



A multiscale method for nonlocal mechanics and diffusion and for the approximation of discontinuous functions

Feifei Xu^{*}, Max Gunzburger, John Burkardt

Department of Scientific Computing, Florida State University, Tallahassee, FL, 32306, United States

Received 16 October 2015; received in revised form 14 April 2016; accepted 18 April 2016

Available online 29 April 2016

Abstract

A multiscale implementation of hybrid continuous/discontinuous finite element discretizations of nonlocal models for mechanics and diffusion in two dimensions is developed. The implementation features adaptive mesh refinement based on the detection of defects and results in an abrupt transition between refined elements that contain defects and unrefined elements free of defects. An additional difficulty overcome in the implementation is the design of accurate quadrature rules for stiffness matrix construction that are valid for any combination of the grid size and horizon parameter, the latter being the extent of nonlocal interactions. As a result, the methodology developed can attain optimal accuracy at very modest additional costs relative to situations for which the solution is smooth. Portions of the methodology can also be used for the optimal approximation, by piecewise linear polynomials, of given functions containing discontinuities. Several numerical examples are provided to illustrate the efficacy of the multiscale methodology.

© 2016 Elsevier B.V. All rights reserved.

Keywords: Peridynamics; Multiscale methods; Nonlocal models; Anomalous diffusion; Finite element methods; Discontinuous displacements

1. Introduction

Classical partial differential mechanics models for solid mechanics feature local interactions, i.e., a point only interacts with points within an infinitesimal distance. In contrast, in the peridynamics (PD) model, points interact with points within a finite influence horizon δ . This nonlocal approach has significant advantages for studying defects such as the nucleation and propagation of cracks [1–6], problems for which the classical approach breaks down because the necessary derivatives do not exist.

The peridynamics model admits a variational formulation that in turn suggests the construction of a discretized model via a finite element method (FEM). For problems with smooth data, FEMs have well-understood convergence behavior with respect to the typical element size h . This allows for the efficient solution of complex systems. However,

^{*} Corresponding author.

E-mail addresses: fx11@my.fsu.edu (F. Xu), mgunzburger@fsu.edu (M. Gunzburger), jburkardt@fsu.edu (J. Burkardt).

in cases in which discontinuities are expected, the rapid convergence of FEMs may be lost unless suitable precautions are taken. Such precautions might involve careful adjustments of the mesh or the use of a discontinuous basis functions. For these remedies to help, it must also be possible to determine, to reasonable accuracy, the location and extent of the discontinuities.

In [7], a one-dimensional PD model was considered, using a piecewise linear polynomial basis for which a problem with smooth data would typically have an L^2 convergence rate of $O(h^2)$. When discontinuous solutions were considered, the convergence rate deteriorated to $O(h^{1/2})$. However, the optimal rate was recovered by adjusting the mesh to bracket the discontinuity by a single tiny element whose characteristic length was $O(h^4)$, whereas the remaining mesh was essentially unchanged. In [7], the locations of the discontinuities were assumed to be explicitly available to the algorithm; this made it easy to tailor the mesh or element choice in the optimal way. In practical problems, however, the detection of discontinuities and the estimation of their extent and severity are not simple tasks. An adaptive approach must be taken which is able to repeatedly adjust the size, shape, and element type of a local portion of the mesh, based on some computable numerical error indicators.

Whereas some regions of a solid may contain singularities, there may also exist large regions of smoothness and regularity, a behavior that is well and efficiently handled by classical partial differential equations (PDEs) approaches. A multiscale implementation of the peridynamics model in one dimension was developed in [8]; the solution interval was divided according to the observed behavior, and then a PDE model was applied in subintervals in which the displacement was detected to be smooth, and an PD model was applied only in the vicinity of discontinuities of the displacement. The application of the PD model was further divided into the use of discontinuous Galerkin (DG) discretizations in the elements containing the discontinuity and continuous Galerkin (CG) discretization in immediately neighboring elements. Thus, elements of the PD-CG type would form an intermediate layer between the PDE regions of smoothness and the PD-DG regions of sharp local discontinuity. The flexibility rendered by using multiple models in this way requires, however, a strategy for correctly coupling them over transition regions.

The work in [7] and [8] was restricted to the one-dimensional case. It is the purpose of this paper to consider the issues involved in implementing a multiscale PD model in two-dimensional regions. We assume that there are some curves across which the displacement is discontinuous and which are separated by relative large regions within which the displacement is smooth. The goal is to implement a finite element discretization that accurately and efficiently approximates the solution with, if possible, a convergence rate that is comparable to the optimal convergence rate observed when there are no discontinuities present.

As often occurs when moving from one to two dimensions, the proper treatment of the geometry becomes significantly more difficult. The most obvious change is that adaptive remeshing becomes much more complicated, and requires attention to element shape (no small angles) and element connectivity (no hanging nodes).

Another geometric issue involves the treatment of discontinuities. In one dimension, a local discontinuity occurs at a single point, and isolating that point just requires determining a very small element that contains it. A discontinuity in two dimensions might, however, constitute a point, a curve intersecting the boundary, or a closed curve contained within the region. Depending on the geometric complexity and curvature of the discontinuity, the technique of “covering” by a very small element may become unfeasible.

A third geometric issue arises because the PD model seeks to integrate interactions over a local circular region defined by the horizon δ . An FEM model approximates such integrals using numerical quadrature over the collection of triangles that form the mesh. However, an effect of the horizon is to render the integrand not smooth across the horizon circumference, resulting in possibly disastrous failures of standard quadrature methods in triangles that intersect the horizon circumference.

In this paper, we consider how to handle these and several other obstacles. Our goal is to implement a multiscale PD finite element method in two dimensions for a problem in which discontinuities are likely to occur but whose locations are not known. We concentrate on the response necessary for a multiscale PD implementation to adapt to the discovery of a discontinuity. This response will include local refinement, remodeling, remeshing, new quadrature rules, and a seamless transition from the nonlocal PD model to a local PDE one. Our goal is an accurate and efficient computation of approximate solution despite the presence of discontinuities.

Clarification about what is meant by *multiscale* is perhaps needed. In this paper, multiscale features of solutions, i.e., large regions over which solutions are smooth vs. small regions over which solutions suffer defects, are resolved by using grid sizes that are large (small) with respect to the horizon for the former (latter). Thus, our approach is

multiscale in nature both as a discretization algorithm and as a means for resolving physical phenomena that occur at different scales.

Although we focus on solid mechanics problems, much of the multiscale methodology we develop also applies to two other settings. First, it applies to nonlocal models for diffusion because they have a similar structure to that of PD models. Second, a large portion of the multiscale methodology applies directly to the problem of interpolating given functions that have jump discontinuities.

The paper is organized as follows. The formulation of the nonlocal problem is given in Section 2. The local grid refinement method is explained in Section 3. Numerical quadrature issues are discussed in Section 4. The coupling between local and nonlocal models is discussed in Section 5 and the multiscale implementation strategy is outlined in Section 6. Numerical results for the interpolation of functions are given in Section 7 and for the PD setting in Section 8. Finally, in Section 9, remarks are provided about the current work and plans for future work.

2. Nonlocal model problem

For the sake of simplicity, we consider a two-dimensional nonlocal model for a scalar-valued function. Peridynamic (PD) models are similar except that the solution is a vector-valued function; however, in every significant respect, the discussions in this work apply to the PD setting.

The action of the operator $\mathcal{L} : \mathbb{R}^n \rightarrow \mathbb{R}$ on $u \in \mathbb{R}$ is defined as

$$\mathcal{L}u(\mathbf{x}) = -2 \int_{\mathbb{R}^n} (u(\mathbf{x}') - u(\mathbf{x}))\gamma(\mathbf{x}, \mathbf{x}') d\mathbf{x}' \quad \forall \mathbf{x} \in \mathbb{R}^n,$$

where γ denotes a symmetric kernel, i.e., $\gamma(\mathbf{x}, \mathbf{x}') = \gamma(\mathbf{x}', \mathbf{x})$ for all \mathbf{x}, \mathbf{x}' . The *interaction domain* corresponding to a given open subset $\Omega \subset \mathbb{R}^n$ is defined as

$$\Omega_{\mathcal{I}} = \{\mathbf{x}' \in \mathbb{R}^n \setminus \Omega : \gamma(\mathbf{x}, \mathbf{x}') \neq 0 \text{ for some } \mathbf{x} \in \Omega\}, \tag{1}$$

i.e., $\Omega_{\mathcal{I}}$ consists of all points in \mathbb{R}^n that interact with points in Ω but are located outside of Ω . In general, $\Omega_{\mathcal{I}}$ has nonzero volume in \mathbb{R}^n .

With these notations, the *nonlocal volume-constrained problem*¹ we consider is given by

$$\begin{cases} \mathcal{L}u(\mathbf{x}) = -2 \int_{\Omega \cup \Omega_{\mathcal{I}}} (u(\mathbf{x}') - u(\mathbf{x}))\gamma(\mathbf{x}, \mathbf{x}') d\mathbf{x}' = f(\mathbf{x}) & \forall \mathbf{x} \in \Omega \\ u(\mathbf{x}) = g(\mathbf{x}) & \forall \mathbf{x} \in \Omega_{\mathcal{I}}, \end{cases} \tag{2}$$

where $f(\mathbf{x}) : \Omega \rightarrow \mathbb{R}$ and $g(\mathbf{x}) : \Omega_{\mathcal{I}} \rightarrow \mathbb{R}$ are given functions. Note that the constraint $u = g$ is applied on the set $\Omega_{\mathcal{I}}$ having finite volume and not on the boundary of Ω ; hence, the terminology “volume-constrained problem”.

We define the bilinear form

$$A(u, v) = \int_{\Omega \cup \Omega_{\mathcal{I}}} \int_{\Omega \cup \Omega_{\mathcal{I}}} (u(\mathbf{x}') - u(\mathbf{x}))(v(\mathbf{x}') - v(\mathbf{x}))\gamma(\mathbf{x}, \mathbf{x}') d\mathbf{x}' d\mathbf{x}, \tag{3}$$

the “energy” norm $\|u\| = (A(u, u))^{1/2}$, the energy space

$$V(\Omega \cup \Omega_{\mathcal{I}}) = \{u \in L^2(\Omega \cup \Omega_{\mathcal{I}}) : \|u\| < \infty\},$$

and the constrained energy space

$$V_c(\Omega \cup \Omega_{\mathcal{I}}) = \{v \in V : v(\mathbf{x}) = 0 \quad \forall \mathbf{x} \in \Omega_{\mathcal{I}}\}.$$

¹ Because this paper deals with problems in two dimensions, we might more appropriately use the terminology *area*-constrained problems. However, in the nonlocal modeling community, it is standard practice for constraints of this type to be referred as volume constraints, regardless of dimension, so we do so as well here and use “volume” instead of “area” in other instances as well.

It is shown in, e.g., [9,10], that (2) has the variational formulation

$$\begin{cases} \text{given } f \in L^2(\Omega) \text{ and } g \in L^2(\Omega_T), \text{ seek } u \in V(\Omega \cup \Omega_T) \text{ such that} \\ u(\mathbf{x}) = g(\mathbf{x}) \quad \forall \mathbf{x} \in \Omega_T \text{ and} \\ A(u, v) = \int_{\Omega} f v \, d\mathbf{x} \quad \forall v \in V_c(\Omega \cup \Omega_T) \end{cases} \quad (4)$$

and that the problem (4) is well posed with respect to the energy space $V(\Omega \cup \Omega_T)$.

We are interested in localized kernels, that is, we have, for some $\delta > 0$, referred to as the *horizon*,

$$\gamma(\mathbf{x}, \mathbf{x}') = 0 \quad \text{whenever } \mathbf{x} \notin H_{\delta}(\mathbf{x}), \quad (5)$$

where $H_{\delta}(\mathbf{x}) = \{\mathbf{x}' \in \mathbb{R}^n : |\mathbf{x}' - \mathbf{x}| < \delta\}$. In this case, we have that the interaction domain is given by $\Omega_T = \{\mathbf{x}' \in \mathbb{R}^n \setminus \Omega : |\mathbf{x}' - \mathbf{x}| < \delta \text{ for } \mathbf{x} \in \Omega\}$.

Many kernels, integrable or non-integrable and singular or non-singular, have been proposed in the literature; see, e.g., [9,10]. For simplicity, we only consider nonsingular localized kernels that are integrable, i.e., for some constant $c^*(\delta) > 0$ whose value depends on δ ,

$$\int_{\mathbb{R}^n} \gamma(\mathbf{x}, \mathbf{x}') d\mathbf{x}' = \int_{H_{\delta}(\mathbf{x})} \gamma(\mathbf{x}, \mathbf{x}') d\mathbf{x}' \leq c^*(\delta) < \infty \quad \forall \mathbf{x} \in \Omega. \quad (6)$$

Again, in every significant aspect, the discussions in this work apply to more general kernels.

3. Local, abrupt mesh refinement in two dimensions

When a discontinuity in a function has been detected, the multiscale implementation responds by carrying out a local adaptive abrupt mesh refinement. For two-dimensional problems, the geometry or shape of the discontinuity zone becomes an issue that, by itself, can present severe challenges to any modeling approach. In general, we restrict our consideration to cases in which discontinuities occur across curves that are piecewise smooth and connected and that either form a closed loop or begin and end at distinct points on the boundary. Additional cases, e.g., discontinuities occurring on curves beginning and ending at different points in the interior, can be handled in a similar manner. In this section, for the sake of simplicity of exposition, we further restrict attention to settings in which discontinuities occur across a single curve which splits the domain into two subregions; the discussion directly applies to settings with multiple discontinuity curves.

Suppose that the region Ω has been subdivided into an initial mesh with a characteristic mesh parameter, i.e., grid length, h which is nominally chosen so that a desired accuracy would be achieved were solutions to be smooth everywhere in Ω . Assume that there is some indicator, such as a posterior error estimator, which can be used to determine which elements contain a portion of the discontinuity curve. Our local mesh refinement strategy adjusts the initial mesh so that the discontinuity curve is, in a sense that will be explained later, contained within a collection of long thin elements, whereas the remainder of the region is still covered by elements of regular shape having characteristic size h . Thus, there is an abrupt transition between regularly-shaped and thin-shaped elements, rather than the more common approach of having a substantial layer of elements that gradually decrease in size.

The refinement strategy we use defines an indicator function that allows us to identify the set of “discontinuous elements”, each of which is traversed by the discontinuity curve. Each discontinuous element may then be split into four small triangles. It is likely that the discontinuity does not affect one or more of the four subelements, which we may then regard as “continuous elements”. Subsequently, the discontinuous elements are repeatedly split until the discontinuous elements reach a size of $O(\epsilon)$ for some prescribed $\epsilon > 0$. As a result, one has defined a subregion in which refinement has occurred and which consists of many small elements of size $O(\epsilon)$ that cover the discontinuity, and, in the rest of Ω , a collection of elements of various sizes that do not contain any part of the discontinuity. The discontinuous elements may be merged into long quadrilaterals of width $O(\epsilon)$ and length $O(h)$ and then split to form triangles. The area covered by the continuous elements can then be newly remeshed using, e.g., a constrained centroidal Voronoi tessellation [11–13].

The mesh refinement algorithm is summarized as follows.

1. *Split*: subdivide discontinuous elements until their size reduces to $O(\epsilon)$; along the way, continuous subelements are set aside.

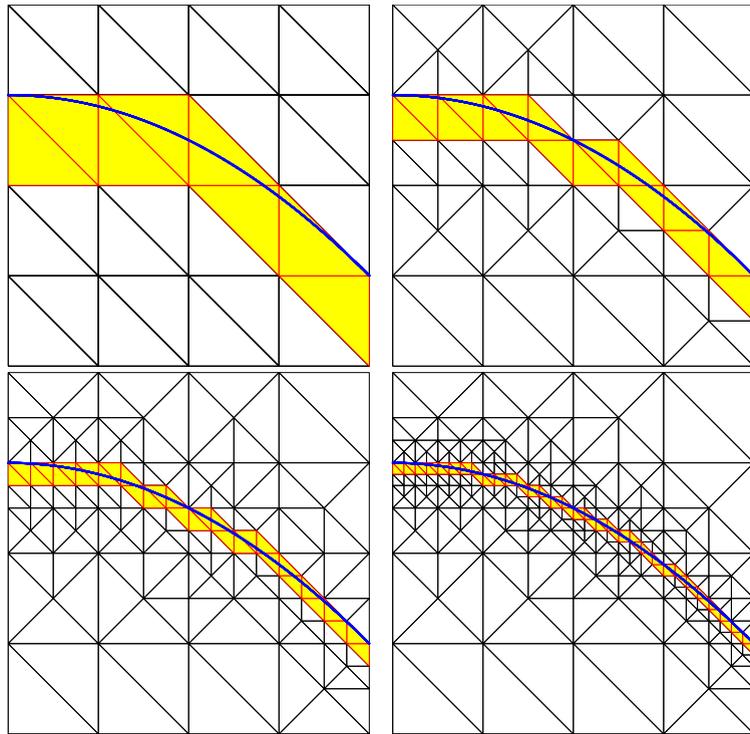


Fig. 1. Illustration of the local splitting process. Left to right and top to bottom: initial mesh and meshes after first, second, and third splittings. In all cases, discontinuous elements are identified in yellow. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

2. *Merge*: merge the tiny discontinuous subelements into long thin quadrilaterals of width $O(\epsilon)$ and length $O(h)$ then split each quadrilateral into two long thin triangles.
3. *Remesh*: remesh subregions that do not contain discontinuous elements into quasi-uniform meshes of grid size $O(h)$.

Fig. 1 illustrates the splitting step in which discontinuous elements (the yellow triangles in the top-left figure), are subdivided into four smaller triangles (the top-right figure). Note that immediately adjacent elements may also need to be modified, to avoid hanging nodes. The nodes of the subdivided elements are reexamined to identify which smaller triangles contain the discontinuity curve (the yellow triangles in the top-right figure) and then those elements are split again, this process being repeated until the discontinuous elements have all been reduced to the desired size of ϵ . The bottom figures illustrate the results of two further splittings.

Fig. 2 illustrates the merging step in which the tiny elements of size $O(\epsilon)$ are merged to create long quadrilateral elements of length $O(h)$ and width $O(\epsilon)$. For general discontinuity curve shapes and very small ϵ (relative to h), it may not be possible to cover the discontinuity curve by the set of long thin quadrilaterals. We are content to simply ensure that the discontinuity curve does intersect with the two short $O(\epsilon)$ sides of each quadrilateral; see Fig. 3 for an illustration. Further discussion of these issues, including how ϵ is chosen, is given in Section 6. Note that the merging step may not be possible in high curvature regions of the discontinuity curve, in which case the long side of the quadrilateral may have to be shorter than $O(h)$ or at corners in the discontinuity curve which would have to remain covered by a regularly shaped element of size $O(\epsilon)$.

Fig. 4 illustrates the remeshing step for the subregions outside the set of discontinuous elements as determined during the merging step. This region consists of two subregions, one on either side of the discontinuity curve, which are separately remeshed. We first restore the grid points of the original uniform mesh that belong to a subregion; to these nodes are added the nodes of the long thin elements that also belong to the subregion; see the left plot in Fig. 4. Then, a subregion is remeshed by adjusting the grid points so that a more regular mesh is produced; see the right plot in Fig. 4. One such means for doing this is to construct a constrained centroidal Voronoi tessellation (CCVT) [11–13]

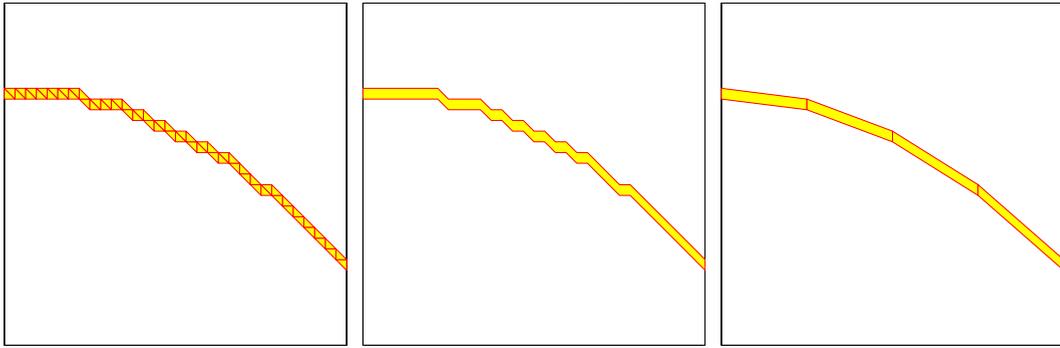


Fig. 2. Illustration of the merging of small isotropic elements containing the discontinuity curve into longer thin quadrilaterals.

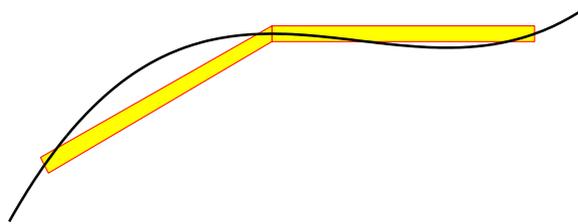


Fig. 3. Illustration of the discontinuity curve passing through the short sides of the thin quadrilaterals.

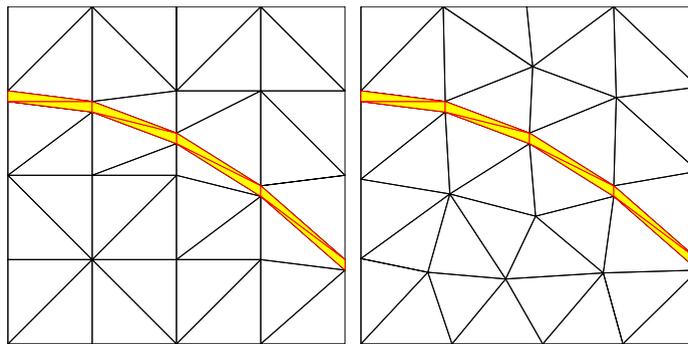


Fig. 4. Illustration of the remeshing of the white regions which do not include discontinuous elements. Left: initial mesh constructed after the merging step. Right: mesh after CCVT remeshing.

in which the nodes on the thin triangles are held fixed. Depending on the situation, some grid points on the boundary of the patch could be allowed to slide slightly along the boundary to automatically improve mesh quality, although our default choice of holding those grid points fixed often works in practice. In Fig. 4, the first image shows the state of the mesh at the beginning of the remeshing process. The second image is the result after the CCVT process and shows how the elements other than the thin elements have nearly recovered the optimal size and shape they had before the discontinuity was discovered and separately handled. In the right-plot of Fig. 4, note the abrupt transition from the set of thin elements to the set of regularly shaped elements.

Quite apart from solving nonlocal problems, this refinement method may be suitable for the interpolation of discontinuous functions. In Section 7, we consider the error for the case of piecewise linear interpolation and compare results with and without mesh refinement. The results provided in that section suggest that this approach can yield near optimal convergence rates in spite of the discontinuity. In Section 8, we present a similar study of the refinement method in the context of the multiscale implementation of the nonlocal problem (2).

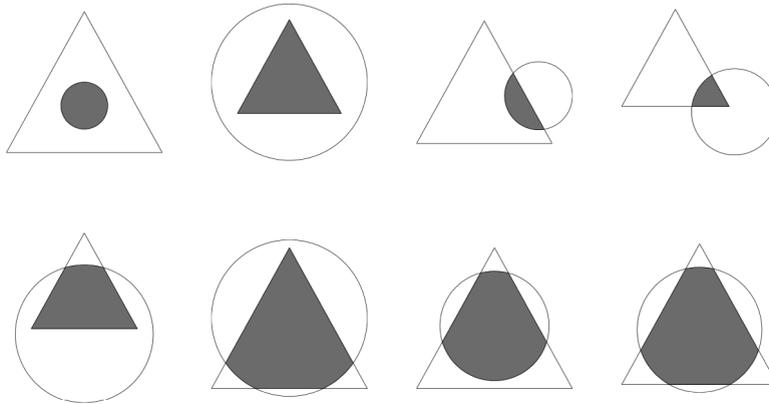


Fig. 5. The eight types of intersections of a disk and a triangle.

4. Quadrature rules for finite element discretizations

Implementation of a finite element method requires the evaluation of integrals. In the nonlocal case, a double integral arises; see (3). Because the kernels we consider are supported on a disk of radius δ (see (5)), the inner integral in (3) has a non-smooth integrand, or, equivalently, has the disk $H_\delta(\mathbf{x})$ as the integration domain. In particular, if $\{\phi_i(\mathbf{x})\}_{i=1}^M$ denotes a finite element basis and $\{E_k\}_{k=1}^K$ denotes the set of finite elements, then, ignoring the volume constraint, each entry of the stiffness matrix A has the general form

$$\begin{aligned}
 A_{ij} &= A(\phi_j, \phi_i) \\
 &= \int_{\Omega \cup \Omega_{\mathcal{I}}} \int_{\Omega \cup \Omega_{\mathcal{I}} \cap H_\delta(\mathbf{x})} \gamma(\mathbf{x}, \mathbf{x}') (\phi_i(\mathbf{x}') - \phi_i(\mathbf{x})) (\phi_j(\mathbf{x}') - \phi_j(\mathbf{x})) d\mathbf{x}' d\mathbf{x} \\
 &= \sum_{k=1}^K \int_{E_k} \int_{\Omega \cup \Omega_{\mathcal{I}} \cap H_\delta(\mathbf{x})} \gamma(\mathbf{x}, \mathbf{x}') (\phi_i(\mathbf{x}') - \phi_i(\mathbf{x})) (\phi_j(\mathbf{x}') - \phi_j(\mathbf{x})) d\mathbf{x}' d\mathbf{x} \\
 &= \sum_{k=1}^K \int_{E_k} \left(\sum_{t=1}^K \int_{E_t \cap H_\delta(\mathbf{x})} \gamma(\mathbf{x}, \mathbf{x}') (\phi_i(\mathbf{x}') - \phi_i(\mathbf{x})) (\phi_j(\mathbf{x}') - \phi_j(\mathbf{x})) d\mathbf{x}' \right) d\mathbf{x}. \tag{7}
 \end{aligned}$$

Thus, the inner integral, when expressed as a sum of integrals over the elements which overlap with the support of the kernel, includes integrals over which $E_t \cap H_\delta(\mathbf{x})$ is strictly contained within E_t .

In finite element implementations, integrals are typically approximated using a suitable quadrature rule over each element E_t . Using such quadrature rules over an element for which $E_t \cap H_\delta(\mathbf{x}) \neq E_t$ can result in a serious loss of accuracy because the integrand may not be smooth. Alternately, in such cases, one can use a quadrature rule specifically designed for integration over the intersection of an element E_t and a disk $H_\delta(\mathbf{x})$. In Fig. 5, we show the eight possible types of domains generated when a triangle and a disk have a non-empty intersection and $E_t \cap H_\delta(\mathbf{x}) \neq E_t$. The variety of cases seem overwhelming, but a little thought shows that, in every case, the intersection region is the union of zero or one polygon and zero to three circular segments (the subarea of the disk between a chord and the circle); because polygons can be easily divided into triangles, we see that, in the end, we only need quadratures over triangles and circular segments. Of course, quadrature rules for triangles are readily available. Quadrature rules for an arbitrary circular segment are given in [14], with a MATLAB implementation given in [15]. An implementation modified for the particular needs of problems such as those treated in this paper is available at [16] and a C implementation is also available. We note that the type of complicated geometries discussed here is also considered in [17] where details about how to deal with them are given.

Using the approach just described, the integrals appearing in the finite element formulation of the problems we consider can be approximated to a sufficiently high accuracy so that the overall accuracy of the approximations is not compromised. The steps needed are to first determine the type of intersection region, decompose that region into sub-triangles and sub-circular segments, apply appropriate quadrature rules, and summing. Note that other issues related to quadrature arise if the integrands are singular; here, so as not to obfuscate our presentation of the multiscale

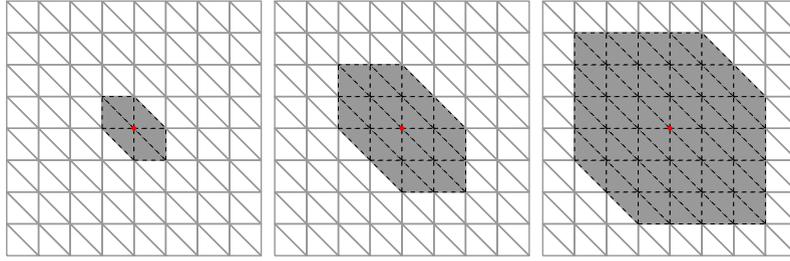


Fig. 6. For $h = 1/8$, the shaded areas show the points that interact with the central point and also the stencil of the finite element stiffness matrix for the local PDE model (left) and for the nonlocal model with $\delta = 1/100 < h$ (middle) and $h < \delta = 1/5 < 2h$ (right).

implementation, we do not address this issue which rears its head in any type of finite element discretization of nonlocal problems. Singular integrals can be treated by methods such as those given in, e.g., [18–20].

5. Local–nonlocal coupling

As part of the multiscale implementation, we wish to apply both a nonlocal (PD) model and a local (PDE) model in different parts of the domain Ω , depending on the presence or lack thereof of solution discontinuities, respectively. Thus, there will be neighboring points at which distinct models are applied so that seemingly a local–nonlocal coupling scheme has to be applied.

Let $\Omega_n \subset \Omega$ and $\Omega_\ell = \Omega \setminus \Omega_n$ denote the regions in which the nonlocal and local models are applied, respectively. The nonlocal–local model, given by

$$\begin{cases} -2 \int_{H_\delta(\mathbf{x})} (u(\mathbf{x}') - u(\mathbf{x})) \gamma(\mathbf{x}, \mathbf{x}') d\mathbf{x}' = f(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega_n \\ -\Delta u(\mathbf{x}) = f(\mathbf{x}) & \text{for } \mathbf{x} \in \Omega_\ell, \end{cases} \quad (8)$$

has to be supplemented with constraints. For simplicity, we only consider Dirichlet type constraints, i.e., we specify $u(\mathbf{x})$ on the interaction domain $\Omega_{\mathcal{I}}$ or on the boundary of Ω . If defects occur within Ω within a distance δ from its boundary $\partial\Omega$, we apply a volume constraint of the form $u(\mathbf{x}) = g(\mathbf{x})$ in a δ -neighborhood $\Omega_{\mathcal{I}}$ exterior to the boundary. However, if there are no defects near the boundary, the condition $u(\mathbf{x}) = g(\mathbf{x})$ is applied only at points precisely on the boundary. Note that when discontinuities are restricted to the interior of the domain, this method has significant advantages over the use of a nonlocal model everywhere, as it avoids dealing with the need to specify data over a volumetric region, data which must be known a priori and thus could perhaps present practical difficulties.

To motivate our strategy for the coupling of local and nonlocal models, we first provide some information about the finite element stiffness matrix generated for the nonlocal model with the use of continuous piecewise linear (CL) elements. First, we examine the effects that nonlocality have on the structure of that matrix. Specifically, in Fig. 6, the shaded areas are indicative of the stencil of that matrix, i.e., the points that interact with a given point and thus engender nonzero entries in the stiffness matrix. We first plot the familiar stencil for the local PDE model that features only nearest neighbor interactions. For the $\delta < h$ case, we see that second-nearest neighbors are also part of the stencil. For the case $h < \delta < 2h$ we see that third nearest neighbors also enter into the picture. As the ratio δ/h increases further, one would see further losses in the sparsity of the stiffness matrix due to increasing nonlocality.

We next examine the effects that changes in the ratio δ/h have on the actual entries of the stiffness matrix. We are particularly interested in what happens for $\delta \ll h$. For simplicity, here we consider the case in which we use a piecewise constant kernel $\gamma = \frac{1}{4\pi\delta^4} \mathbb{1}_{H_\delta(\mathbf{x})}$, where $\mathbb{1}$ denotes the indicator function; the scaling constant $1/4\pi\delta^4$ guarantees that the nonlocal model reduces to the local one as $\delta \rightarrow 0$; see, e.g., [21,22]. The results are qualitatively valid for the other kernels considered in this paper.

For Fig. 7, we have $\Omega = (0, 1)^2$ and fix $h = 1/4$ and vary the values of $\delta < h$. The results in that figure suggest that, for $\delta < h$, the rows of the stiffness matrix associated with the nonlocal model tend to have values similar to those for the local model, and the discrepancy tends to decrease in magnitude linearly with δ . This is hardly a surprising result because it is known that, as $\delta \rightarrow 0$, the nonlocal operator (in the first equation in (8)) converges to that of the local problem (in the second equation in (8)), and moreover, the CL element is asymptotically compatible, as shown

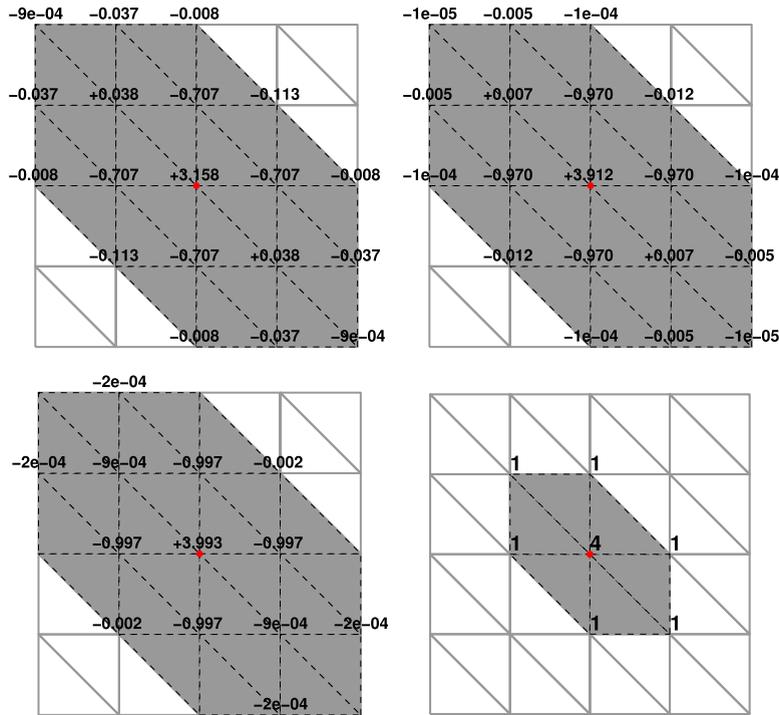


Fig. 7. For $h = 1/4$, the nonzero entries of the stiffness matrix for the nonlocal problem with $\delta = 1/10, 1/100$, and $1/1000$ and for the local partial differential equation model (left to right and top to bottom).

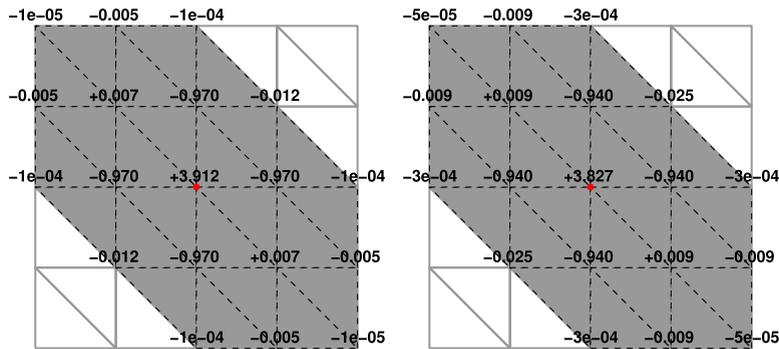


Fig. 8. For $\delta = 1/100$, the nonzero entries of the stiffness matrix for the nonlocal problem with $h = 1/4$ (left) and $1/8$ (right).

in [21,22]. Thus one expects the nonlocal stiffness matrix, i.e., the discretized nonlocal operator, to converge to the local stiffness matrix as $\delta \rightarrow 0$.

We also examine, in Fig. 8, the effect of varying h for fixed $\delta < h$. We observe that because $0 < \delta < h$ for both cases that the two stencils are the same, i.e., they only involve first and second-nearest neighbor interactions. However, for the larger value of h , i.e., for the case of lesser relative nonlocality as indicated by the smaller ratio δ/h , the discrepancy from the local PDE stiffness matrix case is smaller. This again shows that as δ becomes smaller with respect to h , the nonlocal stiffness matrix converges to the local one.

These observations inform our local–nonlocal coupling strategy. We begin with the following requirements: transitions from the nonlocal to the local model only occur in regions where:

1. the solution is smooth;
2. $h \gg \delta$, i.e., the local grid size is much greater than the horizon δ ;
3. continuous finite element spaces are used for both models.

The first requirement ensures that the local model is valid in the transition region. The other two ensure that the stiffness matrices are essentially the same, i.e., that the differences in the entries of the nonlocal and local stiffness matrices corresponding to adjacent nodes are negligible relative to the overall error incurred in discretization. Specifically, with respect to the second requirement, we see from Fig. 7 that the difference between the nonlocal and local stiffness matrices seems to be of $O(\delta)$ as $\delta \rightarrow 0$. Thus, in a practical setting, if h denotes a desired grid size in regions where the solution is smooth, one should choose $\delta = O(h^2)$ to insure that the overall approximation error is of $O(h^2)$.

The coupling strategy we use is very simple. Let $G_N = \{x_j\}_{j=1}^N$ denote the set of interior nodes of the grid in Ω . Then, let $G_{loc} \subset G_N$ denote the subset of “local” nodes; these should be chosen in regions of Ω in which the solution is smooth. Then, $G_{nonlocal} = G \setminus G_{loc}$ denotes the subset of “nonlocal” nodes. The row in the global stiffness matrix corresponding to a local (respectively, nonlocal) node is determined from the weak form of the local (respectively, nonlocal) model. For example, suppose we have $\delta = 0.01$ and a uniform Cartesian grid with $h = 1/4$; note that $\delta \ll h$. Let G_{loc} consist of the leftmost and rightmost pairs of interior nodes so that $G_{nonlocal}$ consists of the three central interior nodes. Using our coupling strategy, the resulting global stiffness matrix is then given by

$$\begin{pmatrix} 4 & -1 & 0 & -1 & 0 & 0 & 0 \\ -1 & 4 & -1 & 0 & -1 & 0 & 0 \\ 0.01 & -0.97 & 3.91 & -0.97 & 0.01 & -0.97 & -0.01 \\ -0.97 & 0.01 & -0.97 & 3.91 & -0.97 & 0.01 & -0.97 \\ 0.01 & -0.97 & 0.01 & -0.97 & 3.91 & -0.97 & 0.01 \\ 0 & 0 & -1 & 0 & -1 & 4 & -1 \\ 0 & 0 & 0 & -1 & 0 & -1 & 4 \end{pmatrix}.$$

A finite element “crime” is committed by our strategy, namely the loss of symmetry of the stiffness matrix. However, because $\delta \ll h$, this crime does not appreciably affect the approximation. For example, we may apply the patch test to our coupling scheme. Thus, if we try to reproduce the solution $u(x, y) = x + y$, the error incurred is of the order of the machine precision with respect to any of the L^2 , L^∞ , or H^1 norms and for a range of grid sizes from $1/2$ to $1/128$. The loss of symmetry can impose a higher cost in solving the discretized problem; if this is a burden, there are ways to recover the symmetry. At the moment, we will not concern ourselves with the symmetry issue because our primary goal is to demonstrate the effectiveness of the multiscale implementation.

It is important to note that, in the mechanics setting, the discussion of the sparsity pattern of the stiffness matrix applies to the bond-based peridynamic model of [4]. For the more generally applicable state-based peridynamic model of [6], the stiffness matrices are considerably less sparse due to the fact that there is more complex interaction behavior between material points.

6. Multiscale implementation

We now have in hand the ingredients needed to define the multiscale implementation of the two-dimensional peridynamics (PD) model. We assume that the solution is a smooth function of position except for jump discontinuities across one or more separate curves, that the discontinuity curves are well separated except perhaps at isolated intersection points, and that each discontinuity curve forms a closed loop entirely within the domain or is a segment that intersects the boundary at exactly two distinct points.

We assume we are given a grid to which the local refinement strategy has already been applied, with a mesh size parameter h , peridynamics horizon δ , and refinement parameter ϵ . We also assume that we have classified every element as either a discontinuous or continuous one, i.e., the thin elements following the discontinuities or the regularly shaped elements, respectively. Starting from this information, we classify every node in the grid. First, every node in a discontinuous element is classified as a PD-DG node (peridynamics-discontinuous Galerkin). Secondly, every node that is not yet classified, but which is contained in an element less than δ in distance from a discontinuous element is classified as PD-CG (peridynamics-continuous Galerkin). The remaining nodes are classified as PDE-CG (local PDE model-continuous Galerkin).

Assuming the grid refinement strategy was carried out, the grid now consists of discontinuous elements, all of whose nodes are of the PD-DG class, with length $O(h)$ and width $O(\epsilon)$. Furthermore, there are one or more transition layers of continuous elements that have at least some PD-CG nodes, whereas the remaining continuous elements

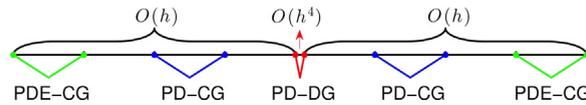


Fig. 9. A multiscale implementation of peridynamics in one dimension [8]. The discontinuity in the displacement occurs in the element of size $\epsilon = O(h^4)$.

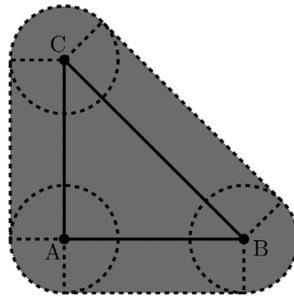


Fig. 10. The interaction domain of a triangle ΔABC is the shaded regions which can be regarded as the union of the triangle, three disks, and three rectangles.

consist entirely of PDE-CG nodes; both of these types of elements have all sides of length $O(h)$. For $\epsilon \ll h$, the layer of PD-CG elements is one element thick. Thus, we obtain a grid that is logically similar to the one-dimensional grid shown in Fig. 9, which is given in [8].

Note that at any stage of the classification process, we wish to identify all continuous elements that are within δ of any discontinuous element. Whereas such a test is obvious to the eye, computationally, it requires a bit of care. Referring to Fig. 10, suppose that triangle ΔABC is a discontinuous element; then the δ -neighborhood of ΔABC for $\delta \ll h$ can be thought of as the union of the triangle itself, three rectangular strips, and three disks, resulting in a convex domain. Instead of checking whether any point in an unclassified triangle is within the δ neighborhood of ΔABC , we choose the quicker approximate test based on checking whether or not that triangle shares any vertices with ΔABC . This rough estimate can save a lot of time and indeed works well in our numerical experiments.

6.1. Choice of the refinement parameter ϵ

We still have to discuss the choice of ϵ and, in particular, how it is related to the grid size h used in regions where the solution is smooth. In the one-dimensional case, we can choose $\epsilon = h^4$ and contain the points at which discontinuity occurs within intervals of that length [8]. This ensures that the multiscale implementation can recover, even for discontinuous solutions, the asymptotic $O(h^2)$ decay rate of the L^2 -norm of the error for piecewise linear elements, which is the optimal rate even for smooth solutions. In two dimensions, if the discontinuity curves are straight-line segments, then one can again contain the discontinuity within thin elements of thickness $\epsilon = h^4$ and again recover the full $O(h^2)L^2$ -norm convergence rate. However, for general curved discontinuity paths, even if they are smooth, it is not possible to contain the discontinuity within straight-sided elements of thickness $\epsilon = O(h^4)$; the best one can do is to have $\epsilon = O(h^2)$. Thus, instead, we make sure that the thin elements are indeed of thickness $\epsilon = O(h^4)$ and that the discontinuity curve passes through the short sides of the thin elements, as illustrated in Fig. 3. As illustrated in the numerical examples provided in Sections 7 and 8, this seems to suffice to recover the $O(h^2)L^2$ -norm convergence rate for piecewise linear approximations.

7. Piecewise linear interpolation of discontinuous functions

As already mentioned, the refinement method described in Section 3 may be used independently of its application to nonlocal problems, e.g., it may be used as a procedure for improving the accuracy of interpolation of given functions having jump discontinuities. Studying the performance of the refinement method in this setting is also useful because it gives us a best-case scenario to compare to when solving nonlocal problems using our multiscale finite element implementation.

Table 1

Example 1. L^2 errors of the piecewise linear interpolant of $\phi_1(x, y)$ corresponding to initial uniform meshes of size h with no refinement and adaptive abrupt mesh refinement.

h	Uniform meshes		Refined meshes	
	L^2 error	Rate	L^2 error	Rate
1/4	5.683e-01	–	5.101e-02	–
1/6	4.665e-01	0.487	2.083e-02	2.209
1/8	4.050e-01	0.491	1.276e-02	1.705
1/10	3.629e-01	0.493	8.069e-03	2.053
1/12	3.316e-01	0.494	5.208e-03	2.401

For a given function, one knows where discontinuities occur so that, in principle, one can use that information to explicitly guide the refinement by, e.g., defining indicator functions that determine which nodes are on which side of a discontinuity. However, for complicated discontinuity curves, this process may be cumbersome, so that we can ignore the fact we know where the discontinuities occur and apply our adaptive abrupt refinement strategy, i.e., at each stage of the refinement process, we evaluate the function values collected at mesh nodes and then determine whether more data may be needed in certain locations. The strategy we use detects a discontinuity curve crossing an element if it detects a large directional derivative, as approximated by a finite difference quotient, of the function along an element edge. For both strategies, only function evaluations at the nodes are needed. The discovery of the location of a discontinuity affects the local sampling rate, the mesh refinement, and the merging step, which seek to isolate the discontinuity within a thin transition zone.

We consider three illustrative examples for the continuous piecewise linear interpolation of discontinuous functions defined over the square $\Omega = (-1, 1)^2$.

Example 1. Consider the function

$$\phi_1(x, y) = \begin{cases} 1 & \text{if } (x < 0 \text{ and } y < 0) \text{ or } (x \geq 0 \text{ and } y \geq 0) \\ 0 & \text{otherwise} \end{cases}$$

for which discontinuities occur across straight line segments. The top-left image of Fig. 11 shows this function which is constant over each of the four unit sub-squares but has a jump of magnitude 1 between any pair of abutting sub-squares. The remaining plots in Fig. 11 show the result of the refinement process starting from several initial uniform meshes with different grid sizes h . This is in keeping with our goal of choosing a grid size h based on the accuracy we would like to achieve for the piecewise linear interpolation of a smooth function and then effect refinement so that, even in the presence of discontinuities, we still achieve an accuracy of $O(h^2)$. Note that the sequence of grid plots in Fig. 11 corresponds to h -refinement whereas each individual plot also shows, for each h , the abrupt grid refinement in the vicinity of the discontinuity curve.

We choose $\epsilon = h^4$; as discussed in Section 6.1, because we have straight discontinuity curves, the refinement process results in the discontinuity line segments being covered by elements of thickness $O(h^4)$ and length $O(h)$ whereas the rest of the mesh has an essentially undisturbed grid size of $O(h)$. Note that the red lines in Fig. 11 are actually made up of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$, except for two isotropic elements of length and width $O(h^4)$ surrounding the intersection of the two discontinuity lines; see Fig. 12. That figure along with Fig. 11 also illustrates the abrupt transition between thin and regularly shaped elements, exactly as desired. Table 1 shows that without applying our adaptive abrupt grid refinement strategy, i.e., for each h simply using a uniform grid, the approximation is only $O(h^{1/2})$ accurate, but with our grid refinement strategy, we recover, for the discontinuous function $\phi_1(x, y)$, the $O(h^2)$ accuracy of piecewise linear approximations of smooth functions using quasi-uniform grids of size h .

Example 2. Consider the function

$$\phi_2(x, y) = \begin{cases} xy + \cos(2\pi x^2) - \sin(2\pi x^2) & \text{if } x^2 + y^2 \leq 0.25 \\ xy + \cos(2\pi x^2) - \sin(2\pi x^2) + (10x - 10) & \text{if } x^2 + y^2 > 0.25 \end{cases}$$

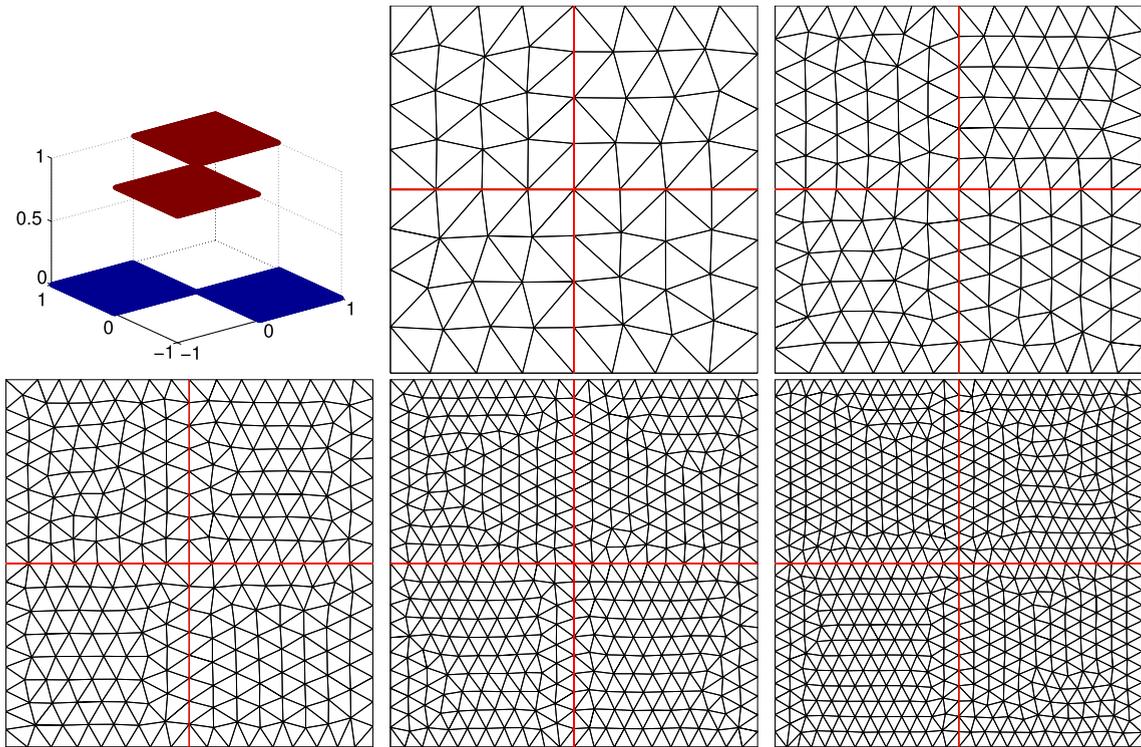


Fig. 11. **Example 1.** Top left: plot of $\phi_1(x, y)$. The remaining figures, from left to right and top to bottom, display refined meshes generated from initial uniform meshes of size $h = 1/4, 1/6, 1/8, 1/10,$ and $1/12$. The red lines actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

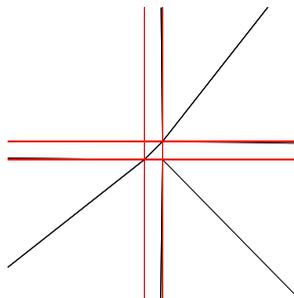


Fig. 12. **Example 1.** Zoom-in near the point $(0, 0)$ of the refined mesh obtained from the initial uniform mesh of size $h = 1/4$.

for which discontinuities occur across a curve, specifically a circle. Fig. 13 provides a plot of $\phi_2(x, y)$ and shows several refined meshes resulting from initial uniform meshes with different grid sizes h . We again choose $\epsilon = O(h^4)$. As discussed in Section 6.1, because we now have curved discontinuities, the thin elements no longer contain the circle across which $\phi_2(x, y)$ is discontinuous, but we have made sure that circle passes through the short sides of those elements.

We also use this example to illustrate some of the influences that the initial mesh may have on the final refined one. In particular, the domain Ω and the discontinuity curve for the function $\phi_2(x, y)$ are such that, geometrically speaking, the problem is symmetric under midline and diagonal reflections and 90° rotations. One may want the refined mesh to share such symmetries, at least with respect to 90° rotations. However, if the initial uniform mesh does not have this

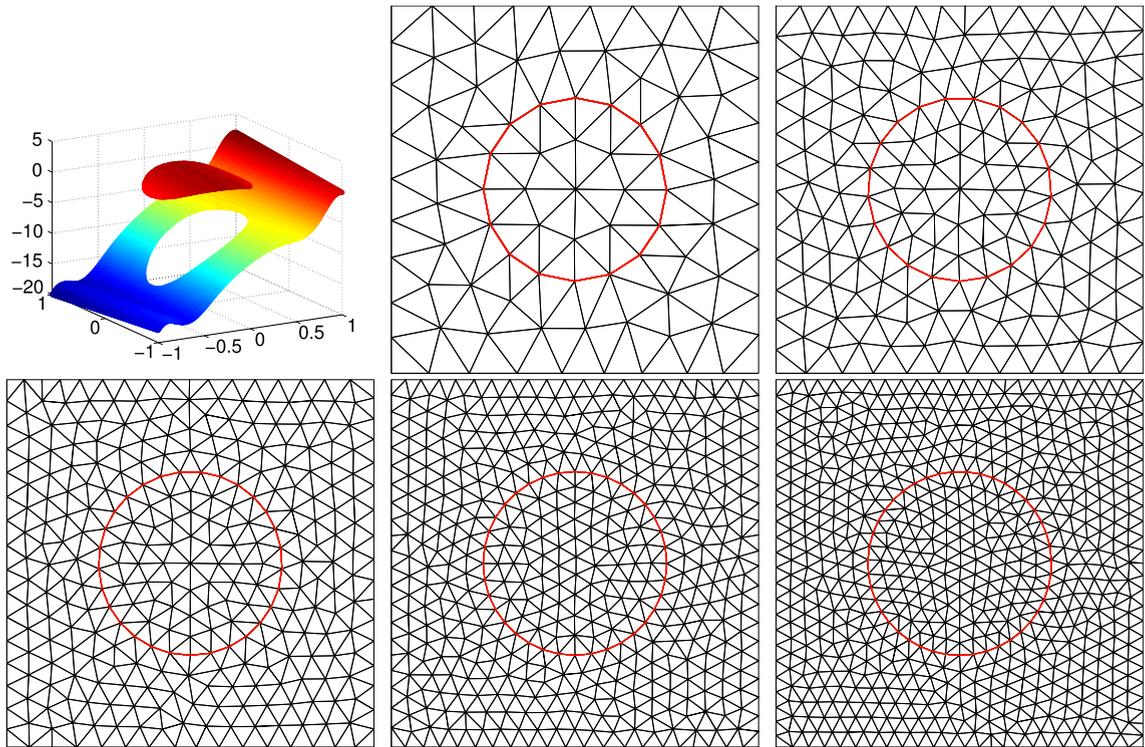


Fig. 13. Example 2. Top left: plot of the function $\phi_2(x, y)$. Remaining plots, left to right and top to bottom: refined meshes generated from initial uniform meshes of size $h = 1/4, 1/6, 1/8, 1/10,$ and $1/12$. The red lines actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2

Example 2. L^2 errors of the piecewise linear interpolant of $\phi_2(x, y)$ corresponding to initial uniform meshes of size h with no refinement and adaptive abrupt mesh refinement.

h	Uniform meshes		Refined meshes	
	L^2 error	Rate	L^2 error	Rate
1/4	4.243e+00	–	8.366e–01	–
1/6	2.566e+00	1.241	3.662e–01	2.037
1/8	2.789e+00	–0.289	2.141e–01	1.866
1/10	2.743e+00	0.0731	1.434e–01	1.794
1/12	2.181e+00	1.259	9.938e–02	2.013

symmetry, there is no mechanism in the refinement process that will bring about symmetry. Fig. 14, however, shows that the refinement procedure will do a much better job at preserving symmetry if the initial mesh shares symmetries with the domain and function.

Fig. 15 illustrates the refined meshes resulting from the use of two different approaches for the detection of the discontinuity curve, namely, the direct use of the indicator function and using finite difference quotient approximations along element edges for derivative estimation. We observe that the resulting meshes are very similar.

Table 2 shows that without grid refinement the error behavior is erratic and certainly does not converge as $O(h^2)$. However, with our adaptive abrupt grid refinement strategy, we, for the most part, recover, for the discontinuous function $\phi_2(x, y)$, the $O(h^2)$ accuracy of piecewise linear approximations of smooth functions using quasi-uniform grids of size h .

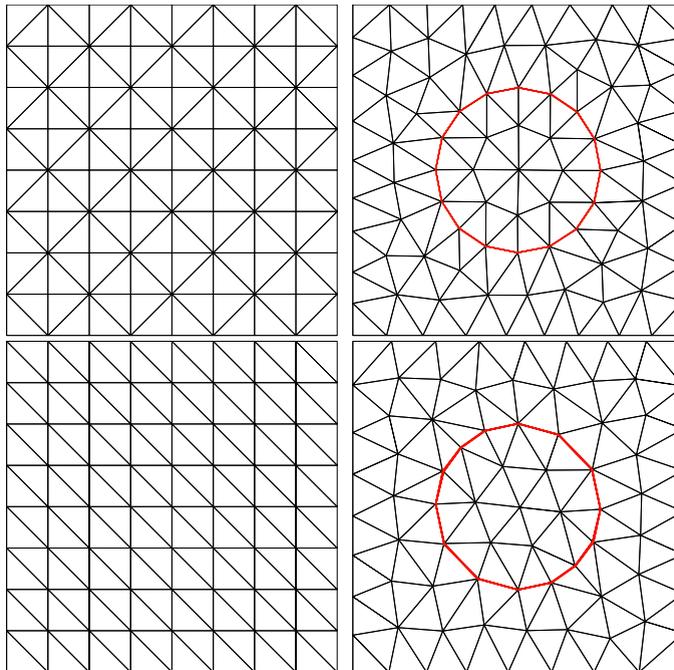


Fig. 14. **Example 2.** Using an initial mesh with 90° rotational symmetry (top left) results in better rotational symmetry of the refined mesh (top right) compared to the analogous result (bottom right) for an initial mesh lacking in such symmetry (bottom left). The red lines actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

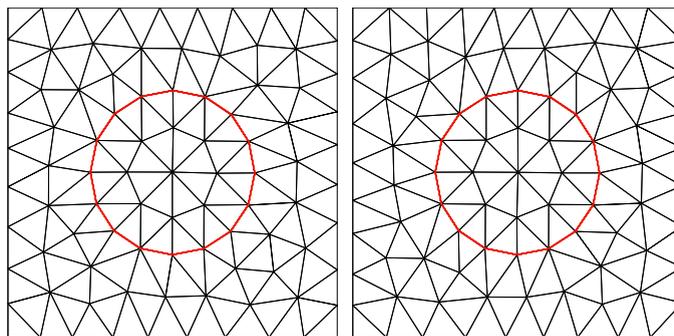


Fig. 15. **Example 2.** Refined meshes resulting from the use of the indicator function (left) and difference quotients along element edges (right). The red lines actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Example 3. Consider the function

$$\phi_3(x, y) = \begin{cases} x & \text{if } y < 0.2 \sin(\pi x) \text{ and } (x - 0.5)^2 + (y + 0.3)^2 \geq 0.01 \\ x - 1 & \text{if } y < 0.2 \sin(\pi x) \text{ and } (x - 0.5)^2 + (y + 0.3)^2 < 0.01 \\ 1 + \sin(x) & \text{if } y \geq 0.2 \sin(\pi x) \text{ and } (x + 0.5)^2 + (y - 0.3)^2 \geq 0.01 \\ \sin(x) & \text{if } y \geq 0.2 \sin(\pi x) \text{ and } (x + 0.5)^2 + (y - 0.3)^2 < 0.01. \end{cases}$$

We again choose $\epsilon = O(h^4)$. This example also serves to illustrate that even away from the discontinuity, one may want to use different mesh sizes in different regions.

Fig. 16 provides plots of $\phi_3(x, y)$ and of several refined meshes resulting from initial meshes with different grid sizes h . For this function, away from discontinuity curves, the two small circular regions interior to the circles call for

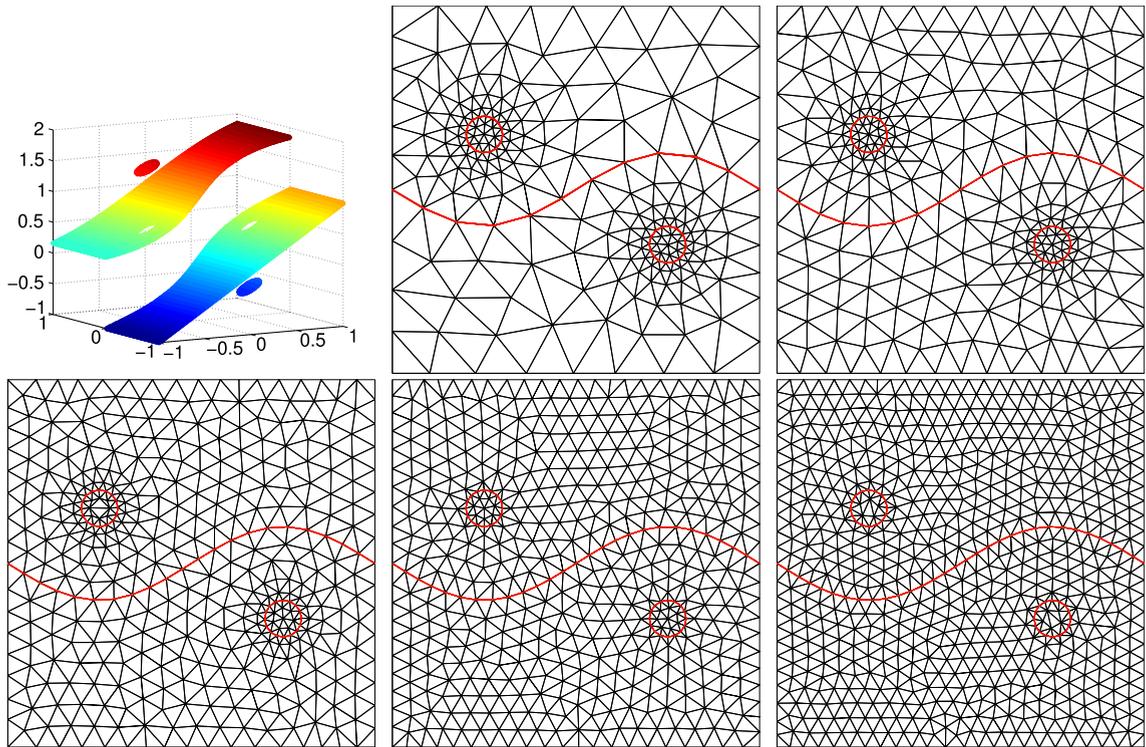


Fig. 16. Example 3. Top left: plot of the function $\phi_3(x, y)$. Remaining plots, left to right and top to bottom: refined meshes generated from initial uniform meshes of size $h = 1/4, 1/6, 1/8, 1/10,$ and $1/12$. The red lines actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3

Example 3. L^2 errors of the piecewise linear interpolant of $\phi_3(x, y)$ corresponding to initial uniform meshes of size h with no refinement and adaptive abrupt mesh refinement.

h	Uniform meshes		Refined meshes	
	L^2 error	Rate	L^2 error	Rate
1/4	3.089e-01	–	4.726e-02	–
1/6	2.882e-01	0.171	2.029e-02	2.086
1/8	2.349e-01	0.711	1.245e-02	1.696
1/10	2.547e-01	–0.362	7.994e-03	1.987
1/12	2.188e-01	0.837	5.211e-03	2.348

a finer mesh compared to the mesh for the regions exterior to the circles; however, both meshes are quasi-uniform and of grid size $O(h)$. For large h , the refinement procedure produces a transition zone of several layers, but this effect diminishes as h decreases to the local h imposed in the circle interiors. The red lines again consist of thin elements of length $O(h)$ and width $O(\epsilon) = O(h^4)$.

Table 3 shows errors and rates for uniform meshes and refined meshes. The comments made for Table 2 apply to this case as well.

8. Examples of the multiscale implementation

We now turn to the implementation of the multiscale method for nonlocal problems in two dimensions. In Section 8.2, we present an example for the full implementation as described in Sections 2–6 in which we pretend we do not know the location of the discontinuity curves and allow our adaptive process to identify them and then to effect the needed refinement. Before we do so, we first present, in Section 8.1, a partial implementation in which,

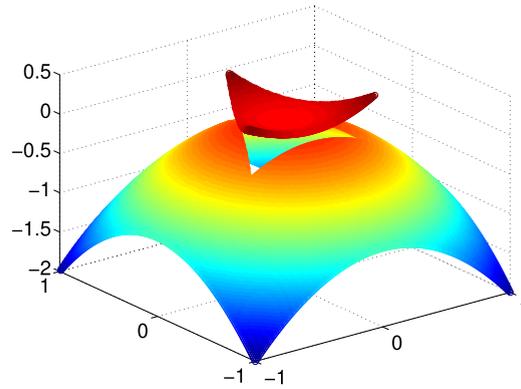


Fig. 17. Plot of the exact solution (9).

instead of using adaptivity to effect refinement, we assume we know where the discontinuities are located and then manually construct the needed refined meshes. We do so to determine a second baseline (in addition to that given by Section 7) for our refined meshes which we then use to judge the effectiveness of the full implementation which is what one would apply in practice.

8.1. Multiscale implementation using manual refinement

8.1.1. Discontinuity across edges of a triangle

Consider the manufactured solution in the square domain $\Omega = (-1, 1)^2$ given by

$$u(x, y) = \begin{cases} x^2 + y^2 & \text{if } (x, y) \in \Delta \\ -x^2 - y^2 & \text{otherwise,} \end{cases} \tag{9}$$

where Δ denotes the equilateral triangle having vertices $(-1/2, -\sqrt{3}/6)$, $(1/2, -\sqrt{3}/6)$, and $(0, \sqrt{3}/3)$ and across whose edges the function (9) experiences a jump discontinuity. Note that discontinuity path itself has a discontinuous geometric derivative. A plot of the function (9) is provided in Fig. 17.

The manual refinement process proceeds as follows. Let h denote a measure of the unrefined grid size which, in this example, can be taken to be the mesh size along the edges of the square domain Ω . First, we cover the discontinuity path, i.e., the equilateral triangle Δ , with congruent quadrilaterals of length $O(h)$ and width $O(h^4)$; this is possible because the discontinuity path consists of straight line segments. The remainder of the interior of Δ is then subdivided into smaller equilateral triangles having sides of length $O(h)$ whereas the remainder of its exterior in Ω is remeshed using a CCVT approach. We also manually assign nodes to the different node classes. Specifically, PD-DG nodes are restricted to the vertices of the elements that contain the discontinuity path, PD-CG nodes form a transition layer around them, and most of the nodes in the region both inside and outside of Δ are PDE-CG nodes, which is exactly the desired distribution for the multiscale model. Fig. 18 shows, for several grid sizes h , the results of this meshing and the assignment of nodes to the different node classes. Also plotted in that figure are the “boundary” nodes, i.e., the nodes at which the Dirichlet constraint on the solution is applied. Note that for the values of $h < 1/2$, those nodes are located simply at the boundary of the region Ω , as is expected because those nodes are separated from the nonlocal PD-DG nodes by CG nodes. However, for $h = 1/2$, we see that additional boundary nodes are needed outside of Ω because the nonlocal PD-CG nodes are connected to the nodes on the boundary of Ω . Of course, in practice, one would almost surely use a grid size smaller than $h = 1/2$, so that this anomaly does not occur unless discontinuity curves intersect the boundary.

Table 4 lists the exact counts of the PD-DG, PD-CG, and PDE-CG nodes for each of the meshes; these are the nodes that correspond to the nodal degrees of freedom whereas, at the remaining “boundary” nodes, the approximate solution is specified by the Dirichlet constraint. The number N_{cg}^{pde} of PDE-CG nodes grows quadratically (proportional to area) whereas the numbers N_{dg}^{pd} and N_{cg}^{pd} of PD-DG and PD-CG nodes, respectively, grow linearly (proportional to length). Table 4 also lists the total degrees of freedom used for solving the nonlocal problem using the abruptly refined grids and the judicious use of CG and DG elements in different regions, the degrees of freedom one would

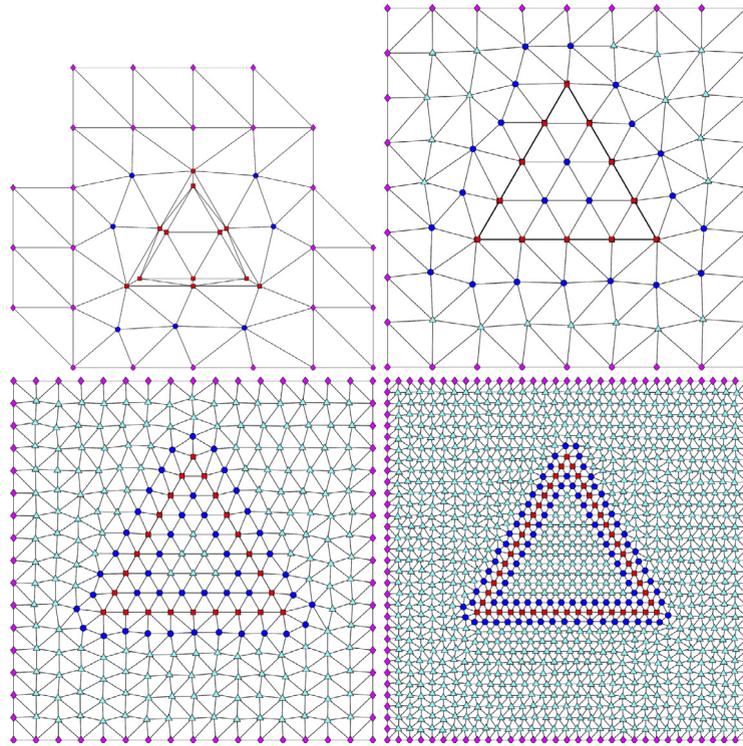


Fig. 18. For the exact solution (9) with manual grid refinement and for $h = 1/2, 1/4, 1/8,$ and $1/16$ (left to right and top to bottom), manually refined meshes and distribution of node types. Red nodes are PD-DG nodes, blue ones are PD-CG nodes, cyan ones are PDE-CG nodes, and magenta ones are boundary nodes. The lines linked by red nodes actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

use in determining an approximation to a smooth solution of the related PDE problem on a uniform grid of size h , and the percentage of additional degrees of freedom used to solve the problem with a discontinuous solution using our approach. We see that the percentage overhead decreases linearly in h as the problem size grows, as is expected from the growth rates for the number of each class of nodes. The grid sizes h used for Table 4 are rather large so that the overhead is significant. However, the smaller grid sizes used in most practical situations, we can expect much lower overheads in the number of degrees of freedom; for example, for $h = 1/100$, one can expect an approximately 10% overhead whereas for $h = 1/1000$ it would be approximately 1%. Thus, we do observe that the multiscale implementation efficiently limits the additional costs incurred for treating the discontinuity using our refinement approach.

We now use grids such as those illustrated in Fig. 18 to solve the coupled local–nonlocal problem (8) where, for the nonlocal problem, we choose the kernel $\gamma(\mathbf{x}, \mathbf{x}') = \frac{40}{3\pi\delta^4} (1 - \frac{r^2}{\delta^2})^2 (1 + \frac{2r^2}{\delta^2}) \mathbb{1}_{H_\delta(\mathbf{x})}$, where $\mathbb{1}$ denotes the indicator function and $r = |\mathbf{x} - \mathbf{x}'|$ the Euclidean distance between \mathbf{x} and \mathbf{x}' . The scaling constant that appears in the kernel guarantees that the nonlocal model reduces to the local one as $\delta \rightarrow 0$. This kernel and its first derivatives vanish at the boundary of $H_\delta(\mathbf{x})$. Kernels that are less smooth require careful treatment, especially with regard to the quadrature rules used to construct stiffness matrices.

Table 5 shows, for δ fixed at 0.01, the convergence behavior of the L^2 and L^∞ norms of the errors as h decreases from $1/2$ to $1/32$. The L^2 error approaches the optimal rate of 2, but the L^∞ exhibits no convergence. This is all to be expected; our goal was to achieve optimal convergence with respect to the L^2 norm, which is what was achieved. However, no amount of grid refinement is going to help with the L^∞ norm because regardless of the degree of polynomials used or of the grid size, the error in the approximation of a function having a jump discontinuity will remain of $O(1)$; see [7] for a further discussion. These results are also consistent with what was reported in [8] for the one-dimensional case. However, if when calculating errors we simply exclude the elements known to contain the

Table 4

For the exact solution (9) with manual grid refinement and for several initial grid sizes h , the distribution of node types and the degrees of freedom resulting from the multiscale implementation. Also listed are the number of degrees of freedom for the local PDE model using a uniform unrefined grid of grid size h . Percentage overhead refers to the additional degrees of freedom (DOF) used for the nonlocal model with abruptly refined grids relative to a local model with a uniform grid.

h	1/2	1/4	1/8	1/16	1/32
N_{dg}^{pd}	12	24	48	96	192
N_{cg}^{pd}	7	22	45	93	191
N_{cg}^{pde}	0	20	170	861	3785
Nonlocal DOF	81	187	503	1530	5130
Local DOF	9	49	225	961	3969
Percentage overhead	800.0	281.6	123.6	59.2	29.3

Table 5

For the exact solution (9) with manual grid refinement, errors in the approximate solutions obtained using the multiscale implementation with manually generated refined meshes.

Errors over all elements					
h	L^∞ error	Rate	L^2 error	Rate	
1/2	1.057e+01	–	1.058e+01	–	
1/4	3.255e–01	5.021	2.352e–01	5.491	
1/8	1.496e–01	1.121	2.625e–02	3.163	
1/16	1.458e–01	0.0372	5.962e–03	2.139	
1/32	1.492e–01	–0.0336	1.545e–03	1.949	

Errors excluding thin elements						
h	L^∞ error	Rate	L^2 error	Rate	H^1 error	rate
1/2	9.001e+00	–	9.714e+00	–	3.092e+01	–
1/4	2.514e–01	5.162	2.343e–01	5.374	9.769e–01	4.984
1/8	4.037e–02	2.639	2.616e–02	3.163	9.871e–02	3.307
1/16	6.735e–03	2.583	5.935e–03	2.140	3.105e–02	1.669
1/32	1.788e–03	1.913	1.538e–03	1.948	1.397e–02	1.152

discontinuity, the convergence rates measured over the remaining elements now approach the optimal rates for all norms, even for the H^1 norm of the error which we can now report on because that norm is now bounded for the exact solution, the discontinuity now being excluded. This illustrates that the combination of abrupt refinement and using DG elements near the discontinuity does not cause a spread of errors away from the elements containing the discontinuity path.

Fig. 19 illustrates different aspects of the approximate solution for the case $h = 1/8$. Suppressing the elements containing the discontinuity results in a plot that accurately approximates the solution. The numerical solution over the discontinuity elements illustrates how the jump in the solution is handled. The bottom-right plot illustrates how well our method does at capturing the discontinuity in the solution while also preserving the smoothness of the solution away from the discontinuity without any numerical artifacts such as Gibbs phenomena.

Table 6 provides further insight by listing the number of nonzeros in the stiffness matrices resulting from using the nonlocal model everywhere, using the multiscale implementation of the nonlocal model for which the nonlocal model is used only in the vicinity of discontinuities, and using a local PDE model everywhere on a uniform grids (with no refinement) of grid size h . We also provide the percentage overhead, i.e., the percentage of additional nonzero entries, incurred by the multiscale implementation over the local one. For the relatively coarse grid sizes h used for that table, the overhead is significant but, because that overhead reduces linearly in h , for the smaller grid sizes used in practical situations, the percentage overhead in the number of nonzero stiffness matrix entries will be much

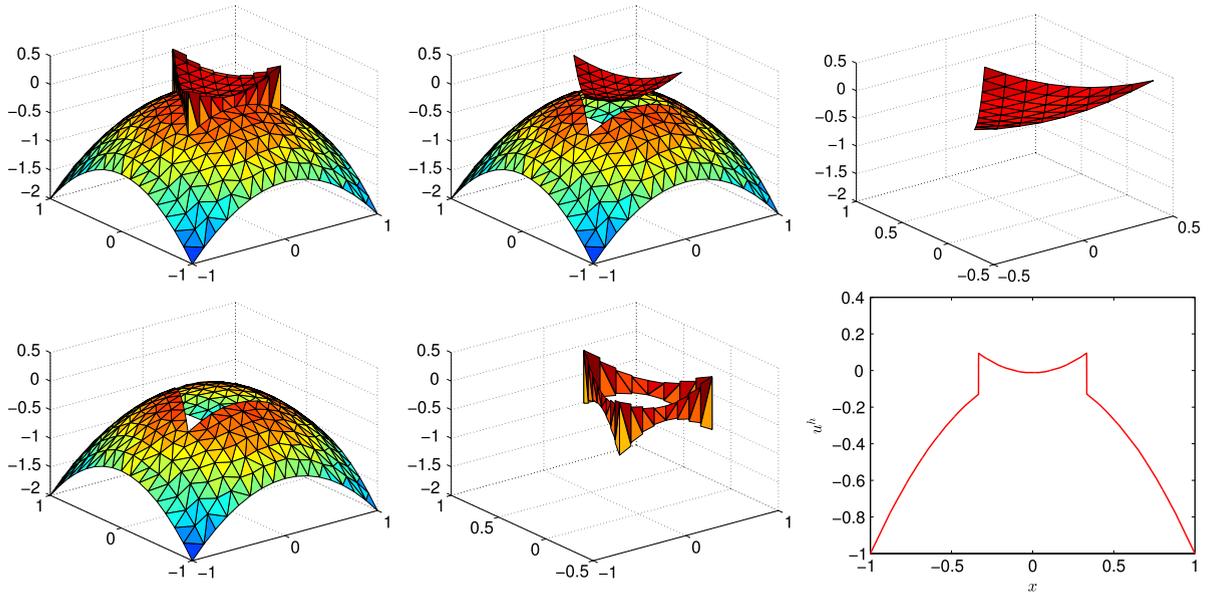


Fig. 19. For the exact solution (9) with manual grid refinement, plots of approximate solution for $h = 1/8$. Top-left: the complete approximate solution. Top-middle: the solution over elements not containing the discontinuity. Top-right: the solution over elements inside the triangle. Bottom-left: solution over elements outside the triangle. Bottom-middle: solution over elements covering the edges of the triangle. Bottom-right: cross section of solution at $y = 0$.

Table 6

For the exact solution (9) with manual grid refinement and for several initial grid sizes h , the number of nonzero stiffness matrix entries resulting from applying the nonlocal model everywhere and from the multiscale implementation of the nonlocal model. Also given are the number of nonzero entries in the stiffness matrix corresponding to the local PDE model with a uniform unrefined grid of grid size h . Here, percentage overhead refers to the additional nonzero entries resulting from the multiscale implementation relative to that for the local model.

h	1/2	1/4	1/8	1/16	1/32
Nonlocal	2139	6831	16 943	43 610	127 004
Multiscale	2139	6652	15 150	33 681	81 789
Local	63	343	1 575	6 727	27 783
Percentage overhead	3295	1839	862	401	194

smaller. For example, for $h = 1/100$ and $1/1000$, one can expect percentage overheads of approximately 60% and 6%, respectively. Clearly, for practical values of h , the multiscale implementation involves significantly fewer nonzero entries compared to the full nonlocal treatment and very little overhead compared to the local model.

8.1.2. A solution with discontinuity across a circle

As a second test, we consider a manufactured solution within the square domain $\Omega = (-1/2, 1/2)^2$ given by

$$u = \begin{cases} 2(x^2 + y^2) & \text{if } x^2 + y^2 > 1/9 \\ x^2 + y^2 & \text{otherwise} \end{cases} \tag{10}$$

for which the discontinuity path is a circle of radius $1/3$ centered at the origin. A plot of the exact solution (10) is given in Fig. 20.

We again manually generate refined meshes with $\epsilon = O(h^4)$. Because of the curvature of the discontinuity path, with this choice of ϵ , the thin elements cannot cover that curve so that we again just make sure that the path passes through the short sides of those elements. We use the kernel $\gamma(\mathbf{x}, \mathbf{x}') = \frac{24}{\pi \delta^4} (1 - \frac{r^2}{\delta^2})^2 \mathbb{1}_{H_\delta(\mathbf{x})}$; the scaling constant again guarantees that as $\delta \rightarrow 0$, the nonlocal problem reduces to the local PDE problem. We set $\delta = 0.01$. Fig. 21 displays

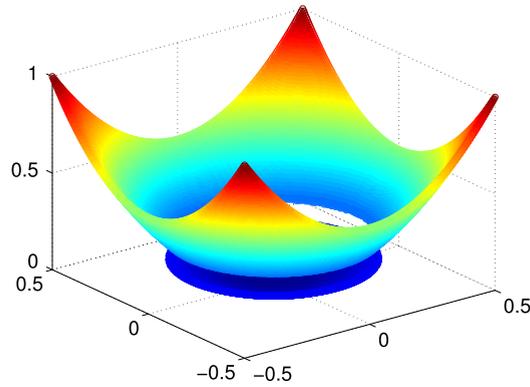


Fig. 20. Plot of the exact solution (10).

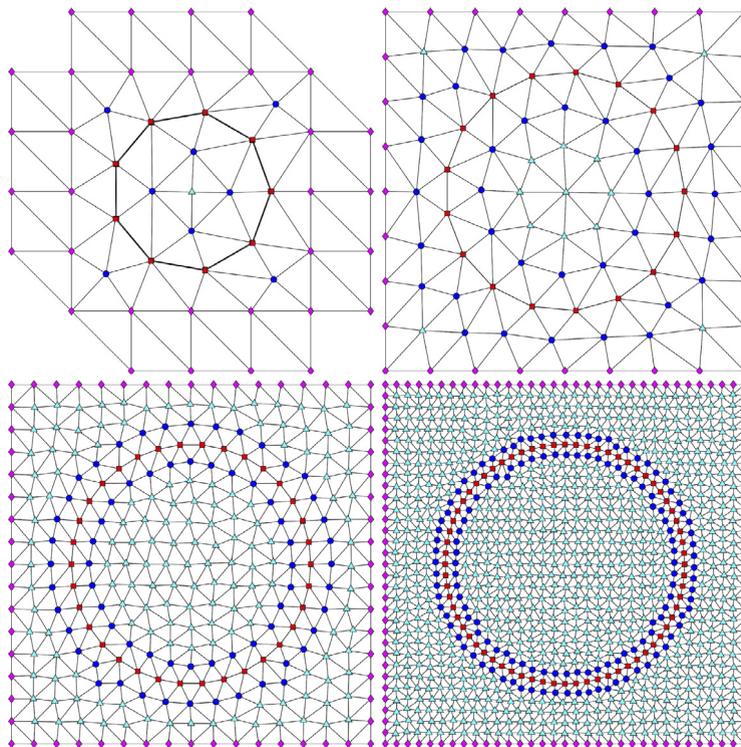


Fig. 21. For the exact solution (10) with manual grid refinement and for $h = 1/4, 1/8, 1/16,$ and $1/32$ (left to right and top to bottom), manually refined meshes and distribution of node types. Red nodes are PD-DG nodes, blue ones are PD-CG nodes, cyan ones are PDE-CG nodes, and magenta ones are boundary nodes. The lines linked by red nodes actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the distribution of node types for this problem for refined meshes with several values of h whereas Table 7 provides the actual node counts. Note that in the upper-left plot in Fig. 21 the thin elements are not anywhere near as visible as they were in the correspondingly positioned plot in Fig. 18. This is simply due to the fact that the former is for $h = 1/4$ so that $h^4 = 1/256$ whereas the latter is for $h = 1/2$ so that $h^4 = 1/16$. Table 8 displays the convergence behavior of approximate solutions generated for several values of h with δ fixed at 0.01. Fig. 22 illustrates features of the numerical solution obtained for $h = 1/32$. Again, plotting over just the continuous elements shows excellent agreement with the true solution. Even though now the discontinuity curve is not completely contained within the

Table 7

For the exact solution (10) with manual grid refinement and for several initial grid sizes h , the distribution of node types and the degrees of freedom resulting from the multiscale implementation. Also listed are the number of degrees of freedom for the local PDE model using a uniform unrefined grid of grid size h . Percentage overhead refers to the additional degrees of freedom (DOF) used for the nonlocal model with abruptly refined grids relative to a local model with a uniform grid.

h	1/4	1/8	1/16	1/32	1/64
N_{dg}^{pd}	18	34	68	136	270
N_{cg}^{pd}	9	36	66	135	263
N_{cg}^{pde}	0	13	151	818	3682
Nonlocal DOF	119	255	623	1768	5558
Local DOF	9	49	225	961	3969
Percent overhead	1222	420	177	84	40

Table 8

For the exact solution (10) with manual grid refinement, errors in the approximate solutions obtained using the multiscale implementation with manually generated refined meshes.

Errors over all elements						
h	L^∞ error	Rate	L^2 error	Rate		
1/4	2.518e-01	–	1.1249e-01	–		
1/8	1.188e-01	1.084	1.429e-02	2.977		
1/16	1.082e-01	0.135	3.985e-03	1.842		
1/32	1.051e-01	0.042	7.340e-04	2.441		
1/64	1.047e-01	0.005	2.041e-04	1.847		
Errors excluding thin elements						
h	L^∞ error	Rate	L^2 error	Rate	H^1 error	Rate
1/4	8.184e-01	–	5.086e-01	–	2.638e+00	–
1/4	1.572e-01	–	1.108e-01	–	4.328e-01	–
1/8	1.831e-02	3.102	1.412e-02	2.9731	6.338e-02	2.772
1/16	5.329e-03	1.781	3.952e-03	1.8366	2.875e-02	1.141
1/32	1.055e-03	2.337	7.235e-04	2.4495	1.306e-02	1.138
1/64	3.093e-04	1.770	2.017e-04	1.8426	6.243e-03	1.065

Table 9

For the exact solution (10) with manual grid refinement and for several initial grid sizes h , the number of nonzero stiffness matrix entries resulting from applying the nonlocal model everywhere and from the multiscale implementation of the nonlocal model. Also given are the number of nonzero entries in the stiffness matrix corresponding to the local PDE model with a uniform unrefined grid of grid size h . Here, percentage overhead refers to the additional nonzero entries resulting from the multiscale implementation relative to that for the local model.

	1/4	1/8	1/16	1/32	1/64
Nonlocal	4445	10 037	22 043	54 340	147 792
Multiscale	4445	9 839	20 322	44 698	103 350
Local	63	343	1 575	6 727	27 783
Percent overhead	6956	2 769	1 190	564	272

thin elements, the comments made for the corresponding tables and figures of Section 8.1.1 apply here as well (see Table 9).

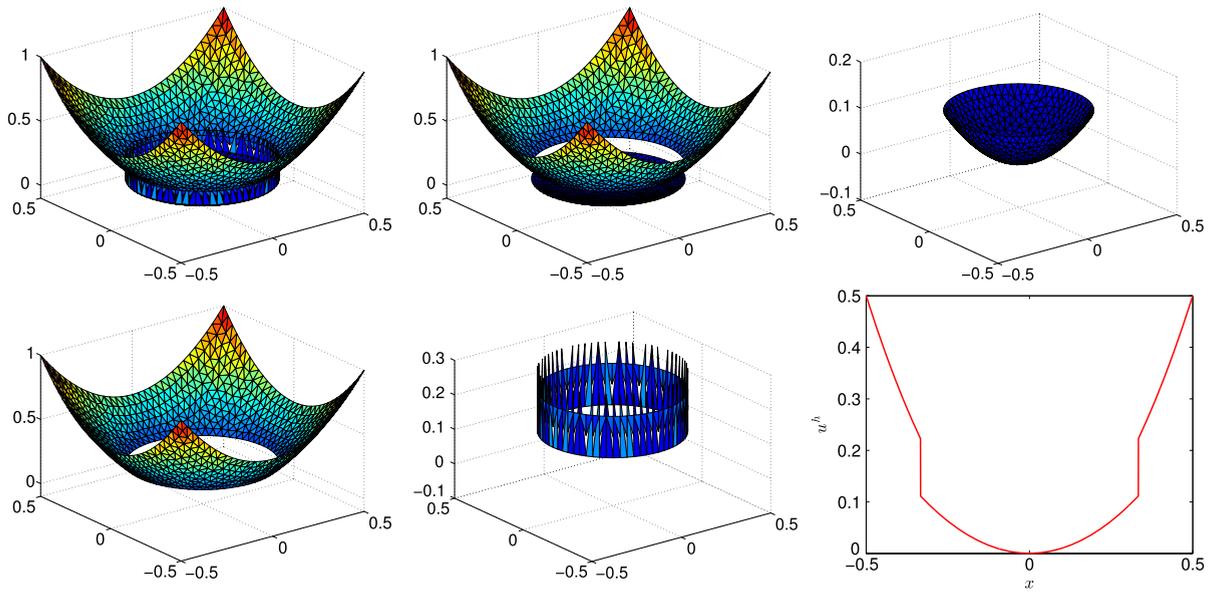


Fig. 22. For the exact solution (10) with manual grid refinement, plots of approximate solution for $h = 1/32$. Top-left: the complete approximate solution. Top-middle: the solution over elements not containing the discontinuity. Top-right: the solution over elements inside the circle. Bottom-left: solution over elements outside the circle. Bottom-middle: solution over elements covering the circle. Bottom-right: cross section of solution at $y = 0$.

Table 10

For the exact solution (10) with adaptive abrupt grid refinement and for several initial grid sizes h , the distribution of node types and the degrees of freedom resulting from the multiscale implementation. Also listed are the number of degrees of freedom for the local PDE model using a uniform unrefined grid of grid size h . Percentage overhead refers to the additional degrees of freedom (DOF) used for the nonlocal model with adaptively refined grids relative to a local model with a uniform grid.

h	1/4	1/8	1/16	1/32
N_{dg}^{pd}	32	38	78	156
N_{cg}^{pd}	21	36	69	135
N_{cg}^{pde}	28	85	156	826
Nonlocal DOF	229	275	684	1876
Local DOF	9	49	225	961
Percent overhead	2444	461	204	95

8.2. Multiscale implementation with adaptive abrupt mesh refinement

Now we consider the full-fledged implementation of the multiscale model with adaptive abrupt grid refinement, as outlined in Section 3. We again use the exact solution (10), choose δ fixed at 0.01, use the kernel $\gamma(\mathbf{x}, \mathbf{x}') = \frac{24}{\pi \delta^4} (1 - \frac{r^2}{\delta^2})^2 \mathbb{1}_{H_\delta(\mathbf{x})}$, and set $\epsilon = O(h^4)$. We consider the solution for a sequence of values h from 1/4 to 1/32. The thin elements resulting from the adaptive abrupt grid refinement process again do not entirely contain the discontinuity curve but their short sides do so, as discussed in Section 8.1.2.

Fig. 23 presents the distribution of node types, Table 10 provides the node type counts, Table 11 records the convergence behavior, Fig. 24 provides plots of different aspects of the approximate solution, and Table 12 provides counts of the number of nonzero entries of stiffness matrices. These results are well in line with the behavior already seen for the interpolation examples and for manual grid refinement; see, e.g., Section 8.1.2. In particular, the adaptive abrupt grid refinement process that constitutes our multiscale implementation described in Section 3 results in optimal accuracy even in the presence of discontinuities in the solution.

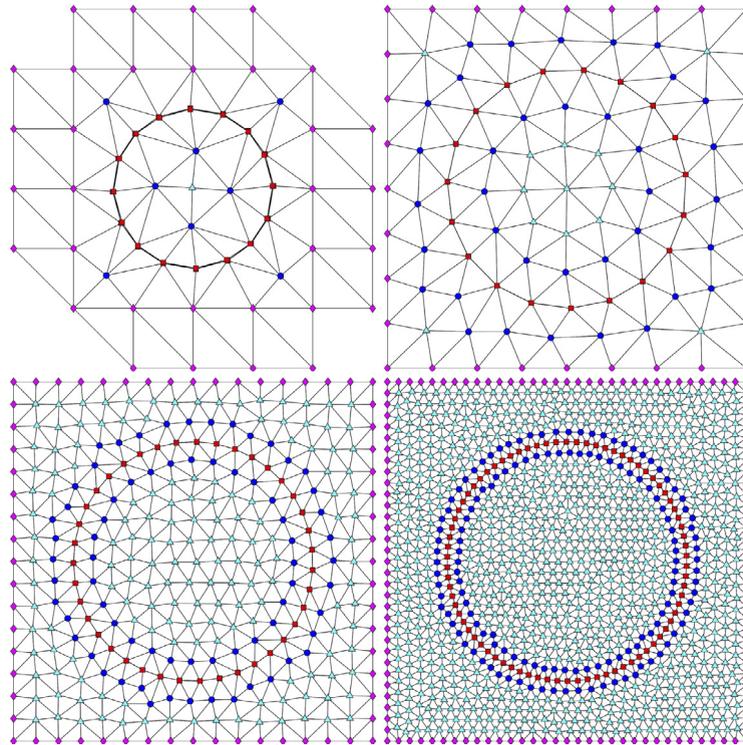


Fig. 23. For the exact solution (10) with adaptive abrupt grid refinement and for $h = 1/4, 1/8, 1/16,$ and $1/32$ (left to right and top to bottom), adaptively refined meshes and distribution of node types. Red nodes are PD-DG nodes, blue ones are PD-CG nodes, cyan ones are PDE-CG nodes, and magenta ones are boundary nodes. The red lines actually consist of pairs of triangular grid cells forming rectangles having long side $O(h)$ and short side $O(h^4)$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 11

For the exact solution (10) with adaptive abrupt grid refinement, errors in the approximate solutions obtained using the multiscale implementation with adaptively generated refined meshes.

Errors over all elements						
h	L^∞ error	Rate	L^2 error	Rate		
1/4	5.623e-02	–	2.738e-02	–		
1/8	9.818e-02	–0.804	9.192e-03	1.575		
1/16	9.573e-02	0.036	2.902e-03	1.663		
1/32	9.421e-02	0.023	7.114e-04	2.029		
Errors excluding thin elements						
h	L^∞ error	Rate	L^2 error	Rate	H^1 error	Rate
1/8	2.362e-01	–	7.767e-02	–	8.559e-01	–
1/4	4.784e-02	–	2.724e-02	–	2.742e-01	–
1/8	1.307e-02	1.872	9.052e-03	1.589	4.916e-02	2.480
1/16	4.060e-03	1.687	2.877e-03	1.654	2.783e-02	0.821
1/32	1.019e-03	1.995	7.050e-04	2.029	6.838e-03	2.025

9. Concluding remarks

We have developed a multiscale implementation of finite element discretizations of nonlocal models in two dimensions that appear in diffusion and mechanics problems. Using piecewise linear approximations and an adaptive abrupt mesh refinement process, the methodology relies on abruptly refined meshes, seamless coupling of local to nonlocal discretized models, and judicious application of discontinuous and continuous Galerkin methods. As a result,

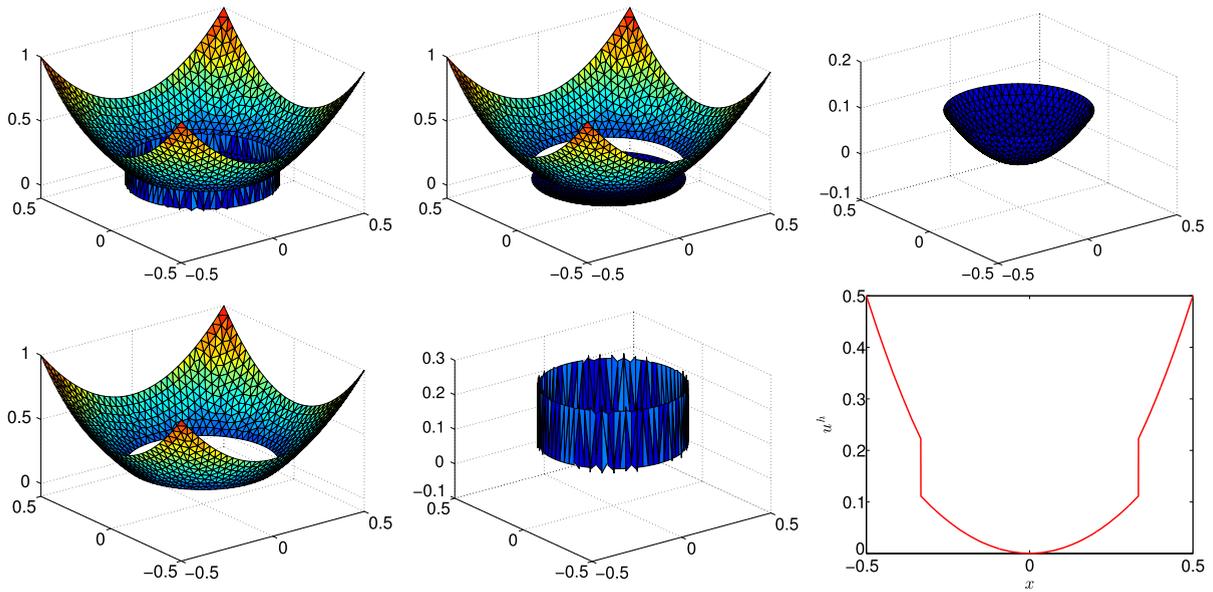


Fig. 24. For the exact solution (10) with adaptive abrupt grid refinement, plots of approximate solution for $h = 1/32$. Top-left: the complete approximate solution. Top-middle: the solution over elements not containing the discontinuity. Top-right: the solution over elements inside the circle. Bottom-left: solution over elements outside the circle. Bottom-middle: solution over elements covering the circle. Bottom-right: cross section of solution at $y = 0$.

Table 12

For the exact solution (10) with adaptive abrupt grid refinement and for several initial grid sizes h , the number of nonzero stiffness matrix entries resulting from applying the nonlocal model everywhere and from the multiscale implementation of the nonlocal model. Also given are the number of nonzero entries in the stiffness matrix corresponding to the local PDE model with a uniform unrefined grid of grid size h . Here, percentage overhead refers to the additional nonzero entries resulting from the multiscale implementation relative to that for the local model.

h	1/4	1/8	1/16	1/32
Nonlocal	7 033	10 761	24 378	58 132
Multiscale	7 033	10 553	22 553	48 371
Local	63	343	1 575	6 727
Percent overhead	11 063	2 977	1 332	619

even for discontinuous solutions, second-order L^2 convergence rates are achieved with little additional costs over the use of local models everywhere.

Here, we make brief comments about future efforts related to making our methodology more efficient and for extending it beyond the two-dimensional, time-independent setting.

A step in the mesh refinement process, as illustrated in the left plot of Fig. 2, is the construction of a string of small isotropic elements that contains the curve across which the solution is discontinuous. Ultimately, we refine these elements so that their linear size is of $O(h^4)$. Then, if the domain and discontinuity curve have linear dimension $\approx L$ and the non-refined grid is of size $h \approx L/N$ for some positive integer N , the number of tiny elements would grow to be of $O(1/N^4)$. Such growth is prohibitive with respect to memory and compute costs. A possible way to mitigate this situation is to not wait until the refinement of the elements containing the discontinuity is completed to combine the tiny isotropic elements into long thin elements as illustrated in the right plot of Fig. 2. Instead that combination step is effected after several steps, after which one could just refine the length and width of the quadrilaterals themselves.

The use of DG methods in the elements containing the discontinuity is a major contributor to the overheads incurred in the number of degrees of freedom and in the number of nonzero stiffness matrix entries. Of course, for small grid sizes this may not matter much because the number of such elements would be small compared to the total number of elements. On the other hand, using CG methods everywhere would greatly reduce the overhead because then, e.g., the total number of degrees of freedom would be the same as the total number of nodes in the grid. CG discretizations,

even in elements containing the discontinuity curve, would incur greater errors, but the $O(h^2)$ convergence rates would be preserved.

Extending our methodology to three-dimensional problems featuring solutions that are discontinuous across surfaces is conceptually straightforward, but does raise serious implementation difficulties. It is true that the growth in the additional degrees of freedom is still linear in $1/h$ because the growth in the number of nodes is of $O(1/h^3)$ in unrefined regions and is of $O(1/h^2)$ for the thin flat elements in the vicinity of the discontinuity. However, the development of quadrature rules in three dimension analogous to those discussed in Section 4 for two dimensions is a daunting task. Instead, one would surely want to use localized kernels that are smooth in \mathbb{R}^n but vanish outside of balls of radius δ so that quadrature rules on whole elements can be used instead of the cumbersome rules for regions defined by the intersection of a ball with a tetrahedron or hexahedron.

Extension of our methodology to problems for which discontinuities in the solution evolve in time also poses implementation difficulties. For one thing, our abruptly refined meshes would have to also evolve in time so that the thin elements follow the discontinuity. Fortunately, one would not have to start from scratch at each time step, but instead locally adjust the mesh from the previous time step.

The models as well the time-independent case investigated in the paper provide a rather simple setting within computational mechanics and in particular, pose little difficulty with regard to adaptive mesh refinement. However, this would not be the case for more complex constitutive models involving, e.g., model transfer such as determining state variables at newly-created material evaluation points. This is another interesting direction for future work.

Acknowledgment

The financial support by the U.S. National Science Foundation under Grant Number DMS1315259 is gratefully acknowledged.

References

- [1] F. Bobaru, J.T. Foster, P.H. Geubelle, S.A. Silling, *Handbook of Peridynamic Modeling*, CRC Press, 2015.
- [2] M. Breitenfeld, P. Geubelle, O. Weckner, S. Silling, Non-ordinary state-based peridynamic analysis of stationary crack problems, *Comput. Methods Appl. Mech. Engrg.* 272 (2014) 233–250.
- [3] Y.D. Ha, F. Bobaru, Characteristics of dynamic brittle fracture captured with peridynamics, *Eng. Fract. Mech.* 78 (6) (2011) 1156–1168.
- [4] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, *J. Mech. Phys. Solids* 48 (1) (2000) 175–209.
- [5] S.A. Silling, F. Bobaru, Peridynamic modeling of membranes and fibers, *Int. J. Non-Linear Mech.* 40 (2) (2005) 395–409.
- [6] S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling, *J. Elasticity* 88 (2) (2007) 151–184.
- [7] X. Chen, M. Gunzburger, Continuous and discontinuous finite element methods for a peridynamics model of mechanics, *Comput. Methods Appl. Mech. Engrg.* 200 (9) (2011) 1237–1250.
- [8] F. Xu, M. Gunzburger, J. Burkardt, Q. Du, A multiscale implementation based on adaptive mesh refinement for the nonlocal peridynamics model in one dimension, *Multiscale Model. Simul.* 14 (1) (2016) 398–429.
- [9] Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, Analysis and approximation of nonlocal diffusion problems with volume constraints, *SIAM Rev.* 54 (4) (2012) 667–696.
- [10] Q. Du, M. Gunzburger, R. Lehoucq, K. Zhou, A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws, *Math. Models Methods Appl. Sci.* 23 (03) (2013) 493–540.
- [11] Q. Du, M. Gunzburger, Grid generation and optimization based on centroidal voronoi tessellations, *Appl. Math. Comput.* 133 (2) (2002) 591–607.
- [12] Q. Du, M.D. Gunzburger, L. Ju, Constrained centroidal voronoi tessellations for surfaces, *SIAM J. Sci. Comput.* 24 (5) (2003) 1488–1506.
- [13] H. Nguyen, J. Burkardt, M. Gunzburger, L. Ju, Y. Saka, Constrained CVT meshes and a comparison of triangular mesh generators, *Comput. Geom.* 42 (1) (2009) 1–19.
- [14] G. Da Fies, M. Vianello, Trigonometric gaussian quadrature on subintervals of the period, *Electro. Trans. Numer. Anal.* 39 (2012) 102–112.
- [15] G. Da Fies, M. Vianello, Numerical quadrature produced by the CAA group: multivariate interpolation and quadrature, <http://www.math.unipd.it/~marcov/CAAssoft.html>.
- [16] J. Burkardt, CIRCLE SEGMENT: Area, Height, Angle, Sampling and Quadrature, http://people.sc.fsu.edu/~jburkardt/m.src/circle_segment/circle_segment.html.
- [17] P. Seleson, Improved one-point quadrature algorithms for two-dimensional peridynamic models based on analytical calculations, *Comput. Methods Appl. Mech. Engrg.* 282 (2014) 184–217.
- [18] A. Aimi, A. Carini, M. Diligenti, G. Monegato, Numerical integration schemes for evaluation of (hyper) singular integrals in 2d bem, *Computational mechanics* 22 (1) (1998) 1–11.
- [19] B. Aksoylu, Z. Unlu, Conditioning analysis of nonlocal integral operators in fractional sobolev spaces, *SIAM J. Numer. Anal.* 52 (2) (2014) 653–677.

- [20] G. Vainikko, I. Lifanov, On the notion of the finite part of divergent integrals in integral equations, *Differential Equations* 38 (9) (2002) 1313–1326.
- [21] X. Tian, Q. Du, Analysis and comparison of different approximations to nonlocal diffusion and linear peridynamic equations, *SIAM J. Numer. Anal.* 51 (6) (2013) 3458–3482.
- [22] X. Tian, Q. Du, Asymptotically compatible schemes and applications to robust discretization of nonlocal models, *SIAM J. Numer. Anal.* 52 (4) (2014) 1641–1665.