters (density ρ, Young's modulus E and Poisson's ratio ν), numbers of coarse elements and coarse nodes, time step, and average timing per time step.

Example	Fig	particles	ρ (kg/m^3), E (Pa), ν	coarse element type	elems, nodes	Δt (s)	timing (s) (per timestep)
Cube shearing	18(a)	216.0K	$1e^3$, $1e^5$, 0.4	boundary B-spline	216, 2.9K	0.01	25.9
Bending beam	20(a)	674.1K	10, -, 0.3	boundary B-spline	1.0K, 12.0K	0.025	77.5
Lattice bar-1	23(a)	992.7K	10, $1e^5$, 0.4	boundary B-spline	2.0K, 20.8K	0.025	142.4
Lattice bar-2	23(a)	992.7K	10, $1e^5$, 0.4	B-spline	2.0K, 23.8K	0.025	345.1
Heterogeneous cube	25(a)	68.9K	10, -, 0.3	boundary B-spline	80, 1.5K	0.01	22.2
Wire rope	33(a)	4.9M	-, -, -	boundary B-spline	11.2K, 100.6K	0.0002	799.3

- denoting parameters with multiple values, whose details are given in Section 8.

8. Numerical examples

Performance of the coarsened shape functions is demonstrated through a variety of numerical examples on a PC with Intel Core i7-11700 2.5 GHz CPU and 64GB RAM. The cMPM was implemented in C++, using Eigen [66] for linear algebra operations, Intel TBB for parallelization, and AMGCL [67] for linear system solving. Planar Dirichlet boundaries were assumed for easy enforcement. Unless specified otherwise, the coarse node span $L = H/2$ and fine node number $n = 7$ per direction were set by default. The performance statistics and problem parameters for all the tested examples are presented in Table 2. The quadratic B-spline MPM (BSMPM) using fine mesh D^h was taken as the benchmark.

The accuracy was assessed by measuring the L^2 norm error of computed displacement and stress as,

$$r_u = \sqrt{\frac{\int_\Omega \|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 \, d\Omega}{\int_\Omega \|\hat{\mathbf{u}}\|_2^2 \, d\Omega}}, \quad r_\sigma = \sqrt{\frac{\int_\Omega \|\sigma - \hat{\sigma}\|_2^2 \, d\Omega}{\int_\Omega \|\hat{\sigma}\|_2^2 \, d\Omega}} \tag{38}$$

where $\hat{\mathbf{u}}$ and $\mathbf{u}$ denote the benchmark and computed displacements, $\hat{\sigma}$ and σ denote the benchmark and computed von Mises stresses respectively. Its performance is extensively tested and compared with related classical approaches, such as BSMPM on nonlinear simulation and EMsFEM on multiscale simulation.

8.1. Performance in simulating heterogeneous nonlinear materials

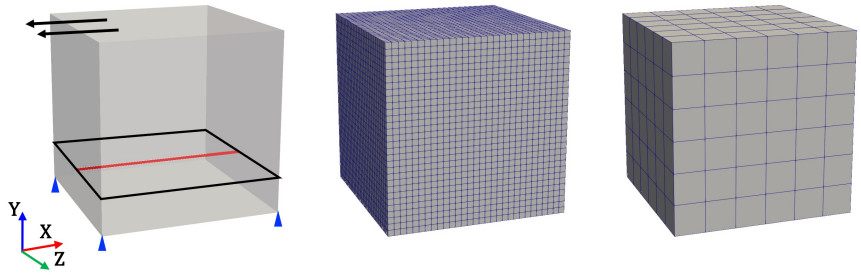
In this section, we tested the performance of the proposed coarsened shape functions in simulating heterogeneous nonlinear materials based on the cMPM framework in Section 7.2.

8.1.1. Effect of C^1-continuity in nonlinear simulation

The coarsened shape functions are of C^1-continuity. This was further tested by observing the stress distribution obtained by cMPM, and its ability in eliminating the cell-crossing artifact. The test was conducted using a simple homogeneous cube example in Fig. 18(a), which was subject to a shear force on its top face along the negative x-axis direction with pressure $p = 3600\ N/m^2$. The solution was also compared with the results obtained from FEM, MPM (conventional MPM [23] using linear shape functions), and BSMPM. The fine element size h of MPM/BSMPM was set equal to the fine knot span size s of the cMPM for fair comparisons, and all other settings (e.g. the particle numbers, time steps) were consistent as well; the strategy was also applied to comparisons in other subsequent examples.

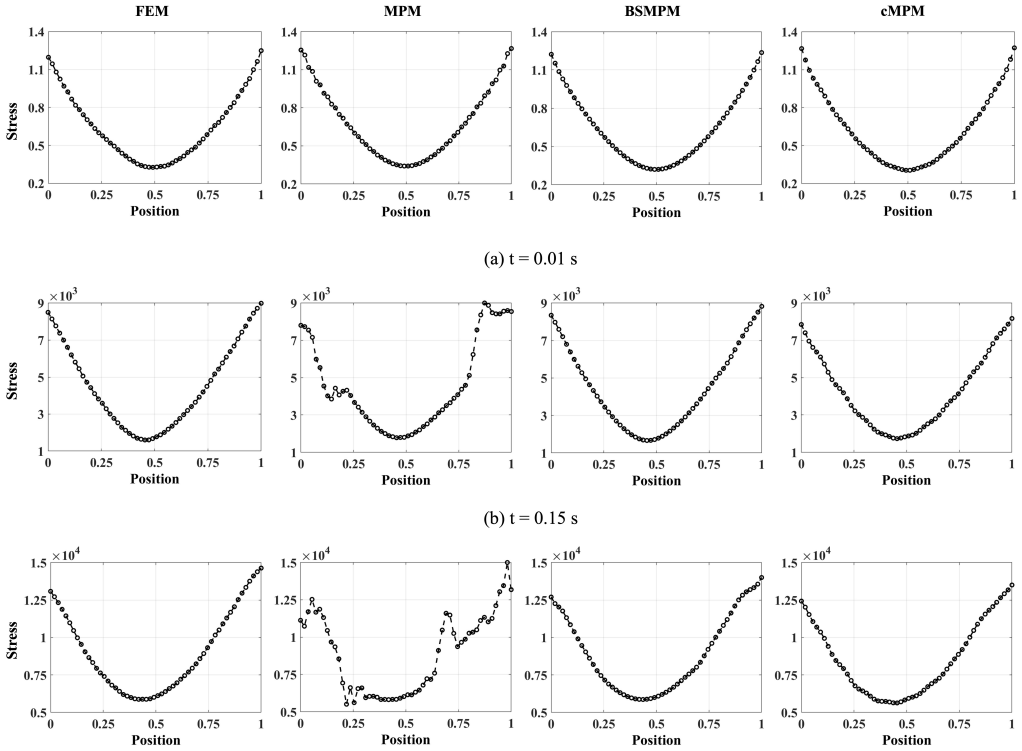
Figs. 18(b)(c) show the active meshes (excluding elements with no particles) for MPM, BSMPM, and cMPM. Their stress solutions along a red line (Fig. 18(a)) in the cube are compared in Fig. 19 at time $t = 0.01,\ 0.15$, and 0.21 s. At time $t = 0.01$ s, before any particle crossed the cell boundary, all four approaches displayed continuous stresses. At $t = 0.15$ s, shortly after some particles crossed cell boundaries, these crossing particles considerably disturbed the solutions and caused oscillating stresses in MPM but not in BSMPM. The cMPM using the proposed coarsened shape functions eliminated the cell-crossing artifact and showed continuous stresses. At $t = 0.21$ s, with even larger deformations, the MPM showed more severe oscillations and large accuracy loss while the solutions obtained by BSMPM and cMPM remained reliable.

Furthermore, the stress solutions on a black plane (Fig. 18(a)) in the cube are compared for four different shape functions: linear, B-spline, CBN [21], and the proposed coarsened B-spline patch shape functions, and shown in Fig. 2 at time $t = 0.21$ s. The latter two shared the same mesh sizes for coarse mesh and local fine mesh. It can be observed that the linear and CBN shape functions produced perturbed stress fields while the B-spline and coarsened shape functions produced continuous solutions due to their C^1-continuity.



(a) Cube (b) MPM and BSMPM (c) cMPM

Fig. 18. Cube shearing example: (a) boundary conditions, where a red line along the x-axis and a black plane along the xz-axes were used for observing the stress fields, (b) fine mesh with 27.0K elements and 34.8K nodes for MPM and BSMPM, and (c) coarse mesh with 216 elements and 2.9K nodes for our cMPM.



(a) t = 0.01 s

(b) t = 0.15 s

(c) t = 0.21 s

Fig. 19. Plots of the three stress snapshots (0.01, 0.15, and 0.21s) for the red line of the cube example (Fig. 18(a)) for the FEM, MPM, BSMPM, and our cMPM.

8.1.2. Comparisons with BSMPM

We tested the accuracy and efficiency of our coarsened shape functions (and the cMPM) in simulating heterogeneous nonlinear materials compared with BSMPM, using a heterogeneous bending beam in Fig. 20(a): having a size of 20×2×2 and containing 40 softer elliptical column inclusions of varied shapes. The Young's moduli were respectively $1e^2$ and $1e^5$ Pa for the inclusions and the remaining. The beam was fixed at its two sides and subject to the self-gravity of $g = 25\ m/s^2$.

Three tests were conducted: cMPM on coarse mesh D^H, BSMPM on global fine mesh D^h, and BSMPM on coarse mesh D^H, denoted by cMPM, BSMPM-fine, and BSMPM-coarse for simplicity. The BSMPM-fine was taken as the benchmark. Figs. 20(b)-(d) plot their corresponding active meshes and Fig. 21 plots their deformations at $t = 0.3$ and 0.6 s. It was observed that the BSMPM-coarse failed to reflect the deformations of softer inclusions, while cMPM produced very close deformations to the benchmark due to its capabilities of capturing fine-scale heterogeneities. Their displacement errors indicated similar phenomena: the maximum error across all time steps was $\max(r_u) = 0.095$ and

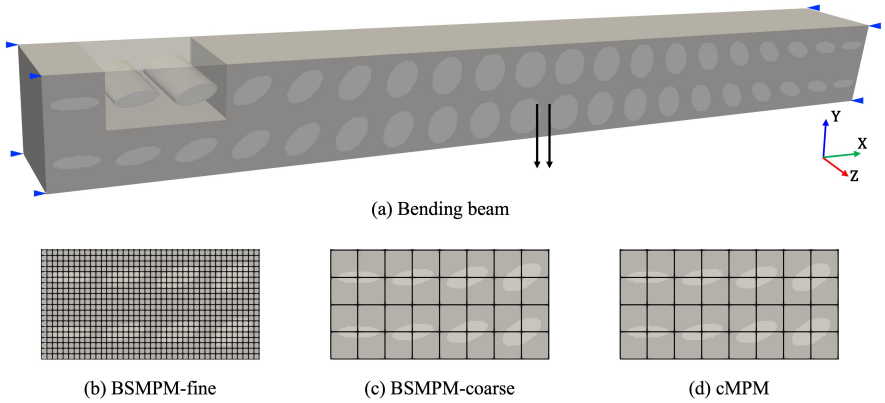


Fig. 20. Bending beam example: (a) boundary conditions, (b) part of the fine mesh with 88.2K elements and 115.8K nodes for BSMPM-fine, (c) part of the coarse mesh with 1.0K elements and 2.6K nodes for the BSMPM-coarse, and (d) part of the coarse mesh with 1.0K elements and 12.0K nodes for the cMPM.

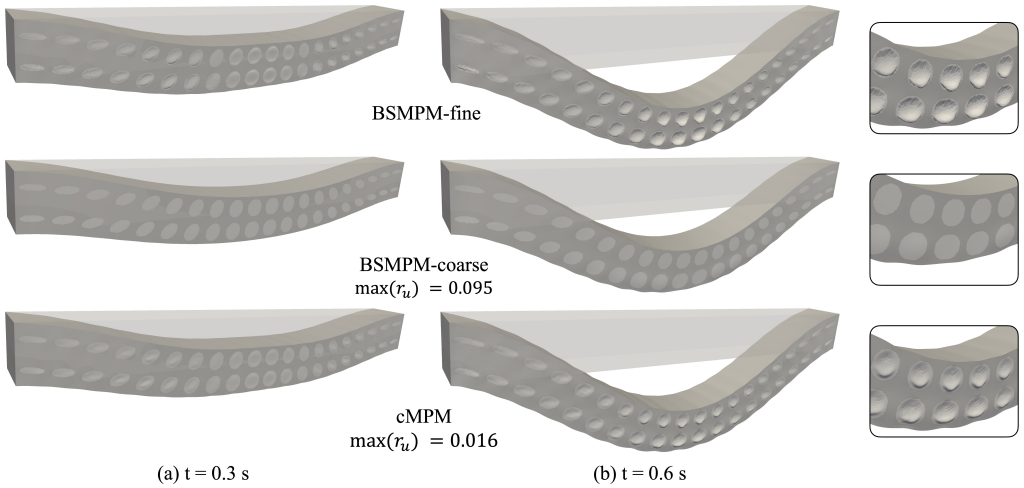


Fig. 21. Deformation plots of the bending beam (Fig. 20(a)) of the BSMPM-fine (above), BSMPM-coarse (middle), and cMPM (below), at time step (a) $t = 0.3$ s and (b) $t = 0.6$ s. The maximum displacement errors of the BSMPM-coarse and cMPM were 0.095 and 0.016 respectively. The local magnified views at $t = 0.6$ s are also provided.

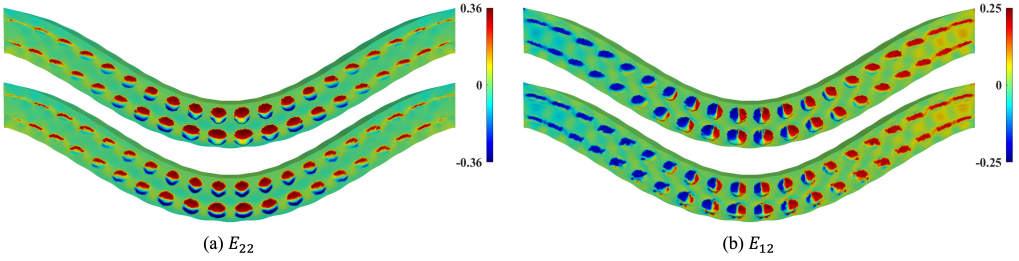
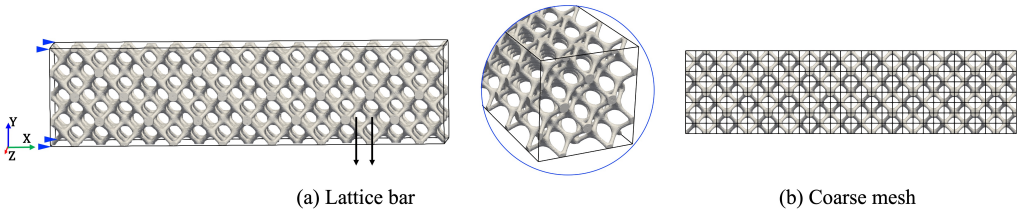


Fig. 22. The Green strain diagrams of the bending beam (Fig. 20(a)) of the BSMPM-fine case (above) and cMPM (below) at $t = 0.6$ s, by showing components (a) E_{22} and (b) E_{12} respectively.

0.016 respectively for BSMPM-coarse and cMPM. We also noticed that the cMPM produced a slightly stiffer deformation than the BSMPM due to its coarsening. Similar phenomenon was also observed in [20].

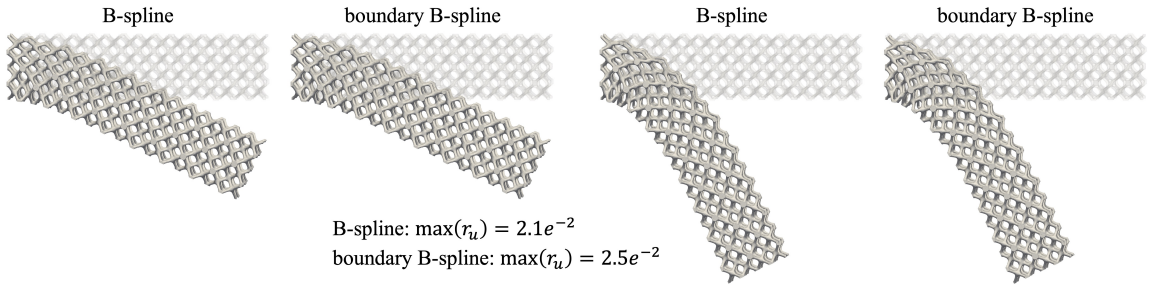
The fields of Green strain $\mathbf{E}$ of BSMPM-fine case and cMPM are also shown in Fig. 22, specifically including the normal component E_{22} of the y-axis and shear component E_{12} of the x- and y-axes. The detailed anisotropies and continuous strain distributions of this heterogeneous beam were observed using the cMPM, demonstrating the ability of the coarsened shape functions in material awareness. Their detailed timing statistics are listed in Table 3, where cMPM achieved almost 11.0× speedup compared with the BSMPM; it would further increase when the problem scale increases.

The *computing integrals* and *computing shape functions* respectively occupied almost 68.4% and 20.3% in cMPM while *solving linear system* occupied almost 88.6% in BSMPM. These results are consistent with the analysis in Section 7.2.



(a) Lattice bar (b) Coarse mesh

Fig. 23. Lattice bar example: (a) boundary conditions, (b) coarse mesh.



(a) t = 1.0 s (b) t = 1.6 s

Fig. 24. Deformations of the lattice bar in Fig. 23(a) using the B-spline (left) and boundary B-spline (right) coarse elements, at time step (a) t = 1.0 s and (b) t = 1.6 s. The maximum displacement errors of the B-spline and boundary B-spline cases were $2.1e^{-2}$ and $2.5e^{-2}$ respectively.

Table 3
Timing statistics in detail for bending beam (Fig. 20(a)) and lattice bar example (Fig. 23(a)). Total timings (for one time step) and timings of three computationally expensive parts (in seconds), including the *computing shape functions* (*SFs*), *computing integrals*, and *solving linear system (linSys)*, are presented.

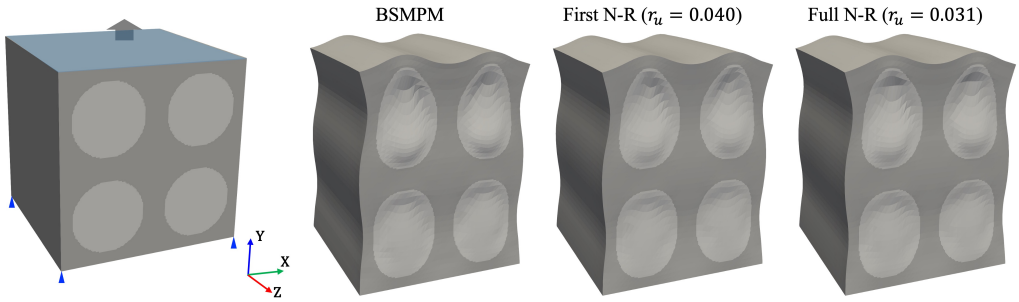
		total timing	compute SFs	compute integrals	solve linSys
Bending beam	BSMPM	855.5	-	40.5	758.4
	cMPM	77.5	15.7	53.0	5.8
Lattice bar	B-spline [1]	345.1	47.2	265.8	15.8
	boundary [1]	142.4	37.0	83.8	9.5

[1] "B-spline" and "boundary" mean the cMPM using B-spline coarse elements and boundary B-spline coarse elements.

8.1.3. Performances at different types of coarse elements

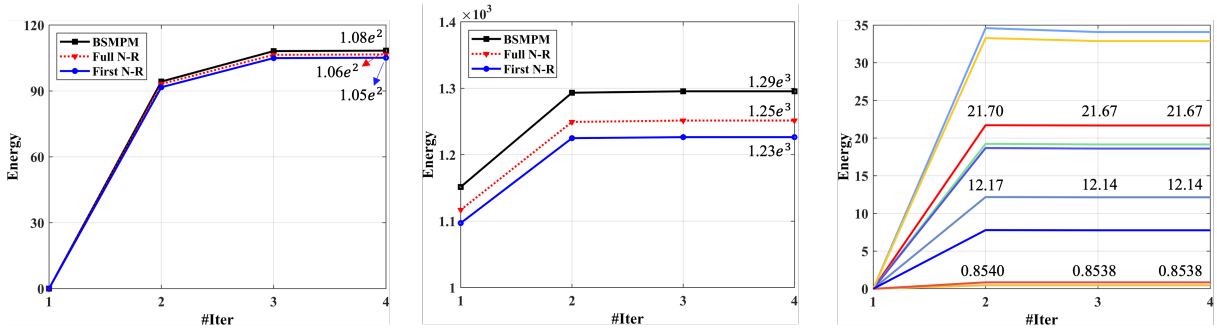
Differences between the proposed shape functions brought by two types of coarse elements were tested here, that is, the B-spline and boundary B-spline coarse elements explained in Section 5.1. A more complex lattice bar example in Fig. 23(a) containing many void regions and fine-scale geometry details, was used here. The lattice bar had a size of $8 \times 2 \times 2$, and it was fixed on its left side face and subject to the self-gravity of $g = 5\ m/s^2$. As listed in Table 2, the two cases having all consistent settings except for different types of coarse elements, were set. Fig. 23(b) plots their common active coarse mesh.

Fig. 24 plots their deformation snapshots at t = 1.0 and 1.6 s. The boundary B-spline case showed close deformation approximation to the B-spline case, respectively of maximum displacement errors $2.5e^{-2}$ and $2.1e^{-2}$. On the other hand, the boundary B-spline case achieved around 3.2 × speedup in *computing integrals* compared with the B-spline case, as seen from the timing statistics in Table 3. This phenomenon can be explained by the fact that boundary B-spline had a decreased coarse node number r per coarse element, or specifically r = 98 and 125 for the two cases respectively. In summary, the boundary B-spline case improved the efficiency while maintaining very close accuracy to the B-spline case.



(a) Heterogeneous cube (b) Deformations at $t = 0.05\ s$

Fig. 25. Heterogeneous cube example: (a) boundary conditions, (b) deformation plots at $t = 0.05$ s for BSMPM, and cMPM using first N-R and full N-R.



(a) Global iterations at $t = 0.01$ s (b) Global iterations at $t = 0.05$ s (c) Local iterations

Fig. 26. Heterogeneous cube's (Fig. 25(a)) convergence results of the strain energy (a) the global coarse simulation at $t = 0.01$ s, (b) the global coarse simulation at $t = 0.05$ s, and (c) the local analysis problem of a coarse element under 10 sampled boundary displacement constraints at $t = 0.01$ s.

8.1.4. Evaluation of non-iterative shape function computation

The coarsened shape functions are computed via a local analysis for each coarse element, as explained in Section 5.2 and included in the overall algorithm in Fig. 15. Although a nonlinear problem, the local analysis is computed using only the first iteration of the Newton-Raphson method. Its performance is tested against using a full Newton-Raphson method, respectively denoted by first N-R and full N-R for simplicity. Results from the BSMPM were taken as benchmark.

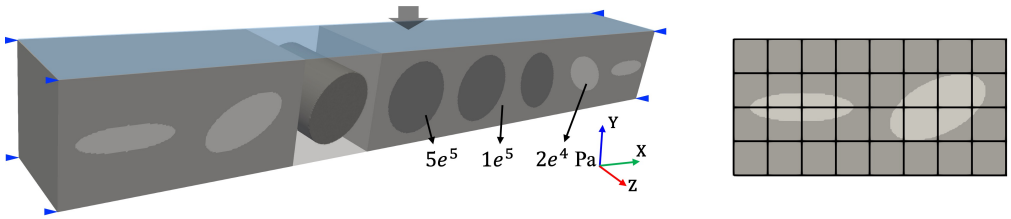
The heterogeneous cube example in Fig. 25(a) was first used, a simplified version from the example in Fig. 20(a). The cube was fixed at its bottom face and subject to an upward force of pressure $p = 5000N/m^2$ on the upper face. Fig. 25(b) plots the largest global deformations at $t = 0.05$ s. Under the two different local computation strategies, using the first N-R and full N-R, the cube exhibited similar global deformations, respectively of errors $r_u = 0.040$ and 0.031. Note that the first N-R strategy is much cheaper for the local analysis, taking around 2.96 s per timestep while the full N-R taking around 392.20 s.

Detailed iteration processes are further shown in Fig. 26 in terms of the strain energy variations. It was observed in (a) and (b), respectively at time step $t = 0.01,\ 0.05$, either using the first N-R or using the full N-R, we had reliable convergence at a iteration step as small as 4. In addition, the two strategies both closely approximated the benchmark results. The nice phenomenon can perhaps be explained from the results in Fig. 26(c) on the iteration history of the local analysis problem of a coarse element under 10 sampled boundary displacement constraints, where the strain energy has deduced drastically at the first iteration.

8.2. Comparisons with related EMsFEM approaches

The performance of our C^1-continuous coarsened shape functions was compared with closely related EMsFEM approaches on linear elastic problem. All these approaches construct a set of numerical shape functions for each coarse element by solving the local problem. Their main difference lies in the forms of constructed shape functions and imposed displacement constraints in solving the local problem. The approaches include:

- *Diagonal* [15], adopting piecewise linear shape functions of diagonal matrix-form, and imposing displacement constraints generated from linear interpolation.



(a) Bending beam (b) Coarse mesh

Fig. 27. Heterogeneous bending beam example: (a) material distributions and boundary conditions, where Young's moduli were $5e^5$, $2e^4$, and $1e^5$ Pa for the four middle elliptical column inclusions, other four inclusions on both sides, and the remaining matrix, (b) part of the employed $32 \times 4 \times 4$ coarse mesh.

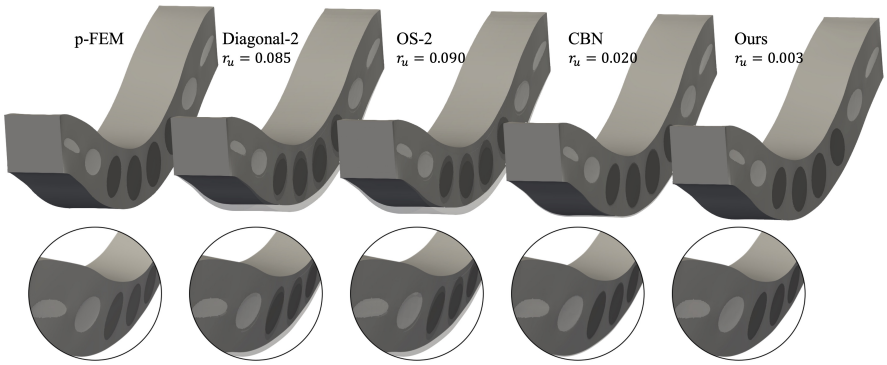


Fig. 28. Deformations of the bending beam (Fig. 27(a)) for approaches: p-FEM, Diagonal-2, OS-2, CBN, and ours, where the shadow volumes denote the benchmark deformations.

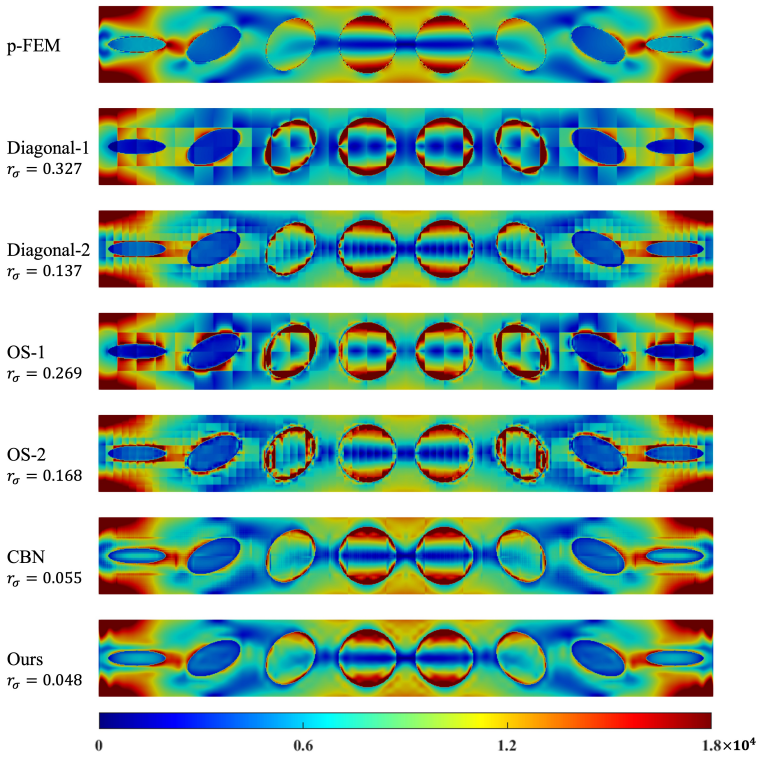


Fig. 29. The von Mises stress fields of the bending beam (Fig. 27(a)) for approaches: p-FEM, Diagonal-1, Diagonal-2, OS-1, OS-2, CBN, and ours.

- *Oversampling (OS)* [46], adopting piecewise linear shape functions of matrix-form, and imposing oscillating dis-

placement constraints generated from the local problem on the oversampling domain.

- *CBN* [21], adopting piecewise linear shape functions of matrix-form, and imposing displacement constraints generated from cubic Bézier interpolation.
- *Ours*, adopting B-spline patch shape functions of matrix-form, and imposing displacement constraints generated from intermediate interpolation volume (IIV).

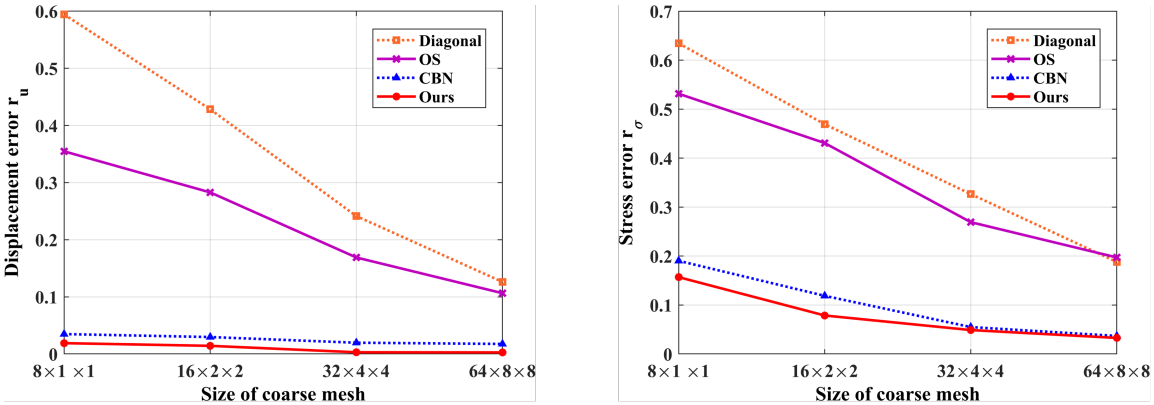
Results from the hexhedral FEM (polynomial degree $p = 2$) using fine mesh D^h were taken as the benchmark.

The comparison results are summarized in Table 4. Ours always produced the closest approximation to the benchmark at a relatively expensive cost while *Diagonal* and *OS* took the least time at a much accuracy reduction.

Table 4
Simulation statistics: p-FEM, Diagonal, OS, CBN, and ours for the heterogeneous bending beam in Fig. 27(a) and lattice bar in (Fig. 23(a)). The timings include local analysis (per coarse element) and global coarse simulation, and the unit of timing is seconds.

Example	Approach	r_u	r_σ	#DOF	Timing (local)	Timing (global)
Bending beam (Fig. 27(a))	p-FEM	–	–	12.6M	–	16609.04
	Diagonal-1	0.241	0.327	2.5K	0.02	0.07
	Diagonal-2 [1]	0.085	0.137	29.4K	$2.6e^{-4}$	0.10
	OS-1	0.169	0.269	2.5K	0.19	0.07
	OS-2 [1]	0.090	0.168	29.4K	$9.8e^{-3}$	0.10
	CBN	0.020	0.055	36.9K	0.05	0.25
	Ours	0.003	0.048	65.5K	0.34	2.71
Lattice bar (Fig. 23(a))	p-FEM	–	–	10.2M	–	14346.94
	Diagonal	0.954	0.700	188.3K	$2.3e^{-4}$	0.26
	OS	0.270	0.269	188.3K	$9.2e^{-3}$	0.26
	CBN	0.013	0.208	132.7K	0.14	2.56
	Ours	0.007	0.146	152.4K	0.45	17.77

[1] The coarse mesh size for Diagonal-2 and OS-2 was refined to $80 \times 10 \times 10$ for obtaining a comparable number of DOFs as CBN and ours.



(a) Displacement error variations

(b) Stress error variations

Fig. 30. Error variations of the bending beam (Fig. 27(a)) at different coarse mesh sizes for approaches: Diagonal, OS, CBN, and ours.

8.2.1. A heterogeneous beam example of three different materials

The first test was conducted on a heterogeneous bending beam in Fig. 27(a), consisting of three different materials of different Young's moduli: $5e^5$, $2e^4$, and $1e^5$ Pa for the four middle elliptical column inclusions, other four inclusions on both sides, and the remaining matrix. The beam was of $8 \times 1 \times 1$, fixed at its two sides and subject to a downward pressure $p = 1000\ N/m^2$ on its upper face. The coarse mesh was of size $32 \times 4 \times 4$ (Fig. 27(b)), and the fine mesh was of size $10 \times 10 \times 10$. We also tested *Diagonal* and *OS* approaches at coarse mesh size of $80 \times 10 \times 10$ and local fine mesh size of $4 \times 4 \times 4$, named *Diagonal-2* and *OS-2*, producing a comparable number of DOFs as *CBN* and ours for a more fair comparison. The coarse node had a span of $L = H/3$ in our approach.

The global deformations and stress distributions are compared in Figs. 28 and 29; see also Table 4. All of these approaches showed close deformation approximation to the benchmark. Still, different accuracies were still observed for *Diagonal-1*, *OS-1*, *Diagonal-2*, *OS-2*, *CBN* and ours at relative errors r_u of 0.241, 0.169, 0.085, 0.090, 0.020 and 0.003. On the other hand, large stress discontinuities were clearly observed in the results of *Diagonal-1* and *OS-1* in Fig. 29 respectively at errors 0.327 and 0.269, which was due to the limited linear displacement constraints at the coarse element boundaries of *Diagonal* or oversampling domain boundaries of *OS*. The phenomenon has been improved in *Diagonal-2* and *OS-2* but still at large errors of 0.137 and 0.168 respectively. *CBN* showed more continuous stress distributions at smaller error r_σ of 0.055. However, stress discontinuity was still observed, perhaps due to its C^0-continuous shape functions. Ours demonstrated continuous stress distribution with the smallest error of 0.048.

Fig. 30 also compares performances of the approaches at different coarse mesh sizes: $8\times1\times1$, $16\times2\times2$, $32\times4\times4$, and $64\times8\times8$, where the global fine mesh size was unchanged as $320\times40\times40$. The errors decreased rapidly as the coarse mesh size increased (producing a smaller-sized local fine mesh). In all settings, our coarsened shape functions always had the smallest errors.

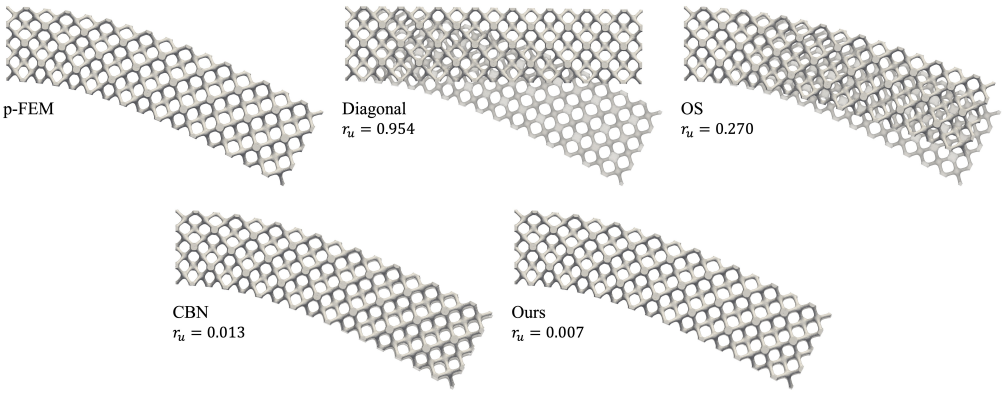


Fig. 31. Deformations of the lattice bar (Fig. 23(a)) for approaches: p-FEM, Diagonal, OS, CBN, and ours, where the shadow volumes denote the benchmark deformations.

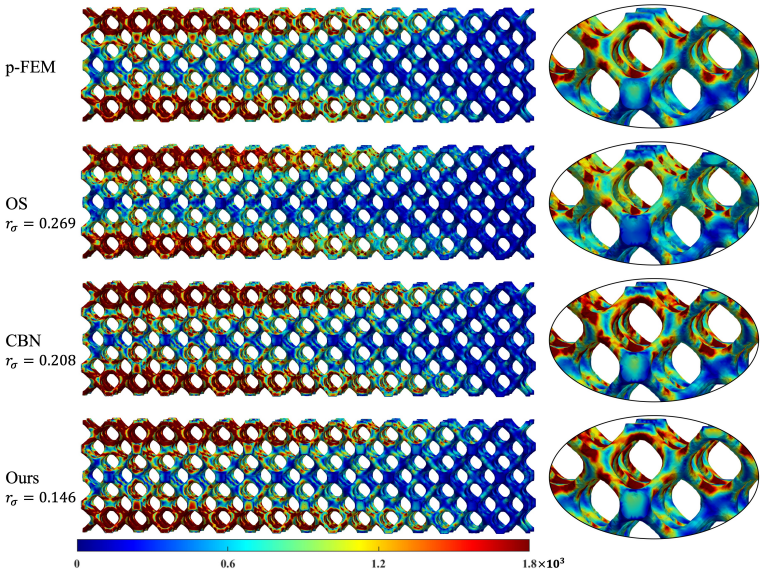


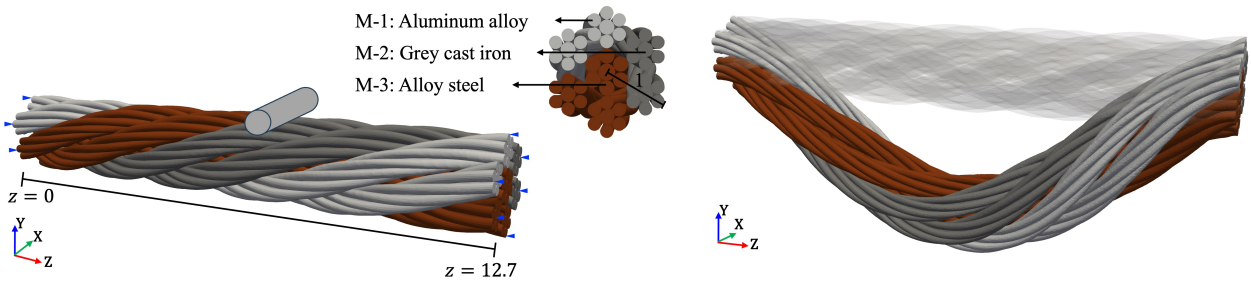
Fig. 32. The von Mises stress fields of the lattice bar (Fig. 23(a)) for approaches: p-FEM, OS, CBN, and ours.

8.2.2. A complex example of lattice bar

The comparisons were also conducted on a complex lattice bar example in Fig. 23(a), being fixed at its left face and subject to a downward pressure $p = 60\ N/m^2$ on its right face. The example contained many void regions and fine-scale geometry details, raising more challenges to effective shape function construction. In our approach, the coarse node had

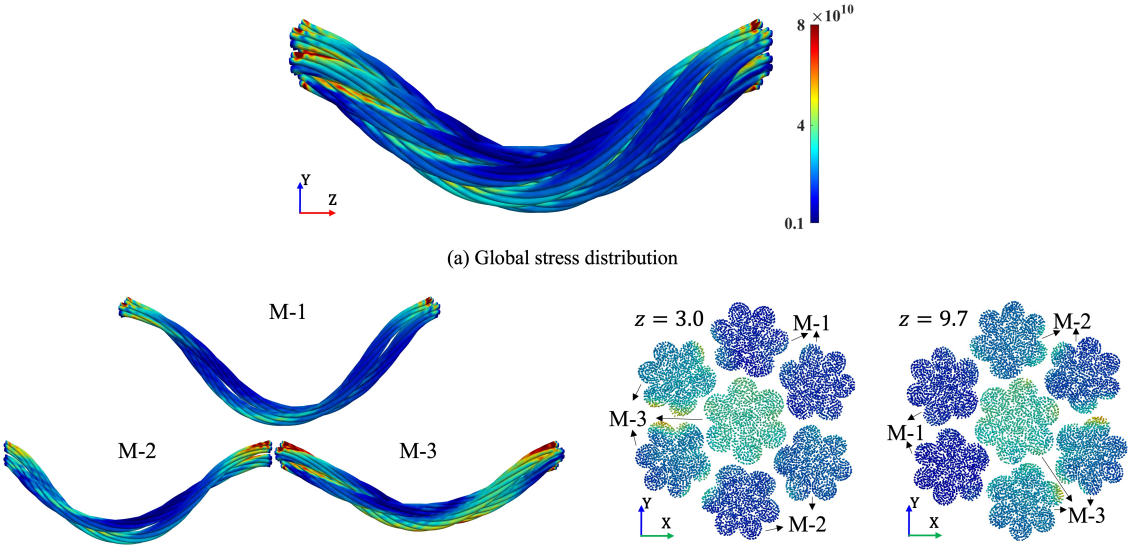
a span of $L = H/3$. The sizes of coarse mesh and local fine mesh were $32 \times 8 \times 8$ and $12 \times 12 \times 12$ respectively for *CBN* and ours while $128 \times 32 \times 32$ and $3 \times 3 \times 3$ for *Diagonal* and *OS*.

The deformations and stress distributions are summarized in Figs. 31 and 32; see also Table 4. *Diagonal* failed to produce a reliable deformation, perhaps due to its shape functions' lack of coupling terms amongst different axes. *OS* tended to stiffen the structure due to the employed linear interpolation at oversampling domain boundaries. *CBN* and ours exhibited deformations almost identical to the benchmark. In addition, our approach achieved more accurate stress results and exhibited better continuity in comparison with other approaches.



(a) Wire rope (b) Deformation plot

Fig. 33. Wire rope example: (a) boundary conditions and material distributions, (b) deformation plot using our cMPM at $t = 0.0072$ s.



(a) Global stress distribution

(b) Stress distributions of the three materials (c) Stress on particles of the two sections

Fig. 34. The von Mises stress fields of the wire rope in Fig. 33(a) computed via our cMPM, including diagrams of (a) the global structure, (b) each individual material and (c) particles of two sections at $z = 3$ and $z = 9.7$.

Table 5
Parameters for the three materials in the nonlinear hybrid ropes (Fig. 33(a)), including density (kg/m^3), Young's modulus (Pa), and Poisson's ratio.

	Material	Density	Young's modulus	Poisson's ratio
M-1 (silver)	Aluminum alloy	$2.7e^3$	$7e^{10}$	0.33
M-2 (grey)	Grey cast iron	$7e^3$	$1.18e^{11}$	0.30
M-3 (brown)	Alloy steel	$7.9e^3$	$2.06e^{11}$	0.30

8.3. An example of nonlinear hybrid ropes under contacts

We also tested the performance of the cMPM in handling heterogeneous structures with fine-scale geometric and material details, even with contacts, on an example of complex hybrid wire ropes in Fig. 33(a). The example had a

length of 12.7 and a radius of 1, and contained three types of rope bundles of different materials: aluminum alloy (M-1, silver color), grey cast iron (M-2, grey), and alloy steel (M-3, brown); the material parameters are listed in Table 5. The ropes were fixed at both its left and right sides, and subject to a downward force $1.5e^{10}N$ on its top.

In the cMPM approach, 4.9M MPM particles and a coarse mesh with 11.2K elements and 100.6K coarse nodes were used. It resulted in 1.4M fine elements (knot spans), a size too huge to be conducted on a regular PC using fine mesh based BSMPM. On average, the cMPM took 799.3 s per time step.

The overall deformation at time step $t = 0.0072$ s is plotted in Fig. 33(b). And several associated von Mises stress fields are shown in Fig. 34, including diagrams of (a) the global structure, (b) each individual material, and (c) particles of the two sections at $z = 3$ and $z = 9.7$. Different stress magnitudes were observed in the three different materials: the softer (M-1) material showed relatively small stress while the stiffer (M-3) material showed larger stress, demonstrating the coarsened shape functions' ability to capture the structure's fine-scale heterogeneities. We also observed that cMPM effectively handled collisions between these different rope bundles from its inherent automatic no-penetration contact algorithm, in a similar performance as the MPM.

9. Conclusion

In this study, a type of matrix-valued coarsened shape functions of C^1-continuity are proposed and constructed for the first time as free-form B-spline patches, being able to capture the fine-scale heterogeneities within the coarse element that allows for continuous stress/strain fields. It much improves the simulation efficiency with an order of magnitude and simultaneously maintains high simulation accuracy. The coarsened shape functions satisfy the important property of partition of unity and are efficiently derived via consequent coarse–boundary mapping and boundary–interior mapping, which only involves small efforts in solving a linear system of small size for each coarse element. The shape functions' application in MPM-based simulation for nonlinear heterogeneous structures was also implemented and tested, which demonstrated its reliable performance, high accuracy, and improved efficiency.

While our approach is able to simulate nonlinear heterogeneous structures with high accuracy and efficiency, the following topics are worthy of future exploration to further alleviate the current limitations. First, the C^1-continuous coarsened shape functions can be applied directly to those numerical analysis problems involving second derivatives of the unknown function, such as the Kirchhoff-Love shell theory [25] and strain gradient elasticity [26, 27]. Second, only potential energy but kinetic energy is considered in the local equilibrium problem for constructing the boundary–interior mapping, which may be improved by introducing substructural inertial effect condensation [68]. In addition, the approach can still be extended in handling more complicated issues, such as frictional contact, separate parts that are close to each other [15], weak enforcement of essential boundary conditions [65]. Last by not least, we are also working on topics on extending the approach to resolve the critical seamless CAD/CAE integration issue [7, 8, 9, 10, 11].

Acknowledgements

The work described in this paper is partially supported by the National Key Research and Development Program of China (No. 2020YFC2201303), the NSF of China (No. 62372401), and Zhejiang Provincial Science and Technology Plan Project (2022C01052).

Appendix A. Proof to Lemma 3

A brief proof is given here. More properties of B-spline are referred to [63]. Consider two adjacent shape functions $\Phi_w^{\alpha,I}$ and $\Phi_w^{\beta,I}$ on coarse elements D^α and D^β, abbreviated as $\Phi^\alpha(x,y)$ and $\Phi^\beta(x,y)$ here, for $0 \le x,y \le 1$, let $p_{i,j}^\alpha$ and $p_{i,j}^\beta$, $i,j = 1,2,\ldots,n$ be their control values, respectively of the control vectors $\mathbf{p}_w^{\alpha,I}$ and $\mathbf{p}_w^{\beta,I}$ in Eq. (12). The values of Φ^α at edge $x = 1$ and Φ^β at edge $x = 0$ are respectively given by,

$$\begin{aligned} \Phi^\alpha(1,y) &= \frac{1}{2}\sum_j^n N_{j,2}(y)(p_{n-1,j}^\alpha + p_{n,j}^\alpha), \\ \Phi^\beta(0,y) &= \frac{1}{2}\sum_j^n N_{j,2}(y)(p_{1,j}^\beta + p_{2,j}^\beta). \end{aligned} \tag{A.1}$$

The first derivatives of Φ^α at edge $x = 1$ and Φ^β at edge $x = 0$, w.r.t. parameters x and y, are respectively computed as,

$$\begin{aligned}
\frac{\partial \Phi^\alpha(1,y)}{\partial x} &= \frac{1}{s}\sum_j^n N_{j,2}(y)\,(p^\alpha_{n,j} - p^\alpha_{n-1,j}),\\
\frac{\partial \Phi^\alpha(1,y)}{\partial y} &= \frac{1}{2}\sum_j^n \frac{\partial N_{j,2}(y)}{\partial y}\,(p^\alpha_{n-1,j} + p^\alpha_{n,j}),\\
\frac{\partial \Phi^\beta(0,y)}{\partial x} &= \frac{1}{s}\sum_j^n N_{j,2}(y)\,(p^\beta_{2,j} - p^\beta_{1,j}),\\
\frac{\partial \Phi^\beta(0,y)}{\partial y} &= \frac{1}{2}\sum_j^n \frac{\partial N_{j,2}(y)}{\partial y}\,(p^\beta_{1,j} + p^\beta_{2,j}).
\end{aligned} \tag{A.2}$$

Consequently, the two surfaces Φ^α and Φ^β are C^1-continuous if they share the same values and first derivatives along the common boundary,

$$\begin{aligned}
\Phi^\alpha(1,y) &= \Phi^\beta(0,y),\\
\frac{\partial \Phi^\alpha(1,y)}{\partial x} &= \frac{\partial \Phi^\beta(0,y)}{\partial x},\\
\frac{\partial \Phi^\alpha(1,y)}{\partial y} &= \frac{\partial \Phi^\beta(0,y)}{\partial y}.
\end{aligned} \tag{A.3}$$

Combining Eqs.(A.1), (A.2), and (A.3), the C^1-continuity condition between the pair of adjacent surfaces Φ^α and Φ^β becomes,

$$p^\alpha_{n-1,j} = p^\beta_{1,j}, \quad p^\alpha_{n,j} = p^\beta_{2,j}, \quad j = 1, 2, \ldots, n, \tag{A.4}$$

which means the last two rows of the control values of Φ^α and the first two rows of the control values of Φ^β, or equivalently their control values corresponding to the common boundary fine nodes, are equal; See blue points in Fig. 11.

It is clear from Lemma 2 that the above conditions hold, and consequently the two shape functions Φ^α and Φ^β are C^1-continuous. □

References

[1] A. Heinz, A. Haszler, C. Keidel, S. Moldenhauer, R. Benedictus, W. Miller, Recent development in aluminium alloys for aerospace applications, Materials Science and Engineering: A 280 (1) (2000) 102–107.

[2] J. T. Oden, K. S. Vemaganti, Estimation of local modeling error and goal-oriented adaptive modeling of heterogeneous materials: I. error estimates and adaptive algorithms, Journal of Computational Physics 164 (1) (2000) 22–47.

[3] H. Liu, T. Maekawa, N. Patrikalakis, E. Sachs, W. Cho, Methods for feature-based design of heterogeneous solids, Computer-Aided Design 36 (12) (2004) 1141–1159.

[4] I. Temizer, T. Zohdi, A numerical method for homogenization in non-linear elasticity, Computational Mechanics 40 (2007) 281–298.

[5] X. Y. Kou, S. T. Tan, Heterogeneous object modeling: A review, Computer-Aided Design 39 (4) (2007) 284–301.

[6] D. K. Rajak, D. D. Pagar, P. L. Menezes, E. Linul, Fiber-reinforced polymer composites: Manufacturing, properties, and applications, Polymers 11 (10) (2019) 1667.

[7] N. Patrikalakis, T. Maekawa, Shape Interrogation for Computer Aided Design and Manufacturing, Vol. 15, Springer, 2002.

[8] V. Shapiro, I. Tsukanov, A. Grishin, Geometric issues in computer aided design/computer aided engineering integration, Journal of Computing and Information Science in Engineering 11 (2011) 021005.

[9] E. Rank, M. Ruess, S. Kollmannsberger, D. Schillinger, A. Düster, Geometric modeling, isogeometric analysis and the finite cell method, Computer Methods in Applied Mechanics and Engineering 249-252 (2012) 104–115.

[10] B. Marussig, T. J. R. Hughes, A review of trimming in isogeometric analysis: Challenges, data exchange and simulation aspects, Archives of Computational Methods in Engineering 25 (2017) 1059 – 1127.

[11] B. Wassermann, S. Kollmannsberger, S. Yin, L. Kudela, E. Rank, Integrating cad and numerical analysis: 'dirty geometry' handling using the finite cell method, Computer Methods in Applied Mechanics and Engineering 351 (1) (2019) 808–835.

[12] J. C. Teo, C.-K. Chui, Z. L. Wang, S. H. Ong, C. H. Yan, S. C. Wang, H.-K. Wong, H. S. Teoh, Heterogeneous meshing and biomechanical modeling of human spine, Medical Engineering and Physics 29 (2) (2007) 277–290.

[13] 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, Archives of Computational Methods in Engineering 22 (3) (2015) 391–455.

[14] T. Y. Hou, X. Wu, A multiscale finite element method for elliptic problems in composite materials and porous media, Journal of Computational Physics 134 (1) (1997) 169–189.

[15] M. Nesme, P. G. Kry, L. Jeřábková, F. Faure, Preserving topology and elasticity for embedded deformable models, ACM Transactions on Graphics 28 (3) (2009) 52.

[16] Y. Efendiev, J. Galvis, X. Wu, Multiscale finite element methods for high-contrast problems using local spectral basis functions, Journal of Computational Physics 230 (4) (2011) 937–955.
[17] H. W. Zhang, Y. Liu, S. Zhang, J. Tao, J. K. Wu, B. S. Chen, Extended multiscale finite element method: Its basis and applications for mechanical analysis of heterogeneous materials, Computational Mechanics 53 (4) (2014) 659–685.
[18] G. Papanicolau, A. Bensoussan, J.-L. Lions, Asymptotic Analysis for Periodic Structures, Vol. 5, Elsevier, 1978.
[19] P. W. Chung, K. K. Tamma, R. R. Namburu, Asymptotic expansion homogenization for heterogeneous media: computational issues and applications, Composites Part A: Applied Science and Manufacturing 32 (9) (2001) 1291–1301.
[20] J. Chen, H. Bao, T. Wang, M. Desbrun, J. Huang, Numerical coarsening using discontinuous shape functions, ACM Transactions on Graphics 37 (4) (2018) 1–12.
[21] M. Li, J. Hu, Analysis of heterogeneous structures of non-separated scales using curved bridge nodes, Computer Methods in Applied Mechanics and Engineering 392 (2022) 114582.
[22] X. Tang, C. Zheng, S. Wu, J. Zhang, A novel four-node quadrilateral element with continuous nodal stress, Applied Mathematics and Mechanics 30 (2009) 1519–1532.
[23] D. Sulsky, Z. Chen, H. L. Schreyer, A particle method for history-dependent materials, Computer Methods in Applied Mechanics and Engineering 118 (1-2) (1994) 179–196.
[24] S. G. Bardenhagen, E. M. Kober, The generalized interpolation material point method, Computer Modeling in Engineering and Sciences 5 (6) (2004) 477–496.
[25] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu, The Finite Element Method: Its Basis and Fundamentals, Elsevier, 2005.
[26] P. Fischer, J. Mergheim, P. Steinmann, On the C^1 continuous discretization of non-linear gradient elasticity: a comparison of nem and fem based on bernstein–bézier patches, International Journal for Numerical Methods in Engineering 82 (10) (2010) 1282–1307.
[27] S. Hosseini, J. Niiranen, 3D strain gradient elasticity: Variational formulations, isogeometric analysis and model peculiarities, Computer Methods in Applied Mechanics and Engineering 389 (2022) 114324.
[28] M. V. Le, J. Yvonnet, N. Feld, F. Detrez, The coarse mesh condensation multiscale method for parallel computation of heterogeneous linear structures without scale separation, Computer Methods in Applied Mechanics and Engineering 363 (2020) 112877.
[29] K. Stüben, Algebraic multigrid (AMG): experiences and comparisons, Applied Mathematics and Computation 13 (3) (1983) 419–451.
[30] W. L. Briggs, V. E. Henson, S. F. McCormick, A Multigrid Tutorial, 2nd Edition, Society for Industrial and Applied Mathematics, 2000.
[31] Z. Wu, L. Xia, S. Wang, T. Shi, Topology optimization of hierarchical lattice structures with substructuring, Computer Methods in Applied Mechanics and Engineering 345 (2019) 602–617.
[32] Z. Liu, L. Xia, Q. Xia, T. Shi, Data-driven design approach to hierarchical hybrid structures with multiple lattice configurations, Structural and Multidisciplinary Optimization (2020) 1–9.
[33] R. Smit, W. Brekelmans, H. Meijer, Prediction of the mechanical behavior of nonlinear heterogeneous systems by multi-level finite element modeling, Computer Methods in Applied Mechanics and Engineering 155 (1) (1998) 181–192.
[34] L. Xia, P. Breitkopf, Design of materials using topology optimization and energy-based homogenization approach in MATLAB, Structural and Multidisciplinary Optimization 52 (6) (2015) 1229–1241.
[35] A. Tognevi, M. Guerich, J. Yvonnet, A multi-scale modeling method for heterogeneous structures without scale separation using a filter-based homogenization scheme, International Journal for Numerical Methods in Engineering 108 (1) (2016) 3–25.
[36] J. Yvonnet, Computational Homogenization of Heterogeneous Materials with Finite Elements, Springer, 2019.
[37] V. B. Tan, K. Raju, H. P. Lee, Direct FE^2 for concurrent multilevel modelling of heterogeneous structures, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112694.
[38] T. Lesičar, Z. Tonković, J. Sorić, A second-order two-scale homogenization procedure using C^1 macrolevel discretization, Computational Mechanics 54 (2) (2014) 425–441.
[39] H. Zhang, J. Wu, J. Lü, Z. Fu, Extended multiscale finite element method for mechanical analysis of heterogeneous materials, Acta Mechanica Sinica 26 (6) (2010) 899–920.
[40] H. Liu, X. Sun, Y. Xu, X. Chu, A hierarchical multilevel finite element method for mechanical analyses of periodical composite structures, Composite Structures 131 (2015) 115–127.
[41] H. Liu, L. Zhang, D. Yang, H. Zhang, An efficient multiscale method for 2D large displacement–small strain analysis of heterogeneous materials, Computational Materials Science 83 (2014) 443–456.
[42] H. Zhang, H. Liu, J. Wu, A uniform multiscale method for 2D static and dynamic analyses of heterogeneous materials, International Journal for Numerical Methods in Engineering 93 (7) (2013) 714–746.
[43] H. Liu, H. Zhang, A uniform multiscale method for 3D static and dynamic analyses of heterogeneous materials, Computational Materials Science 79 (2013) 159–173.
[44] P. Fu, H. Liu, X. Chu, An efficient multiscale computational formulation for geometric nonlinear analysis of heterogeneous piezoelectric composite, Composite Structures 167 (2017) 191–206.
[45] H. Liu, J. Lv, An equivalent continuum multiscale formulation for 2D geometrical nonlinear analysis of lattice truss structure, Composite Structures 160 (2017) 335–348.
[46] H. Liu, D. Yang, J. Wu, Y. Zheng, H. Zhang, An open-source matlab implementation for elastic alalyses of heterogeneous materials using the extended multiscale finite element method, International Journal for Multiscale Computational Engineering 19 (2021) 19–43.
[47] B. Cockburn, Discontinuous galerkin methods, ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik: Applied Mathematics and Mechanics 83 (11) (2003) 731–754.
[48] M. Klimczak, W. Cecot, Higher order multiscale finite element method for heat transfer modeling, Materials 14 (14) (2021) 3827.
[49] D. A. White, J. Kudo, K. E. Swartz, D. A. Tortorelli, S. E. Watts, A reduced order model approach for finite element analysis of cellular structures, Finite Elements in Analysis and Design 214 (2023) 103855.
[50] J. M. Melenk, I. Babuska, The partition of unity finite element method: basic theory and applications, Computer Methods in Applied Mechanics and Engineering 139 (1) (1996) 289–314.
[51] D. Sulsky, S.-J. Zhou, H. L. Schreyer, Application of a particle-in-cell method to solid mechanics, Computer Physics Communications 87 (1-2) (1995) 236–252.
[52] S. Ma, X. Zhang, X. Qiu, Comparison study of mpm and sph in modeling hypervelocity impact problems, International Journal of Impact

Engineering 36 (2) (2009) 272–282.
[53] A. Sadeghirad, R. M. Brannon, J. Burghardt, A convected particle domain interpolation technique to extend applicability of the material point method for problems involving massive deformations, International Journal for Numerical Methods in Engineering 86 (12) (2011) 1435–1456.
[54] A. Sadeghirad, R. M. Brannon, J. Guilkey, Second-order convected particle domain interpolation (CPDI2) with enrichment for weak discontinuities at material interfaces, International Journal for Numerical Methods in Engineering 95 (11) (2013) 928–952.
[55] V. P. Nguyen, C. T. Nguyen, T. Rabczuk, S. Natarajan, On a family of convected particle domain interpolations in the material point method, Finite Elements in Analysis and Design 126 (2017) 50–64.
[56] A. de Vaucorbeil, V. P. Nguyen, C. R. Hutchinson, A total-lagrangian material point method for solid mechanics problems involving large deformations, Computer Methods in Applied Mechanics and Engineering 360 (2020) 112783.
[57] D. Z. Zhang, X. Ma, P. T. Giguere, Material point method enhanced by modified gradient of shape function, Journal of Computational Physics 230 (16) (2011) 6379–6398.
[58] M. Steffen, P. Wallstedt, J. Guilkey, R. Kirby, M. Berzins, Examination and analysis of implementation choices within the material point method (mpm), Computer Modeling in Engineering and Sciences 31 (2) (2008) 107–127.
[59] Y. Gan, Z. Sun, Z. Chen, X. Zhang, Y. Liu, Enhancement of the material point method using B-spline basis functions, International Journal for Numerical Methods in Engineering 113 (3) (2018) 411–431.
[60] A. de Vaucorbeil, V. P. Nguyen, S. Sinaie, J. Y. Wu, Material point method after 25 years: Theory, implementation, and applications, Advances in Applied Mechanics 53 (2020) 185–398.
[61] D. J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Computer Methods in Applied Mechanics and Engineering 99 (2-3) (1992) 235–394.
[62] K.-J. Bathe, Finite element procedures, Klaus-Jurgen Bathe, 2006.
[63] L. Piegl, W. Tiller, The NURBS book, Springer Science & Business Media, 1996.
[64] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, Computer Methods in Applied Mechanics and Engineering 194 (39-41) (2005) 4135–4195.
[65] Y. Yamaguchi, S. Moriguchi, K. Terada, Extended B-spline-based implicit material point method, International Journal for Numerical Methods in Engineering 122 (7) (2021) 1746–1769.
[66] G. Guennebaud, B. Jacob, et al., Eigen v3, http://eigen.tuxfamily.org (2010).
[67] D. Demidov, Amgcl: An efficient, flexible, and extensible algebraic multigrid implementation, Lobachevskii Journal of Mathematics 40 (2019) 535–546.
[68] S.-H. Boo, P.-S. Lee, A dynamic condensation method using algebraic substructuring, International Journal for Numerical Methods in Engineering 109 (12) (2017) 1701–1720.