

A residual-driven local iterative corrector scheme for the multiscale finite element method

Lam H. Nguyen^{a,*}, Dominik Schillinger^a

^a*Department of Civil, Environmental, and Geo- Engineering, University of Minnesota, Twin Cities, USA*

Abstract

We describe a local iterative corrector scheme that significantly improves the accuracy of the multiscale finite element method (MsFEM). Our technique is based on the definition of a local corrector problem for each multiscale basis function that is driven by the residual of the previous multiscale solution. Each corrector problem results in a local corrector solution that improves the accuracy of the corresponding multiscale basis function at element interfaces. We cast the strategy of residual-driven correction in an iterative scheme that is straightforward to implement and, due to the locality of corrector problems, well-suited for parallel computing. We show that the iterative scheme converges to the best possible fine-mesh solution. Finally, we illustrate the effectiveness of our approach with multiscale benchmarks characterized by missing scale separation, including the microCT-based stress analysis of a vertebra with trabecular microstructure.

Keywords: Multiscale finite element method, residual-driven correction, iterative corrector scheme, parallel computing, heterogeneous materials

*Corresponding author;

Lam H. Nguyen, Department of Civil Engineering, University of Minnesota, 500 Pillsbury Drive S.E., Minneapolis, MN 55455, USA; Phone: +1 612 625 5522; Fax: +1 612 626 7750; E-mail: nguy2451@umn.edu

Contents

1	Introduction	3
2	The multiscale finite element method	4
2.1	Multiscale basis functions	5
2.2	Boundary conditions for the multiscale basis	6
3	An iterative local corrector scheme	8
3.1	A series of residual-driven local corrector problems	8
3.2	Iterative algorithm	9
3.3	Updating multiscale basis functions	10
3.4	Local imposition of global Dirichlet data	11
3.5	Convergence aspects	12
3.5.1	Optimality of residual-driven correction	12
3.5.2	Convergence to the fine-mesh solution	13
3.6	Parallel implementation	14
3.6.1	Patterning nodes with non-overlapping corrector regions	14
3.6.2	Adding coarse-scale edge and face nodes	16
4	Extension of the corrector scheme to vector problems	17
4.1	MsFEM for elasticity	17
4.2	A residual-driven corrector scheme for elasticity	18
4.2.1	Weighted corrector distribution	19
4.2.2	Corrector degrees of freedom	20
4.3	Local imposition of Dirichlet constraints at embedded surfaces	21
5	Numerical examples	23
5.1	Laplace equation with a highly oscillatory coefficient	23
5.2	Advection-diffusion with strong local solution gradients	25
5.3	L-shaped specimen with randomly distributed Young's modulus	29
5.4	MicroCT-based stress analysis of a vertebra	31
6	Summary, conclusions and outlook	35
Appendix A	Multiscale finite element discretization and implementation	37
Appendix A.1	Approximation of bubble part and multiscale basis functions	37
Appendix A.2	Implementation of the multiscale solution procedure	38
Appendix B	Identifying corrector patterns on general unstructured meshes	40

1. Introduction

Heterogeneous materials such as bone [1–3], composites [4, 5] or soils [6–8] involve interaction between multiple scales. When modeling their behavior, one has to incorporate rapidly varying material properties on the fine scales, since they can significantly affect the macro-scale response. Specific assumptions such as scale separation or periodicity can be used to simplify modeling and reduce computational cost, which has been the basis for developing a number of successful homogenization methods [5, 9–21], see also the recent reviews in [22, 23]. When these assumptions do not hold, for example for random microstructures with unclear scale separation or localization in plasticity, fracture and damage, one possibility to obtain accurate results is to fully resolve all scales [24, 25]. The computational cost, however, associated with the full resolution of contemporary multiscale problems involving heterogeneous materials, e.g. in biomedical applications, easily exceeds current computing resources, even with the rapid development in supercomputing [18, 26, 27]. Therefore, there is a demand for alternative multi-resolution methods not based on scale separation or periodicity that are significantly less expensive and achieve comparable accuracy with respect to full resolution.

One attractive class of methods in this direction is based on appropriately enriching the coarse-scale solution in the variational formulation [28–34]. One example of this type is the variational multiscale method [29, 30], which splits the variational form into a coarse-scale and a fine-scale problem. The core idea is to use the fine-scale problem to find an (inexpensive) approximation of the fine-scale solution that can then be inserted into the coarse-scale problem to represent the impact of fine-scale features on the macro-scale behavior. To reduce computational cost, the fine-scale solution is typically assumed to vanish on the boundaries of coarse-scale elements. This assumption, however, becomes a significant obstacle in situations that involve localized features that are crossing element interfaces, such as cracks or plastic bands [24, 35–37].

Another example of this type is the multiscale finite element method (MsFEM), originally introduced by HOU and WU [28] and further elaborated by EFENDIEV, HOU and collaborators [26, 38–41]. The basic idea of MsFEM is to incorporate the fine-scale information into the coarse-scale basis functions, creating a so-called multiscale basis. The solution of the multiscale system requires significantly less memory compared to a fully resolved discretization, while the computation of multiscale basis functions can be carried out efficiently in parallel. HOU et al. [28, 42] showed, however, that the accuracy of the approach can be adversely affected by the local boundary conditions that are imposed on the multiscale basis functions at the interfaces of coarse-scale elements. When multiscale basis functions are constrained at element interfaces to the standard basis functions of the conforming coarse-scale mesh, MsFEM can be interpreted as a special case of VMS, yielding similar results and suffering from the same restrictions [18]. To counteract this problem, several techniques have been proposed to improve the fine-scale accuracy at element interfaces [28, 38, 39, 43–50], many based on the idea of oversampling. The MsFEM has also been influenced by ideas from model order reduction [40, 51–53].

In this paper, we focus on MsFEM and attempt to devise a technique that enables the best possible fine-mesh accuracy, while preserving the core computational advantages of MsFEM. The core idea of our approach is based on the definition of a local corrector problem for each multiscale basis function that mitigates the error introduced by local boundary conditions. Each corrector

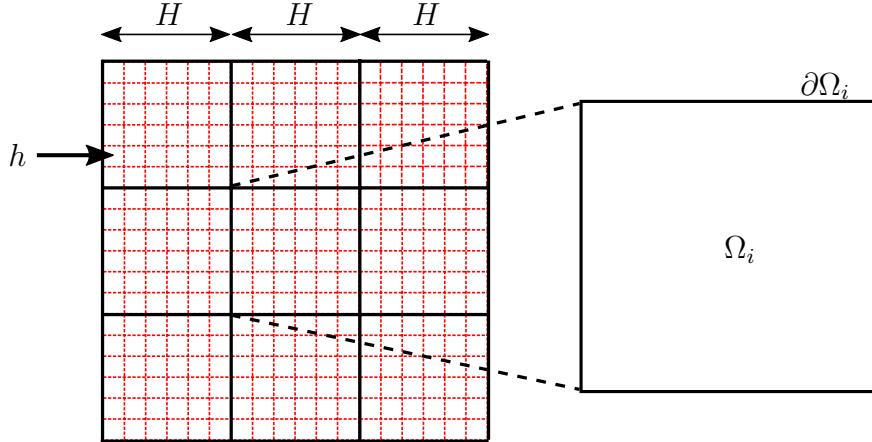


Figure 1: Discretization of the domain Ω .

problem is driven by the local residual of the previous multiscale solution [38, 40, 41]. As the associated corrector regions, we identify the local support of each multiscale basis function, which typically extend over a few coarse-scale elements. It can be expected that the corrected multiscale basis significantly improves the accuracy at coarse-scale element interfaces. When cast into an iterative scheme, the residual-driven correction of multiscale basis functions combines a number of attractive properties. First, the MsFEM solution converges to the same solution as a fully resolved discretization irrespective of the ratio between fine and coarse scales, on any small region of interest. Second, the residual of the multiscale solution is the optimal driver for correction that achieves the full accuracy of the fine-mesh resolution with a few iterations. Third, the scheme is simple to implement and can be efficiently parallelized due to the locality of the corrector problems.

Our paper is organized as follows: In Section 2, we briefly review the basic concepts and implementation of standard MsFEM. In Section 3, we introduce our corrector approach for a scalar Laplace equation and discuss convergence and parallel implementation aspects. In Section 4, we extend our corrector scheme to vector equations, focusing on elasticity, and introduce the concept of additional corrector degrees of freedom. In Section 5, we present a series of numerical examples in two and three dimensions that illustrate accuracy and computational efficiency of our approach. Finally, we draw conclusions in Section 6.

2. The multiscale finite element method

We begin with a brief review of the standard multiscale finite element method (MsFEM). To this end, we consider an elliptic equation with a highly oscillatory coefficient $a(\mathbf{x})$,

$$-\nabla \cdot (a(\mathbf{x}) \nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \quad (1)$$

$$u(\mathbf{x})|_{\partial\Omega} = g(\mathbf{x}) \quad (2)$$

The domain Ω is discretized by coarse elements, $\Omega = \cup_i \Omega_i$, with mesh size H , and by an underlying fine mesh of size h (see Fig. 1).

2.1. Multiscale basis functions

In MsFEM, we consider the following decomposition of the solution u [38, 39],

$$u(\mathbf{x}) = u_a(\mathbf{x}) + u_b(\mathbf{x}) \quad (3)$$

The first component, u_a , is $a(\mathbf{x})$ -harmonic and satisfies the homogeneous differential equation on the element domain Ω_i ,

$$-\nabla \cdot (a(\mathbf{x}) \nabla u_a(\mathbf{x})) = 0, \quad \mathbf{x} \in \Omega_i \quad (4)$$

$$u_a(\mathbf{x})|_{\partial\Omega_i} = u(\mathbf{x})|_{\partial\Omega_i} \quad (5)$$

with Dirichlet boundary conditions that correspond to the exact (unknown) solution. The second component, u_b , represents the solution of a local bubble problem on each element domain,

$$-\nabla \cdot (a(\mathbf{x}) \nabla u_b(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in \Omega_i \quad (6)$$

$$u_b(\mathbf{x})|_{\partial\Omega_i} = 0 \quad (7)$$

with homogeneous boundary conditions. It is straightforward to show that u_a and u_b are $a(\mathbf{x})$ -orthogonal, $\int_{\Omega_i} \nabla u_b \cdot a(\mathbf{x}) \nabla u_a \, d\Omega = 0$ [38]. Based on this property, the multiscale basis functions $\phi_i \in H_0^1(\Omega)$ are defined as $a(\mathbf{x})$ -harmonic functions on the element domain Ω_i ,

$$-\nabla \cdot (a(\mathbf{x}) \nabla \phi_i(\mathbf{x})) = 0, \quad \mathbf{x} \in \Omega_i \quad (8)$$

With appropriate local boundary conditions, equation (8) can be solved analytically or numerically. It is shown in [38] that multiscale basis functions achieve the same performance in terms of accuracy and convergence as a set of standard basis functions defined on the coarse-scale mesh, given they are the same at element interfaces and only $O(H)$ accuracy in the energy norm is desired. With the multiscale basis ϕ_i , $i = 1, \dots, n$, MsFEM searches the numerical solution $u_{ms}^H \in V^H$, such that

$$a(u_{ms}^H(\mathbf{x}), v(\mathbf{x})) = \langle f(\mathbf{x}), v(\mathbf{x}) \rangle, \text{ for all } v(\mathbf{x}) \in V^H \quad (9)$$

$$V^H = \text{span} \{ \phi_1(\mathbf{x}), \phi_2(\mathbf{x}), \dots, \phi_n(\mathbf{x}) \} \subset H_0^1(\Omega) \quad (10)$$

where

$$a(u(\mathbf{x}), v(\mathbf{x})) = \int_{\Omega} \nabla u(\mathbf{x}) \cdot a(\mathbf{x}) \nabla v(\mathbf{x}) \, d\Omega, \quad \langle f(\mathbf{x}), v(\mathbf{x}) \rangle = \int_{\Omega} f(\mathbf{x}) v(\mathbf{x}) \, d\Omega \quad (11)$$

We add the local bubble part u_b to the multiscale solution u_{ms}^H to obtain the complete multiscale finite element solution, \bar{u} , as

$$\bar{u}(\mathbf{x}) = u_{ms}^H(\mathbf{x}) + u_b(\mathbf{x}) \quad (12)$$

When using the finite element method to find the multiscale basis and the bubble part, the solution \bar{u} in (12) is represented by the underlying fine mesh. If V^h and u^h are denoted as the space of

piecewise basis functions on the fine mesh and the Galerkin numerical solution using this space, the accuracy of the solution \bar{u} is bounded by the fine-mesh solution u^h [38, 39],

$$\| \bar{u}(\mathbf{x}) - u(\mathbf{x}) \|_E \geq \| u^h(\mathbf{x}) - u(\mathbf{x}) \|_E \quad (13)$$

where the energy norm is defined as

$$\| u(\mathbf{x}) \|_E^2 = a(u(\mathbf{x}), u(\mathbf{x})) \quad (14)$$

Therefore, the fine mesh is required to resolve all fine-scale features of the multiscale problem, represented here by the variation of the highly oscillatory coefficient $a(\mathbf{x})$. We emphasize that in this paper we do not distinguish between the exact solution u and the fine-mesh solution u^h , since the latter is the best possible accuracy we can achieve with a full resolution discretization due to (13). To keep the paper self-contained, we briefly outline relevant aspects of finding fine-scale finite element approximations of the local problems (6) and (8) in [Appendix A](#).

2.2. Boundary conditions for the multiscale basis

To determine multiscale basis functions, the local boundary value problems (8) must be closed by appropriate local boundary conditions. Since we do not know the exact solution u in advance, we have to impose boundary conditions based on assumptions. This step plays a crucial role for the performance of MsFEM, since appropriate boundary conditions can alleviate the error at element interfaces and improve the accuracy of MsFEM [28]. There are different techniques to find appropriate local boundary conditions for the multiscale basis functions, which we briefly summarize in the following.

The simplest option is to use standard polynomial basis functions defined on the coarse mesh as the boundary conditions for the multiscale basis functions on element interfaces [26],

$$\phi_i(\mathbf{x}) = N_i(\mathbf{x}), \quad \mathbf{x} \text{ on } \partial\Omega_i \quad (15)$$

where N_i denotes a polynomial function associated with the same coefficient i on the coarse mesh that can be part of a Lagrange, Bézier or hierarchical basis. This approach, however, neglects fine-scale heterogeneous variations of the true multiscale solution on element interfaces. This can lead to large errors, in particular caused by resonance effects that typically occur when the element size H of the coarse mesh and the characteristic length scale of the fine-scale variation are of the same order [26, 28].

An alternative option that counteracts resonance effects is the oversampling method, originally proposed in [28]. The error caused by resonance concentrates at element interfaces and decays quickly as we move away from the interfaces. Based on this observation, the oversampling method uses the interior information of a larger domain to construct multiscale basis functions. Hou and collaborators [28] showed that this approach can significantly reduce resonance errors. A disadvantage of oversampling, however, is that the multiscale basis functions are not conforming across element interfaces, which may want further considerations [46, 54].

To improve the accuracy of the local boundary conditions of the multiscale basis, global information of the solution can be used in such a way that the multiscale basis from the oversampling

method can be iteratively modified to obtain a conforming basis [43, 44]. Although the iteration converges, the multiscale solution does not approach the fine-mesh solution [39]. Multigrid algorithms have been applied to improve the MsFEM solution [39, 45], where the smoothing step is performed with the full resolution system for a certain number of iterations. The information can then be used to update the multiscale basis. The multiscale solution converges to the fine-mesh solution, but at the cost that a full fine system is assembled and iteratively processed.

Remark 1: It is instructive to notice that MsFEM can be interpreted as a special case of the variational multiscale method (VMS) [29]. To illustrate this relation, we decompose the solution and weighting functions, y and w , in coarse- and fine-scale parts,

$$y = y^1 + y^2 \quad (16)$$

$$w = w^1 + w^2 \quad (17)$$

where $y^1 \in V^H \subset H^1(\Omega)$, $y^2 \in V^h \subset H^1(\Omega)$, $w^1 \in V^H \subset H_0^1(\Omega)$, and $w^2 \in V^h \subset H_0^1(\Omega)$. Substituting (16) and (17) into the Galerkin projection, see for instance (9), we have

$$a(w^1 + w^2, y^1 + y^2) = \langle f, w^1 + w^2 \rangle \quad (18)$$

We can find y^1 and y^2 by solving the following coupled system,

$$a(w^1, y^1) + a(w^1, y^2) = \langle w^1, f \rangle \quad (19)$$

$$a(w^2, y^1) + a(w^2, y^2) = \langle w^2, f \rangle \quad (20)$$

Due to the coupling terms in (19) and (20), we need to solve two equations together or resort to a staggered approach. However, if we choose the function spaces V^h and V^H to be $a(\mathbf{x})$ -orthogonal, we automatically eliminate the coupling terms in (19) and (20). Therefore, the two systems become independent of each other,

$$a(w^1, y^1) = \langle f, w^1 \rangle \quad (21)$$

$$a(w^2, y^2) = \langle f, w^2 \rangle \quad (22)$$

and can therefore be solved independently. Based on the $a(\mathbf{x})$ -orthogonal property of its multiscale basis functions, the MsFEM can therefore be interpreted as a special VMS procedure that leverages $a(\mathbf{x})$ -orthogonality to independently solve the coarse-scale and fine-scale variational problems in the sense of (21) and (22). In MsFEM, we apply local boundary conditions according to (15), which in the VMS framework translates to

$$y^2 = 0 \text{ on } \partial\Omega_i \quad (23)$$

at coarse-scale element interfaces. We note that the same constraint (23) is also imposed in many VMS methods to enable the local solution of (22).

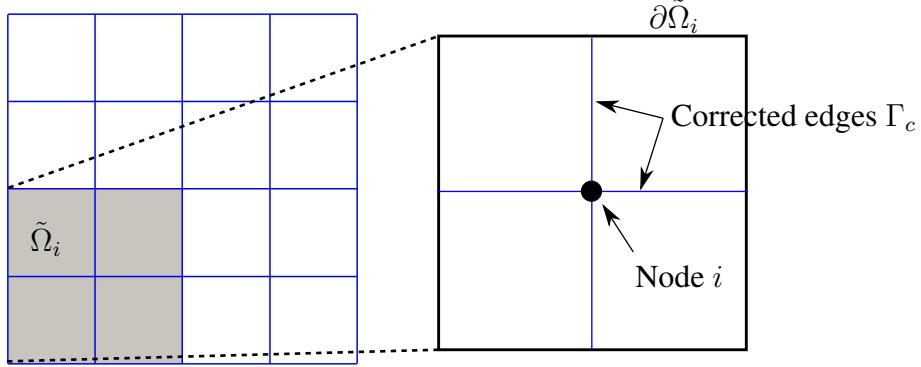


Figure 2: Local corrector region that corresponds to the support $\tilde{\Omega}_i$ of the multiscale basis function ϕ_i .

3. An iterative local corrector scheme

In the following, we establish an iterative local corrector scheme for MsFEM that removes the error in the multiscale basis functions due to the assumptions involved in choosing local boundary conditions, restoring the full fine-mesh accuracy at coarse-scale element interfaces. In this section, we focus on scalar multiscale problems first, using (1) and (2) as a model problem, before extending the corrector scheme to vector problems in the next section. In the scope of this paper, we illustrate and derive the corrector scheme for quadrilateral and hexahedral elements. We note, however, that it can be directly generalized for other types of elements, such as triangles and tetrahedra.

3.1. A series of residual-driven local corrector problems

Let us consider the support $\tilde{\Omega}_i$ of the multiscale basis function ϕ_i with node i and boundary $\partial\tilde{\Omega}_i$, illustrated as the gray-shaded region in Fig. 2. We write a local variant of the elliptic model problem (1) and (2) with the multiscale solution \bar{u} defined in (12) as

$$-\nabla \cdot (a(\mathbf{x}) \nabla \bar{u}(\mathbf{x})) = \bar{f}(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Omega} \quad (24)$$

$$\bar{u}(\mathbf{x}) = \bar{u}(\mathbf{x})|_{\partial\tilde{\Omega}}, \quad \mathbf{x} \in \partial\tilde{\Omega} \quad (25)$$

Note that the right-hand side of (24) does not represent the exact source term f , since the multiscale approximation of the true solution at the local boundaries in (25) in general causes an error. We now assume that we know the exact solution, so that we can write the local variant of (1) and (2) on the support $\tilde{\Omega}_i$ of the multiscale basis function as

$$-\nabla \cdot (a(\mathbf{x}) \nabla u(\mathbf{x})) = f(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Omega}_i \quad (26)$$

$$u(\mathbf{x}) = u(\mathbf{x})|_{\partial\tilde{\Omega}_i}, \quad \mathbf{x} \in \partial\tilde{\Omega}_i \quad (27)$$

We now define u_c as the local corrector for \bar{u} , such that

$$-\nabla \cdot (a(\mathbf{x}) \nabla (\bar{u} + u_c)) = -\nabla \cdot (a(\mathbf{x}) \nabla u) = f(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Omega}_i \quad (28)$$

where f is the source term that corresponds to the exact solution u of the problem. Since the multiscale basis function we would like to correct must be zero on the boundary of its support

region $\tilde{\Omega}_i$, the difference between the multiscale solution \bar{u} and the exact solution u at the local boundary cannot be corrected by that basis function. We therefore assume that

$$|\bar{u}(\mathbf{x}) - u(\mathbf{x})| \approx 0, \quad \mathbf{x} \in \partial\tilde{\Omega}_i \quad (29)$$

such that the corrector solution u_c is zero at the boundary $\partial\tilde{\Omega}_i$. Subtracting (24) from (28) and using (29), we obtain the following local problem that determines the corrector,

$$-\nabla \cdot (a(\mathbf{x}) \nabla u_c(\mathbf{x})) = r(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Omega}_i \quad (30)$$

$$u_c(\mathbf{x}) = 0, \quad \mathbf{x} \in \partial\tilde{\Omega}_i \quad (31)$$

We observe that the corrector solution is directly driven by the local residual defined as

$$r(\mathbf{x}) = f(\mathbf{x}) - \bar{f}(\mathbf{x}) \quad (32)$$

The corrector solution of each local corrector problem can be found via a finite element discretization in a straightforward way. To this end, we cast (30) and (31) into the following variational form

$$\int_{\tilde{\Omega}_i} \nabla v^h \cdot a(\mathbf{x}) \nabla u_c^{i,h} \, d\Omega = \int_{\tilde{\Omega}_i} v^h f \, d\Omega - \int_{\tilde{\Omega}_i} \nabla v^h \cdot a(\mathbf{x}) \nabla \bar{u}^h \, d\Omega \quad (33)$$

where v^h and $u_c^{i,h}$ are finite element approximations of the test functions and the corrector solution for the coarse-scale node i . They are represented on the corresponding local corrector region $\tilde{\Omega}_i$ by the fine mesh available in each coarse-scale element in the sense of Fig. 1. The finite element approximation of the complete multiscale solution \bar{u}^h to be corrected is known from the previous MsFEM computation. We refer readers interested in more details to [Appendix A](#).

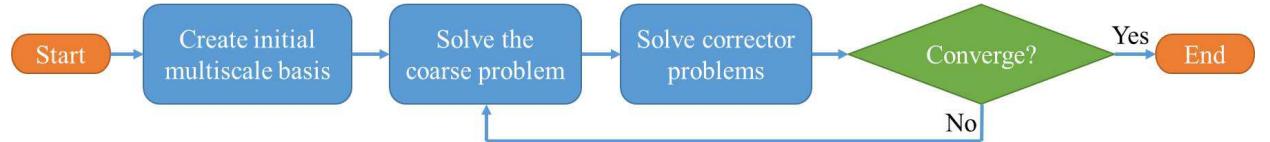


Figure 3: Iterative corrector scheme

3.2. Iterative algorithm

Equations (30), (31) and (32) and their finite element treatment can now be used to set up a straightforward corrector scheme. We first define a local corrector region for each multiscale basis function ϕ_i in the domain that we want to correct. In this work, we choose the local support $\tilde{\Omega}_i$ of each basis function. We then compute a series of local corrector problems to iteratively improve the accuracy of multiscale basis functions until the MsFEM solution \bar{u} matches the full fine-mesh solution to the desired level of accuracy.

The resulting iterative corrector algorithm can be summarized as follows (see Fig. 3 for a graphical illustration):

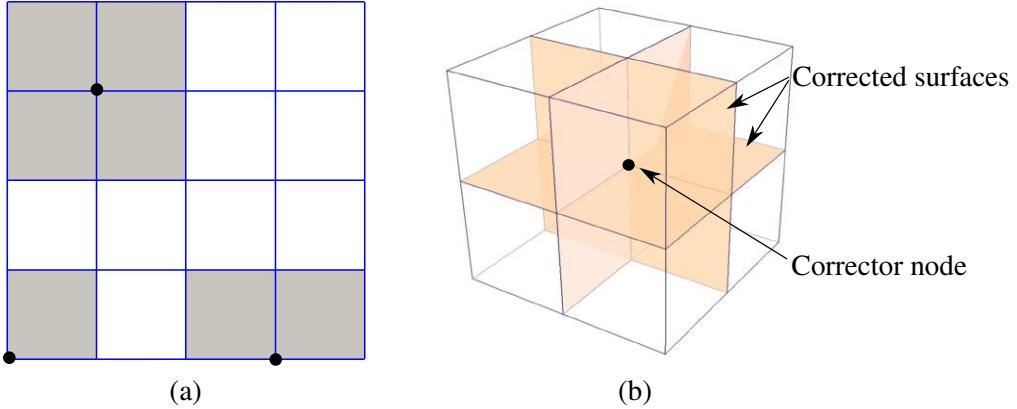


Figure 4: Illustrations of different corrector regions in (a) two and (b) three dimensions.

1. Standard MsFEM

(a) Compute the initial basis functions with some assumed local boundary conditions

2. Corrector scheme

(a) Obtain a solution on the coarse mesh with the current multiscale basis

(b) Compute the residual r in local corrector regions

(c) Solve local corrector problems where $\|r\| > \text{tolerance}$ (user-defined)

(d) Update the current multiscale basis functions and return to step (a)

Depending on the location of a corrector node, we can differentiate between different corrector regions, which are illustrated in Fig. 4a for a two-dimensional mesh. In addition to regular corrector regions in the interior of the mesh, we also obtain regions on the edge with two elements and corner regions with only one element. We note that edge and corner regions require an adaptation of the corrector scheme to incorporate given boundary data, which we discuss below. The notion of a corrector region directly transfers to three dimensions. A typical corrector region from the interior of a 3D hexahedral mesh is shown in Fig. 4b, where the faces to be corrected are highlighted.

3.3. Updating multiscale basis functions

It is important to note that the residual (32) is nonzero only on element interfaces Γ_c (see Fig. 2). Within the element domains, the residual must be zero, since \bar{f} is computed from \bar{u} , which includes the bubble solution u_b . This means that

$$-\nabla \cdot (a(\mathbf{x}) \nabla u_c(\mathbf{x})) = 0, \quad \mathbf{x} \in \Omega_i \quad (34)$$

In addition, the corrector solution u_c is also $a(\mathbf{x})$ -orthogonal with respect to the bubble solution u_b ,

$$a_{\Omega_i}(u_c(\mathbf{x}), u_b(\mathbf{x})) = \int_{\Omega_i} \nabla u_c(\mathbf{x}) \cdot a(\mathbf{x}) \nabla u_b(\mathbf{x}) \, d\Omega = 0, \quad \mathbf{x} \in \Omega_i \quad (35)$$

Due to (35), the current multiscale basis function ϕ_i can be updated by simply adding u_c to its current value. The new basis function will be used as part of the updated multiscale basis in the next iteration.

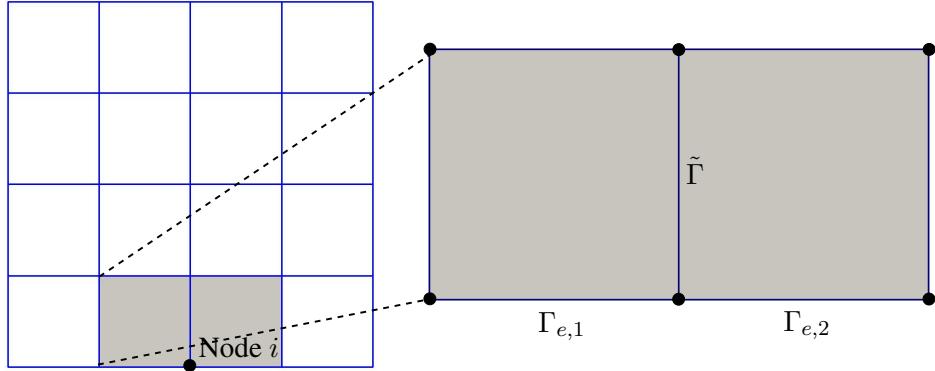


Figure 5: Corrector problem involving the global Dirichlet boundary.

The updating of basis functions is carried out in the following way. Let ϕ_i , c_i , and $u_{c,i}$ denote the multiscale basis function at node i , its corresponding solution coefficient in the current MsFEM solution, and its corrector solution, respectively. The new multiscale basis function at node i , ϕ_i^{new} , should satisfy

$$\phi_i^{\text{new}} c_i = \phi_i c_i + u_{c,i} \quad (36)$$

Therefore, the new basis function can be computed as

$$\phi_i^{\text{new}} = \phi_i + \frac{u_{c,i}}{c_i} \quad (37)$$

It is straightforward to see that (37) is not well-defined in the case of zero-valued coefficients in the multiscale solution. To avoid this issue, we propose to blend the coefficient into the new basis function, using the new contribution of node i as the new basis function. The new basis function is thus computed as

$$\phi_i^{\text{new}} = \phi_i c_i + u_{c,i} \quad (38)$$

Since we need to update the solution \bar{u} when we construct subsequent corrector problems for other basis functions, we immediately set c_i to one after the update (38).

Remark 2: The initial set of multiscale basis functions that are computed by imposing the standard coarse-scale basis functions as boundary conditions satisfy a partition of unity. After updating by corrector solutions, the new multiscale basis does in general not satisfy a partition of unity. We can, however, restore this property for the multiscale basis, if needed. We emphasize that for convergence, it is only important that the basis of the fine mesh satisfies a partition of unity.

3.4. Local imposition of global Dirichlet data

Dirichlet boundary data on the coarse mesh can only be imposed via coarse-scale nodes. The solution at the boundary segments between fixed coarse-scale nodal values is obtained by interpolating with multiscale basis functions determined with local boundary conditions in (8).

Therefore, the multiscale solution \bar{u} will only converge to the fine-mesh solution if we impose the correct global Dirichlet boundary data in corrector problems that involve the Dirichlet boundary of the global domain.

To illustrate this, we consider the corrector region shown in Fig. 5, where $\tilde{\Gamma}_D = \Gamma_{e,1} \cup \Gamma_{e,2}$ is a part of the global boundary $\partial\Omega$ with Dirichlet boundary conditions. In the corrector problem corresponding to node i , we impose the following boundary condition

$$u_c(\mathbf{x}) = g(\mathbf{x}) - \bar{u}(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Gamma}_D \quad (39)$$

We call (39) global-local condition, since global boundary information is used for imposing constraints in local corrector problems. When we compute the multiscale solution using the new multiscale basis, the coefficients of all multiscale basis functions affected by the global-local condition (39) are set to one. Due to (38), the value of each boundary basis functions at its node is equal to the value of the global Dirichlet boundary data at that specific point.

Remark 3: In the case of a homogeneous Dirichlet data, that is $g = 0$, we still set the coefficients of the new basis functions affected by the global-local condition to one. This naturally accommodates situations where a multiscale basis function has support on boundary sections with both zero and non-zero Dirichlet data.

Remark 4: Since our corrector scheme based on (30) is driven by the residual, it automatically improves the solution at the Neumann boundary. We therefore do not need any special treatment for global Neumann boundary data.

3.5. Convergence aspects

In this subsection, we show that the residual is the optimal choice for driving corrector solutions and that the sequence of solutions of the iterative scheme converges to the fine-mesh solution. In what follows, we always consider only one local corrector region $\tilde{\Omega}_i$ that corresponds to a single multiscale basis function at an arbitrary coarse-scale node i . This point of view does not limit the generality of our analysis, as we could apply the same considerations consecutively to all multiscale basis functions and associated corrector regions.

3.5.1. Optimality of residual-driven correction

Let us assume that we are in the q -th iteration of the corrector scheme, updating the current multiscale basis function ϕ_i to the new multiscale basis function ϕ_i^{new} , using the corrector solution $u_{c,i}$ on the corrector region $\tilde{\Omega}_i$. Let $V_{q+1}^H = V_q^H - \text{span}\{\phi_i\} + \text{span}\{\phi_i^{new}\}$ be the multiscale approximation space at the $(q+1)$ -th iteration, \bar{u}_{q+1} be the corresponding solution of the Galerkin projection on V_{q+1}^H , and (again) u^h be the fine-mesh solution. According to Céa's lemma, we know

$$\|u^h - \bar{u}_{q+1}\|_E^2 = \inf_{v \in V_{q+1}^H} \|u^h - v\|_E^2 \quad (40)$$

with the energy norm defined in (14). Following the definition of ϕ_i^{new} in (38), one possible choice for v is $v = \bar{u}_q + u_{c,i}$. Let us further assume for the moment that the corrector solution $u_{c,i} \in H^1(\tilde{\Omega}_i)$

can be any function in $\tilde{\Omega}_i$. Substituting our choice of v into (40), we find

$$\|u^h - \bar{u}_{q+1}\|_E^2 \leq \|u^h - \bar{u}_q - u_{c,i}\|_E^2 = \|u^h - \bar{u}_q\|_E^2 - (2a_{\tilde{\Omega}_i}(u^h - \bar{u}_q, u_{c,i}) - a_{\tilde{\Omega}_i}(u_{c,i}, u_{c,i})) \quad (41)$$

in which $a_{\tilde{\Omega}_i}(u, v) = \int_{\tilde{\Omega}_i} \nabla u \cdot a(\mathbf{x}) \nabla v \, d\Omega$. The optimal choice for $u_{c,i}$ is the function that reduces the error at the q -th iteration the most. According to the final expression in (41), this is the function that maximizes the following functional

$$I(u_{c,i}) = (2a_{\tilde{\Omega}_i}(u^h - \bar{u}_q, u_{c,i}) - a_{\tilde{\Omega}_i}(u_{c,i}, u_{c,i})) \quad (42)$$

We note that we still impose the constraint $u_{c,i}|_{\partial\tilde{\Omega}_i} = 0$. To find the stationary solution, we set the functional derivative of (42) to zero, leading to the following variational form,

$$a_{\tilde{\Omega}_i}(u_{c,i}, \delta u_{c,i}) = a_{\tilde{\Omega}_i}(u^h - \bar{u}_q, \delta u_{c,i}), \quad \forall \delta u_{c,i} \in H_0^1(\tilde{\Omega}_i) \quad (43)$$

where $\delta u_{c,i}$ is the variation of $u_{c,i}$. Since the fine-mesh solution u^h satisfies

$$a_{\tilde{\Omega}_i}(u^h, \delta u_{c,i}) = \langle f, \delta u_{c,i} \rangle_{\tilde{\Omega}_i}, \quad \forall \delta u_{c,i} \in H_0^1(\tilde{\Omega}_i) \quad (44)$$

we can rewrite the right-hand side of (43) in the following way

$$r = \langle f, \delta u_{c,i} \rangle_{\tilde{\Omega}_i} - a_{\tilde{\Omega}_i}(\bar{u}_q, \delta u_{c,i}) \quad (45)$$

which can be easily identified as the residual in the corrector region $\tilde{\Omega}_i$. Therefore, the solution of the residual-driven corrector problem that we defined in (30) to (32) is indeed the maximizer of (42) under the given constraint $u_{c,i}|_{\partial\tilde{\Omega}_i} = 0$, thus being the optimal corrector solution. A similar analysis leading to this optimal choice was made for generalized MsFEM, where additional reduced order basis functions can be found, driven by the residual [41, 55].

Remark 5: Due to the constraint $u_{c,i}|_{\partial\tilde{\Omega}_i} = 0$, the corrector solution $u_{c,i}$ obtained from (43) will not be able to recover the fine-mesh solution in one step. In turn, if we knew the fine-mesh solution on the boundary of the corrector region $\tilde{\Omega}_i$, the iterative scheme would converge to the fine-mesh solution with a single corrector iteration. If we take the complete domain of the boundary value problem as our corrector region, all boundary values are given as data, so that the corrector solution $u_{c,i}$ obtained from (43) will be able to recover the fine-mesh solution in one single step.

3.5.2. Convergence to the fine-mesh solution

Having obtained $u_{c,i}$ from (43), we can write Céa's lemma (41) as

$$\|u^h - \bar{u}_{q+1}\|_E^2 \leq \|u^h - \bar{u}_q\|_E^2 - \|u_{c,i}\|_E^2 \quad (46)$$

If we discretize (43) by the fine mesh such that $\{u_{c,i}, \delta u_{c,i}\} \in V_{\tilde{\Omega}_i}^h \subset H_0^1(\tilde{\Omega}_i)$, we can solve for $u_{c,i}$ as the Galerkin projection of $u^h - \bar{u}_q$ on the space of piecewise basis functions in $V_{\tilde{\Omega}_i}^h$. There exists a function $u'_c \in H^1(\Omega)$ in the global domain, which satisfies

$$u_{c,i} + u'_c = u^h - \bar{u}_q \quad (47)$$

Combining (43) and (47), we also know that

$$a_{\tilde{\Omega}_i}(u_{c,i}, u'_c) = 0 \quad (48)$$

due to Galerkin orthogonality. Using (47), (48) and Pythagoras' theorem, we have

$$\|u^h - \bar{u}_q\|_E^2 = \|u_{c,i} + u'_c\|_E^2 = \|u_{c,i}\|_E^2 + 2a_{\tilde{\Omega}_i}(u_{c,i}, u'_c) + \|u'_c\|_E^2 = \|u_{c,i}\|_E^2 + \|u'_c\|_E^2 \quad (49)$$

We observe from (49), that the corrector solution $u_{c,i}$ mitigates part of the current error, where the improvement can lie anywhere between the complete error ($\|u'_c\|_E^2 = 0$) and an arbitrarily small portion of it ($\|u'_c\|_E^2 \rightarrow \|u^h - \bar{u}_q\|_E^2$). Using a constant $\alpha \in \mathbb{R}$, we can therefore re-express (49) as

$$\|u_{c,i}\|_E^2 = \alpha \|u^h - \bar{u}_q\|_E^2, \quad 0 < \alpha < 1 \quad (50)$$

Substituting (50) into (46), we find

$$\|u^h - \bar{u}_{q+1}\|_E^2 \leq (1 - \alpha) \|u^h - \bar{u}_q\|_E^2 \quad (51)$$

Since $1 - \alpha < 1$, we conclude that the sequence of solutions produced by our corrector scheme must converge to the fine-mesh solution u^h . In addition, the more local corrector problems are solved, the better is the solution in the next iteration step. We note that in our analysis, we have neglected the solution of the coarse-scale problem with the new multiscale basis. We emphasize that without this coarse-scale solve (only using the corrector solutions in the local corrector regions), the convergence of the multiscale solution to the fine-mesh solution will be slower.

3.6. Parallel implementation

Each corrector problem is driven by the current residual and therefore relies on the computation of the source term \bar{f} of the current multiscale solution \bar{u} . Updating a multiscale basis function to its new form ϕ_i^{new} , however, leads at the same time to a change of \bar{u} over the local support of that multiscale basis function. In a parallel implementation, corrector problems defined on overlapping corrector regions cannot be processed at the same time without creating race conditions. Therefore, a trivial parallel implementation of the corrector procedure is not possible.

3.6.1. Patterning nodes with non-overlapping corrector regions

The first idea to circumvent this issue is to group multiscale basis functions and associated nodes in such a way that the corrector regions of each group do not overlap and multiscale basis functions of each group can be corrected in parallel. This idea gives rise to so-called corrector patterns that are formed by the nodes of each group of corrector problems. Figure 6 illustrates four possible patterns for a two-dimensional mesh with 4×4 elements. This results in the following parallel corrector procedure: work on all nodes in one pattern in parallel, then move to the next pattern. We work on all four patterns to improve the accuracy of all basis functions. This guarantees that no overlapping corrector regions are being considered at the same time.

This principle directly extends to meshes in 3D. Figure 7 shows a possible corrector pattern for a structured hexahedral mesh. Using this pattern, local correction for all nodes of the same type (I, II, III, or IV) is performed in parallel. The procedure is repeated in parallel for the nodes of each group. For general unstructured meshes, a simple algorithm to establish similar patterns is given in Appendix B. All numerical examples in this paper are computed in parallel based on this idea.

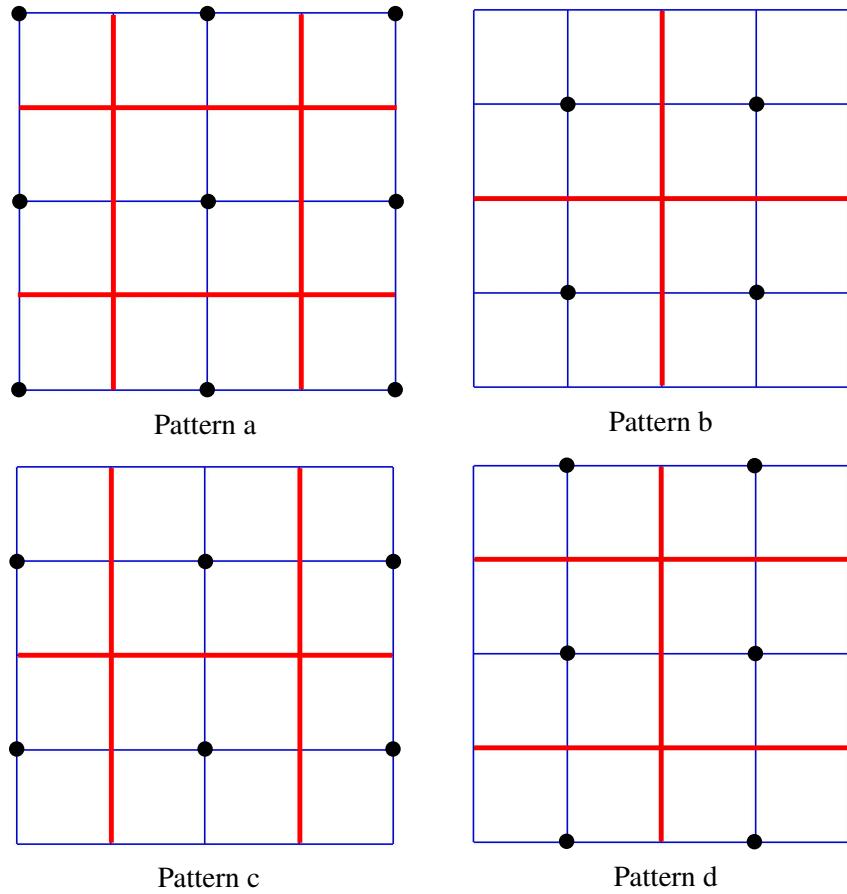


Figure 6: Four patterns in 2D (the red lines show the local boundaries of nonoverlapping corrector regions).

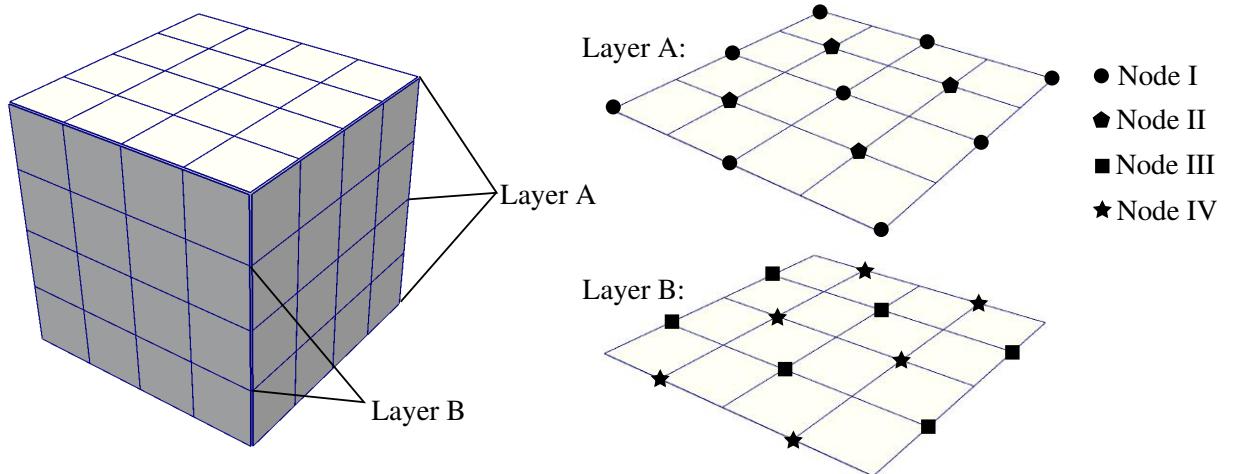


Figure 7: One possible corrector pattern in a 3D example.

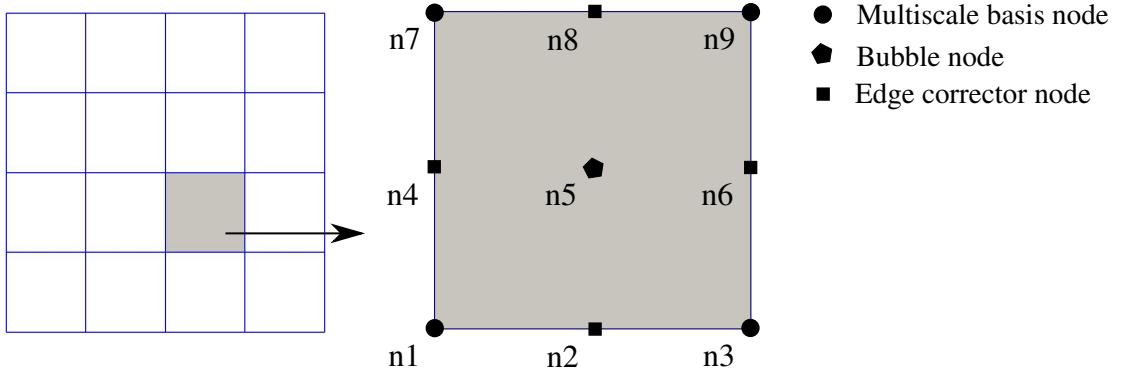


Figure 8: A coarse quadrilateral element with edge corrector nodes.

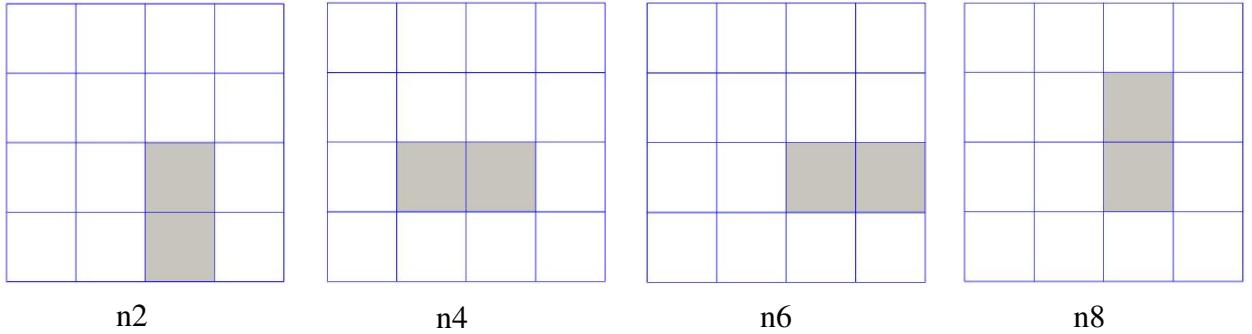


Figure 9: Corresponding corrector regions for the four edges nodes.

3.6.2. Adding coarse-scale edge and face nodes

An alternative strategy to enable simple parallelization is to add nodes on each edge to be corrected [38], illustrated in Fig. 8 for a quadrilateral element. For computing the multiscale basis functions of the additional nodes, standard quadratic polynomial basis functions can be used as local boundary conditions. Figure 9 shows the corresponding local corrector regions of these nodes. We emphasize that correcting the multiscale basis functions of the edge nodes is sufficient and the multiscale basis functions of the vertex nodes do not need to be touched. Since each edge node deals with the correction of its edge only, the associated corrector problems can be performed at once and in parallel. In addition, the reduced support of the edge modes leads to a smaller size of the corrector regions compared to the vertex nodes.

We note that this strategy directly extends to meshes in three dimensions. Individual nodes can be added to the edges and faces to be corrected, with the same beneficial effect on the properties of locality and parallelizability of associated corrector problems. Figure 10 illustrates the additional nodes for a hexahedral element. In the scope of this paper, we did not thoroughly test the opportunities of this strategy with respect to parallel efficiency, which will be the subject of future work.

Remark 6: In case a partition of unity is required for some reason, one can add coarse-scale edge or face nodes and impose local boundary conditions for multiscale basis functions in the sense

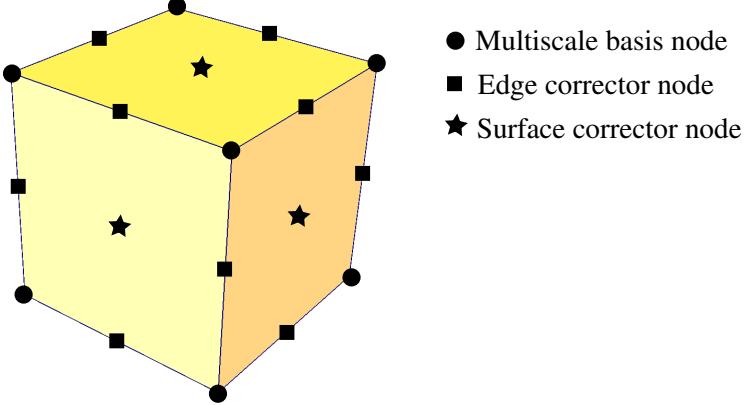


Figure 10: A hexahedral element with additional corrector nodes on edges and faces for localized corrector problems.

of hierarchical basis functions. The basis functions associated with the vertex nodes stay linear functions, guaranteeing a partition of unity, while the edge nodes absorb all correction terms.

4. Extension of the corrector scheme to vector problems

In the next step, we briefly illustrate the MsFEM formulation of vector problems for the example of linear elasticity and discuss how the corrector scheme can be extended to vector problems. In particular, we address ways to distribute the components of the corrector solution on several multiscale basis functions that occur in the MsFEM discretization of vector problems.

4.1. MsFEM for elasticity

In this work, we follow the displacement-based MsFEM formulation for elasticity derived in [56]. To this end, we first consider a two-dimensional problem with the solution vector $\mathbf{u} = (u_x \ u_y)^T$, where u_x and u_y are functions of displacements in x - and y -directions, respectively. In MsFEM, due to the Poisson effect in elasticity (Poisson's ratio $\nu \neq 0$), the multiscale contributions of each basis function of one displacement direction in the other displacement directions need to be explicitly taken into account. Therefore, the coarse-scale interpolation of the displacement vector is

$$\mathbf{u}(\mathbf{x}) = \sum_{i=1}^m \begin{pmatrix} \phi_{ixx} \\ \phi_{ixy} \end{pmatrix} u_{ix} + \sum_{i=1}^m \begin{pmatrix} \phi_{ixy} \\ \phi_{iyy} \end{pmatrix} u_{iy} = \sum_{i=1}^m (\phi_{ix} u_{ix} + \phi_{iy} u_{iy}) \quad (52)$$

At node i , the multiscale basis functions ϕ_{ixx} and ϕ_{ixy} , for instance, represent the multiscale displacement in x -direction and the associated multiscale Poisson effect in y -direction, respectively.

We now consider again a multiscale discretization in two dimensions as shown in Fig. 1. To define multiscale basis functions, we go through a similar procedure as in Section 2, but now using the vector equation of linear elasticity,

$$-\nabla \cdot (\mathbb{D} : \nabla_s \mathbf{u}(\mathbf{x})) = \mathbf{f}(\mathbf{x}) \quad (53)$$

where \mathbb{D} is an elasticity matrix (e.g., for plane stress) and ∇_s is the symmetric gradient operator. We first decompose the solution in analogy to (3) as

$$\mathbf{u}(\mathbf{x}) = \mathbf{u}_a(\mathbf{x}) + \mathbf{u}_b(\mathbf{x}) \quad (54)$$

In analogy to (4) and (6), the harmonic part \mathbf{u}_a is the solution of the local boundary value problem:

$$-\nabla \cdot (\mathbb{D} : \nabla_s \mathbf{u}_a(\mathbf{x})) = \mathbf{0}, \quad \mathbf{x} \in \Omega_i \quad (55)$$

$$\mathbf{u}_a(\mathbf{x})|_{\partial\Omega_i} = \mathbf{u}(\mathbf{x})|_{\partial\Omega_i} \quad (56)$$

and the bubble part \mathbf{u}_b is the solution of the local boundary value problem:

$$-\nabla \cdot (\mathbb{D} : \nabla_s \mathbf{u}_b(\mathbf{x})) = \mathbf{f}(\mathbf{x}), \quad \mathbf{x} \in \Omega_i \quad (57)$$

$$\mathbf{u}_b(\mathbf{x})|_{\partial\Omega_i} = \mathbf{0} \quad (58)$$

The boundary value problems (55) through (58) are defined on each coarse-scale element domain Ω_i . Substituting the approximation of \mathbf{u}_a based on (52) into (55), we obtain

$$-\sum_{i=1}^m (\nabla \cdot (\mathbb{D} : \nabla_s \phi_{ix}) u_{a,ix}) - \sum_{i=1}^m (\nabla \cdot (\mathbb{D} : \nabla_s \phi_{iy}) u_{a,iy}) = \mathbf{0}, \quad \mathbf{x} \in \Omega_i \quad (59)$$

The vectors ϕ_{ix} and ϕ_{iy} contain all multiscale basis functions that depend on the corresponding nodal displacements in x - and y -directions at node i , respectively. Using the fine-scale mesh, we can discretize and solve for the multiscale basis functions in x - and y -directions at node i from the following element-wise boundary values problems

$$-\nabla \cdot (\mathbb{D} : \nabla_s \phi_{ix}) = \mathbf{0} \quad \text{and} \quad -\nabla \cdot (\mathbb{D} : \nabla_s \phi_{iy}) = \mathbf{0}, \quad \mathbf{x} \in \Omega_i \quad (60)$$

To close (60), we apply simple boundary conditions, which are illustrated in Fig. 11. For the multiscale basis function ϕ_{ix} at node i , we fix all solution components in y -direction and impose the standard basis function on the displacement solution in x -direction at the element boundary. We can apply the same procedure for ϕ_{iy} . Having obtained the complete set of multiscale basis functions, we can find the multiscale solution as discussed in Section 2. The bubble part of the solution, \mathbf{u}_b , can be approximated separately on each coarse element from (58). We note that this procedure directly extends to three dimensions and can be readily adapted to other vector problems.

4.2. A residual-driven corrector scheme for elasticity

We now adapt the iterative corrector scheme introduced in Section 3 to linear elasticity. To this end, we define local corrector problems over the support $\tilde{\Omega}_i$ of the group of multiscale basis functions at node i . This results in

$$-\nabla \cdot (\mathbb{D} : \nabla_s \mathbf{u}_c(\mathbf{x})) = \mathbf{r}(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Omega}_i \quad (61)$$

$$\mathbf{u}_c(\mathbf{x})|_{\partial\tilde{\Omega}_i} = \mathbf{0}, \quad \mathbf{x} \in \partial\tilde{\Omega}_i \quad (62)$$

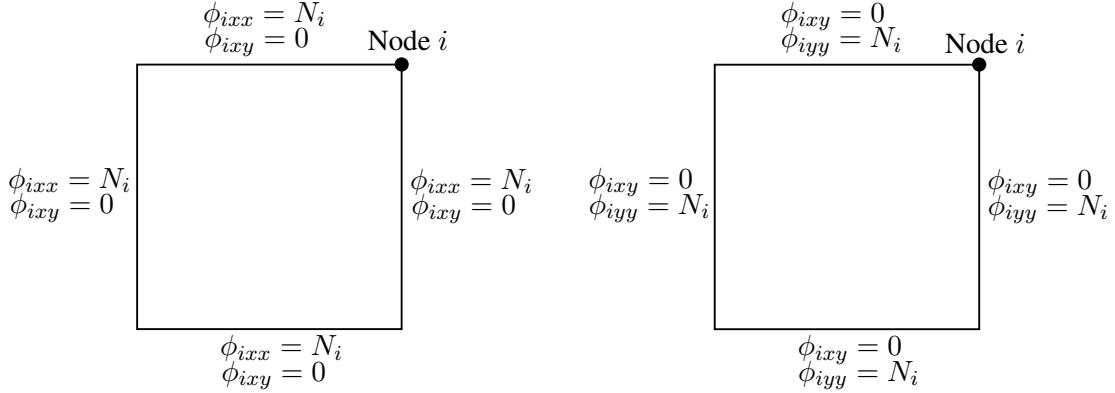


Figure 11: Local boundary conditions of the multiscale basis functions at node i in a coarse element.

from which corrector solutions \mathbf{u}_c can be locally determined. The corrector problem of (61) and (62) is again driven by the local residual force vector,

$$\mathbf{r}(\mathbf{x}) = \mathbf{f}(\mathbf{x}) - \bar{\mathbf{f}}(\mathbf{x}) \quad (63)$$

that is defined as the difference between the external force vector, $\mathbf{f}(\mathbf{x})$, and the current internal force vector computed as

$$\bar{\mathbf{f}}(\mathbf{x}) = -\nabla \cdot (\mathbb{D} : \nabla_s \bar{\mathbf{u}}(\mathbf{x})) \quad (64)$$

where $\bar{\mathbf{u}}$ is the current multiscale solution. Following Section 3.4, we emphasize again that for corrector problems with support at the global Dirichlet boundary, global Dirichlet data \mathbf{g} needs to be imposed through the following global-local condition

$$\mathbf{u}_c(\mathbf{x}) = \mathbf{g}(\mathbf{x}) - \bar{\mathbf{u}}(\mathbf{x}), \quad \mathbf{x} \in \tilde{\Gamma}_D \quad (65)$$

where $\tilde{\Gamma}_D$ denotes the boundary of $\tilde{\Omega}_i$ that coincides with the global Dirichlet boundary.

In a vector problem, the main source of difficulty is the fact that there are generally more multiscale basis functions than corrector solutions. Let us again focus on the example of 2D elasticity. There are two multiscale basis vector functions at each node, while there is only one vector function arising from the solution of the corrector problem. This raises the question of how to distribute the corrector solution. In the following, we discuss two possible strategies.

4.2.1. Weighted corrector distribution

The first strategy is to find a suitable way of splitting the corrector solution into several parts that correspond to the available multiscale basis functions to be corrected. Before motivating a split, it is helpful to recall two conditions that the updated multiscale basis functions need to satisfy:

- After each correction step, the new solution component of node i , represented by the sum of the updated multiscale basis functions times the new coefficients of node i , needs to be equal

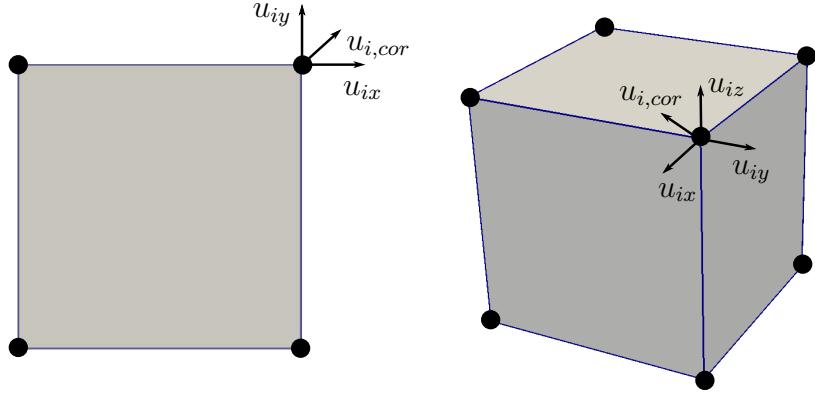


Figure 12: One additional corrector degrees of freedom at vertex nodes in the 2D and 3D case.

to the old solution component of node i plus the corresponding corrector solution, associated with node i . This requirement can be summarized as

$$\phi_{ix}^{\text{new}} u_{ix}^{\text{new}} + \phi_{iy}^{\text{new}} u_{iy}^{\text{new}} = \mathbf{u}_{c,i} + \phi_{ix} u_{ix} + \phi_{iy} u_{iy} \quad (66)$$

where $\mathbf{u}_{c,i}$ is the corrector solution associated with node i , ϕ_{ix}^{new} and ϕ_{iy}^{new} are the two updated multiscale basis functions in x - and y -directions, and u_{ix}^{new} and u_{iy}^{new} are the corresponding new nodal coefficients.

- The new multiscale basis functions need to be orthogonal to the bubble part of the solution.

We assume that the solution component in the current multiscale solution which has a larger displacement should receive a larger portion of the corrector solution. To this end, we propose the following splitting rule. First, we compute a normalized weight for each direction,

$$w_x = \frac{|u_x(\mathbf{x}_i)|}{|u_x(\mathbf{x}_i)| + |u_y(\mathbf{x}_i)|} \quad \text{and} \quad w_y = \frac{|u_y(\mathbf{x}_i)|}{|u_x(\mathbf{x}_i)| + |u_y(\mathbf{x}_i)|} \quad (67)$$

such that the weights sum up to one. We can then use these weights to determine a distribution of the corrector solution to the available multiscale basis functions as follows

$$\phi_{ix}^{\text{new}} u_{ix}^{\text{new}} = \mathbf{u}_{c,i} w_x + \phi_{ix} u_{ix} \quad (68)$$

$$\phi_{iy}^{\text{new}} u_{iy}^{\text{new}} = \mathbf{u}_{c,i} w_y + \phi_{iy} u_{iy} \quad (69)$$

Following the discussion in Section 3.3, we can then use the results of (68) and (69) as the new multiscale basis functions, setting the new coefficients $u_{ix}^{\text{new}} = u_{iy}^{\text{new}} = 1$. It is straightforward to verify that the new multiscale basis functions satisfy the above two conditions, since $w_x + w_y = 1$ and $\mathbf{u}_{c,i}$ is orthogonal to the bubble part of the solution.

4.2.2. Corrector degrees of freedom

The core idea of the second strategy is to introduce additional corrector degrees of freedom at the nodes of the coarse mesh. As illustrated in Fig. 12, there is only one additional corrector degree

of freedom per node, irrespective of the dimensionality of the problem. This new degree of freedom, $\hat{u}_{\text{cor},i}$, is assigned to the corrector solution, which is updated during corrector iterations, while the initial multiscale basis functions obtained in Section 4.1 stay unchanged during corrector iterations. Due to the additional corrector degree of freedom, we establish the final corrector solution as an additional multiscale basis function, which consists of components in x - and y -directions.

This strategy has a number of implications. On the one hand, the number of degrees of freedom in the MsFEM analysis with multiscale basis functions is larger. With the corrector solution being an individual basis function, linear dependencies become possible, for instance if the corrector solution is zero, and algorithms need to be put in place to suitably catch and mitigate these cases. In addition, establishing corrector solutions as individual basis functions is likely to affect the conditioning of the system. On the other hand, as the initial multiscale basis functions form a partition of unity, this strategy maintains a partition of unity at all times.

Despite many potentially problematic properties, we still advertise the second strategy in the scope of the present work. On the one hand, we will show in the next section that the iterative corrector scheme based on additional corrector degrees of freedom converges significantly faster than the corrector scheme based on corrector distribution, since the latter in general does not find an optimal weighting. The smaller number of iterations typically outweighs the increase in the number of basis functions in the coarse-scale MsFEM discretization by a significant margin. On the other hand, although the fine-mesh discretizations can carry a huge number of degrees of freedom, the system size of the coarse-scale MsFEM analysis with multiscale basis functions and the system size of the associated local corrector problems are both relatively small. Therefore, we can always make use of a direct solver efficiently, so that the dependence on conditioning is practically eliminated.

4.3. Local imposition of Dirichlet constraints at embedded surfaces

Cut-cell finite element methods approximate the solution of boundary value problems using non-boundary-fitted meshes [57, 58]. An essential requirement is the variational imposition of Dirichlet boundary conditions at embedded surfaces that arbitrarily cut through finite elements [59–65]. In the scope of this paper, we briefly outline a straightforward variant of an embedded multiscale finite element method that is based on the extension of a penalty method [66]. We will leverage this capability in Section 5.4 for the analysis of a microCT-based vertebra bone with trabecular microstructure, whose outer surface does not align with a structured coarse-scale mesh.

The variational form of the equilibrium equation with the penalty term that enforces Dirichlet boundary conditions can be written as

$$\int_{\Omega} (\mathbb{D} : \nabla_s \mathbf{u}) : \nabla_s \delta \mathbf{u} \, d\Omega + \beta \int_{\Gamma_D} \mathbf{u} \cdot \delta \mathbf{u} \, d\Gamma = \int_{\Omega} \mathbf{f} \cdot \delta \mathbf{u} \, d\Omega + \int_{\Gamma_N} \mathbf{t} \cdot \delta \mathbf{u} \, d\Gamma + \beta \int_{\Gamma_D} \mathbf{g} \cdot \delta \mathbf{u} \, d\Gamma \quad (70)$$

where $\delta \mathbf{u}$ is the virtual displacement vector, \mathbf{g} is the displacement data given on the Dirichlet boundary Γ_D , \mathbf{t} is the traction vector on the Neumann boundary Γ_N , and β is the penalty parameter.

We assume that the Dirichlet boundary Γ_D cuts through a structured mesh of coarse-scale finite elements. Based on (70), multiscale basis functions ϕ_{ix} and ϕ_{iy} in elements Ω_i that are cut by the

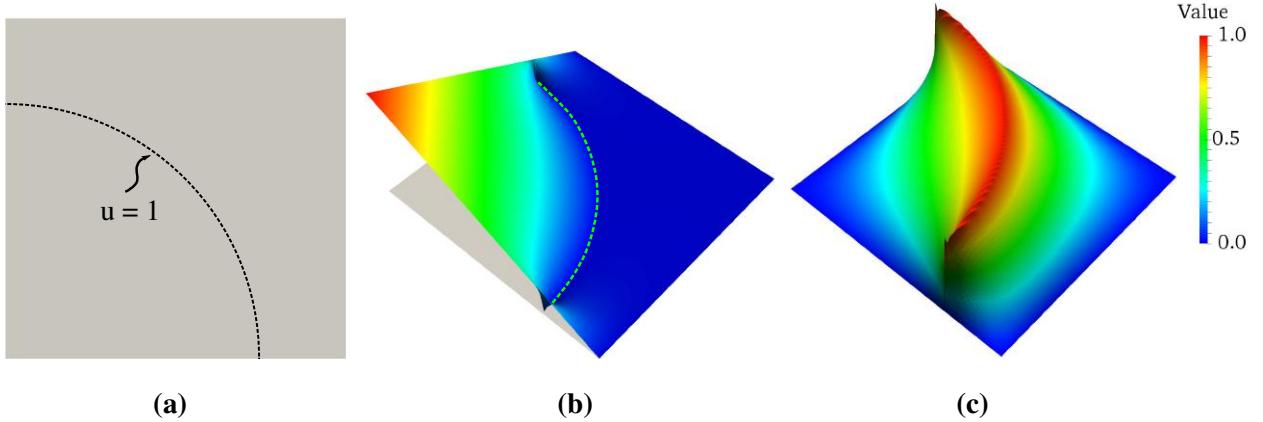


Figure 13: Imposing Dirichlet data at embedded surfaces: (a) a coarse-scale quadrilateral element with an embedded Dirichlet constraint (dashed curve), (b) one of the multiscale basis functions, (c) the bubble part of the solution.

global Dirichlet boundary are the solutions of the following variational statements,

$$\int_{\Omega_i} (\mathbb{D} : \nabla_s \phi_{ix}) : \nabla_s \delta \phi_{ix} \, d\Omega + \beta \int_{\Gamma_{D,i}} \phi_{ix} \cdot \delta \phi_{ix} \, d\Gamma = 0 \quad (71)$$

$$\int_{\Omega_i} (\mathbb{D} : \nabla_s \phi_{iy}) : \nabla_s \delta \phi_{iy} \, d\Omega + \beta \int_{\Gamma_{D,i}} \phi_{iy} \cdot \delta \phi_{iy} \, d\Gamma = 0 \quad (72)$$

where $\delta \phi$ denotes test functions and $\Gamma_{D,i}$ is an embedded surface that is part of the Dirichlet boundary of the global problem. At the element interfaces $\partial\Omega_i$, we impose simple boundary constraints based on standard nodal basis functions as described in Section 4.1, such that we can determine the multiscale basis functions from (71) and (72).

The bubble part of the multiscale solution in each coarse-scale element that is cut by the global Dirichlet boundary is the solution of the following variational statement,

$$\begin{aligned} \int_{\Omega_i} (\mathbb{D} : \nabla_s \mathbf{u}_b) : \nabla_s \delta \mathbf{u}_b \, d\Omega + \beta \int_{\Gamma_{D,i}} \mathbf{u}_b \cdot \delta \mathbf{u}_b \, d\Gamma = \\ \int_{\Omega_i} \mathbf{f} \cdot \delta \mathbf{u}_b \, d\Omega + \int_{\Gamma_{N,i}} \mathbf{t} \cdot \delta \mathbf{u}_b \, d\Gamma + \beta \int_{\Gamma_{D,i}} \mathbf{g} \cdot \delta \mathbf{u}_b \, d\Gamma \end{aligned} \quad (73)$$

with homogeneous boundary conditions on element interfaces $\partial\Omega_i$. In case there is an embedded Neumann boundary $\Gamma_{N,i}$, the expression (73) involves the corresponding traction integral.

Figure 13 illustrates multiscale basis functions and bubble solutions with penalty terms. For clarity of exposition, we consider a 2D Laplace example with a homogeneous material coefficient of $a = 1$ and a penalty parameter $\beta = 10^4$. Figure 13a plots the domain of one coarse-scale element with an embedded Dirichlet boundary, where the solution should be enforced to be $u = 1$. The coarse-scale element is complemented by a fine-scale mesh of 100×100 elements to numerically solve for the multiscale basis functions analogous to (71) and (72) and the bubble part of the solution analogous to (73). In Fig. 13b, we plot one of the multiscale basis functions. We see that it is forced to zero on the embedded boundary by the penalty term, while its values on element

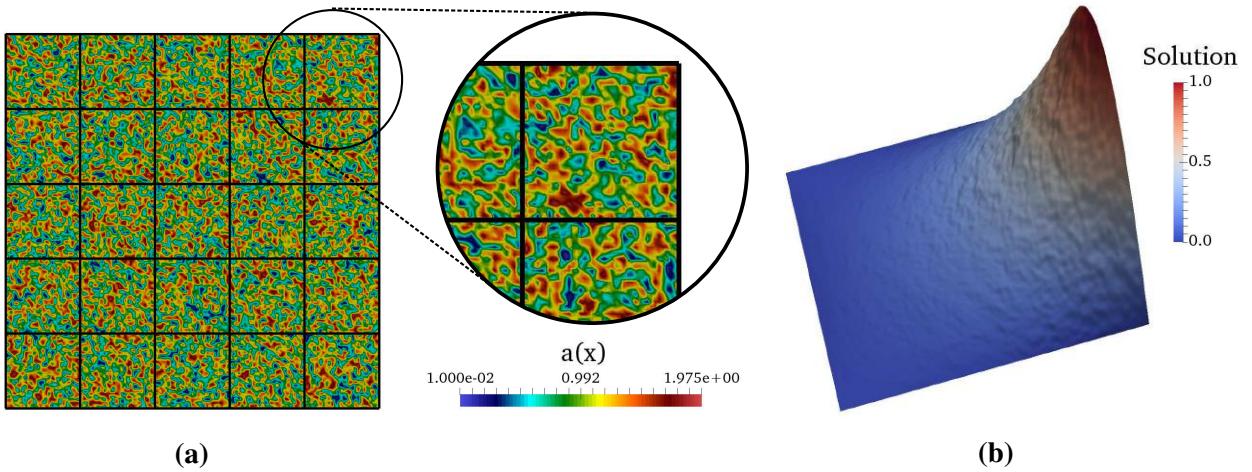


Figure 14: 2D Laplace example: (a) random material distribution on the coarse grid mesh (5×5 elements), (b) reference solution on the fine-scale mesh (200×200 elements).

interfaces correspond to a standard nodal basis function. In Fig. 13c, we observe that within the element domain, the bubble part of the multiscale solution is forced to comply with the Dirichlet data by the penalty term in (73), while its value is zero on the element interfaces. Despite the jump at the element interfaces, this formulation is simple and provides the same fast convergence to the fine-mesh solution when integrated into the corrector scheme. We emphasize again that this is a straightforward variant of embedded multiscale analysis, which serves as the basis for further investigations in current and future research.

5. Numerical examples

In this section, we illustrate the effectiveness of MsFEM enhanced with our iterative corrector scheme in terms of accuracy and computational efficiency by a number of benchmark problems that represent multiscale scenarios with unclear scale separation. These include a Laplace problem with a highly oscillatory coefficient, an advection-diffusion problem with strong local solution gradients, an L-shaped elastic specimen with randomly distributed Young's modulus and a microCT-based vertebra bone characterized by complex trabecular microstructure.

5.1. Laplace equation with a highly oscillatory coefficient

As a first example, we consider the Laplace problem (1) on a unit square domain $[0, 1]^2$ with $f = 0$ and Dirichlet boundary conditions $u(x, 0) = \sin(\pi, x)$ and $u(x, 1) = u(0, y) = u(1, y) = 0$. Figure 14a plots the material coefficient, $a(x)$, that is randomly distributed and highly oscillatory. It is sampled on a four-node quadrilateral mesh of element size $h = 1/128$ and linearly interpolated between nodes. Figure 14b plots the fine-mesh solution obtained with standard FEM and 200×200 four-node quadrilateral elements that will serve as a reference in the following.

To assess the efficiency of our corrector scheme, we compare the solutions of three different methods: (i) standard FEM on a coarse mesh, (ii) MsFEM without the corrector scheme, and (iii) MsFEM with our corrector scheme. All three methods use a coarse mesh with 5×5 four-node

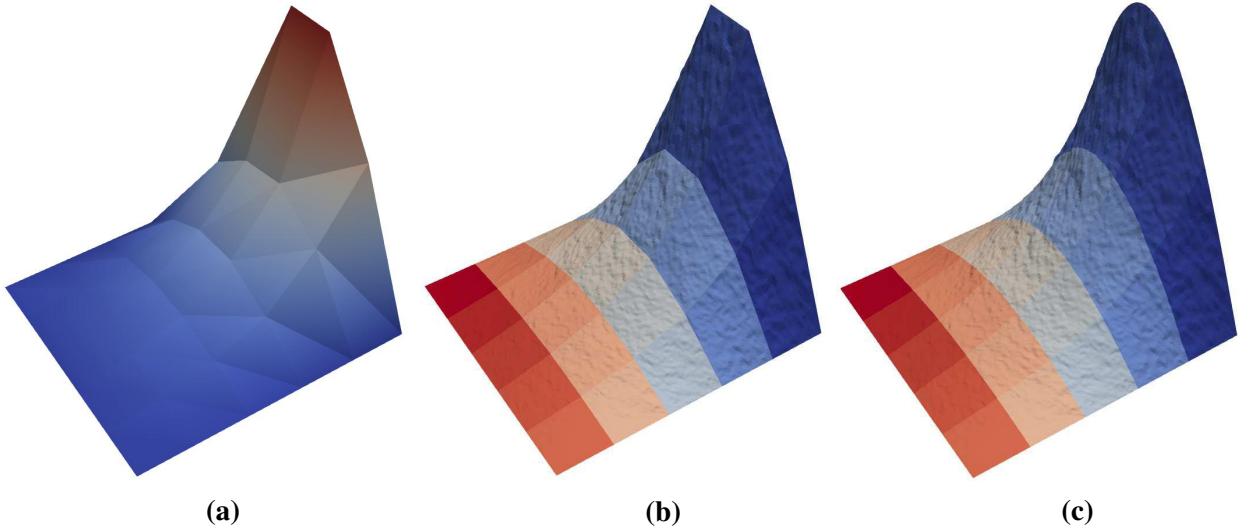


Figure 15: Solutions obtained on a 5×5 coarse-scale quadrilateral mesh with (a) standard FEM, (b) MsFEM without corrector scheme, and (c) MsFEM with the corrector scheme. Note that the solution in (a) is visualized on triangles due to restrictions in our visualization tool. The color in (b) and (c) denotes different element domains.

quadrilateral elements shown in Fig. 14a. In addition, the MsFEM schemes use a fine mesh of 40×40 elements within each coarse-scale element, such that the underlying fine-mesh resolution corresponds to 200×200 elements.

Figure 15 illustrates the solutions obtained with the three methods on the coarse-scale mesh. We can see in Fig. 15a that standard FEM is entirely unable to capture fine-scale oscillations. The MsFEM solution without the corrector scheme in Fig. 15b is able to represent fine-scale solution components inside the element domains. We clearly observe, however, that there is an error on the element interfaces and on the boundary of the domain. Since we use linear interpolations between coarse-scale nodes as local boundary conditions for the multiscale basis functions, we can only interpolate linearly on element interfaces. The MsFEM results in Fig. 15b highlight the important role that local boundary conditions of multiscale basis functions play for the accuracy of MsFEM. The MsFEM solution with our corrector scheme plotted in Fig. 15c demonstrates that our approach is able to overcome the accuracy limit at element interfaces, enabling communication between the fine-scale systems in each coarse-scale element. We can see in Fig. 15c that the corrected MsFEM solution fully captures the fine-scale oscillations both inside the element domains, on local element interfaces and on the boundary of the global domain. We also emphasize that the complete fine-scale solution is fully recovered, which cannot be achieved by standard homogenization methods.

Figure 16 illustrates the convergence of MsFEM with the corrector scheme to the fully resolved fine-mesh solution with increasing number of iterations and for different mesh sizes. The domain is discretized by a series of meshes with 5×5 , 10×10 , 20×20 , and 40×40 coarse-scale elements. At the same time, we fix the fine-scale discretization to 200×200 elements such that in each coarse-scale element, we have 40×40 , 20×20 , 10×10 , and 5×5 fine-scale elements, respectively.

Figure 16a plots the relative error with respect to the fine-mesh solution computed as

$$\text{Relative error} = \frac{\|\bar{\mathbf{u}} - \mathbf{u}_{\text{FEM}}^{\text{fine}}\|_2}{\|\mathbf{u}_{\text{FEM}}^{\text{fine}}\|_2} \quad (74)$$

where $\|\cdot\|_2$ is the L^2 norm, and $\bar{\mathbf{u}}$ and $\mathbf{u}_{\text{FEM}}^{\text{fine}}$ are the solutions obtained by MsFEM and by fine-mesh FEM, respectively. We observe that the multiscale solutions rapidly converge to the solution of the fully resolved fine mesh irrespective of the ratio between the coarse-scale mesh size H and the fine-scale mesh size h .

The first points of the curves represent the error of MsFEM with linear boundary conditions on element interfaces. It is interesting to note that when we refine the coarse mesh from 5×5 to 40×40 elements, and thus reduce the constraints on the fine-mesh, we do not see a significant decrease of the error. This phenomenon is due to the so-called resonance effect [28]. Its impact on the convergence can be summarized as

$$\|\bar{\mathbf{u}} - \mathbf{u}_{\text{FEM}}^{\text{fine}}\|_2 = O(H^2 + \epsilon/H) \quad (75)$$

where ϵ is the characteristic length of the fine scale represented by the fine-scale mesh. When the coarse-scale mesh size H is close to ϵ (but still $H > \epsilon$), the error terms that depend on ϵ/H dominate the total error. When we apply the corrector scheme, we can significantly reduce this error after each iteration.

The corrector scheme can be seen as a fixed-point iteration scheme. We can therefore examine the rate of convergence of the iterative corrector scheme with the help of the following expression,

$$\lim_{q \rightarrow \infty} \frac{\|\bar{\mathbf{u}}_{q+1} - \mathbf{u}_{\text{FEM}}^{\text{fine}}\|_2}{\|\bar{\mathbf{u}}_q - \mathbf{u}_{\text{FEM}}^{\text{fine}}\|_2^\rho} = \lambda \quad (76)$$

where $\bar{\mathbf{u}}_q$ and $\bar{\mathbf{u}}_{q+1}$ denote the solution at the q -th and $(q+1)$ -th iteration, and the exponent ρ is the order of convergence. An iterative scheme converges with the order ρ when $\lambda < \infty$ at all times. In our case, it is easy to check from the results shown in Fig. 16 that this holds for $\rho = 1$, but not for $\rho = 2$. Therefore, as expected, our scheme is first order.

Figure 16b plots the norm of the residual vector that drives the corrector procedure and can therefore be seen as a direct indicator of error. We observe that the norm of the residual vector rapidly decreases to zero (machine accuracy). This is confirmed by Fig. 17 that plots the magnitude of the residual on the domain in the first three iterations. We can see that the corrector scheme effectively decreases the residual within a few iterations. Note that the residual is nonzero only at the element interfaces.

5.2. Advection-diffusion with strong local solution gradients

In the second example, we consider the advection-diffusion equation,

$$\mathbf{a}(\mathbf{x}) \cdot \nabla u(\mathbf{x}) - \nabla \cdot (D \nabla u(\mathbf{x})) = 0 \quad (77)$$

on a quarter of an annular domain [67]. Figure 18a illustrates the problem domain, defines the boundary conditions, and provides values for all problem parameters. We assume a flow field $\mathbf{a}(\mathbf{x})$

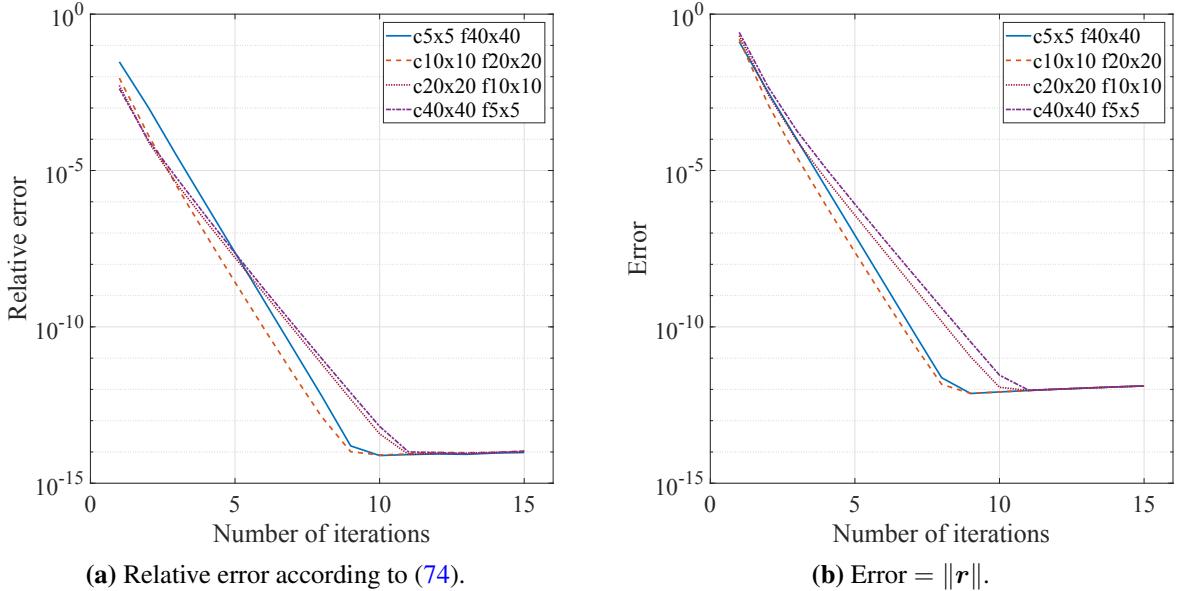


Figure 16: Convergence of MsFEM with the corrector scheme with increasing number of iterations and for different mesh sizes; (a) with respect to the fully resolved fine-mesh solution, (b) in terms of the norm of the residual vector.

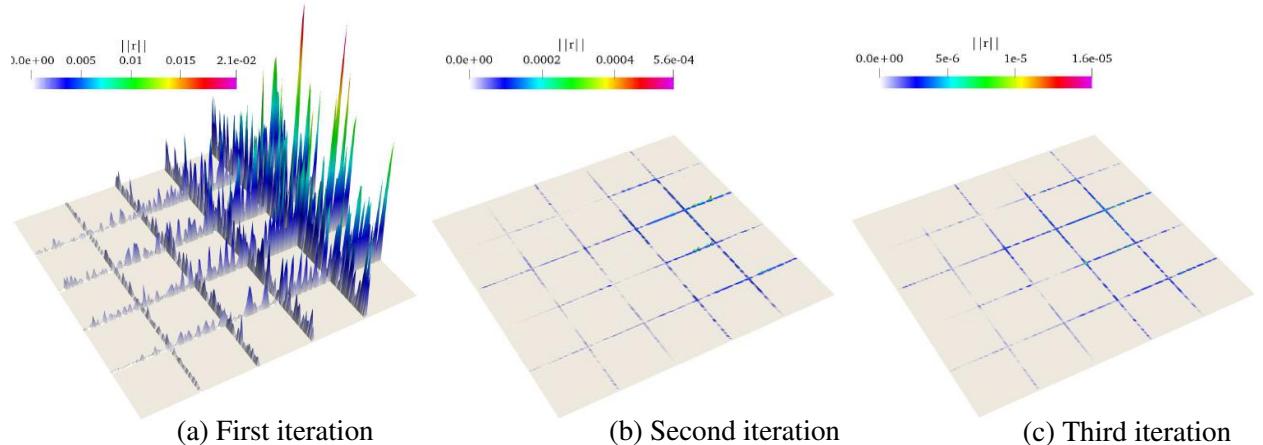


Figure 17: Residual force magnitude in different iterations.

with tangential velocity $a_\theta = \omega r$ and radial velocity $a_r = 0$. The Péclect number of this problem is $\text{Pe} = 10$. The concentration $u = 1$ on the inflow boundary creates boundary and internal layers. The solution by standard FEM with a fine mesh resolution of 200×200 four-node quadrilateral elements and 40,401 degrees of freedom is shown in Fig. 18b.

For our MsFEM computations, we consider a coarse mesh with 20×20 elements and only 441 degrees of freedom. We find multiscale basis functions by numerically solving

$$\mathbf{a}(\mathbf{x}) \cdot \nabla \phi_i(\mathbf{x}) - \nabla \cdot (D \nabla \phi_i(\mathbf{x})) = 0 \quad (78)$$

on each element domain, where ϕ_i is the multiscale basis function of node i . We use linear

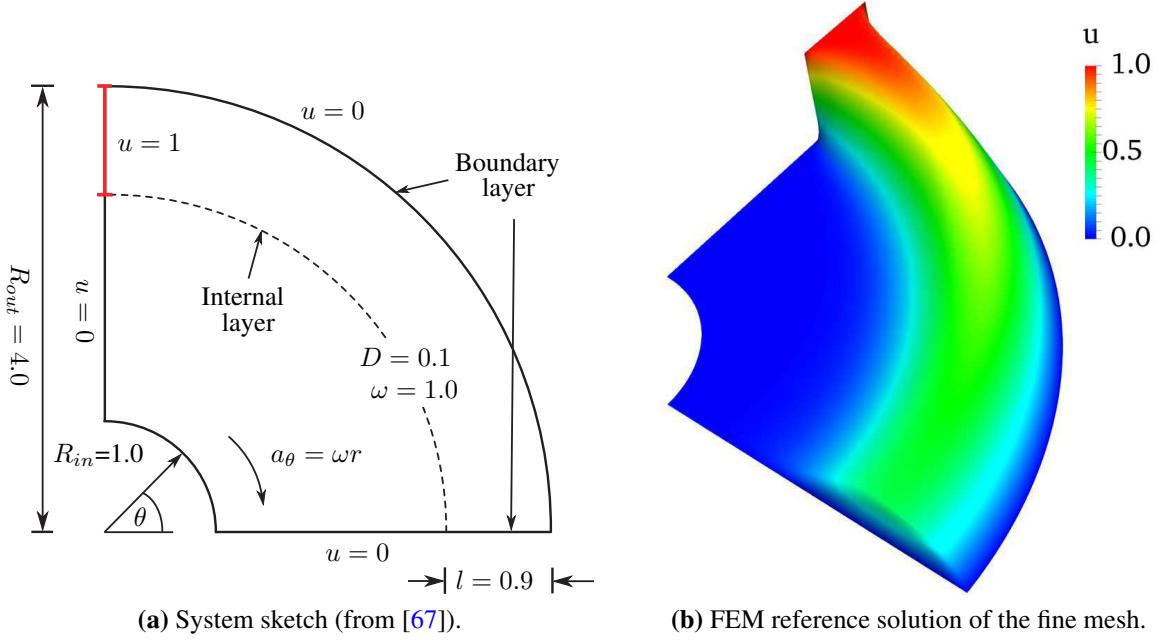


Figure 18: Advection-diffusion with strong solution gradients in an annular section.

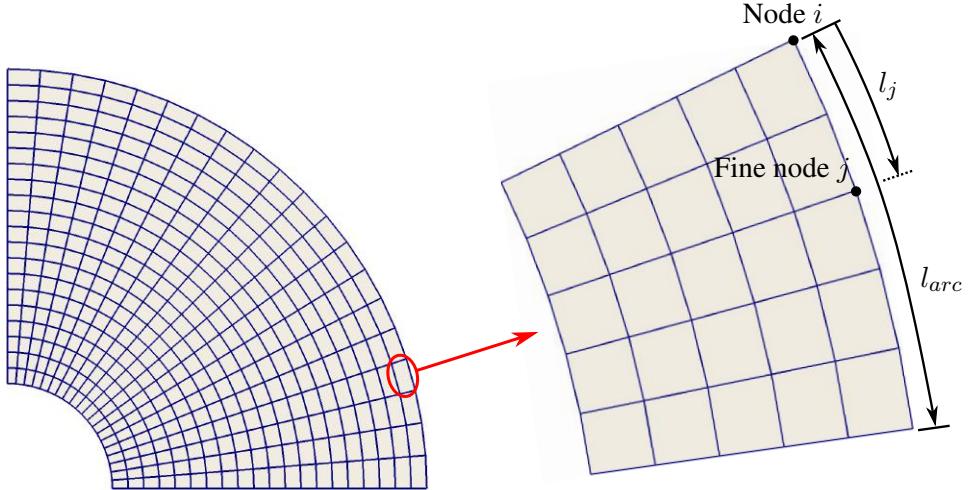


Figure 19: Local imposition of global Dirichlet data on multiscale basis functions of a coarse element.

interpolations between coarse-scale nodes as the initial local boundary conditions for the multiscale basis functions. In Fig. 19, we illustrate the imposition of local boundary conditions on multiscale basis functions of a coarse element. When computing the multiscale basis function at node i , we determine nodal values N_j at fine-scale nodes j located at the curved edge as

$$N_j = 1 - \frac{l_j}{l_{arc}} \quad (79)$$

where l_{arc} and l_j are the arclength of the corresponding side and the arclength from the coarse-scale

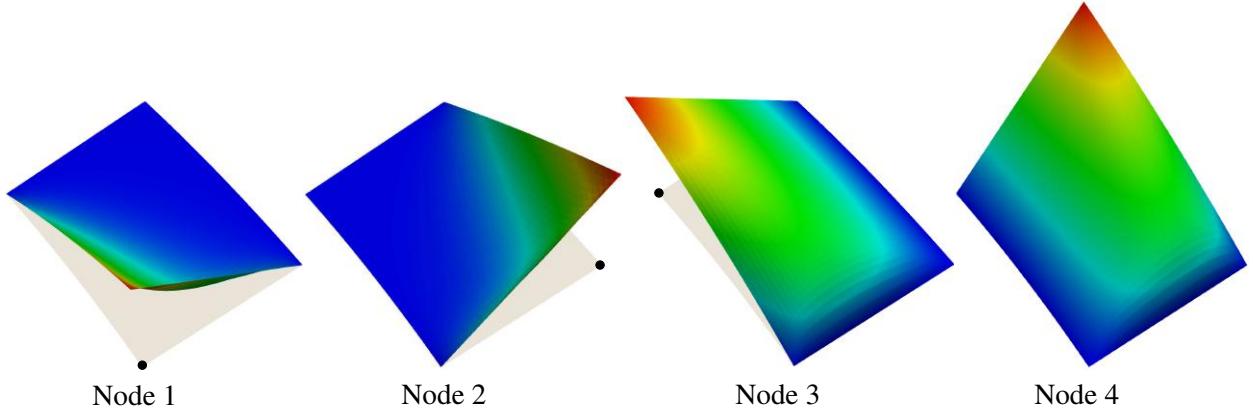


Figure 20: Multiscale basis functions for the advection-diffusion problem, plotted on one of the coarse-scale elements. We observe that they incorporate the advection effect, supporting stable MsFEM analysis on coarse-scale meshes.

node i to the fine-scale node j , respectively. The procedure in (79) demonstrates that as part of our corrector scheme, we can impose any boundary condition, which guarantees conforming basis functions for the initial multiscale basis. The corrector scheme can correct these boundary conditions later through iterations. We note that for geometries with more complex curved boundaries, a mapping algorithm that is more general than (79) is required.

Figure 20 plots the resulting multiscale basis functions of one of the coarse-scale elements. We observe that the multiscale basis functions incorporate the advection effect due to the advection term in (78). Therefore, the multiscale basis functions yield a stable solution on the coarse-scale mesh, while standard FEM leads to spurious oscillations. The incorporation of the advection effect can be perceived of as a stabilization procedure, constituting another link to variational multiscale and stabilized finite element methods [68–70].

The corrector scheme can be applied to all nodes of the coarse mesh. We can see from the fine-mesh solution in Fig. 18b, however, that we only need the fine resolution in high-gradient regions. In other areas, the initial multiscale basis functions of the coarse-scale mesh can provide sufficient accuracy. To identify multiscale basis functions for which correction is required, we can use the residual as an error indicator. The residual is readily available as it is computed as part of the corrector scheme. This gives rise to the following adaptive strategy. We compute the 2-norm of the discrete residual vector $|\mathbf{r}|$ for each corrector node and only solve corrector problems which have $|\mathbf{r}| > \text{tolerance}$. In this particular problem, we set the tolerance to 0.001.

Figures 21 and 22 illustrate the support of the corrected multiscale basis functions and the magnitude of the residual, respectively, in the first three iterations of the corrector scheme. We observe that the adaptive scheme based on the residual automatically chooses high-residual regions which naturally correspond to high-gradient regions, where the accuracy needs to be improved. In addition, the error is reduced significantly after each iteration. We note that in standard FEM, common adaptivity techniques are: (i) generating an adaptive mesh in advance (which requires prior knowledge about solution characteristics), or (ii) creating adaptive meshes on the fly (which is complicated in terms of implementation and parallelization). Our approach offers an attractive alternative that elegantly circumvents these challenges.

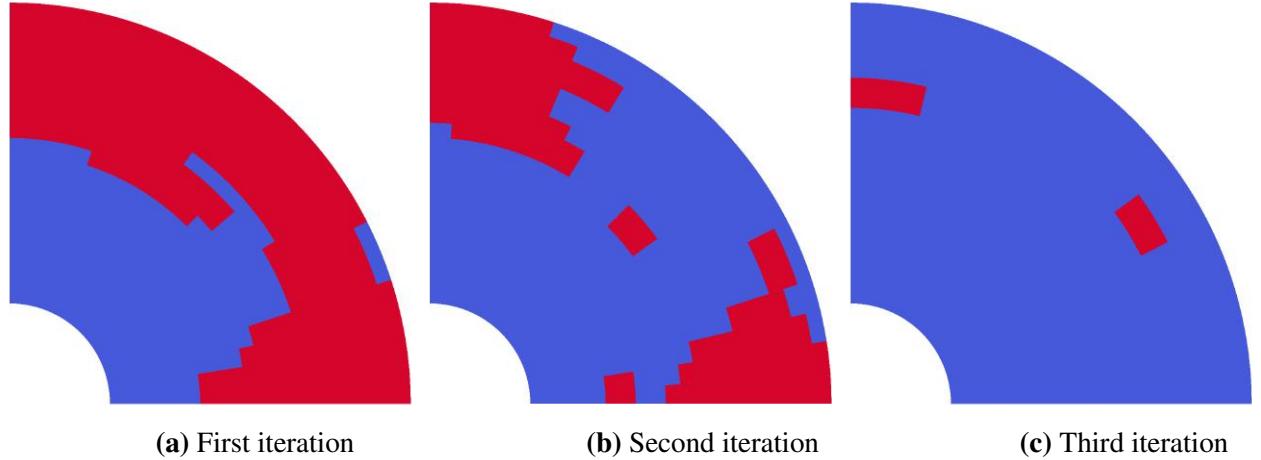


Figure 21: Adaptive corrector regions (plotted in red) in the first three iterations, selected based on the 2-norm of the local residual vector in each potential corrector region.

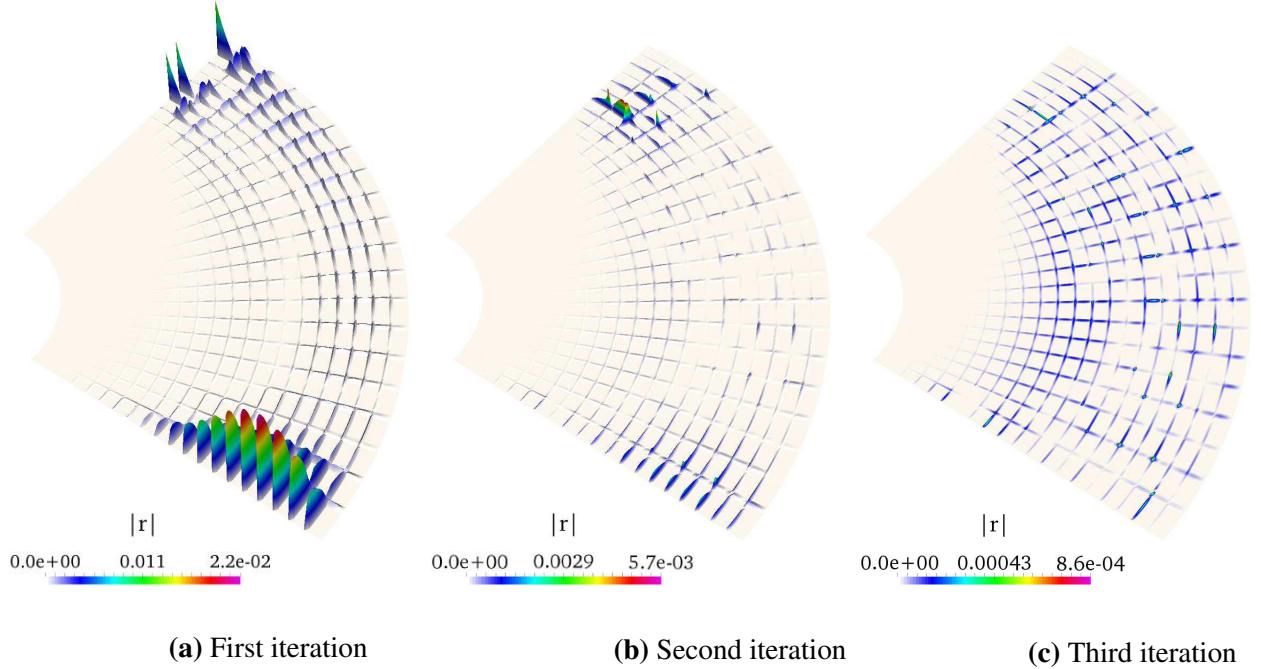


Figure 22: Magnitude of the residual.

5.3. L-shaped specimen with randomly distributed Young's modulus

To illustrate the effectiveness of MsFEM and our corrector scheme for linear elasticity, we consider the L-shaped elastic specimen shown in Fig. 23. The material properties of the L-shaped specimen are described by a constant Poisson's ratio of $\nu = 0.3$ and a randomly distributed Young's modulus that is illustrated in Fig. 23a. Global Dirichlet data and the coarse-scale mesh that corresponds to 20 four-node quadrilateral elements along each main direction are shown in Fig. 23b. Each coarse-scale element contains a local fine mesh of 10×10 four-node quadrilateral elements such that the equivalent fine mesh corresponds to 200 elements in each main direction.

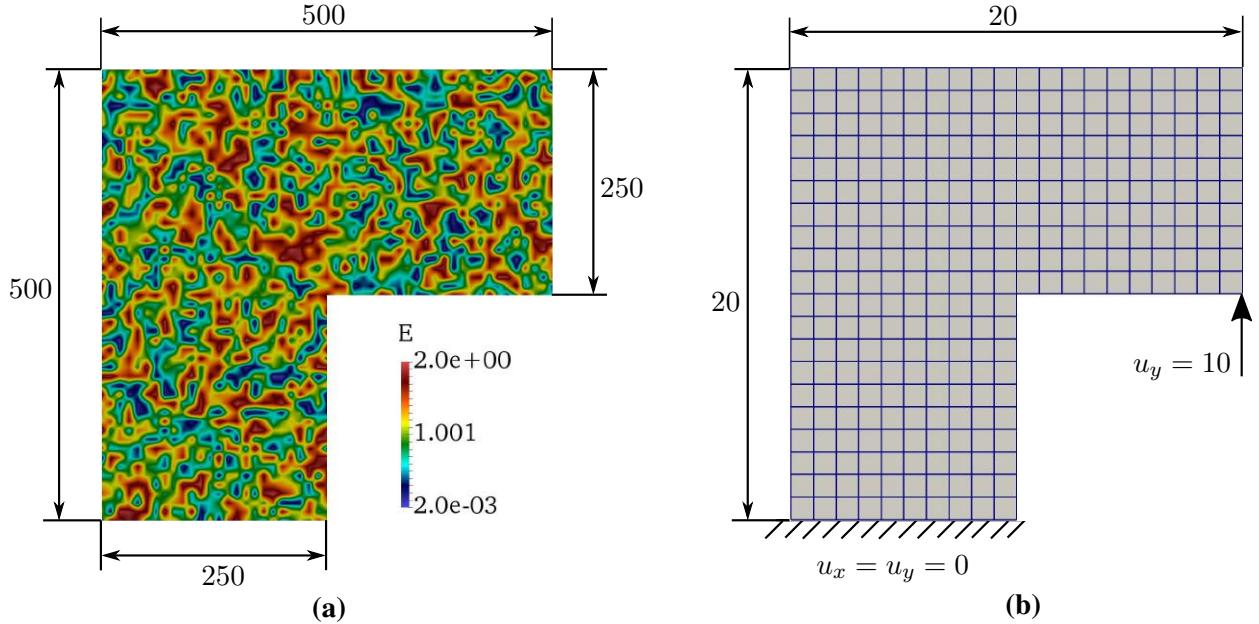


Figure 23: Heterogeneous distribution of Young's modulus, dimensions, global boundary conditions, and coarse-scale mesh size of the L-shaped specimen.

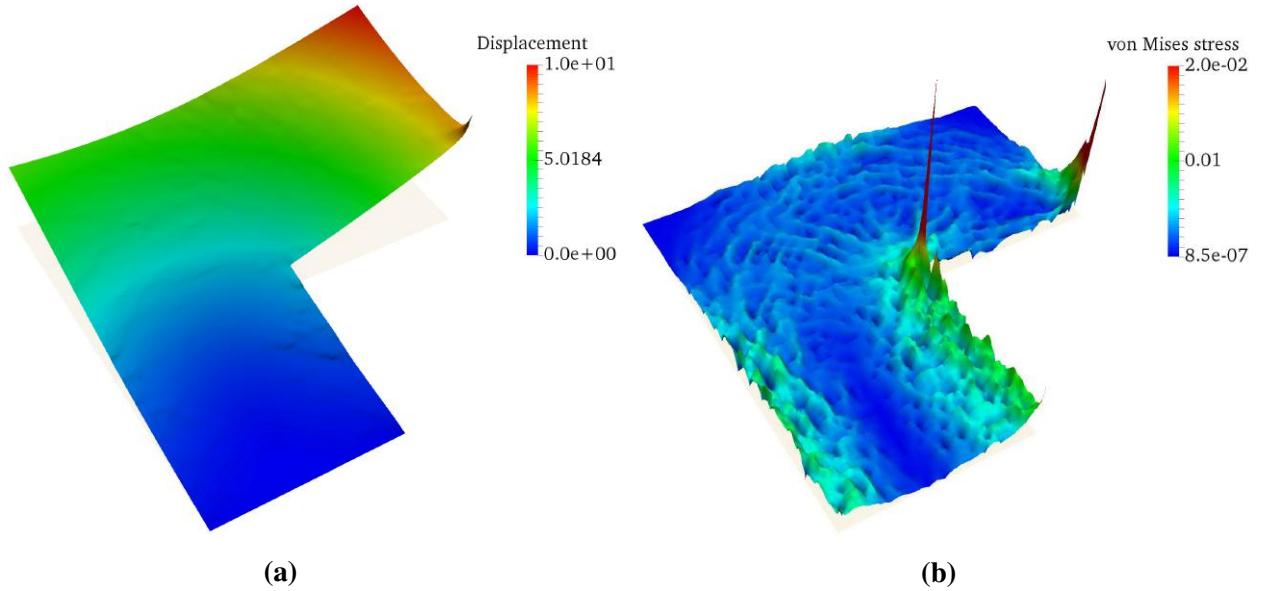


Figure 24: L-shaped specimen: (a) Total displacements and (b) von Mises stresses obtained with MsFEM.

Figure 24 plots the MsFEM solution fields in terms of total displacements and von Mises stress. We observe that the effect of the material variability on the von Mises stress is significant. Standard homogenization methods cannot take into account this effect at the same level of accuracy, since the material variability is not sufficiently separated from the global length scale. An interesting side note is that regions of increased stress select paths of increased stiffness to form a truss-like pattern.

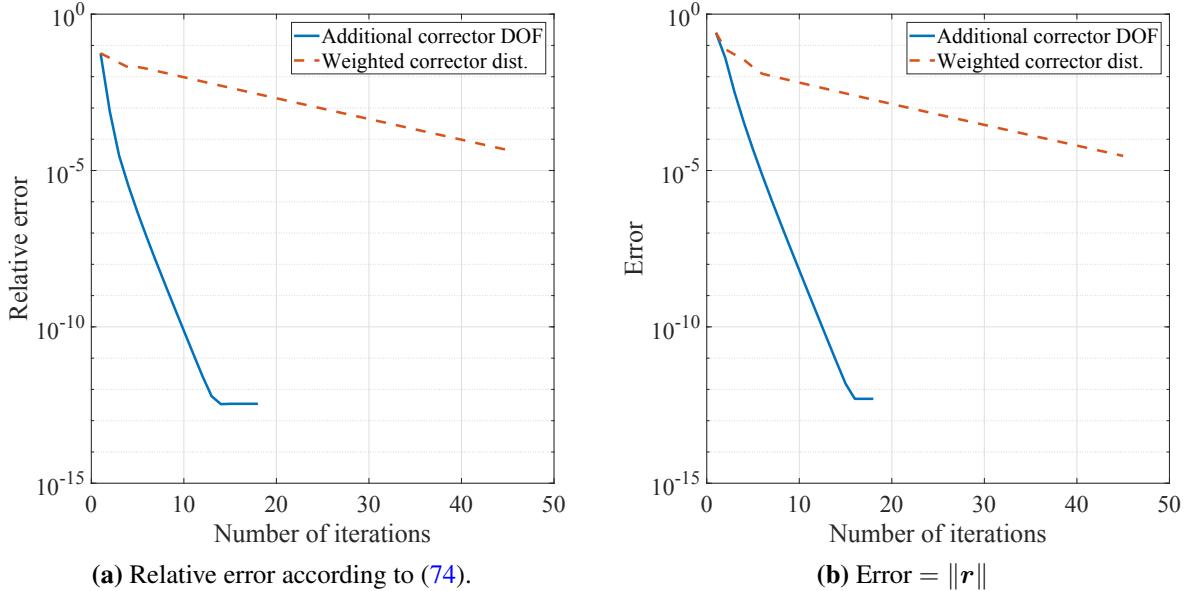


Figure 25: Comparison of MsFEM with the additional corrector degree of freedom and MsFEM with weighted corrector distribution in terms of convergence of the relative error of the solution and the norm of the residual.

In the following, we demonstrate that in our corrector scheme, adding a corrector degree of freedom is computationally superior to the weighted distribution of the corrector onto the vector multiscale basis functions. To this end, we employ MsFEM and the corrector scheme with each approach, introduced in Section 4.1. Figure 25 plots the convergence of the MsFEM solution with respect to the number of iterations for both options. We note that we adopt again the fine-mesh solution obtained with standard FEM as a reference. We clearly observe that the MsFEM solution obtained with a separate corrector degree of freedom converges significantly faster than the MsFEM solution obtained with the weighted corrector distribution. This indicates that the split as defined in (67) is far from optimal for elasticity. In this example, the number of degrees of freedom for MsFEM with weighted corrector distribution is 682, while for MsFEM with corrector degrees of freedom it amounts to 1,023. With respect to the fully resolved fine mesh that carries 60,802 degrees of freedom, this increase in degrees of freedom is negligible and compensated for by far by the reduction in iterations. We therefore recommend to use an additional corrector degree of freedom at each coarse-scale node.

5.4. MicroCT-based stress analysis of a vertebra

Our final example addresses MsFEM analysis of heterogeneous bone structures. To this end, we consider a vertebral body given by microCT data that has been separated from the surrounding imaging data with the help of robust variational segmentation technology [71]. Figure 26a shows a segmented CT slice, illustrating the available image resolution that captures the trabecular microstructure of the vertebra. Stacking slices on top of each other, we construct a volumetric pixel (voxel) model of the vertebral body, illustrated in Fig. 26b. The spacing corresponds to the pixel spacing $\Delta_x = \Delta_y = 0.1465$ mm in each slice and the distance $\Delta_z = 0.3$ mm between slices. One of

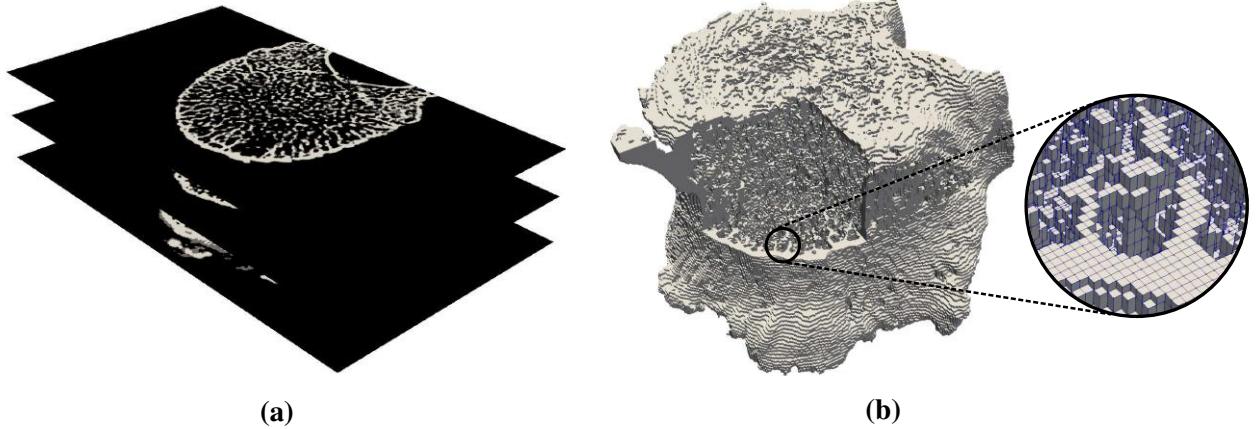


Figure 26: Image-based representation of a vertebra: (a) slices of segmented CT scans and (b) voxel reconstruction of the three-dimensional geometry. The mesh that identifies each voxel with one hexahedral element serves as our fine-mesh reference.

the most common methods for the direct analysis of imaging data is the voxel finite element method, which identifies each voxel as a linear hexahedral element [72–77]. For our vertebra sample, the hexahedral mesh that corresponds to the voxel model of Fig. 26b consists of 8,435,574 degrees of freedom. While this fine-scale voxel mesh fully resolves the effect of the trabecular microstructure, it requires significant memory and computing time.

MsFEM combined with our corrector scheme constitutes an opportunity to reduce the computational cost, while achieving the same accuracy as the full-resolution voxel FEM. Figure 27a plots the coarse-scale mesh for the MsFEM analysis that consists of 11,817 nodes. With one additional corrector degree of freedom per node, the total number of degrees of freedom of the coarse-scale problem amounts to 47,268. On the fine scale, we use the same voxel elements as voxel FEM. They align with the coarse-scale mesh such that $10 \times 10 \times 5$ voxel elements are covered by one coarse element (see Fig. 27b). Hence, one coarse-scale element includes 500 fine-scale elements. We note that MsFEM in this form directly operates on the voxel data and does not require the (potentially labor-intensive) generation of a boundary-fitted finite element mesh. As a consequence, it enables the same seamless integration of voxel data into analysis as standard voxel FEM.

Although bone, in general, is an anisotropic material [78], it has been shown and experimentally verified that in the case of vertebral simulations, an isotropic material model is a reasonable approximation [79–82]. In the voxel representation of the vertebra, we can therefore assume that each voxel is associated with two material parameters (Young's modulus E , Poisson's ratio ν), where voxels with bone material carry (10 GPa, 0.3) [83, 84] and void voxels carry (10^{-5} GPa, 0.3) [85, 86]. The trabecular microstructure is implicitly represented by the stiffness variation from one voxel to the next. This is illustrated in Fig. 27c that plots voxels with bone material for one coarse-scale element. The small stiffness value in void voxels guarantees that their effect on the mechanical response of the vertebra is negligible. At the same time, this effectively prevents a singular stiffness matrix in case of single voxels with bone material that are disconnected from the main body. We note that we remove all coarse-scale elements from the initial mesh that do not contain at least one voxel with bone material.

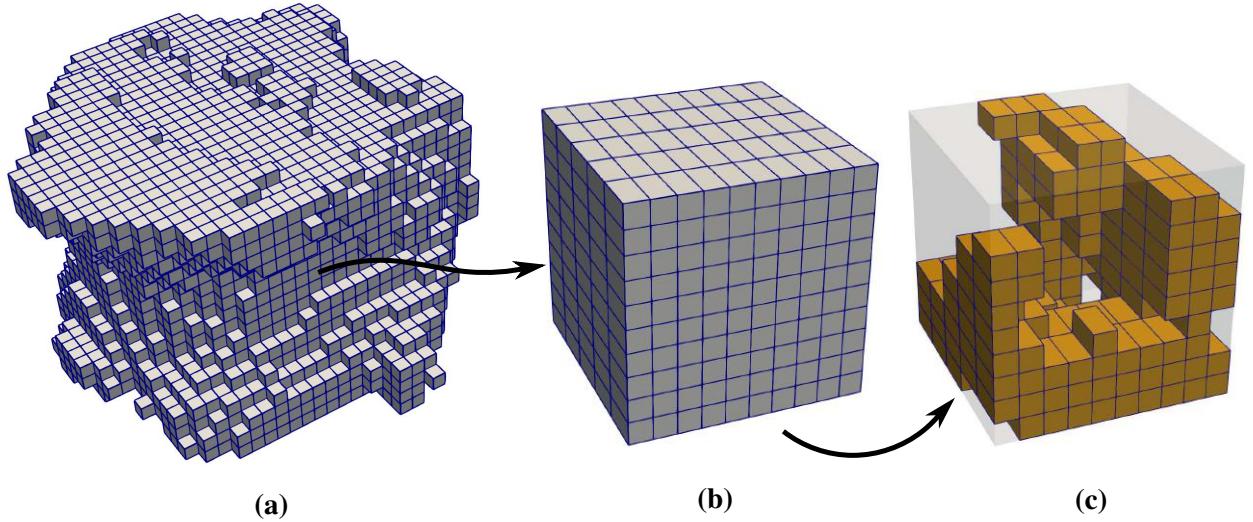


Figure 27: Image-based discretization of the vertebra: (a) complete coarse-scale mesh, (b) voxel mesh in one coarse-scale element, and (c) voxel elements with bone material, encoding the trabecular microstructure at the given CT resolution. We note that the voxel elements in (b) and (c) have been rotated for better visualization.

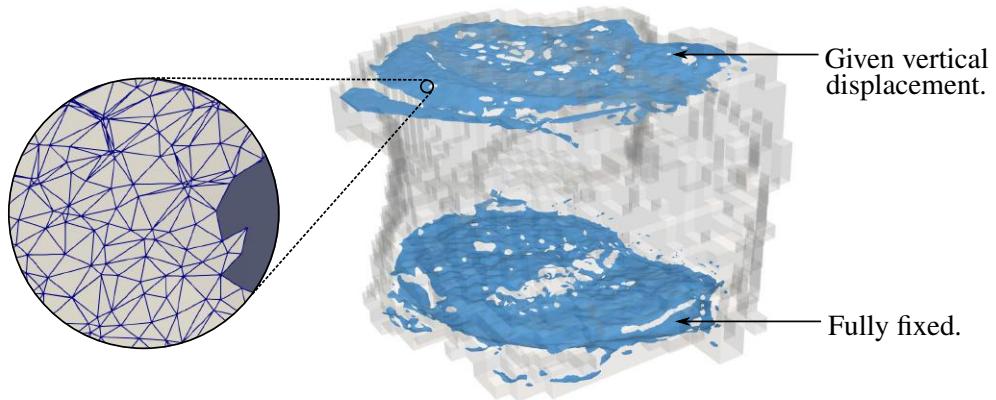


Figure 28: Dirichlet boundary data are imposed variationally on the upper and lower loading surfaces.

To illustrate the effectiveness of our corrector scheme for this application, we carry out a virtual compression test of the vertebra, for which we fix the displacements at the bottom surface and apply a uniform downward vertical displacement of 1 mm at the top surface. The global boundary conditions illustrated in Fig. 28 are imposed variationally on embedded explicit triangulations of the upper and lower vertebra surfaces, based on the embedded penalty method outlined in Section 4.3. We note that these explicit triangulations were obtained as part of our variational segmentation process [71]. We use the same fine-scale voxel mesh and the same global Dirichlet boundary data imposed over the same triangulated top and bottom surfaces in the full-resolution voxel FEM and the MsFEM computations. We can therefore expect that MsFEM leads to exactly the same results as the voxel FEM, given the former achieves the full fine-mesh accuracy.

For visualization purposes, the solution fields described by the multiscale basis are projected

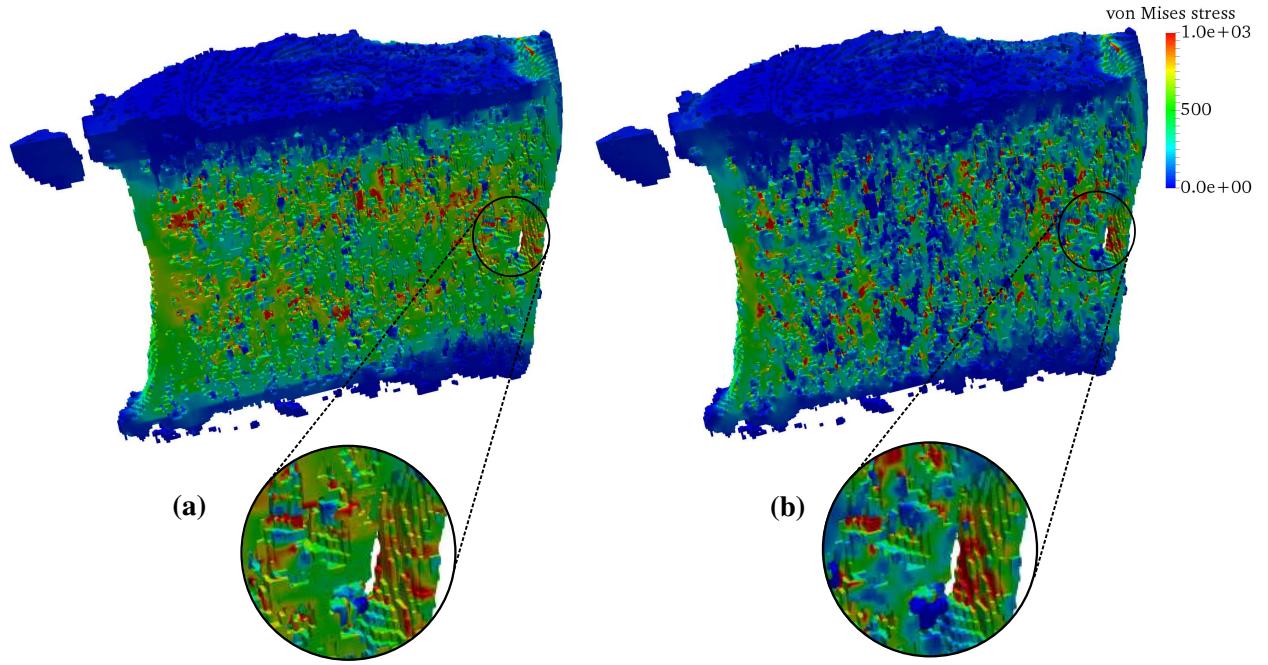


Figure 29: Von Mises stress in the trabecular microstructure: (a) initial MsFEM result and (b) after 10 corrector iterations (the coarse-scale MsFEM mesh has approx. 45,000 dofs).

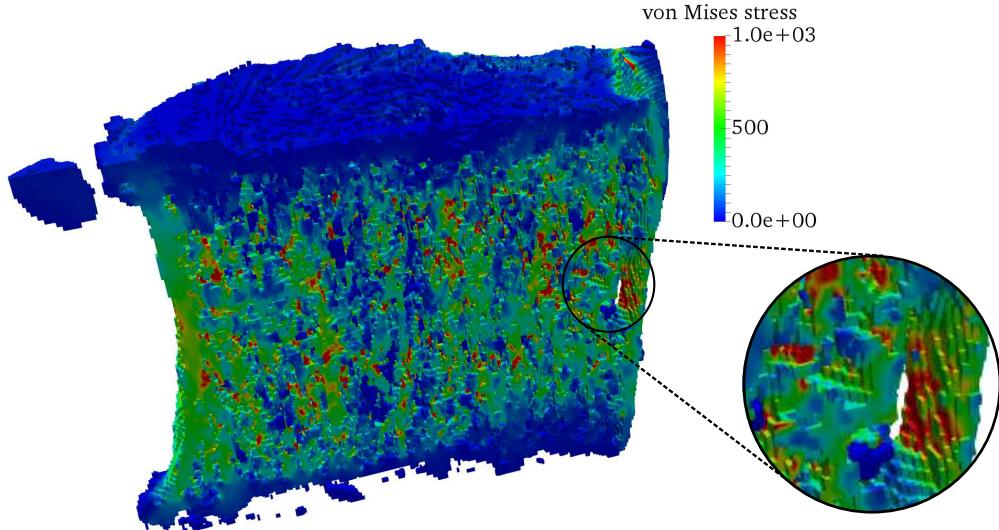


Figure 30: Von Mises stress in the trabecular microstructure, computed with full voxel FEM (8.5 million dofs).

back on the voxel representation and plotted voxel by voxel across a vertical plane that cuts through the center of the vertebral body. Figures 29a and 29b plot the von Mises stresses in the vertebral body obtained with MsFEM without correction and with MsFEM and ten corrector iterations, respectively. We observe a significant difference between the initial uncorrected solution and the solution after a few iteration steps. Figure 30 plots the von Mises stress results obtained with

full-resolution voxel FEM. When we compare the MsFEM results with results obtained with voxel FEM, we can see that MsFEM with iterative correction leads to stresses that are virtually indistinguishable from the fine-mesh reference.

We conclude that MsFEM with our iterative corrector scheme achieves the full fine-mesh accuracy of the voxel FEM. At the same time, it reduces the computational cost of voxel FEM, in particular if the multiscale basis functions can be pre-computed in parallel and re-used several times. For example, MsFEM (approx. 45,000 degrees of freedom) could be used for real-time parametric studies, for instance to determine the maximum stress for a large number of different loading conditions, reaching the same accuracy as full-resolution voxel FEM (approx. 8.5 million degrees of freedom) at any point in the microstructure. In the present case, corrector iterations are carried out for each coarse-scale element until the global norm of the residual reaches machine accuracy. We emphasize that in practical engineering computations, one or two corrector iterations carried out in a few critical elements are sufficient to guarantee fine-scale high-fidelity accuracy.

6. Summary, conclusions and outlook

While many multiscale finite element (MsFEM) schemes can handle up to 1,000 times more elements than standard FEM with the same amount of memory [26, 28], their accuracy often remains affected by constraints at coarse-scale element interfaces. In this paper, we presented a corrector scheme that enables MsFEM to provide the best possible fine-mesh accuracy, while preserving its core computational advantages. Its main idea is to incorporate the fine-mesh solution information into multiscale basis functions on element interfaces through iterations. We showed analytically that the residual of the multiscale solution is the optimal choice to drive the iterative correction procedure and that its solution will converge to the global fine-mesh result of a full-resolution discretization. The corrector scheme thus enables communication between fine-scale systems in coarse-scale elements. Since the corrector scheme only operates on local domains, it is well-suited for parallel computing. To avoid race conditions due to overlapping corrector regions, we proposed its implementation using non-overlapping node patterns that can be parallelized individually.

We extended our approach to vector field problems, focusing on elasticity. To updated multiscale basis functions, corrector solutions must be distributed, since at each node there are multiple basis functions, but only one corrector solution. To overcome this issue, we introduced the concept of an additional corrector degree of freedom at each coarse-scale node, thus completely avoiding a distribution of the corrector solution. The local corrector solutions represent additional multiscale basis functions that are updated during the iterative procedure. In addition, we also discussed a straightforward variant to impose Dirichlet constraints on embedded surfaces in the context of MsFEM and the corrector scheme.

We presented several numerical examples, including a 2D Laplace equation with a highly oscillatory coefficient, an advection-diffusion problem with strong local solution gradients, and an elasticity problem with randomly distributed Young's modulus. In all examples, the iterative procedure converges rapidly to the fine-mesh solution irrespective of the ratio between coarse mesh size H and fine mesh size h . We also observed that our approach can iteratively eliminate the resonance effect. In addition, we showed that the corrector scheme can be naturally applied in an adaptive way to improve the accuracy of any specific regions of interest. In particular, since it

is driven by residuals, our scheme enables automatic adaptivity without further error estimation steps. We also observed that applying the concept of corrector degrees of freedom in vector field problems preserves the same fast convergence of the iterative scheme that we had observed in scalar problems, where multiscale basis functions can be directly updated by corrector solutions.

We finally applied our MsFEM approach with local correction for microCT-based stress analysis of a vertebra with complex trabecular microstructure. This example highlighted the potential of MsFEM for effective imaging-driven analysis of heterogeneous materials with random microstructure and unclear scale separation. We showed that the coarse-scale MsFEM mesh can handle the full-resolution vertebra problem with only 45,000 degrees of freedom, while the standard voxel FEM requires 8.5 million degrees of freedom. A comparison of the solution fields, however, revealed that MsFEM with simple linear constraints on element interfaces and the full-resolution voxel FEM produce solutions with significant differences in stresses. We demonstrated that with a few iterations, our corrector approach is able to restore the full accuracy with respect to the reference, illustrated by voxel-wise stress concentrations at coarse-scale element interfaces that are practically indistinguishable from the voxel FEM result.

In the future, we will extend MsFEM with iterative correction to nonlinear problems, where we anticipate several significant advantages. First, MsFEM with residual-driven correction enables the fully accurate simulation of localization up to the fine mesh size h . This is for example crucial for accurately tracing cracks that advance across coarse-scale element interfaces. Second, we anticipate that the iterative corrector algorithm presented in this paper can be merged synergistically with the Newton-Raphson algorithm in such a way that iterative correction does not increase the overall computational cost of nonlinear MsFEM. A core issue to be addressed is how to maintain the orthogonality of multiscale basis functions, corrector basis functions, and bubble solutions.

Acknowledgments: The authors gratefully acknowledge support from the National Science Foundation via the NSF grant CISE-1565997. The first author (L.H. Nguyen) was partially supported by a Doctoral Dissertation Fellowship awarded by the University of Minnesota for the academic year 2017-2018. The second author (D. Schillinger) gratefully acknowledges support from the National Science Foundation through the NSF CAREER Award No. 1651577.

The authors also acknowledge the Minnesota Supercomputing Institute (MSI) of the University of Minnesota for providing computing resources that have contributed to the research results reported within this paper (<https://www.msi.umn.edu/>). The authors are grateful to Thomas Baum and Jan S. Kirschke (Dept. of Neuroradiology, Technische Universität München, Germany) for providing access to the medical imaging data of the vertebra.

Appendix A. Multiscale finite element discretization and implementation

In the following, we briefly discuss some more details on the discretization process in MsFEM, using the two-dimensional Laplace equation and the multiscale discretization shown in Fig. 1 as an illustrative example.

Appendix A.1. Approximation of bubble part and multiscale basis functions

A finite element approximation of the bubble part u_b of the solution can be computed by solving the local boundary value problem (6) and (7) for each coarse-scale element. The fine mesh in each coarse-scale element domain consists of a set of m basis functions $\mathbf{N}(\mathbf{x}) = \{N_{i,p}(\mathbf{x})\}_{i=1}^m$ of polynomial degree p . We can represent (6) and (7) in variational form as

$$\int_{\Omega_i} \nabla v^h(\mathbf{x}) \cdot a(\mathbf{x}) \nabla u_b^h(\mathbf{x}) \, d\Omega = 0 \quad (\text{A.1})$$

where the bubble part u_b and weighting functions are discretized as

$$u_b^h(\mathbf{x}) = \sum_{i=1}^m N_{i,p}(\mathbf{x}) \hat{u}_i^b \quad (\text{A.2})$$

$$v^h(\mathbf{x}) = \sum_{i=1}^m N_{i,p}(\mathbf{x}) \hat{v}_i \quad (\text{A.3})$$

with \hat{u}_i^b and \hat{v}_i being the coefficients of discrete bubble solution and the test functions. Taking into account the homogeneous boundary conditions (7), the finite element problem (A.1) can be locally solved for \hat{u}_i for each coarse-scale element.

In the same way, a finite element approximation of the multiscale basis functions ϕ_i at node i of the coarse-scale mesh can be computed by solving (8) with suitable boundary conditions for each coarse-scale element. We recall that different options for choosing boundary conditions have been briefly discussed in Section 2.2. We can write (8) in variational form,

$$\int_{\Omega_i} \nabla v^h(\mathbf{x}) \cdot a(\mathbf{x}) \nabla \phi_i^h(\mathbf{x}) \, d\Omega = 0 \quad (\text{A.4})$$

where v^h are weighting functions (A.3) and the multiscale basis function are discretized by the fine-mesh basis as

$$\phi_i^h(\mathbf{x}) = \sum_{j=1}^m N_{j,p}(\mathbf{x}) d_{ij} \quad (\text{A.5})$$

with discrete coefficients d_{ij} . We emphasize again that computing the solutions of (A.1) and (A.4) are entirely local tasks that can be accomplished independently on each coarse-scale element. This part of the MsFEM procedure that can therefore be efficiently parallelized is often referred to as *offline* step, inspired by its conceptual similarity to model order reduction.

Appendix A.2. Implementation of the multiscale solution procedure

With the offline step completed, the multiscale solution $u_{ms}^H(\mathbf{x})$ and the corresponding weighting functions $v_{ms}(\mathbf{x})$ can be written in terms of the multiscale basis as

$$u_{ms}^H(\mathbf{x}) = \sum_{i=1}^n \phi_i^h(\mathbf{x}) \hat{u}_i^{ms} \quad (\text{A.6})$$

$$v_{ms}(\mathbf{x}) = \sum_{i=1}^n \phi_i^h(\mathbf{x}) \hat{v}_i^{ms} \quad (\text{A.7})$$

where \hat{u}_i^{ms} and \hat{v}_i^{ms} are corresponding discrete coefficients. When we substitute (A.7) and (A.6) into (9) and use the inner product defined in (11), we can find $u_{ms}^H(\mathbf{x})$ by solving

$$\mathbf{K}^{ms} \hat{\mathbf{u}}^{ms} = \mathbf{f}^{ms} \quad (\text{A.8})$$

where the multiscale stiffness matrix $\mathbf{K}^{ms} = (K_{ij}^{ms})$ and the multiscale force vector $\mathbf{f}^{ms} = (f_i^{ms})$ are defined as

$$K_{ij}^{ms} = \int_{\Omega} \nabla \phi_i^h(\mathbf{x}) \cdot a(\mathbf{x}) \nabla \phi_j^h(\mathbf{x}) \, d\Omega \quad (\text{A.9})$$

$$f_i^{ms} = \int_{\Omega} f(\mathbf{x}) \phi_i^h(\mathbf{x}) \, d\Omega \quad (\text{A.10})$$

When we substitute (A.5) into (A.9), we can compute the multiscale stiffness matrix K_{ij}^{ms} as

$$K_{ij}^{ms} = \sum_{k=1}^n \sum_{l=1}^n d_{ik} \int_{\Omega} \nabla N_{k,p}(\mathbf{x}) \cdot a(\mathbf{x}) \nabla N_{l,p}(\mathbf{x}) \, d\Omega \, d_{jl} \quad (\text{A.11})$$

Following the exposition in [26], we can compute the global stiffness matrix \mathbf{K}^{ms} by assembling stiffness matrices of the fine-scale mesh in n_e coarse-scale elements:

$$\mathbf{K}^{ms} = \bigwedge_{i=1}^{n_e} \mathbf{K}^{c,i} \quad \text{with} \quad \mathbf{K}^{c,i} = \mathbf{D}_i \mathbf{K}^{f,i} \mathbf{D}_i^T \quad (\text{A.12})$$

$$\text{and} \quad \mathbf{K}^{f,i} = \left(K_{kl}^{f,i} \right), \quad K_{kl}^{f,i} = \int_{\Omega_i} \nabla N_{k,p}(\mathbf{x}) \cdot a(\mathbf{x}) \nabla N_{l,p}(\mathbf{x}) \, d\Omega \quad (\text{A.13})$$

The matrix of coefficients $\mathbf{D} = (d_{ij})$ is constructed such that its i^{th} row consists of the fine-mesh representation of the i^{th} multiscale basis function and is the number of coarse-scale elements. We emphasize that the approach in (A.12) re-uses the fine-scale stiffness matrices $\mathbf{K}^{f,i}$ that have been previously computed for the determination of the multiscale basis functions via (A.4).

The coarse-scale element stiffness matrix $\mathbf{K}^{c,i}$ can also be computed directly from each fine-scale element without prior assembly of $\mathbf{K}^{f,i}$. To this end, we further split $\mathbf{K}^{c,i}$ in (A.12) into

$$\mathbf{K}^{c,i} = \mathbf{D}_i \mathbf{K}^{f,i} \mathbf{D}_i^T = \sum_{j=1}^{m_i} \mathbf{D}_{i,j} \mathbf{K}_j^{f,i} \mathbf{D}_{i,j}^T \quad (\text{A.14})$$

where m_i is the number of fine-scale elements in the i^{th} coarse-scale element, $\mathbf{K}_j^{f,i}$ is the local stiffness matrix of the j^{th} fine-scale element in the i^{th} coarse-scale element, and $\mathbf{D}_{i,j}$ is a matrix which contains the fine-scale mesh information corresponding to the j^{th} fine-scale element. We note that the approach (A.14) might have advantages over (A.12) in terms of parallelization on modern hybrid computing architectures.

We can apply the same approach to compute the global multiscale force vector \mathbf{f}^{ms} as

$$\mathbf{f}^{ms} = \bigwedge_{i=1}^{n_e} \mathbf{f}^{c,i} \text{ with } \mathbf{f}^{c,i} = \mathbf{D}_i \mathbf{f}^{f,i} \text{ or } \mathbf{f}^{c,i} = \sum_{j=1}^{m_i} \mathbf{D}_{i,j} \mathbf{f}_j^{f,i} \quad (\text{A.15})$$

$$\text{and } \mathbf{f}^{f,i} = \left(f_j^{f,i} \right), \quad f_j^{f,i} = \int_{\Omega_i} N_{j,p} f(\mathbf{x}) \, d\Omega \quad (\text{A.16})$$

The solution u_{ms}^H fully represents the fine-mesh solution of the $a(\mathbf{x})$ -harmonic part of the complete solution in each coarse-scale element. It can therefore be projected back onto the fine mesh, for instance for visualization purposes. With identities (A.5) to (A.6), we find

$$u_{ms}^H(\mathbf{x}) = \sum_{i=1}^{n_e} \sum_{j=1}^{m_i} N_{j,p}(\mathbf{x}) d_{ij} \hat{u}_i^{ms} = \sum_{j=1}^{m_i} N_{j,p}(\mathbf{x}) \left(\sum_{i=1}^{n_e} d_{ij} \hat{u}_i^{ms} \right) = \sum_{j=1}^{m_i} N_{j,p}(\mathbf{x}) \hat{u}_j \quad (\text{A.17})$$

$$\text{such that } \hat{\mathbf{u}} = (\hat{u}_i) = \mathbf{D}^T \hat{\mathbf{u}}^{ms} \quad (\text{A.18})$$

The coefficients \hat{u}_i represent the fine-mesh approximation of the multiscale solution and can be obtained by simply multiplying the matrix \mathbf{D}^T with coarse-scale coefficients. We recall that the final multiscale solution (12) requires the addition of the bubble part u_b .

Appendix B. Identifying corrector patterns on general unstructured meshes

Algorithm 1: Finding corrector patterns

Result: All nodes are assigned to a set of k patterns

```
1 Set pattern index  $k=1$ ;  
2 for  $i = 1 \rightarrow \text{numberOfNodes}$  do  
3   if Node  $i$  does not belong to an existing pattern then  
4     Create a new pattern  $k$ ;  
5     Node  $i$  belongs to pattern  $k$ ;  
6     for  $j = i+1 \rightarrow \text{numberOfNodes}$  do  
7       if node  $j$  has already assigned to an existing pattern then  
8         continue in next iteration;  
9       end  
10      if support of basis function at node  $j$  has no overlap with support of basis  
11        functions at any other node in pattern  $k$  then  
12          Node  $j$  belongs to pattern  $k$ ;  
13        end  
14      end  
15      Increment pattern index  $k++$ ;  
16 end
```

References

- [1] L.H. Nguyen and D. Schillinger. A multiscale predictor/corrector scheme for efficient elastoplastic voxel finite element analysis, with application to ct-based bone strength prediction. *Computer Methods in Applied Mechanics and Engineering*, 330:598 – 628, 2018.
- [2] R.O. Ritchie, M.J. Buehler, and P. Hansma. Plasticity and toughness in bone. *Physics Today*, 62(6):41–47, 2009.
- [3] J. Fish and N. Hu. Multiscale modeling of femur fracture. *International Journal for Numerical Methods in Engineering*, 111(1):3–25, 2017. nme.5450.
- [4] S. Ghosh, K. Lee, and S. Moorthy. Multiple scale analysis of heterogeneous elastic structures using homogenization theory and voronoi cell finite element method. *International Journal of Solids and Structures*, 32(1):27 – 62, 1995.
- [5] Z. Liu, M.A. Bessa, and W.K. Liu. Self-consistent clustering analysis: An efficient multi-scale scheme for inelastic heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 306:319 – 341, 2016.
- [6] W.C. Sun, J.E. Andrade, and J.W. Rudnicki. Multiscale method for characterization of porous microstructures and their impact on macroscopic effective permeability. *International Journal for Numerical Methods in Engineering*, 88(12):1260–1279, 2011.
- [7] W. Sun, M.R. Kuhn, and J.W. Rudnicki. A multiscale DEM-LBM analysis on permeability evolutions inside a dilatant shear band. *Acta Geotechnica*, 8(5):465–480, 2013.
- [8] K. Wang and W. Sun. A semi-implicit discrete-continuum coupling method for porous media based on the effective stress principle at finite strain. *Computer Methods in Applied Mechanics and Engineering*, 304:546 – 583, 2016.
- [9] J.C. Michel, H. Moulinec, and P. Suquet. Effective properties of composite materials with periodic microstructure: a computational approach. *Computer Methods in Applied Mechanics and Engineering*, 172(1):109 – 143, 1999.
- [10] F. Feyel and J.L. Chaboche. FE² multiscale approach for modelling the elastoviscoplastic behaviour of long fibre SiC/Ti composite materials. *Computer Methods in Applied Mechanics and Engineering*, 183(3):309 – 330, 2000.
- [11] V. Kouznetsova, W.A.M. Brekelmans, and F.P.T. Baaijens. An approach to micro-macro modeling of heterogeneous materials. *Computational Mechanics*, 27(1):37–48, 2001.
- [12] J.C. Michel and P. Suquet. Nonuniform transformation field analysis. *International Journal of Solids and Structures*, 40(25):6937 – 6955, 2003.
- [13] C. Oskay and J. Fish. Eigendeformation-based reduced order homogenization for failure analysis of heterogeneous materials. *Computer Methods in Applied Mechanics and Engineering*, 196(7):1216 – 1243, 2007.
- [14] J. Yvonnet and Q.-C. He. The reduced model multiscale method (R3M) for the non-linear homogenization of hyperelastic media at finite strains. *Journal of Computational Physics*, 223(1):341–368, 2007.
- [15] J. Fish and R. Fan. Mathematical homogenization of nonperiodic heterogeneous media subjected to large deformation transient loading. *International Journal for Numerical Methods in Engineering*, 76(7):1044–1064, 2008.
- [16] C. Miehe, J. Dettmar, and D. Zäh. Homogenization and two-scale simulations of granular materials for different microstructural constraints. *International Journal for Numerical Methods in Engineering*, 83:1206–1236, 2010.
- [17] V.P. Nguyen, M. Stroeven, and L.J. Sluys. Multiscale continuous and discontinuous modeling of heterogeneous materials: A review on recent developments. *Journal of Multiscale Modelling*, 03(04):229–270, 2011.
- [18] J. Fish. *Practical multiscaling*. John Wiley & Sons, 2013.
- [19] V.D. Nguyen, G. Becker, and L. Noels. Multiscale computational homogenization methods with a gradient enhanced scheme based on the discontinuous galerkin formulation. *Computer Methods in Applied Mechanics and Engineering*, 260:63 – 77, 2013.
- [20] M. Mosby and K. Matouš. Computational homogenization at extreme scales. *Extreme Mechanics Letters*, 6:68 – 74, 2016.
- [21] R. Biswas and L.H. Poh. A micromorphic computational homogenization framework for heterogeneous materials. *Journal of the Mechanics and Physics of Solids*, 102:187 – 208, 2017.
- [22] T.I. Zohdi. Homogenization methods and multiscale modeling. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–24, 2017.

[23] M.G.D. Geers, V.G. Kouznetsova, K. Matouš, and J. Yvonnet. Homogenization methods and multiscale modeling: Nonlinear problems. *Encyclopedia of Computational Mechanics Second Edition*, pages 1–34, 2017.

[24] T. Hettich, A. Hund, and E. Ramm. Modeling of failure in composites by X-FEM and level sets within a multiscale framework. *Computer Methods in Applied Mechanics and Engineering*, 197(5):414–424, 2008.

[25] M.G.D. Geers, V.G. Kouznetsova, and W.A.M. Brekelmans. Multi-scale computational homogenization: Trends and challenges. *Journal of Computational and Applied Mathematics*, 234(7):2175–2182, 2010.

[26] Y. Efendiev and T.Y. Hou. *Multiscale finite element methods: theory and applications*. Springer, 2009.

[27] I. Babuška and J.E. Osborn. Can a finite element method perform arbitrarily badly? *Mathematics of Computation*, 69(230):443–462, 2000.

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

[29] T.J.R. Hughes, G.R. Feijóo, L. Mazzei, and J.-B. Quincy. The variational multiscale method – a paradigm for computational mechanics. *Computer Methods in Applied Mechanics and Engineering*, 166(1):3–24, 1998.

[30] T.J.R. Hughes, L. Mazzei, and K.E. Jansen. Large eddy simulation and the variational multiscale method. *Computing and Visualization in Science*, 3(1-2):47–59, 2000.

[31] C. Farhat, I. Harari, and L.P. Franca. The discontinuous enrichment method. *Computer Methods in Applied Mechanics and Engineering*, 190(48):6455 – 6479, 2001.

[32] J. Fish and Z. Yuan. Multiscale enrichment based on partition of unity. *International Journal for Numerical Methods in Engineering*, 62(10):1341–1359, 2005.

[33] J. Fish and Z. Yuan. Multiscale enrichment based on partition of unity for nonperiodic fields and nonlinear problems. *Computational Mechanics*, 40(2):249–259, Jul 2007.

[34] S. Löhnert and T. Belytschko. A multiscale projection method for macro/microcrack simulations. *International Journal for Numerical Methods in Engineering*, 71(12):1466–1482, 2007.

[35] K. Garikipati and T.J.R. Hughes. A variational multiscale approach to strain localization–formulation for multidimensional problems. *Computer Methods in Applied Mechanics and Engineering*, 188(1):39–60, 2000.

[36] A. Hund and E. Ramm. Locality constraints within multiscale model for non-linear material behaviour. *International Journal for Numerical Methods in Engineering*, 70(13):1613–1632, 2007.

[37] J. Mergheim. A variational multiscale method to model crack propagation at finite strains. *International Journal for Numerical Methods in Engineering*, 80(3):269–289, 2009.

[38] T.Y. Hou and P. Liu. Optimal local multi-scale basis functions for linear elliptic equations with rough coefficients. *Discrete and Continuous Dynamical Systems*, 36(8):4451–4476, 2016.

[39] T.Y. Hou, F.N. Hwang, P. Liu, and C.C. Yao. An iteratively adaptive multi-scale finite element method for elliptic pdes with rough coefficients. *Journal of Computational Physics*, 336:375 – 400, 2017.

[40] Y. Efendiev, J. Galvis, and T.Y. Hou. Generalized multiscale finite element methods (GMsFEM). *Journal of Computational Physics*, 251:116–135, 2013.

[41] E.T. Chung, Y. Efendiev, and W.T. Leung. Residual-driven online generalized multiscale finite element methods. *Journal of Computational Physics*, 302:176 – 190, 2015.

[42] T.Y. Hou, X.H. Wu, and Z. Cai. Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients. *Mathematics of Computation*, 68:913–943, 1999.

[43] Y. Efendiev, V. Ginting, T. Hou, and R. Ewing. Accurate multiscale finite element methods for two-phase flow simulations. *Journal of Computational Physics*, 220(1):155 – 174, 2006.

[44] L.J. Durlofsky, Y. Efendiev, and V. Ginting. An adaptive local-global multiscale finite volume element method for two-phase flow simulations. *Advances in Water Resources*, 30(3):576 – 588, 2007.

[45] H. Hajibeygi, G. Bonfigli, M.A. Hesse, and P. Jenny. Iterative multiscale finite-volume method. *Journal of Computational Physics*, 227(19):8604 – 8621, 2008.

[46] P. Henning and D. Peterseim. Oversampling for the multiscale finite element method. *Multiscale Modeling & Simulation*, 11(4):1149–1175, 2013.

[47] N. Castelletto, H. Hajibeygi, and H.A. Tchelepi. Multiscale finite-element method for linear elastic geomechanics. *Journal of Computational Physics*, 331:337 – 356, 2017.

[48] N. Zhang, B. Yan, Q. Sun, and Y. Wang. Improving multiscale mixed finite element method for flow simulation in highly heterogeneous reservoir using adaptivity. *Journal of Petroleum Science and Engineering*, 154:382 –

388, 2017.

- [49] N. Zhang, Y. Wang, Y. Wang, B. Yan, and Q. Sun. A locally conservative multiscale finite element method for multiphase flow simulation through heterogeneous and fractured porous media. *Journal of Computational and Applied Mathematics*, 2018.
- [50] K. Gao, S. Fu, and E.T. Chung. A high-order multiscale finite-element method for time-domain acoustic-wave modeling. *Journal of Computational Physics*, 360:120 – 136, 2018.
- [51] N.C. Nguyen. A multiscale reduced-basis method for parametrized elliptic partial differential equations with multiple scales. *Journal of Computational Physics*, 227(23):9807–9822, 2008.
- [52] J.S. Hesthaven, S. Zhang, and X. Zhu. Reduced basis multiscale finite element methods for elliptic problems. *Multiscale Modeling & Simulation*, 13(1):316–337, 2015.
- [53] V.M. Calo, Y. Efendiev, J. Galvis, and G. Li. Randomized oversampling for generalized multiscale finite element methods. *Multiscale Modeling & Simulation*, 14(1):482–501, 2016.
- [54] Y.R. Efendiev, T.Y. Hou, and X.H. Wu. Convergence of a nonconforming multiscale finite element method. *SIAM Journal on Numerical Analysis*, 37(3):888–910, 2000.
- [55] E.T. Chung, Y. Efendiev, and G. Li. An adaptive GMsFEM for high-contrast flow problems. *Journal of Computational Physics*, 273:54 – 76, 2014.
- [56] H.W. Zhang, J.K. Wu, J. Lü, and Z.D. Fu. Extended multiscale finite element method for mechanical analysis of heterogeneous materials. *Acta Mechanica Sinica*, 26(6):899–920, 2010.
- [57] D. Schillinger and 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):391–455, 2015.
- [58] E. Burman, S. Claus, P. Hansbo, M.-G. Larson, and A. Massing. CutFEM: discretizing geometry and partial differential equations. *International Journal for Numerical Methods in Engineering*, 104(7):472–501, 2015.
- [59] A. Hansbo and P. Hansbo. An unfitted finite element method, based on Nitsche’s method, for elliptic interface problems. *Computer Methods in Applied Mechanics and Engineering*, 191:537–552, 2002.
- [60] A. Embar, J. Dolbow, and I. Harari. Imposing Dirichlet boundary conditions with Nitsche’s method and spline-based finite elements. *International Journal for Numerical Methods in Engineering*, 83:877–898, 2010.
- [61] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: a stabilized Lagrange multiplier method. *Computer Methods in Applied Mechanics and Engineering*, 62(4):2680–2686, 2010.
- [62] E. Burman and P. Hansbo. Fictitious domain finite element methods using cut elements: a stabilized Nitsche method. *Applied Numerical Mathematics*, 62(4):328–341, 2012.
- [63] D. Schillinger, I. Harari, M.-C. Hsu, D. Kamensky, K.F.S. Stoter, Y. Yu, and Z. Ying. The non-symmetric Nitsche method for the parameter-free imposition of weak boundary and coupling conditions in immersed finite elements. *Computer Methods in Applied Mechanics and Engineering*, 309:625–652, 2016.
- [64] L.H. Nguyen, S.K.F. Stoter, M. Ruess, M.A. Uribe Sanchez, and D. Schillinger. The diffuse Nitsche method: Dirichlet constraints on phase-field boundaries. *International Journal for Numerical Methods in Engineering*, doi:10.1002/nme.5628, 2017.
- [65] A. Massing, B. Schott, and W.A. Wall. A stabilized Nitsche cut finite element method for the Oseen problem. *Computer Methods in Applied Mechanics and Engineering*, 328:262–300, 2018.
- [66] D. Schillinger, M. Ruess, N. Zander, Y. Bazilevs, A. Düster, and E. Rank. Small and large deformation analysis with the p - and B-spline versions of the Finite Cell Method. *Computational Mechanics*, 50(4):445–478, 2012.
- [67] D. Schillinger, L. Dede’, M.A. Scott, J.A. Evans, M.J. Borden, E. Rank, and T.J.R. Hughes. An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces. *Computer Methods in Applied Mechanics and Engineering*, 249-250:116–150, 2012.
- [68] T.J.R. Hughes. Multiscale phenomena: Green’s functions, the Dirichlet-to–Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. *Computer methods in applied mechanics and engineering*, 127(1-4):387–401, 1995.
- [69] Y. Bazilevs, V.M. Calo, J. A. Cottrell, T.J.R. Hughes, A. Reali, and G. Scovazzi. Variational multiscale residual-based turbulence modeling for large eddy simulation of incompressible flows. *Computer Methods in Applied Mechanics and Engineering*, 197:173–201, 2007.

[70] V.M. Calo, E.T. Chung, Y. Efendiev, and W.T. Leung. Multiscale stabilization for convection-dominated diffusion in heterogeneous media. *Computer Methods in Applied Mechanics and Engineering*, 304:359–377, 2016.

[71] T. Gangwar, J. Calder, T. Takahashi, J.E. Bechtold, and D. Schillinger. Robust variational segmentation of 3d bone ct data with thin cartilage interfaces. *Medical Image Analysis*, 47:95–110, 2018.

[72] P. Müller and P. Rüegsegger. Three-dimensional finite element modelling of non-invasively assessed trabecular bone structures. *Medical Engineering & Physics*, 17(2):126 – 133, 1995.

[73] B. van Rietbergen, H. Weinans, R. Huiskes, and A. Odgaard. A new method to determine trabecular bone elastic properties and loading using micromechanical finite-element models. *Journal of Biomechanics*, 28(1):69 – 81, 1995.

[74] J.H. Keyak, S.A. Rossi, K.A. Jones, and H.B. Skinner. Prediction of femoral fracture load using automated finite element modeling. *Journal of Biomechanics*, 31(2):125–133, 1997.

[75] M. Viceconti, M. Davinelli, F. Taddei, and A. Cappello. Automatic generation of accurate subject-specific bone finite element models to be used in clinical studies. *Journal of Biomechanics*, 37(10):1597–1605, 2004.

[76] E. Schileo, F. Taddei, A. Malandrino, L. Cristofolini, and M. Viceconti. Subject-specific finite element models can accurately predict strain levels in long bones. *Journal of Biomechanics*, 40(13):2982–2989, 2007.

[77] L.H. Nguyen, S.K.F. Stoter, T. Baum, J.S. Kirschke, M. Ruess, Z. Yosibash, and D. Schillinger. Phase-field boundary conditions for the voxel finite cell method: Surface-free stress analysis of CT-based bone structures. *International Journal for Numerical Methods in Biomedical Engineering*, doi:10.1002/cnm.2880, 2017.

[78] C. Hellmich, C. Kober, and B. Erdmann. Micromechanics-based conversion of ct data into anisotropic elasticity tensors, applied to fe simulations of a mandible. *Annals of Biomedical Engineering*, 36(1):108–122, 2007.

[79] W. Pistoia, B. Van Rietbergen, E.-M. Lochmüller, C.A. Lill, F. Eckstein, and P. Rüegsegger. Estimation of distal radius failure load with micro-finite element analysis models based on three-dimensional peripheral quantitative computed tomography images. *Bone*, 30(6):842–848, 2002.

[80] T.L. Mueller, D. Christen, S. Sandercott, S.K. Boyd, B. van Rietbergen, F. Eckstein, E.-M. Lochmüller, R. Müller, and G.H. van Lenthe. Computational finite element bone mechanics accurately predicts mechanical competence in the human radius of an elderly population. *Bone*, 48(6):1232 – 1238, 2011.

[81] A.J. Wirth, R. Müller, and G.H. van Lenthe. The discrete nature of trabecular bone microarchitecture affects implant stability. *Journal of Biomechanics*, 45(6):1060 – 1067, 2012.

[82] R. Blanchard, C. Morin, A. Malandrino, A. Vella, Z. Sant, and C. Hellmich. Patient-specific fracture risk assessment of vertebrae: A multiscale approach coupling X-ray physics and continuum micromechanics. *International Journal for Numerical Methods in Biomedical Engineering*, doi:10.1002/cnm.2760, 2016.

[83] H.H. Bayraktar, E.F. Morgan, G.L. Niebur, G.E. Morris, E.K. Wong, and T.M. Keaveny. Comparison of the elastic and yield properties of human femoral trabecular and cortical bone tissue. *Journal of Biomechanics*, 37(1):27 – 35, 2004.

[84] P.H.F. Nicholson, X.G. Cheng, G. Lowet, S. Boonen, M.W.J. Davie, J. Dequeker, and G. Van der Perre. Structural and material mechanical properties of human vertebral cancellous bone. *Medical Engineering & Physics*, 19(8):729–737, 1997.

[85] Z. Yang, M. Ruess, S. Kollmannsberger, A. Düster, and E. Rank. An efficient integration technique for the voxel-based finite cell method. *International Journal for Numerical Methods in Engineering*, 91:457–471, 2012.

[86] M. Ruess, D. Tal, N. Trabelsi, Z. Yosibash, and E. Rank. The finite cell method for bone simulations: Verification and validation. *Biomechanics and Modeling in Mechanobiology*, 11(3):425–437, 2012.