An algorithm for imposing local boundary conditions in peridynamic models of diffusion on arbitrary domains

Jiangming Zhao[a], Siavash Jafarzadeh[a], Ziguang Chen[b, c], Florin Bobaru[a,*]

[a] Department of Mechanical & Materials Engineering, University of Nebraska-Lincoln, Lincoln, NE, 68588-0526, USA

[b] Department of Mechanics, Huazhong University of Science and Technology, Wuhan 430074, China

[c] Hubei Key Laboratory of Engineering Structural Analysis and Safety Assessment, 1037 Luoyu Road, 430074, Wuhan, China

Abstract

Imposing local boundary conditions in nonlocal/peridynamic models is often desired/needed. Fictitious nodes methods (FNMs) are commonly used techniques for this purpose but they are limited, in general, to domains with simple geometry. FNMs also mitigate the well-known peridynamic surface/skin effect at boundaries/surfaces. Here, we introduce a general algorithm that automatically locates mirror nodes for fictitious nodes in the mirror-based FNM, without requiring an explicit mathematical description of the boundary. The algorithm is based on computing a nonlocal gradient, at fictitious nodes, to determine the "generalized" normal direction to the boundary of a domain with arbitrary geometry. We test several FNMs on peridynamic diffusion problems with or without singularities, that exist in the corresponding local models, along the boundaries. We find that the mirror-based FNM works best in agreeing with the classical solutions. We then test the new algorithm with this FNM for diffusion problems in domains with complex geometries, including one with intersecting cracks. The algorithm is general and should work for any type of peridynamic model, including those for problems with moving boundaries and growing cracks, for which enforcement of local-type boundary conditions is desired. Since high accuracy is critical near boundaries of arbitrary shape (including corners, notches, crack tips) in a variety of problems, the new algorithm has potential for high impact.

Keywords

Boundary conditions; Diffusion; Fictitious nodes method; Nonlocal gradient; Peridynamics; Surface effect.

1. Introduction

Two kinds of singularities usually appear in classical PDEs: angular and interface singularities [1]. Angular singularities occur at reentrant corners (including crack tips), while interface singularity happens due to sudden jumps in a boundary condition (BC). While angular singularities are more commonly observed in models of physical systems, there are certain problems in which both may exist. For example, in galvanic corrosion [2], there is a sudden change of boundary condition at the interface between anode and cathode, and at the same location, a reentrant corner forms due to localized corrosion. They lead to singularities in the classical formulations of the Laplace/Poisson equation which governs the electro-potential distribution

* Corresponding author.
E-mail address: fbobaru2@unl.edu (F. Bobaru).

in the solution domain. Classical methods deal with singularities by special treatments such as conforming transformation method [3], local refinement [4,5], singular function methods [6–8], combined methods [9,10], etc. These methods, while working fine for simple cases, have limitations on problems with moving singularities such as crack propagation and corrosion.

The peridynamic (PD) model was introduced by Silling in 2000 [11] and it allows for a natural treatment of discontinuities/singularities by employing integration, over a nonlocal region called the horizon region, rather than differentiation. For a nonlocal formulation, associated BCs are of nonlocal type as well, and they are sometimes referred to as "nonlocal volume constraints" [12,13]. In reality, however, conditions to be imposed (on values and/or derivatives of the unknown function) are known (measurable) only at the surface of a body, not through a finite thickness layer at the surface. The natural representation of such measurement-based conditions is via local boundary conditions. Therefore, imposing local BCs in nonlocal/peridynamic models is often desired/needed. Fictitious nodes methods (FNMs) [14–17] are commonly used to enforce local boundary conditions in PD models. Other names for fictitious nodes are: "dummy particles", "ghost particles", or "imaginary particles" [18,19]. When applied to free boundaries (or crack surfaces) they can also mitigate the PD surface effect [20]. The PD surface effect appears because, unlike in the bulk, points near the surface/boundary do not have a full nonlocal neighborhood. One of the FNMs, which uses mirror nodes and thus is called the mirror-based FNM, has been proven to work efficiently in problems with simple geometries [16,17,21]. For arbitrary geometries such as those with curved boundaries, kinks, corners and cracks, however, the method is difficult to implement because there is no general algorithm to find the mirror nodes (of all fictitious nodes) required by the method.

In this work, we introduce a new algorithm that helps us automatically find mirror nodes in the mirror-based FNM for domains of arbitrary geometry, including for those with cracks. This algorithm approximates the nonlocal normal vector along the boundary for any geometry by computing the PD gradient of the "index" function. The index function, for a generic point $\boldsymbol{x}$, is the proportion of the area of $\boldsymbol{x}$'s horizon that is inside the domain and connected to $\boldsymbol{x}$, to the full area (or volume in 3D) of $\boldsymbol{x}$'s horizon. Note that parts of the horizon region may fall outside of the domain but they are counted in the "full area" of the horizon region. We select the PD diffusion model to test our algorithm. With small modifications, the algorithm presented here is also applicable to PD models solving other types of problems, such as fracture and corrosion damage. We also investigate two other types of FNMs and compare their results with those from the mirror-based FNM: the "naive" FNM [22], and the Taylor FNM [18,23,24]. We compare the performance of different FNMs in enforcing local boundary conditions using two problems: one is a simple problem without singularity and the other is the Motz's problem with a singularity (in fluxes) along a boundary where Dirichlet and Neumann boundary conditions meet [3,25]. We show the capability of the new algorithm, used in the context of mirror-based FNM, by solving examples of diffusion in domains with crack surfaces and curved boundaries.

This paper is organized as follows: in Section 2 we review the PD method for diffusion-type problems; in Section 3 we discuss the fictitious nodes method and introduce the autonomous algorithm to generalize the mirror-based FNM for arbitrary geometries; in Section 4 we compare the performance of three different types of FNMs using examples with and without local singularities, then test the generality and capability of the developed algorithm on the mirror-based FNM for more complicated domains (with cracks); conclusions are given in Section 5.

2. The peridynamic model for diffusion

The peridynamic (PD) theory is a nonlocal extension of the classical continuum mechanics [11]. The PD formulation is in the form of integro-differential equations (IDEs) rather than PDEs used in classical local theories. In PD, discontinuities such as cracks/damages can initiate and propagate naturally and autonomously [15,26–28]. While PD has been primarily used to deal with mechanical behaviors [26,27,29–32], it has also been employed in diffusion-type problems involving cracks and damages, including thermal diffusion [14,33–35] and mass transport (e.g. corrosion) [36–42]. In peridynamics, each spatial point $\boldsymbol{x}$ interacts with other points $\hat{\boldsymbol{x}}$ within its neighborhood $\mathcal{H}_{\boldsymbol{x}}$ which is called the horizon region of $\boldsymbol{x}$ and is usually selected to be a disk in 2D (or sphere in 3D) centered at $\boldsymbol{x}$. The radius of $\mathcal{H}_{\boldsymbol{x}}$ is called horizon size (or simply, the horizon) and denoted by δ. Objects that carry the pairwise nonlocal interactions between points are called bonds. Fig. 1 schematically shows a peridynamic body with a generic point $\boldsymbol{x}$, its family and its horizon.

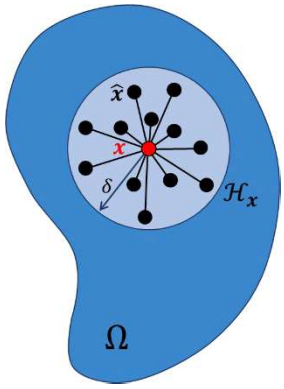

Fig. 1. A peridynamic body with a generic point $\boldsymbol{x}$ and its horizon $\mathcal{H}_{\boldsymbol{x}}$. Nonlocal interactions exist through the bond between two points, e.g., point $\boldsymbol{x}$ and an arbitrary point $\hat{\boldsymbol{x}}$ located in its horizon $\mathcal{H}_{\boldsymbol{x}}$.

In this work we use the bond-based peridynamics for diffusion-type problems. In the context of heat conduction, for example, one assumes that the heat diffusion between two points depends only on their temperature difference. In the more generalized state-based formulation, heat diffusion in a bond is directly influenced by other bonds which connect to the same point [14]. This setting may be beneficial for certain problems but is not considered here.

For diffusion in a homogeneous and isotropic medium with constant diffusivity, the peridynamic transient equation can be written as [12,33,43]:

$$\frac{\partial u(\boldsymbol{x},t)}{\partial t} = \nu \mathcal{L}_{\mu} u(\boldsymbol{x},t) + s(\boldsymbol{x},t) \tag{1}$$

where $u(\boldsymbol{x},t)$ is the target quantity (e.g. temperature, concentration, electric potential, etc.) at point $\boldsymbol{x}$ at time t, ν is the scalar diffusivity constant, s is the source/sink term and $\mathcal{L}_{\mu}$ is the PD Laplacian operator which can be expressed as:

$$\mathcal{L}_{\mu} u(\boldsymbol{x},t) = \int_{\mathcal{H}_{\boldsymbol{x}}} \mu(\hat{\boldsymbol{x}}-\boldsymbol{x})[u(\hat{\boldsymbol{x}},t) - u(\boldsymbol{x},t)]\mathrm{d}V_{\hat{\boldsymbol{x}}}\,, \tag{2}$$

where the kernel function μ is nonnegative and radially symmetric, i.e. $\mu(\hat{\boldsymbol{x}} - \boldsymbol{x}) = \mu(\boldsymbol{x} - \hat{\boldsymbol{x}}) \geq 0$ and $\mathrm{d}V_{\hat{\boldsymbol{x}}}$ is the differential volume (area in 2D, length in 1D) of material point $\hat{\boldsymbol{x}}$.

It has been shown that as the horizon approaches zero, in a continuous domain, the peridynamic solution to a diffusion problem converges to the solution given by the classical Fick's law of diffusion, under certain conditions [22,44,45]. A kernel function that leads to good convergence properties with the one-point Gaussian quadrature discretization of the spatial domain is [22]:

$$\mu(\hat{\boldsymbol{x}} - \boldsymbol{x}) = \frac{A}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2} \tag{3}$$

where A is a constant calibrated by matching PD flux to classical flux for a linearly distributed field. Other options for the kernel function (see [22]) can be selected, but they are not considered here.

The steady state of Eq. (1) without sources/sinks is:

$$\int_{\mathcal{H}_x} \frac{u(\hat{\boldsymbol{x}}, t) - u(\boldsymbol{x}, t)}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|^2} \mathrm{d}V_{\hat{\boldsymbol{x}}} = 0 \tag{4}$$

For a specific problem, nonlocal boundary conditions (sometimes called "volume-constraints" [12,13]) are required to solve Eq. (4). In the next section we discuss local and nonlocal boundary conditions in peridynamics.

3. Fictitious nodes methods to enforce local-type boundary conditions and mitigate the peridynamic surface effect

Unlike classical local methods, the boundary conditions in peridynamics are nonlocal. However, when solving practical problems, imposing local-type boundary conditions in nonlocal/peridynamic models is usually desired/needed because, in reality, conditions (on the unknown function values or its flux) are imposed at the surfaces of a body, not through a finite layer near the surface. The natural representation of such conditions (based on measurements) is via local boundary conditions. Various methods to impose local boundary conditions in PD models have been investigated in [13,14,46]. One such method is the fictitious nodes method (FNM) [14,16,46]. In FNM for peridynamics, instead of imposing local BCs at boundary points, certain volume constraints are specified on fictitious points/nodes in Γ (the "collar" outside of the body shown in Fig. 2, called the fictitious region), so that desired local boundary conditions on $\partial\Omega$ are satisfied or approximately satisfied. The fictitious region is the smallest region that assures full horizon for all domain points. Such volume-constrained peridynamic problems are defined analogous to boundary value problems with PDEs in the local theory according to nonlocal vector calculus [13]. Volume constrained peridynamic steady-state diffusion (or Laplace) problem can be expressed as [13]:

$$\begin{cases} \mathcal{L}_\mu u(\boldsymbol{x}, t) = 0 & \boldsymbol{x} \in \Omega \\ G(u) = 0 \quad \text{(volume constraints)} & \boldsymbol{x} \in \Gamma \end{cases} \tag{5}$$

where G is a function that prescribes the constraints of u on Γ. Fig. 2 schematically shows the solution domain Ω, its boundary $\partial\Omega$, and the constrained volume or fictitious region, Γ.

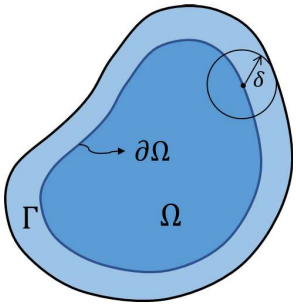

Fig. 2. Schematic of a peridynamic domain (Ω), its boundary ($\partial\Omega$), and its constrained volume (or fictitious region, Γ).

When applied to free boundaries (or crack surfaces), the FNM mitigates the peridynamic surface effect [16]. This is a common issue for nonlocal methods as well as local methods with certain nonlocal treatments such as the SPH method [19,23]. The surface effect appears because, unlike in the bulk, material points near the surface/boundary do not have a full non-local neighborhood. The surface effect leads to slightly different behavior of material points near the surface compared with those in the bulk. This could mean a lower diffusivity near the surface for diffusion problems, and a lower stiffness for problems in elasticity (see [20]). These effects are reduced as one decreases the horizon size and would be "exact" when the size of the neighborhood is the same as the physical nonlocal interaction range, which could be atomistically small. In practical modeling, the size of nonlocal interaction is usually set to match observable physical length-scales (see discussion in [47]), and not larger than relevant geometrical features of the domain (notch widths, etc.).

In this section, we briefly review three different types of FNMs from literature and then introduce the new algorithm for the mirror-based FNM to make it applicable to arbitrary geometries. Comparisons between their performances are shown in Section 4.

3.1. Naïve FNM

A naïve type of FNM is often used in the literature to impose local Dirichlet and homogeneous Neumann (no flux) BCs. This method enforces Dirichlet BCs by assigning the same values to all fictitious points corresponding to a boundary point, while homogeneous Neumann BCs are enforced by simply neglecting all corresponding fictitious points [22]. See Fig. 3 for an illustration of how a Dirichlet BC $u(\boldsymbol{x}, t) = u_b$ for $\boldsymbol{x} \in \partial\Omega_{\mathrm{D}}$ (boundary subjected to Dirichlet BC) is enforced at each fictitious point $\boldsymbol{x}_\Gamma$. An even simpler version of "enforcing" local Dirichlet BCs in the nonlocal model is to dispense entirely with the fictitious points and subject only the surface points to the values of local boundary conditions [33]. Previous work has shown that results by both versions converge to local solutions as the horizon size approaches zero [22,35]. The naïve FNM has the advantage of featuring the simplest implementation and is the most efficient (see Section 4.1). Note, however, that a jump-discontinuity in the solution at the boundary may be generated, leading to possible errors in fluxes near the boundary (see the result in section 4.1 for an example).

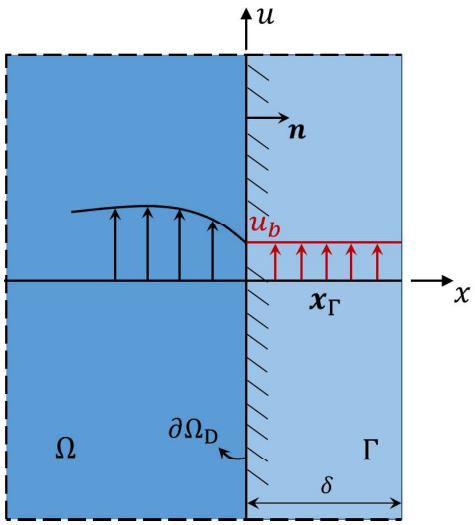

Fig. 3. Illustration of enforcing the local Dirichlet BC in the naïve FNM (redrawn from [19]).

3.2. Taylor-based FNM

The second FNM, used in the PD context first in [24], requires a Taylor expansion (to linear terms) for points in the horizon of the boundary. We call it Taylor-based FNM or simply Taylor FNM. To impose a local Dirichlet BC $u(\boldsymbol{x},t)=u_{\mathrm{D}}(\boldsymbol{x})$ for $\boldsymbol{x}\in\partial\Omega_{\mathrm{D}}$ (boundary subjected to Dirichlet BC), the constraint value at a generic fictitious point $\boldsymbol{x}_\Gamma$, which is located inside the horizon of an generic point $\boldsymbol{x}_\Omega$ in the domain with $\overrightarrow{\boldsymbol{x}_\Omega\boldsymbol{x}_\Gamma}$ intersecting the boundary at $\boldsymbol{x}_b$, is extrapolated from $\boldsymbol{x}_\Omega$ as:

$$\begin{aligned} u_\Gamma &= u_\Omega+\left(\frac{\partial u}{\partial \boldsymbol{n}}\right)_{\boldsymbol{x}_b}(d_\Omega+d_\Gamma)+\mathcal{O}(\Delta x^2)\approx u_\Omega+\frac{(u_b-u_\Omega)}{d_\Omega}(d_\Omega+d_\Gamma)\\ &=\left(1+\frac{d_\Gamma}{d_\Omega}\right)u_b-\frac{d_\Gamma}{d_\Omega}u_\Omega \end{aligned} \tag{6}$$

where u_Ω represents $u(\boldsymbol{x}_\Omega,t)$, u_Γ represents $u(\boldsymbol{x}_\Gamma,t)$, d_Ω and d_Γ represent the distance of point $\boldsymbol{x}_\Omega$ and $\boldsymbol{x}_\Gamma$ to the boundary, respectively. Notice that with this approach, the imposed constraint at a fictitious point $\boldsymbol{x}_\Gamma$ has to be computed for each corresponding domain point $\boldsymbol{x}_\Omega$ anew, and this means that such value may change during computations at the same solution step (see Fig. 4, where u_Γ^Ω is the distribution of $u(\boldsymbol{x})$ for $\boldsymbol{x}\in\mathcal{H}_{\boldsymbol{x}_\Omega}\cap\Gamma$, and is different for each $\boldsymbol{x}_\Omega$). The solution step refers to each call to the Conjugate Gradient (CG) solver, see Fig. A.2 in Appendix A for the flowchart of the simulation.

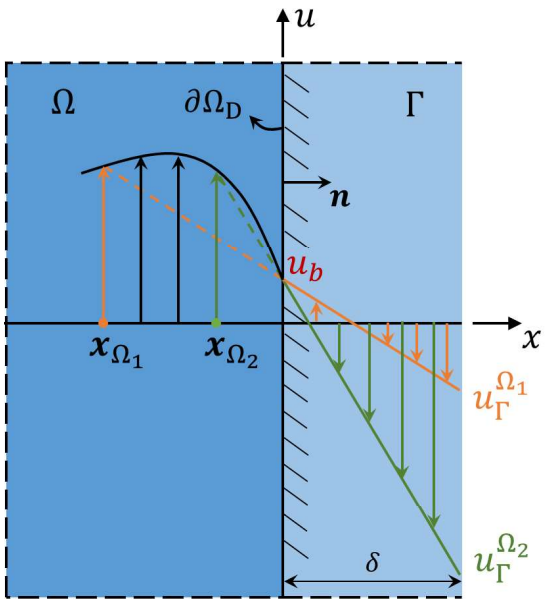

Fig. 4. Illustration of enforcing a local Dirichlet BC $u(\boldsymbol{x},t)=u_b$ for $\boldsymbol{x}\in\partial\Omega_{\mathrm{D}}$ with the Taylor FNM (redrawn from [19]).

Eq. (6) may lead to unstable results if implemented directly. A modification is required as follows [23]:

$$u_\Gamma = u_\Omega + (u_b - u_\Omega)\times \min\left\{\lambda, \left(1+\frac{d_\Gamma}{d_\Omega}\right)\right\} \tag{7}$$

where λ is a parameter determined from numerical tests. Normally λ =1.5 leads to good results [23].

To impose a local Neumann or Robin BC $\boldsymbol{\nabla} u(\boldsymbol{x},t)\cdot\boldsymbol{n}=f\big(u(\boldsymbol{x},t)\big)$ for $\boldsymbol{x}\in\partial\Omega_{\mathrm{NR}}$ (boundary subjected to Neumann or Robin BC), where f is a given function, we can first relate values at a generic domain point $\boldsymbol{x}_\Omega$ and a generic fictitious points $\boldsymbol{x}_\Gamma$ (inside the horizon of $\boldsymbol{x}_\Omega$) to the BC by the following Taylor expansions to linear terms around $\boldsymbol{x}_b$ (intersection of $\overrightarrow{\boldsymbol{x}_\Omega\boldsymbol{x}_\Gamma}$ and $\partial\Omega_{\mathrm{NR}}$) [24]:

$$\begin{aligned} u_\Omega &= u_b + d_\Omega\left(\frac{\partial u}{\partial \boldsymbol{n}}\right)_{\boldsymbol{x}_b} + \mathcal{O}(\Delta x^2) \approx u_b + d_\Omega f(u_b) \\ u_\Gamma &= u_b - d_\Gamma\left(\frac{\partial u}{\partial \boldsymbol{n}}\right)_{\boldsymbol{x}_b} + \mathcal{O}(\Delta x^2) \approx u_b - d_\Gamma f(u_b) \end{aligned} \tag{8}$$

Note that, since $\boldsymbol{x}_b\in\partial\Omega_{\mathrm{NR}}$, $u_b=u(\boldsymbol{x}_b,t)$ is not given and needs to be computed. By subtracting these two equations we have:

$$u_\Gamma \approx u_\Omega - (d_\Omega + d_\Gamma) f(u_b) \tag{9}$$

where u_b is approximated from the first equation in Eq. (8) and then substituted into Eq. (9) to solve for u_Γ. Notice that if $f(u_b)$ is a nonlinear function, a nonlinear equation solver, such as Newton's method, is needed to solve for u_b [48].

Notice that, if $d_\Omega = d_\Gamma$ in Eqs. (6) and (9), the Taylor-based FNM degenerates to the mirror-based FNM which will be introduced in the next section.

3.3. Mirror-based FNM

Another FNM, referred to as the mirror-based FNM or simply "mirror FNM", assigns the constraint at each fictitious point based on the current solution of its mirror point in the domain. Two points are a pair of mirror points if they are "symmetric" to the boundary. In this method, one assumes that for each fictitious point $\boldsymbol{x}_\Gamma$, its mirror point $\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}$ is given [14] as shown in Fig. 5 where $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ is the mid-point of $\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}$ sitting on the boundary. When the boundary is smooth in a neighborhood (of size that depends on geometry) of $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$, $\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}$ is collinear with the normal vector $\boldsymbol{n}_b$ to the boundary at $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$.

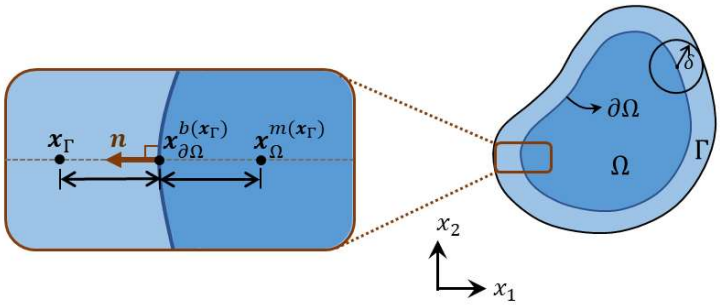

Fig. 5. Mirror-based FNM for imposing local BCs on $\partial\Omega$ by assigning certain constraints at fictitious points in Γ based on current values at their mirror points inside Ω, and the value of the boundary condition to be imposed.

To enforce the local Dirichlet BC $u(\boldsymbol{x},t) = u_{\mathrm{D}}(\boldsymbol{x})$ for $\boldsymbol{x} \in \partial\Omega_{\mathrm{D}}$ and the Neumann BC $\boldsymbol{\nabla_n} u(\boldsymbol{x},t) = q_{\mathrm{N}}(\boldsymbol{x})$ for $\boldsymbol{x} \in \partial\Omega_{\mathrm{N}}$ in the mirror FNM, the constraint at a generic fictitious point $\boldsymbol{x}_\Gamma$ is written as:

$$u_\Gamma = 2u_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} - u_\Omega^{m(\boldsymbol{x}_\Gamma)} \tag{10}$$

and:

$$u_\Gamma = u_\Omega^{m(\boldsymbol{x}_\Gamma)} - \left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}\right\| q_{\mathrm{N}}(\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}) \tag{11}$$

respectively, where $u_\Gamma = u(\boldsymbol{x}_\Gamma, t)$, $u_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} = u\left(\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}, t\right)$, $u_\Omega^{m(\boldsymbol{x}_\Gamma)} = u\left(\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}, t\right)$, and $\left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}\right\| = \left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}}\right\| + \left\|\overrightarrow{\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}\right\| = 2\left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}}\right\| = 2\left\|\overrightarrow{\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}\right\|$ is the distance between $\boldsymbol{x}_\Gamma$ and $\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}$. The local Robin BC $\boldsymbol{\nabla_n} u(\boldsymbol{x},t) = f(u(\boldsymbol{x},t))$ for $\boldsymbol{x} \in \partial\Omega_{\mathrm{R}}$ is enforced on fictitious points as:

$$u_\Gamma = u_\Omega^{m(\boldsymbol{x}_\Gamma)} - \left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}\right\| f(u_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}) = u_\Omega^{m(\boldsymbol{x}_\Gamma)} - \left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}}\right\| f\left(\frac{u_\Gamma + u_\Omega^{m(\boldsymbol{x}_\Gamma)}}{2}\right) \tag{12}$$

in which the approximation $u_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} = \frac{u_\Gamma + u_\Omega^{m(\boldsymbol{x}_\Gamma)}}{2}$ is made according to Eq. (10), and u_Γ requires to be solved using a nonlinear solver if function f is nonlinear.

In both the Taylor and the mirror FNMs, constraints applied at fictitious points vary with the solution step (or time step, see the flowchart in Fig. A.2.). However, in the same solution step, the constraint at each fictitious point in the Taylor FNM may change with the domain point at which a computation is performed.

For the mirror FNM, this does not change because it only depends on the current solution at its mirror point in the domain (uniquely determined by the algorithm) and the value of the imposed boundary condition (at the intersection point between the boundary and the segment connecting the fictitious point and its mirror point, which is the mid-point on the segment). See Fig. 4 and Fig. 6 for illustrations of how local Dirichlet BCs are enforced in the Taylor and the mirror FNMs, respectively.

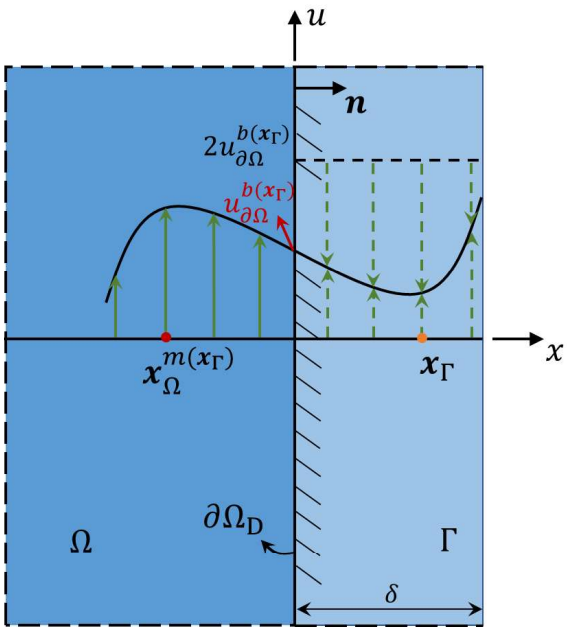

Fig. 6. Illustration of enforcing a local Dirichlet BC in the mirror FNM (redrawn from [14]).

Important Remark: note that in the original Taylor and mirror FNMs in PD, mirror pairs are considered given as inputs, which works for simple geometries. However, for the general case, to find mirror pairs one needs to find the distance between each point and the boundary, and, for each fictitious point, find the mirror point in the domain. While providing such information is simple and straightforward for simple geometries, it is not so for arbitrary ones, for example shapes with corners of various angles, cusps and crack tips, etc. At such locations, the tangent line (and normal vector) may not be well defined. While some ad-hoc choices can be made for overcoming this issue, we aim for a general algorithm that generates a well-defined "generalized" normal vector to the boundary at any of its points, regular or not, so that both the mirror point and the distance from the point to the boundary can be determined without ambiguity. In the next section we introduce such an automatic algorithm so that local BCs can be enforced in PD models using the mirror FNM for arbitrary geometries.

3.4. An algorithm to find mirror points using the peridynamic gradient

Using the mirror FNM for arbitrarily shaped boundaries (with corners, cusps, etc.) requires finding mirror points for all fictitious points. For surfaces of thin cracks, domain points (within a distance δ from the crack surface) on one side of the crack surface can serve as fictitious points for those on the other side. In what follows, $\boldsymbol{x}_\Gamma$ includes all points in the fictitious region Γ as well as domain points (acting as fictitious points) that are required to enforce local boundary conditions.

As shown in Fig. 5, $\boldsymbol{x}_{\Omega}^{m(\boldsymbol{x}_\Gamma)}$ can be determined by $\boldsymbol{x}_{\Omega}^{m(\boldsymbol{x}_\Gamma)} = \boldsymbol{x}_\Gamma + 2(\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} - \boldsymbol{x}_\Gamma)$ once $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$, the mid-point of the segment connecting $\boldsymbol{x}_{\Omega}^{m(\boldsymbol{x}_\Gamma)}$ and $\boldsymbol{x}_\Gamma$, is known. Therefore, we first need to locate $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$. Starting from $\boldsymbol{x}_\Gamma$, one could determine $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ by searching the family points of $\boldsymbol{x}_\Gamma$ for the domain point closest to $\boldsymbol{x}_\Gamma$. However, such a domain point may not be unique. In addition, for fictitious points near corners with an angle smaller than 90° (see regions Γ_1 in Fig. 7), this strategy may find mirror points which are not in the domain. Special treatments would have to be used for these instances, which reduces the generality of the algorithm.

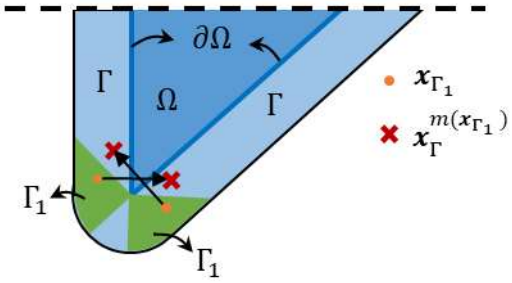

Fig. 7. Schematic of fictitious regions near an acute corner for which mirror points to some fictitious points are not located inside the domain, if $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ is determined to be the domain point closest to $\boldsymbol{x}_\Gamma$.

We introduce a more general approach to compute $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ by computing a modified nonlocal gradient (unit) vector at a fictitious point $\boldsymbol{x}_\Gamma$, $\boldsymbol{n}_\Gamma = \boldsymbol{n}(\boldsymbol{x}_\Gamma, t)$, which will allow us to locate a unique $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$. For smooth boundaries, $\boldsymbol{n}_\Gamma$ has the same direction with the normal to the boundary at the corresponding location on the boundary. For non-smooth boundaries in the horizon of $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ (e.g. at corners, kinks, etc.), $\boldsymbol{n}_\Gamma$ may or may not be collinear with the outward normal vector $\boldsymbol{n}_b$ (which may or may not exist) to the domain boundary.

To compute $\boldsymbol{n}_\Gamma$ in the peridynamic framework, without a mathematical representation of the boundary, we first compute the peridynamic gradient (or flux) [34], at a fictitious point, defined by:

$$\nabla_{\mathrm{PD}} r(\boldsymbol{x}_\Gamma, t) = \int_{\mathcal{H}_{\boldsymbol{x}_\Gamma}} \frac{\widehat{\boldsymbol{x}} - \boldsymbol{x}_\Gamma}{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}_\Gamma\|^2} [r(\widehat{\boldsymbol{x}}, t) - r(\boldsymbol{x}_\Gamma, t)] \mathrm{d}V_{\widehat{\boldsymbol{x}}} \tag{13}$$

where $\mathcal{H}_{\boldsymbol{x}_\Gamma}$ is $\boldsymbol{x}_\Gamma$'s horizon, and $r(\boldsymbol{x}, t)$ is the index function given by:

$$r(\boldsymbol{x}, t) = \frac{\int_{\mathcal{H}_{\boldsymbol{x}}^{\Omega}} \mu(\widehat{\boldsymbol{x}}, \boldsymbol{x}) \mathrm{d}V_{\widehat{\boldsymbol{x}}}}{\int_{\mathcal{H}_{\boldsymbol{x}}} \mathrm{d}V_{\widehat{\boldsymbol{x}}}} \tag{14}$$

where $\mu(\widehat{\boldsymbol{x}}, \boldsymbol{x})$ is the binary connectivity function which equals 1 if $\boldsymbol{x}$ is connected to $\widehat{\boldsymbol{x}}$, and 0 otherwise. Thus, $r(\boldsymbol{x}, t)$ is the ratio of the area of $\mathcal{H}_{\boldsymbol{x}}$ that is inside the domain and connected to $\boldsymbol{x}$, to the total area (or volume in 3D) of $\mathcal{H}_{\boldsymbol{x}}$. Although $r(\boldsymbol{x}, t)$ is dependent on time, it changes only when the domain changes. See Fig. 8 for an illustration of $r(\boldsymbol{x}, t)$.

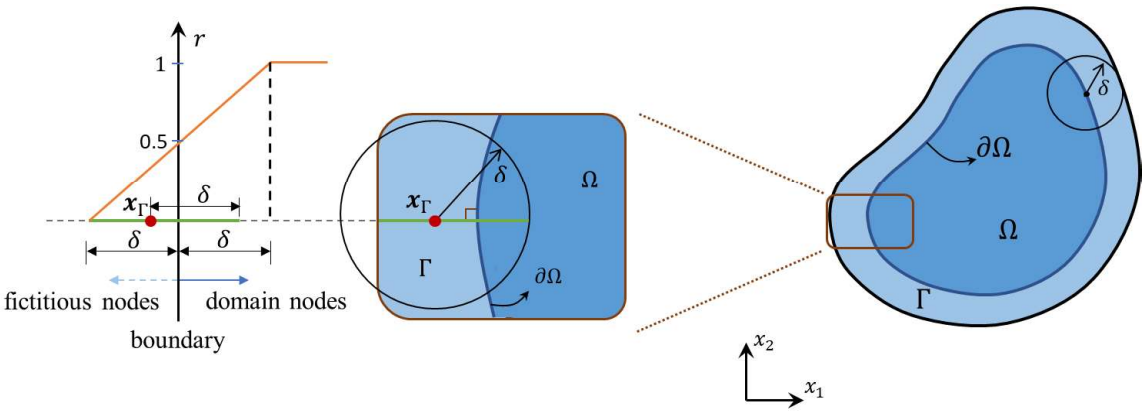

Fig. 8. Index function $r(\boldsymbol{x}, t)$ for points near the boundary. Notice that the function does not have to be linear as shown in the picture.

We note that $r(\boldsymbol{x}, t)$ as $\boldsymbol{x}$ crosses a thin crack is an inverted hat function, instead of the monotonous function shown in Fig. 8 for a generic boundary. In such a case, its peridynamic gradient formulated by Eq. (13) will not guarantee a direction equivalent to the outward normal to the crack surface, which is what we want to have it represent. Modifications of Eq. (13) are then required to resolve this issue as follows:

$$\nabla_{\mathrm{PD}} r(\boldsymbol{x}_\Gamma, t) = \int_{\mathcal{H}_{\boldsymbol{x}_\Gamma}^{\Omega}} \frac{\widehat{\boldsymbol{x}} - \boldsymbol{x}_\Gamma}{\|\widehat{\boldsymbol{x}} - \boldsymbol{x}_\Gamma\|^2} r(\widehat{\boldsymbol{x}}, t) \mathrm{d}V_{\widehat{\boldsymbol{x}}} \tag{15}$$

where $\mathcal{H}_{\boldsymbol{x}_\Gamma}^{\Omega} = \mathcal{H}_{\boldsymbol{x}_\Gamma} \cap \Omega$. For domain points (acting as fictitious points) within a distance δ from a crack surface, $\mathcal{H}_{\boldsymbol{x}_\Gamma}^{\Omega}$ is the part of $\mathcal{H}_{\boldsymbol{x}_\Gamma}$ that is disconnected from $\boldsymbol{x}_\Gamma$. Then $\boldsymbol{n}_\Gamma$ at a fictitious point is obtained by:

$$\boldsymbol{n}_\Gamma = \frac{\nabla_{\mathrm{PD}} r(\boldsymbol{x}_\Gamma, t)}{\|\nabla_{\mathrm{PD}} r(\boldsymbol{x}_\Gamma, t)\|} \tag{16}$$

Notice that $\boldsymbol{n}_\Gamma$ has the opposite sign compared with the external normal at a regular point on the boundary. With $\boldsymbol{n}_\Gamma$ computed, $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ can be determined as the domain point closest to $\boldsymbol{x}_\Gamma$ along $\boldsymbol{n}_\Gamma$ and $\boldsymbol{x}_{\Omega}^{m(\boldsymbol{x}_\Gamma)}$ by $\boldsymbol{x}_\Gamma + 2(\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} - \boldsymbol{x}_\Gamma)$, as shown in Fig. 9.

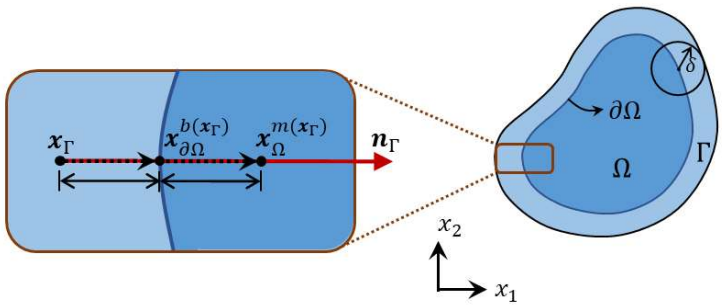

Fig. 9. Schematic diagram for a fictitious point ($\boldsymbol{x}_\Gamma$) and its corresponding projection ($\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$) onto the boundary and mirror point ($\boldsymbol{x}_{\Omega}^{m(\boldsymbol{x}_\Gamma)}$) in the domain, along $\boldsymbol{n}_\Gamma$ direction.

Notice that this algorithm may encounter a problem when finding mirror points at sharp (relative to the horizon size) convex corners, but this only affects some very small portions of the fictitious region (see the green regions in Fig. 10). The issue can be reduced by using smaller horizon sizes but not fully resolved. Approximation of the constraint imposed at such points can be achieved by averaging the constraint of surrounding fictitious points. Another option is to use Naïve FNM for such nodes. A comparison between Fig. 7 and Fig. 10 shows the improvement in this situation when we use the more general normal direction (based on PD gradients) instead of the regular normal to the boundary surfaces.

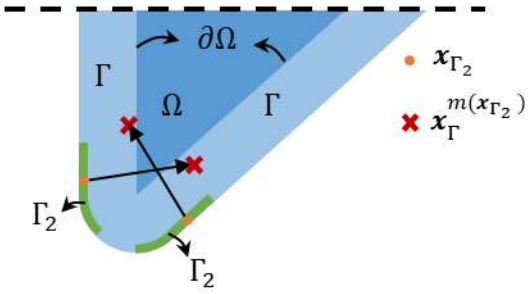

Fig. 10. Schematic of fictitious regions near a convex corner for which mirror points cannot be located appropriately, if $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}$ is determined by the domain point closest to $\boldsymbol{x}_\Gamma$ along $\boldsymbol{n}_\Gamma$.

When implementing the algorithm in the discretized configuration (see Appendix A), as shown in Fig. 11, we first search all the family nodes of $\boldsymbol{x}_\Gamma$ for the closest domain node $\boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}$ to $\boldsymbol{x}_\Gamma$ (or first node with $\mu\left(\boldsymbol{x}_\Gamma, \boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}\right) = 0$ for "fictitious" nodes at crack surface) while the distance between $\boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}$ and $\boldsymbol{n}_\Gamma$ is smaller than $\Delta x/2$. Notice that if a node sits on the boundary, it belongs to the domain node. In certain cases where two such $\boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}$ exist, selecting either one of them is acceptable. Then we have $\boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)} \approx \boldsymbol{x}_\Gamma + (\left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}}\right\| - \Delta x/2)\boldsymbol{n}_\Gamma$ and $\left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}}\right\| = \left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}}\right\| - \Delta x/2$. Ideally, $\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)} = \boldsymbol{q} = \boldsymbol{x}_\Gamma + 2\left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_{\partial\Omega}^{b(\boldsymbol{x}_\Gamma)}}\right\| \boldsymbol{n}_\Gamma$, but since $\boldsymbol{q}$ may not coincide with a node's coordinates, instead, we search the domain node closest to $\boldsymbol{q}$. This implementation works for both uniform and non-uniform grids and $\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}$ converges to the analytical $\boldsymbol{q}$, for a certain choice of δ, as $\delta/\Delta x$ approaches infinity. Since the number of fictitious nodes usually only accounts for a small portion of the total number of nodes, the computational cost to locate mirror nodes for all fictitious nodes is trivial compared with the cost of a complete simulation.

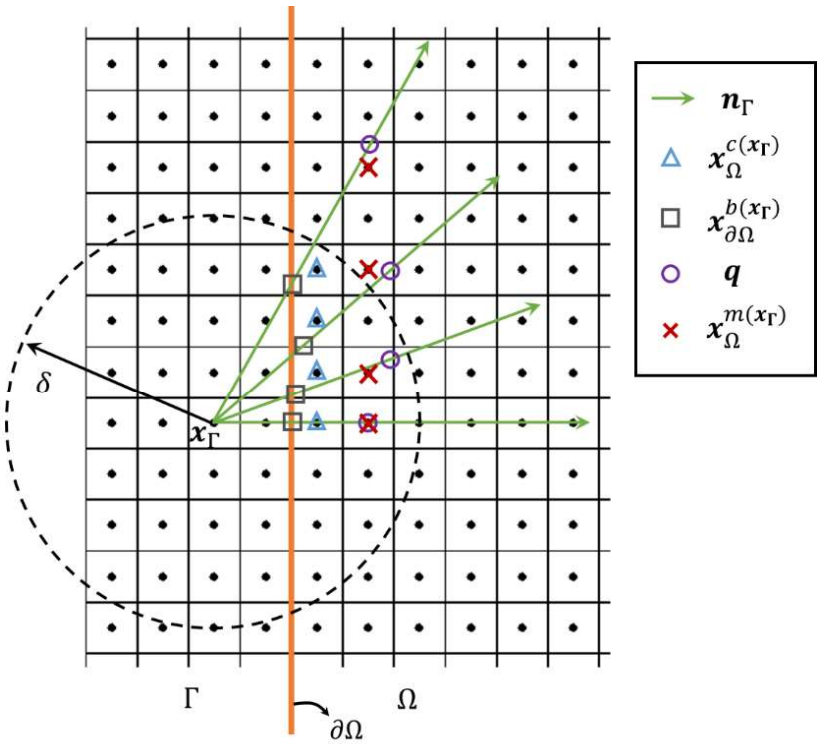

Fig. 11. Skematic diagram for determing $\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}$ for a generic $\boldsymbol{x}_\Gamma$ with four different possible $\boldsymbol{n}_\Gamma$.

The summary of the algorithm's implementation is:

At time t_0, and after any time step at which geometry changes, do:

1) For all nodes $\boldsymbol{x}$ in the domain: compute the index function $r(\boldsymbol{x}, t)$ according to Eq. (14).
2) For all fictitious nodes $\boldsymbol{x}_\Gamma$:
   a. calculate the "generalized" gradient $\boldsymbol{n}_\Gamma$ according to Eq. (15).
   b. compute distances between its family nodes in the domain, and $\boldsymbol{n}_\Gamma$ direction; select only those with this distance smaller than $\Delta x/2$; from this set of nodes, choose the node $\boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}$, closest to $\boldsymbol{x}_\Gamma$.
   c. compute $\boldsymbol{q} = \boldsymbol{x}_\Gamma + (2\left\|\overrightarrow{\boldsymbol{x}_\Gamma \boldsymbol{x}_\Omega^{c(\boldsymbol{x}_\Gamma)}}\right\| - \Delta x)\boldsymbol{n}_\Gamma$, an approximate point to the exact mirror node for $\boldsymbol{x}_\Gamma$; find $\boldsymbol{x}_\Omega^{m(\boldsymbol{x}_\Gamma)}$, the node grid in the domain closest to $\boldsymbol{q}$, by searching in a box near it.

The autonomous algorithm developed works for arbitrary geometries, including corners with different angles, cusps and crack tips, where the peridynamic gradient (or $\boldsymbol{n}_\Gamma$) may not be collinear with $\mathbf{n}_b$. In the original mirror FNM, one treats fictitious nodes at sharp corners either by assuming $\boldsymbol{n}_\Gamma$ to be the average of the normal vector on its two edges [14], or assuming $\boldsymbol{n}_\Gamma$ to be the same as $\mathbf{n}_b$ on one edge [16]. By using the new algorithm, however, the mirror nodes for fictitious nodes at the corner are found automatically and consistently just like the corner is a special case of a curved boundary. As shown in Fig. 12 where those arrows start from fictitious nodes and end at their mirror nodes, $\boldsymbol{n}_\Gamma$ has smooth transitions between the normal vector on two edges of the corner (or crack tip). Not all the arrows are shown in order to illustrate clearly how $\boldsymbol{n}_\Gamma$ transits. The transition zone will narrow down as the horizon shrinks (δ-convergence) and

the transition will be smoother as the ratio of δ and Δx increases (m-convergence). It may be possible that this transition of the peridynamic gradient be used to detect/track crack tips, on the fly, but this idea will be pursued elsewhere.

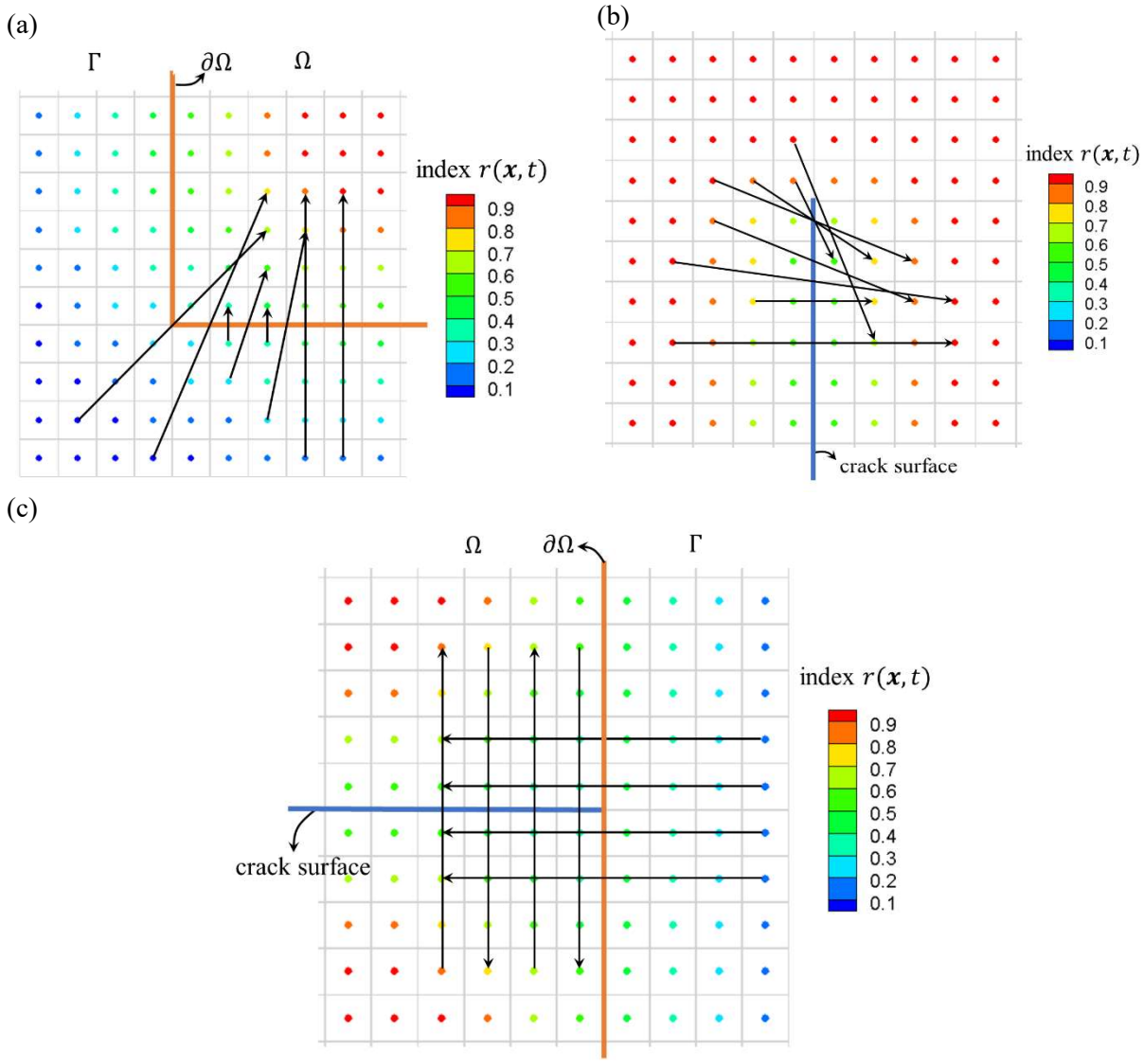

Fig. 12. Ficititious nodes and their mirror nodes (connected by arrows), computed by the new algorithm, for three diffferent configurations: (a) at a corner; (b) at a crack tip; (c) at the edge of a crack.

The algorithm to find mirror nodes can be implemented as a preprocessing subroutine, because mirror nodes do not change if the boundary or crack surface remains unchanged. However, if the solution domain evolves, which happens in problems such as corrosion or fracture, re-finding of mirror nodes is required. Explicit formulas for boundaries or crack surfaces are not required for this algorithm. This algorithm also works for

damage without explicit surfaces, such as distributed damage (a subdomain containing nodes with damage indices smaller than the threshold for a crack surface, e.g. 0.4) [49]). This topic, however, is left for future research.

For problems with complex shapes, the algorithm leads to important time-savings compared to manually determining mirror nodes.

In the following section, the performance of the mirror FNM will be compared with the naïve and the Taylor FNMs. We will also show the capability of the autonomous mirror FNM using problems with curved geometry and cracks.

4. Results and discussion

In this section, we first show how previously discussed three different FNMs perform in enforcing local BCs through the steady-state diffusion in a square domain. Then we will show how these FNMs perform in a problem with a singular point on its boundary. Finally, we are going to show the capability of the autonomous mirror FNM in problems with curved geometry and cracks.

The classical PDE-based formulation for steady-state diffusion problem can be written as:

$$\begin{cases} \nabla^2 u(\boldsymbol{x}) = 0 & \boldsymbol{x} \in \Omega \\ G\big(u(\boldsymbol{x})\big) = 0 & \boldsymbol{x} \in \partial\Omega \end{cases} \tag{17}$$

where $G(u(\boldsymbol{x}))$ defines the boundary conditions (which could be Dirichlet, Neumann, Robin, or mixed). For 2D problems, $\boldsymbol{x} = (x, y)$ in Cartesian coordinates and $\boldsymbol{x} = (\rho, \theta)$ in polar coordinates.

4.1. The performance of the three FNM versions in enforcing local BCs in peridynamics for problems without singularities

In the first problem, we consider a square domain with its side equal to 0.1, subject to the following local boundary conditions:

$$\begin{cases} u(x = -0.05, y) = 0.1 \\ u(x = 0.05, y) = 0.6 \\ \nabla_{\boldsymbol{n}} u(x, y = \pm 0.05) = 0 \end{cases} \tag{18}$$

as shown in Fig. 13. The classical solution to this problem is a linear function written as:

$$u(x, y) = 5x + 0.35 \tag{19}$$

In the PD simulations for this problem, the horizon and grid size are taken to be 0.004 and 0.001, respectively. This leads to about 10,000 nodes for a uniform discretization.

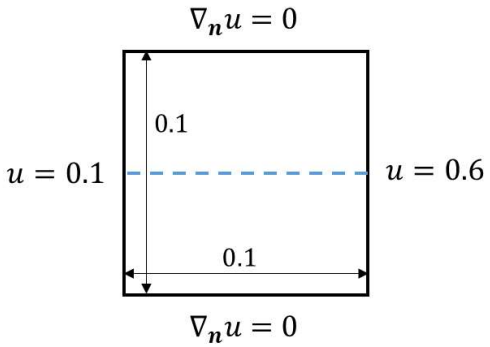

Fig. 13. The geometry and boundary conditions for diffusion in a square domain used to compare different FNMs' capability in enforcing local BCs in the PD formulation.

The solution, obtained with the different FNMs, along the dashed mid-line in Fig. 13 is shown in Fig. 14. All three results match the classical solution very well. However, if we zoom in near the boundary, we can see clearly that the result obtained with the mirror FNM matches the classical analytical solution much better than results obtained with the other two FNMs. The simulation times for the three FNMs are given in Table 1. The simulation with the naïve FNM is at least 10 times faster than other two types of FNM, while the efficiency of Taylor and the mirror FNMs are similar to one another. Therefore, for such a simple problem, if the accuracy of the solution near the boundary is not critical, the naïve FNM could be taken as the "best" option. If high accuracy is needed, then the mirror FNM should be used.

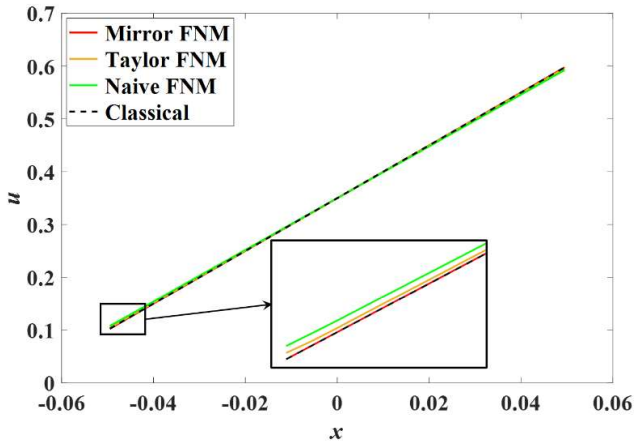

Fig. 14. The classical analytical solution and PD solutions with different FNMs for the steady-state diffusion problem in Fig. 13, along the dashed line shown there.

Table 1

Computational time of the simulation using the three different FNMs, when $\delta = 0.004$ and $\Delta x = 0.001$.

	FNM		
	naïve	Taylor-based	mirror-based
Time (s)	2.0	20.1	20.6

Notice that in [24], the Taylor FNM shows slightly better results than the mirror FNM. This is because for the problem studied in [24], Eq. (7) is not required since there are no stability issues in that case. However, for the example shown in Fig. 13, as well as the next example shown below, the solution obtained with the Taylor FNM does not converge if Eq. (7) is not used. This issue of numerical stability depends on the problem being solved. The mirror FNM, however, does not need such extra treatments for numerical stability.

The next example has a singular point along a straight boundary, and we will use it to further test the performance of different FNMs types.

4.2. The performance of the three FNMs for Motz's problem

To test the capability of the different FNMs in handling local singularities along the boundary, we choose Motz's problem [3,25], which can be seen as a steady-state diffusion problem with the following local boundary conditions:

$$\begin{cases} \nabla_{\boldsymbol{n}} u(x=-1,y)=0 \\ \nabla_{\boldsymbol{n}} u(x,y=1)=0 \\ \nabla_{\boldsymbol{n}} u(x>0,y=0)=0 \\ u(x<0,y=0)=0 \\ u(x=1,y)=500 \end{cases} \tag{20}$$

as shown in Fig. 15. This classical solution for this problem has a strong local singularity $O(\rho^{1/2})$ at the origin O.

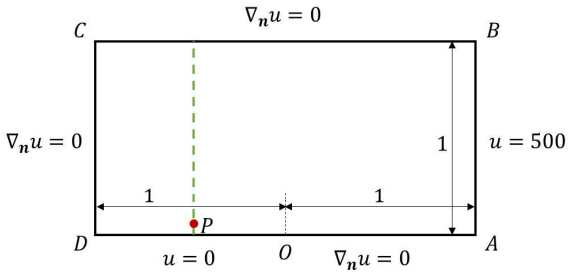

Fig. 15. Domain and boundary conditions for an example of Motz's problem.

The classical solution for Motz's problem can be written as [3]:

$$u(\rho,\theta)=\sum_{i=0}^{\infty} D_i \rho^{i+1/2} \cos\left[\left(i+\frac{1}{2}\right)\theta\right] \tag{21}$$

where D_i's are analogous to the stress intensity factors in linear elastic fracture mechanics, sometimes called "generalized flux intensity factors" [50]. Here we choose 34 terms, as in [51].

In Fig. 16 (a) we show the contours for the classical solution. The point-wise relative differences between the PD solutions and the classical solution are given in Fig. 16 (b)-(d). For the PD simulations, we used δ =0.04 and $\Delta x = 0.01$, respectively. Notice that for the mirror FNM, the non-trivial relative difference (>5%) is restricted to the horizon region of the singular point (point O) and the left-bottom corner (point D). As the horizon size approaches zero (with $\delta/\Delta x$ not decreasing), these areas also converges to zero. For the other two types of FNMs, the relative difference is large not only near the singular point or the corner, but also at locations far away from them. A quantitative comparison of the solutions along the vertical dash

line shown in Fig. 15 between the three types of FNMs is shown in Fig. 17. As we can see in the zoomed-in image, the PD solution obtained with the mirror FNM matches the classical solution much better than the PD solutions obtained with other two FNMs.

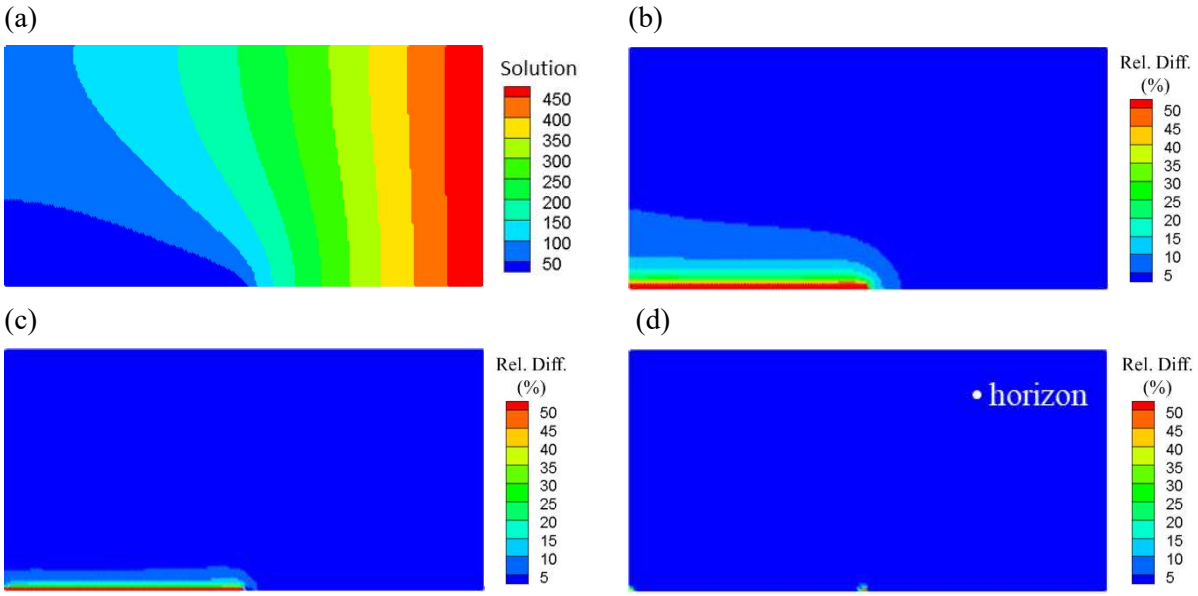

Fig. 16. (a) Contours for the classical solution for Motz's problem; (b)-(d) Relative difference (Rel. Diff.) to classical solution of Motz's problem using the PD model with the naïve, the Taylor and the mirror FNMs, respectively.

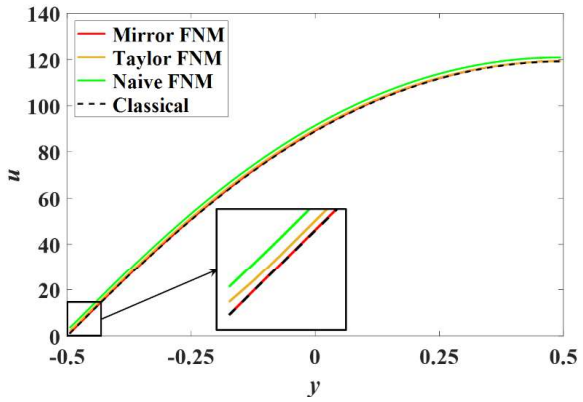

Fig. 17. The classical solution and PD solutions with different FNMs for Motz's problem along the vertical dashed line at $x = -L/2 + \Delta x/2$, as shown in Fig. 15.

The δ-convergence [52] (with $\delta/\Delta x$ fixed to be 4) to the classical solution for the three PD solutions at point P (-0.5, -0.48) (see Fig. 15), is provided in Fig. 18 with the relative difference in log scale. We do not choose a point on the boundary because the analytical solution there is zero (i.e. relative difference does not exist). Notice that as the horizon size changes, the grid also changes and there may not exist a node at

point P. In such cases, we simply obtain the value at point P by averaging the value at four nearest nodes around P. Fig. 18 demonstrates that, as the horizon size decreases, solutions obtained with all three types of FNM approach the classical solution. However, the mirror FNM produces relative differences from the classical solution that are two orders of magnitude smaller than those from the other two FNMs. Moreover, the mirror FNM solution exhibits a convergence rate that is increasing faster than the other two, as the horizon size decreases.

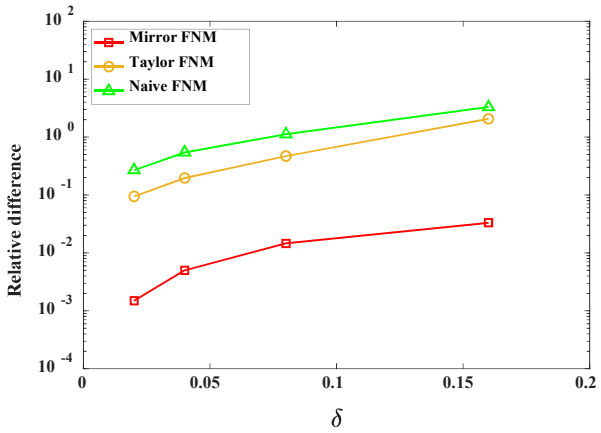

Fig. 18. δ-convergence of PD results (using the different FNMs) to the classical value at point P (-0.5, -0.48) in Fig. 15.

The above two examples show that the mirror FNM works best at accurately enforcing local boundary conditions in PD models, especially for problems with singularities (in local models) along the boundary. In the following section, we will test the autonomous algorithm developed for the mirror FNM for problems with more complex geometries.

4.3. Steady-state diffusion in disks with single and multiple pre-cracks

In this subsection, we will apply the autonomous mirror FNM to solve the PD formulation for diffusion in disks with cracks under Dirichlet boundary conditions. Using the algorithm developed in Section 3.4, the mirror FNM can be easily implemented for curved boundaries and cracks.

4.3.1. Disk with a single crack

We first consider a disk with a single crack and boundary conditions imposed as shown in Fig. 19. The mirror node for each fictitious node, required by the mirror FNM, is determined by the algorithm described in Section 3.4. See Fig. 12 for how mirror nodes are found for those nodes with $r < 1$ (which act as fictitious nodes) near the crack tip.

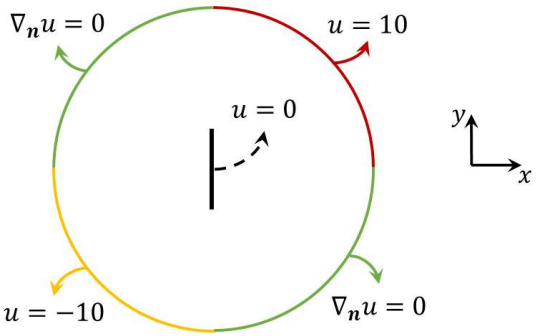

Fig. 19. Local boundary conditions imposed for diffusion in a disk with a pre-crack at the center.

For verification, we also run a simulation of the corresponding PDE-based problem using the steady-state thermal solver in ANSYS workbench. Details of the ANSYS simulation, including how the "crack" is simulated, can be found in Appendix B. In the PD model, the crack is inserted by cutting all bonds that intersect with the pre-crack segment. The differences between the slightly different approaches as representing a crack between the two models should only have a trivial effect on the results.

Contours of the results obtained by ANSYS and PD, with zoomed-ins around the crack region, are given in Fig. 20. The PD results match the FEM results closely, which shows that the autonomous mirror FNM works very well for problems with a curved boundary and with cracks. The jagged shape in the contours of the PD results appears because no smoothing interpolation technique is used in the visualization, we simply plot the obtained values at each PD node.

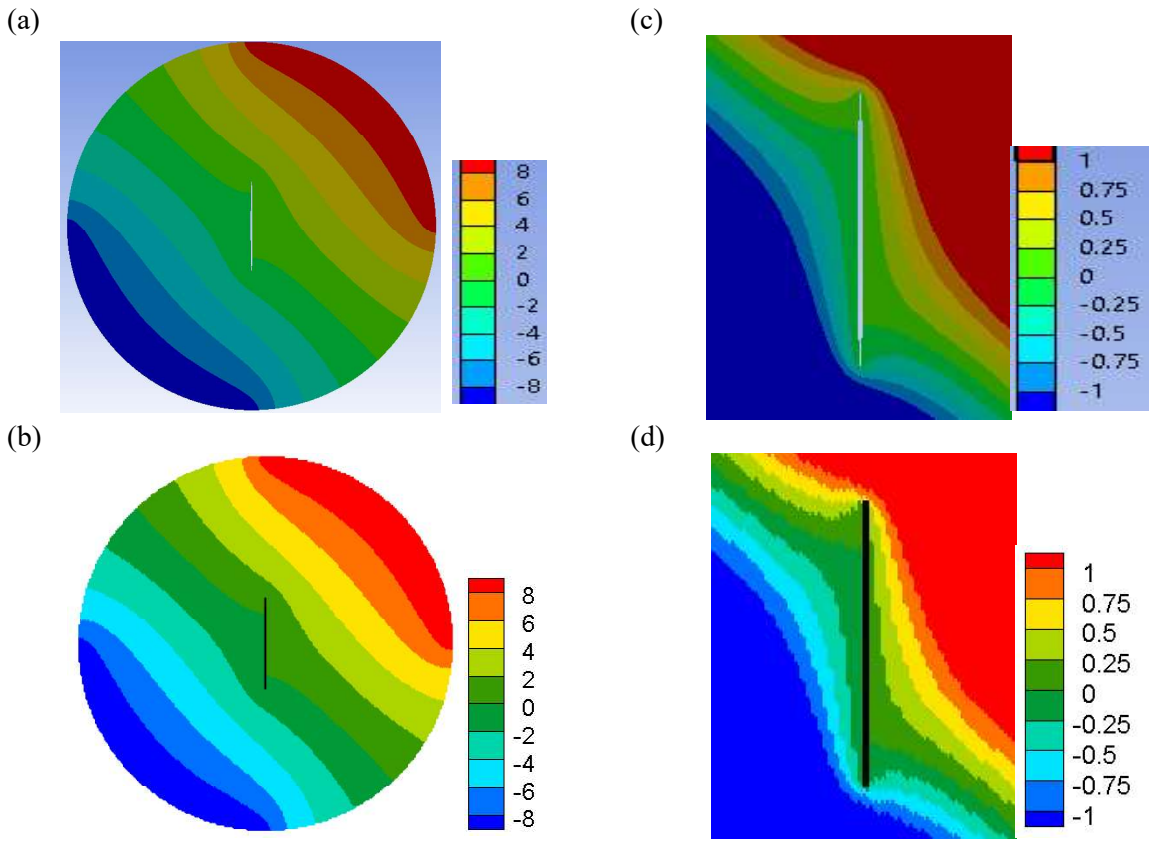

Fig. 20. Contours for the solution to problem shown in Fig. 19 obtained with: (a) ANSYS and (b) PD. In (c) and (d) we show zoomed-in regions around the crack for the corresponding solutions.

In practical simulations, the crack surface may have an angle with respect to the (uniform) discretization grid, whether it is a pre-crack or a new crack formed during the simulation. In order to demonstrate the generality of our algorithm for the mirror FNM, we solve the same problem but we rotate the uniform grid in the counterclockwise direction by 30° and 45° relative to the coordinates shown in Fig. 19, respectively, as shown in Fig. 21. Notice that when the crack is not aligned with the lattice, it may intersect with some PD nodes. These nodes are considered as free nodes (fully damaged) at which the constraint is the average of their family nodes (those not fully damaged). The PD solution with 30°-rotated and 45°-rotated grids are shown in Fig. 22. They match very well the solution obtained with the original lattice shown in Fig. 20. Therefore, the autonomous algorithm can deal with arbitrarily oriented uniform grids effectively.

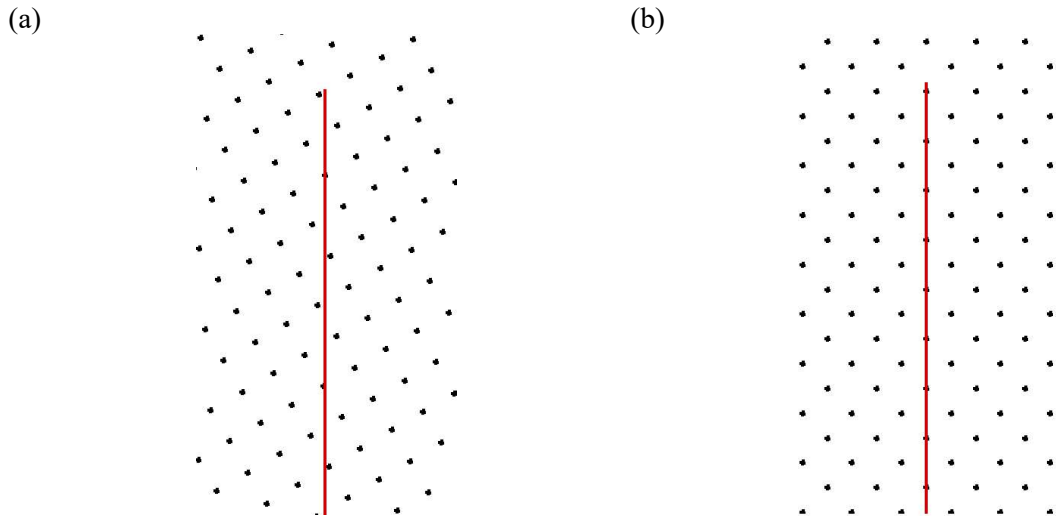

Fig. 21. Part of the crack segment, shown by the red segment, and the PD grid after counterclockwise rotation by (a) 30° and (b) 45°.

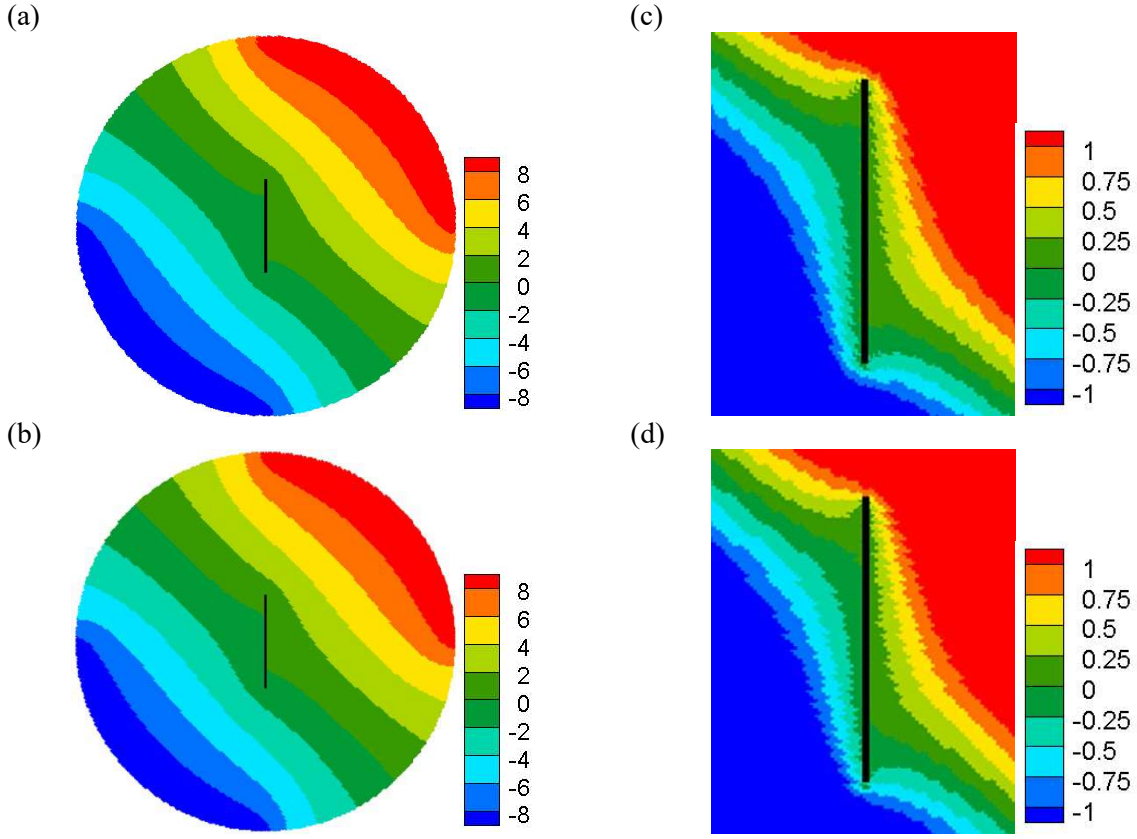

Fig. 22. Contours of PD solutions obtained with the mirror FNM for counterclockwise-rotated grids by (a) 30° and (b) 45°. In (c) and (d) we show zoomed-in views around the crack for the corresponding solutions.

4.3.2. Diffusion in a disk with crossing cracks

A diffusion problem in a disk with two intersecting cracks is shown in Fig. 23. We use the same autonomous algorithm introduced in Section 3.4 without any changes to treat this case. A contour-plot for the PD solution and the zoomed-in picture around the two cracks with imposed Dirichlet BCs are shown in Fig. 24.

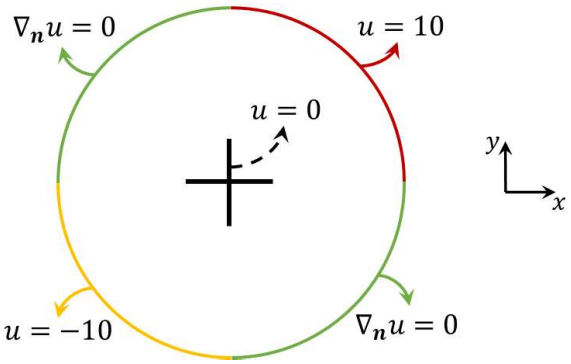

Fig. 23. Diffusion in a disk with two intersecting cracks and associated local boundary conditions.

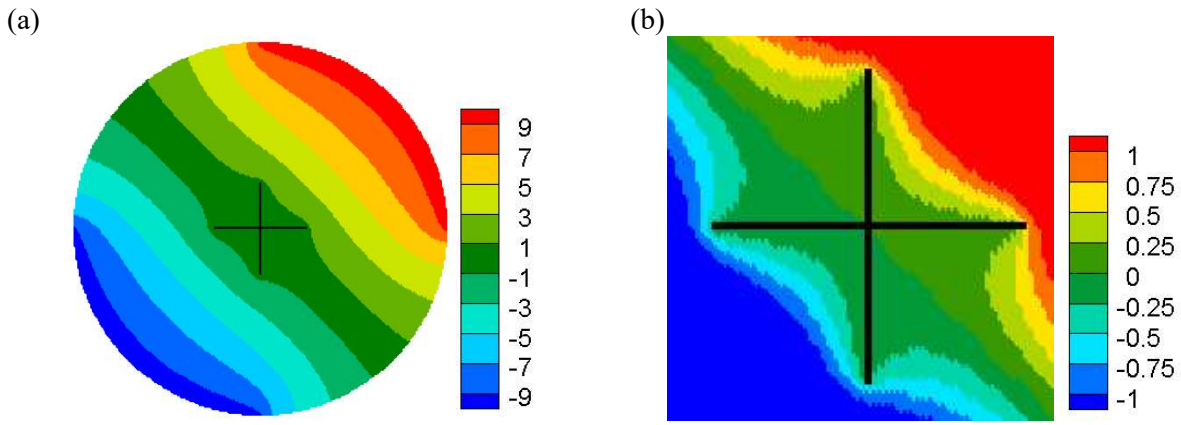

Fig. 24. Contour-plot for the PD solution obtained with the mirror FNM (with the new algorithm) (a) over the disk and (b) over the zoomed-in area around the intersecting cracks (crack lines are drawn only approximately on top of the plot).

The examples shown in this section demonstrated that the autonomous algorithm for the mirror FNM works very well to enforce local BCs for complex geometries, including crack surfaces. Moreover, it can be readily employed for problems with moving boundaries such as corrosion damage, crack propagation, etc.

5. Conclusions

We introduced an autonomous algorithm to find mirror nodes in the mirror-based fictitious nodes method (FNM) for arbitrary geometries, including domains with cracks. Without requiring an explicit mathematical description of the boundary, the algorithm is based on computing a nonlocal gradient, at fictitious nodes, to determine the "generalized" normal direction to the boundary for arbitrary geometry. The new algorithm allows us to easily generate the necessary data for mirror-based FNM to correctly impose the desired local

boundary conditions in peridynamic diffusion models, as well as reduce/eliminate the peridynamic surface effect caused by incomplete nonlocal region near the boundary.

We compared the mirror-based FNM with the naïve and the Taylor-based FNMs for problems with or without singularities (in the corresponding local models) along the boundary, and showed that the peridynamic solution with the mirror-based FNM agrees with the classical solution best, especially for the problem with a singularity in the corresponding classical formulation. The other two methods showed "pollution" of the solution far from the location of the singularity. We applied the new algorithm to diffusion problems in domains with a curved boundary and with cracks. In these cases, too, the peridynamic solution obtained with our mirror-based FNM matched well the FE solution obtained in ANSYS. The same algorithm, with few modifications, should also work for other types of peridynamic models to enforce local-type boundary conditions, including those with moving boundaries and growing cracks.

The new algorithm, which enables a fictitious node method imposition of local boundary conditions in peridynamic models on arbitrary domains, will allow for more accurate peridynamic solutions near the boundaries. High accuracy near arbitrarily-shaped boundaries and material interfaces is crucial in, for example, problems that involve crack initiation and propagation, or evolution of corrosion fronts. In such problems, the new algorithm introduced here will have a great impact.

Acknowledgments

This work has been supported in part by the US National Science Foundation CMMI CDS&E Grant No. 1953346 (program manager Joanne Culbertson) and by a Nebraska System Science award. This work was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

Appendix A. Numerical implementation of peridynamic diffusion model with the fictitious nodes method

For spatial discretization, we discretize the peridynamic body uniformly [26] into cells with nodes in the center of those cells. Fig. A.1 shows a 2D uniform discretization with grid spacing Δx around a node $\boldsymbol{x}_i$. Non-uniform grids are also possible [20,53,54], which may conform better for shapes with, for example, rounded boundaries [49], but this is not pursued in this work.

To discretize the peridynamic integro-differential equations, we use a meshfree method with one-point Gaussian quadrature [26] for the approximation of the integral term. Faster numerical methods such as the boundary-adapted spectral method [17][55] can be alternative options.

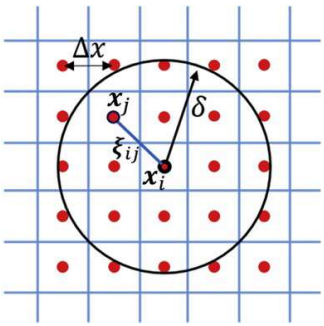

Fig. A.1. Uniform discretization for a peridynamic model. The circular region is the horizon region of node $\boldsymbol{x}_i$.

The discretized PD Laplace's equation (Eq. (4)) becomes:

$$\sum_{\substack{j\in\mathcal{H}_i \\ j\neq i}} \frac{u_j^n - u_i^n}{\xi_{ij}^2} \Delta V_{ij} = 0 \tag{A.1}$$

The superscript n means n^{th} load step. The subscripts i and j denote the current node $\boldsymbol{x}_i$ and its family node $\boldsymbol{x}_j$ respectively, in the discretized domain. $\mathcal{H}_i$ is the horizon region of node $\boldsymbol{x}_i$, $j \in \mathcal{H}_i$ includes all the nodes covered by $\mathcal{H}_i$ (fully or partially), $\xi_{ij} = \|\boldsymbol{x}_j - \boldsymbol{x}_i\|$, ΔV_{ij} is the area of node $\boldsymbol{x}_j$ covered by $\mathcal{H}_i$. The discretized versions for other equations are similar to Eq. (A.1) and are not given here.

In Taylor-based and mirror-based fictitious nodes methods, the equilibrium system can be solved iteratively using the linear Conjugate Gradient (CG) solver combined with additional criteria to check for the convergence of the solution. At each iteration or solution step, the CG solver is called and the solution in the domain and fictitious region is updated, respectively. For the purpose of minimizing the overall computational cost, the tolerance in the CG solver is set to be 1e-2 at first and then decreased with solution steps by a factor until it reaches 1e-6. This treatment could make the simulation 50% more efficient than fixing the tolerance in the CG solver to be 1e-6 during the whole simulation. The system converges when the solution in the domain obtained between two sequential solution steps differ, in terms of norm-2 relative difference, by less than a given tolerance (1e-6 in this work). The detailed workflow for a complete simulation is shown in Fig. A.2.

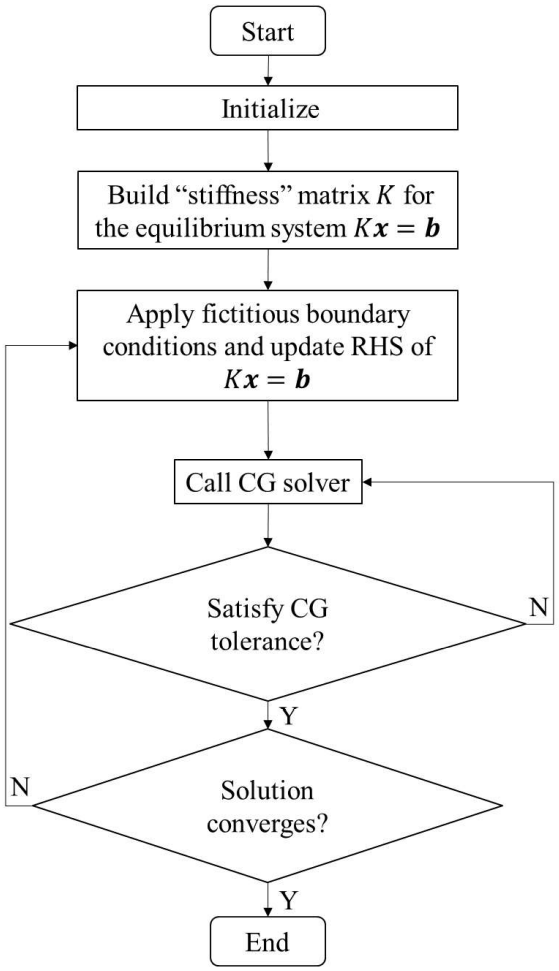

Fig. A.2. Workflow for the peridynamic simulation with Taylor/mirror FNM.

Appendix B. FEM modeling of the steady-state thermal problem

To obtain the classical FEM-based solution for the problem shown in Fig. 19, ANSYS Workbench Steady-State Thermal solver is used. In the FE model, the two crack surfaces are generated by two arcs with the same small curvature and the maximum space between them is 0.01, which equals the grid size in the corresponding PD model. For the mesh, as shown in Fig. B.1, the element order is selected to be program controlled and all elements are triangles with the maximum size equals 0.05. The total number of nodes and elements are 9312 and 4548, respectively. All other options in the solver are set to be defaults.

(a) (b)

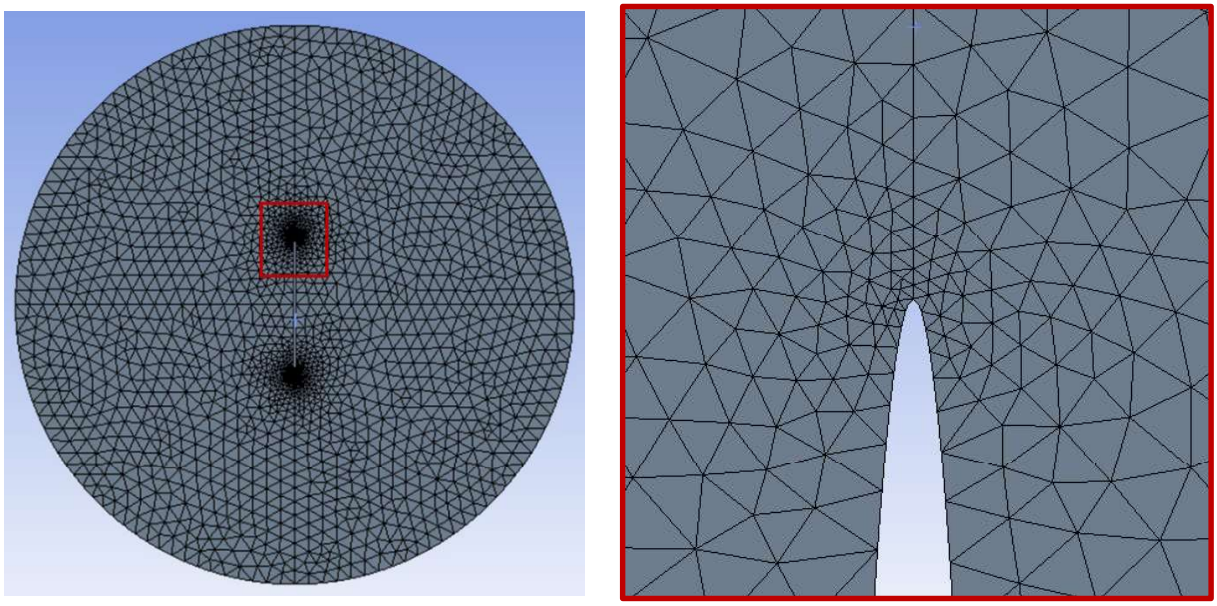

Fig. B.1. FEM mesh (a) over the whole disk; (b) near the crack tip.

References

[1] Z.C. Li, Combined Methods for Elliptic Equations with Singularities, Interfaces and Infinities, Springer US, Boston, MA, 1998. https://doi.org/10.1007/978-1-4613-3338-8.

[2] E. McCafferty, Introduction to Corrosion Science, Springer New York, New York, NY, 2010. https://doi.org/10.1007/978-1-4419-0455-3.

[3] Z.C. Li, T.T. Lu, Singularities and treatments of elliptic boundary value problems, Math. Comput. Model. 31 (2000) 97–145. https://doi.org/10.1016/S0895-7177(00)00062-5.

[4] L.B. Wahlbin, Local behavior in finite element methods, Handb. Numer. Anal. 2 (1991) 353–522. https://doi.org/10.1016/S1570-8659(05)80040-7.

[5] K. Eriksson, C. Johnson, An Adaptive Finite Element Method for Linear Elliptic Problems, Math. Comput. 50 (1988) 361. https://doi.org/10.2307/2008613.

[6] G.J. Fix, S. Gulati, G.I. Wakoff, On the use of singular functions with finite element approximations, J. Comput. Phys. 13 (1973) 209–228. https://doi.org/10.1016/0021-9991(73)90023-5.

[7] N.M. Wigley, On a Method to Subtract off a Singularity at a Corner for the Dirichlet or Neumann Problem, Math. Comput. 23 (1969) 395. https://doi.org/10.2307/2004432.

[8] H. Blum, M. Dobrowolski, On finite element methods for elliptic equations on domains with corners, Computing. 28 (1982) 53–63. https://doi.org/10.1007/BF02237995.

[9] E. Trefftz, Ein Gegenstück zum Ritzschen Verfahren, in: Int. Kongress Für Tech. Mech. 12–17 Sept., 1926: pp. 131–137.

[10] L.M. Delves, C.A. Hall, An implicit matching principle for global element calculations, IMA J. Appl. Math. (Institute Math. Its Appl. 23 (1979) 223–234. https://doi.org/10.1093/imamat/23.2.223.

[11] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, J. Mech. Phys. Solids. 48 (2000) 175–209. https://doi.org/10.1016/S0022-5096(99)00029-0.

[12] Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, Analysis and approximation of nonlocal diffusion problems with volume constraints, SIAM Rev. 54 (2012) 667–696. https://doi.org/10.1137/110833294.

[13] Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws, Math. Model. Methods Appl. Sci. 23 (2013) 493–540. https://doi.org/10.1142/S0218202512500546.

[14] S. Oterkus, E. Madenci, A. Agwai, Peridynamic thermal diffusion, J. Comput. Phys. 265 (2014) 71–96. https://doi.org/10.1016/j.jcp.2014.01.027.

[15] F. Bobaru, J.T. Foster, P.H. Geubelle, S.A. Silling, Handbook of peridynamic modeling, CRC press, 2016. https://doi.org/https://doi.org/10.1201/9781315373331.

[16] Q. V. Le, F. Bobaru, Surface corrections for peridynamic models in elasticity and fracture, Comput. Mech. 61 (2018) 499–518. https://doi.org/10.1007/s00466-017-1469-1.

[17] S. Jafarzadeh, A. Larios, F. Bobaru, Efficient Solutions for Nonlocal Diffusion Problems Via Boundary-Adapted Spectral Methods, J. Peridynamics Nonlocal Model. 2 (2020) 85–110. https://doi.org/10.1007/s42102-019-00026-6.

[18] H. Takeda, S.M. Miyama, M. Sekiya, Numerical Simulation of Viscous Flow by Smoothed Particle Hydrodynamics, Prog. Theor. Phys. 92 (1994) 939–960. https://doi.org/10.1143/ptp/92.5.939.

[19] F. Maciá, M. Antuono, L.M. González, A. Colagrossi, Theoretical analysis of the no-slip boundary condition enforcement in SPH methods, Prog. Theor. Phys. 125 (2011) 1091–1121. https://doi.org/10.1143/PTP.125.1091.

[20] F. Bobaru, Y.D. Ha, Adaptive refinement and multiscale modeling in 2D peridynamics, Int. J. Multiscale Comput. Eng. 9 (2011) 635–659. https://doi.org/10.1615/IntJMultCompEng.2011002793.

[21] S. Oterkus, E. Madenci, A. Agwai, Fully coupled peridynamic thermomechanics, J. Mech. Phys. Solids. 64 (2014) 1–23. https://doi.org/10.1016/j.jmps.2013.10.011.

[22] Z. Chen, F. Bobaru, Selecting the kernel in a peridynamic formulation: A study for transient heat diffusion, Comput. Phys. Commun. 197 (2015) 51–60. https://doi.org/10.1016/j.cpc.2015.08.006.

[23] J.P. Morris, P.J. Fox, Y. Zhu, Modeling low Reynolds number incompressible flows using SPH, J. Comput. Phys. 136 (1997) 214–226. https://doi.org/10.1006/jcph.1997.5776.

[24] J. Wang, W. Hu, X. Zhang, W. Pan, Modeling heat transfer subject to inhomogeneous Neumann boundary conditions by smoothed particle hydrodynamics and peridynamics, Int. J. Heat Mass Transf. 139 (2019) 948–962. https://doi.org/10.1016/j.ijheatmasstransfer.2019.05.054.

[25] H. Motz, The treatment of singularities of partial differential equations by relaxation methods, Q. Appl. Math. 4 (1947) 371–377. https://doi.org/10.1090/qam/18442.

[26] S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics,

in: Comput. Struct., 2005: pp. 1526–1535. https://doi.org/10.1016/j.compstruc.2004.11.026.

[27] F. Bobaru, G. Zhang, Why do cracks branch? A peridynamic investigation of dynamic brittle fracture, Int. J. Fract. 196 (2015) 59–98. https://doi.org/10.1007/s10704-015-0056-8.

[28] E. Madenci, E. Oterkus, Peridynamic Theory and Its Applications, Springer New York, New York, NY, 2014. https://doi.org/10.1007/978-1-4614-8465-3.

[29] S.A. Silling, R.B. Lehoucq, Peridynamic Theory of Solid Mechanics, in: Adv. Appl. Mech., 2010: pp. 73–168. https://doi.org/10.1016/S0065-2156(10)44002-8.

[30] Y.D. Ha, F. Bobaru, Studies of dynamic crack propagation and crack branching with peridynamics, Int. J. Fract. 162 (2010) 229–244. https://doi.org/10.1007/s10704-010-9442-4.

[31] J.T. Foster, S.A. Silling, W.W. Chen, Viscoplasticity using peridynamics, Int. J. Numer. Methods Eng. 81 (2010) 1242–1258. https://doi.org/10.1002/nme.2725.

[32] Y.L. Hu, N. V. De Carvalho, E. Madenci, Peridynamic modeling of delamination growth in composite laminates, Compos. Struct. 132 (2015) 610–620. https://doi.org/10.1016/j.compstruct.2015.05.079.

[33] F. Bobaru, M. Duangpanya, The peridynamic formulation for transient heat conduction, Int. J. Heat Mass Transf. 53 (2010) 4047–4059. https://doi.org/10.1016/j.ijheatmasstransfer.2010.05.024.

[34] F. Bobaru, M. Duangpanya, A peridynamic formulation for transient heat conduction in bodies with evolving discontinuities, J. Comput. Phys. 231 (2012) 2764–2785. https://doi.org/10.1016/j.jcp.2011.12.017.

[35] J. Zhao, Z. Chen, J. Mehrmashhadi, F. Bobaru, Construction of a peridynamic model for transient advection-diffusion problems, Int. J. Heat Mass Transf. 126 (2018) 1253–1266. https://doi.org/10.1016/j.ijheatmasstransfer.2018.06.075.

[36] Z. Chen, F. Bobaru, Peridynamic modeling of pitting corrosion damage, J. Mech. Phys. Solids. 78 (2015) 352–381. https://doi.org/10.1016/j.jmps.2015.02.015.

[37] Z. Chen, G. Zhang, F. Bobaru, The Influence of Passive Film Damage on Pitting Corrosion, J. Electrochem. Soc. 163 (2016) C19–C24. https://doi.org/10.1149/2.0521602jes.

[38] S. Jafarzadeh, Z. Chen, F. Bobaru, Peridynamic modeling of repassivation in pitting corrosion of stainless steel, Corrosion. 74 (2018) 393–414. https://doi.org/10.5006/2615.

[39] S. Jafarzadeh, Z. Chen, F. Bobaru, Peridynamic Modeling of Intergranular Corrosion Damage, J. Electrochem. Soc. 165 (2018) C362–C374. https://doi.org/10.1149/2.0821807jes.

[40] S. Jafarzadeh, Z. Chen, F. Bobaru, Computational modeling of pitting corrosion, Corros. Rev. 37 (2019) 419–439. https://doi.org/10.1515/corrrev-2019-0049.

[41] S. Jafarzadeh, Z. Chen, S. Li, F. Bobaru, A peridynamic mechano-chemical damage model for stress-assisted corrosion, Electrochim. Acta. 323 (2019) 134795. https://doi.org/10.1016/j.electacta.2019.134795.

[42] S. Jafarzadeh, Z. Chen, J. Zhao, F. Bobaru, Pitting, lacy covers, and pit merger in stainless steel: 3D peridynamic models, Corros. Sci. 150 (2019) 17–31. https://doi.org/10.1016/j.corsci.2019.01.006.

[43] P. Radu, K. Wells, A doubly nonlocal laplace operator and its connection to the classical Laplacian, J. Integr. Equations Appl. 31 (2019) 379–409. https://doi.org/10.1216/JIE-2019-31-3-

379.

[44] X. Tian, Q. Du, Asymptotically compatible schemes and applications to robust discretization of nonlocal models, SIAM J. Numer. Anal. 52 (2014) 1641–1665. https://doi.org/10.1137/130942644.

[45] Q. Du, X. Tian, Asymptotically Compatible Schemes for Peridynamics Based on Numerical Quadratures, in: Vol. 1 Adv. Aerosp. Technol., American Society of Mechanical Engineers, 2014. https://doi.org/10.1115/IMECE2014-39620.

[46] Y. Tao, X. Tian, Q. Du, Nonlocal diffusion and peridynamic models with Neumann type constraints and their numerical approximations, Appl. Math. Comput. 305 (2017) 282–298. https://doi.org/10.1016/j.amc.2017.01.061.

[47] F. Bobaru, W. Hu, The meaning, selection, and use of the peridynamic horizon and its relation to crack branching in brittle materials, Int. J. Fract. 176 (2012) 215–222. https://doi.org/10.1007/s10704-012-9725-z.

[48] U.M. Ascher, C. Greif, A First Course in Numerical Methods, Society for Industrial and Applied Mathematics, Philadelphia, PA, 2011. https://doi.org/10.1137/9780898719987.

[49] J. Zhao, Z. Chen, J. Mehrmashhadi, F. Bobaru, A stochastic multiscale peridynamic model for corrosion-induced fracture in reinforced concrete, Eng. Fract. Mech. 229 (2020) 106969. https://doi.org/10.1016/j.engfracmech.2020.106969.

[50] Z. Yosibash, Singularities in elliptic boundary value problems and elasticity and their connection with failure initiation, Springer Science & Business Media, 2011.

[51] H.Y. Hu, Z.C. Li, Collocation methods for Poisson's equation, Comput. Methods Appl. Mech. Eng. 195 (2006) 4139–4160. https://doi.org/10.1016/j.cma.2005.07.019.

[52] F. Bobaru, M. Yang, L.F. Alves, S.A. Silling, E. Askari, J. Xu, Convergence, adaptive refinement, and scaling in 1D peridynamics, Int. J. Numer. Methods Eng. 77 (2009) 852–877. https://doi.org/10.1002/nme.2439.

[53] S.F. Henke, S. Shanbhag, Mesh sensitivity in peridynamic simulations, Comput. Phys. Commun. 185 (2014) 181–193. https://doi.org/10.1016/j.cpc.2013.09.010.

[54] X. Gu, Q. Zhang, X. Xia, Voronoi-based peridynamics and cracking analysis with adaptive refinement, Int. J. Numer. Methods Eng. 112 (2017) 2087–2109. https://doi.org/10.1002/nme.5596.

[55] S. Jafarzadeh, L. Wang, A. Larios, F. Bobaru, A fast Fourier spectral method for peridynamic transient diffusion in arbitrary domains, (2020). (in review).