Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 375 (2021) 113633
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

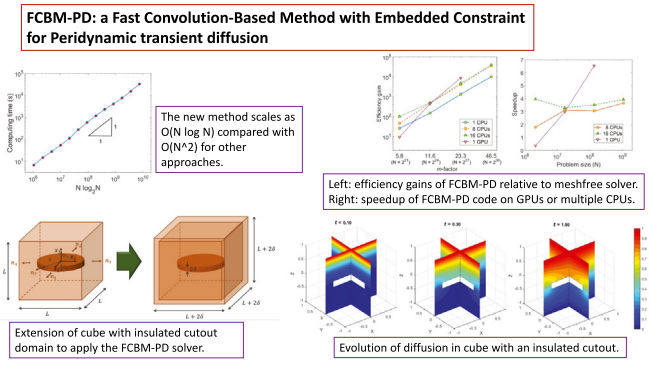
A fast convolution-based method for peridynamic transient diffusion in arbitrary domains

Siavash Jafarzadeh[a], Longzhen Wang[a], Adam Larios[b,*], Florin Bobaru[a,*]

[a] *Department of Mechanical and Materials Engineering, University of Nebraska-Lincoln, Lincoln, NE 68588-0526, USA*
[b] *Department of Mathematics, University of Nebraska-Lincoln, Lincoln, NE 68588-0130, USA*

Received 1 June 2020; received in revised form 3 December 2020; accepted 4 December 2020
Available online 2 January 2021

Graphical Abstract



Abstract

We introduce a fast convolution-based method (FCBM) for solving linear and a certain class of nonlinear peridynamic (PD) transient diffusion problems in 1D, 2D, and 3D. The method exploits the convolutional structure of the PD diffusion operator to compute it efficiently by using the fast Fourier transform (FFT). A new "embedded constraint" (EC) strategy allows using the Fourier transform on irregular domains and imposing arbitrary nonlocal boundary conditions. The complexity of the new method is $O(N \log 2N)$, compared with $O(N^2)$ for the conventional peridynamic meshfree or finite element solvers of the same problem. We find quadratic convergence rates in terms of spatial discretization to the exact nonlocal solutions in 1D and 2D problems. Numerical tests show substantial efficiency gains when using the FCBM-EC method compared to the meshfree discretization method for problems with a larger number of nodes inside the nonlocal region. Further speedup is achieved with Matlab's intrinsic FFT functions for computations on GPUs or multiple CPUs. An example in 3D with over 1 billion degrees of freedom over tens of thousands of time-steps, is solved by the new method on a single CPU in a matter of days. The same problem would have required over a century to complete with the commonly used meshfree discretization.

Keywords: Peridynamics; Fourier spectral methods; Convolution; Nonlocal diffusion; High-performance computing

* Corresponding authors.
E-mail addresses: alarios@unl.edu (A. Larios), fbobaru2@unl.edu (F. Bobaru).

https://doi.org/10.1016/j.cma.2020.113633
0045-7825/

1. Introduction

Nonlocal diffusion-type equations are frequently used to describe various phenomena such as swarm of organisms [1], flocking of birds [2–4], pedestrian traffic [5], delayed reaction–diffusion in biology [6], etc. Peridynamic (PD) diffusion equation is one of such nonlocal formulations that was used for modeling diffusion in domains with evolving discontinuities, for example heat diffusion in a cracking domain [7] or mass-transfer in corrosion damage propagation [8–11]. Peridynamic models replace spatial derivatives in local formulations with integral operators. This allows more general formulations with no smoothness restrictions on the unknown functions, making it easy to solve problems with spontaneous emergence and evolution of discontinuities [12,13]. However, numerical solutions to such nonlocal formulations are relatively expensive due to the computation of volume integrals (area integrals in 2D) required at each node [14,15].

The meshfree method based on one-point Gaussian quadrature [16] is perhaps the most widely used numerical method for discretizing PD equations due to its benefits in modeling damage and fracture [17]. Continuous and discontinuous Galerkin finite element methods (FEM) [18–21] are other approaches used for discretizing PD equations. Continuous FEM models for PD formulations are rarely used in simulating fracture, while the discontinuous Galerkin-based discretizations of PD models have shown some difficulties in correctly predicting dynamic brittle fracture [22]. For a certain size of the nonlocal interaction region (the "horizon" size), the computational complexity of the meshfree and FEM approaches is $O\left(N^2\right)$ with N being the total number of nodes employed to discretize the domain.

Reducing the computational cost for PD models can be achieved with Fourier-based methods. The PD operator in the PD diffusion equation can be expressed in terms of convolution integrals. Fourier methods transform the convolution integral operator into a multiplication in the Fourier space, at a cost of $O(N \log 2N)$, with the bulk of the computations spent on computing the Fourier transform and its inverse via *fast Fourier transform* (FFT) [23,24]. However, Fourier methods are normally only applicable to periodic domains, limiting their use. This type of method has been used to solve the nonlocal Allen–Cahn equation [25], fractional-in-space reaction diffusion [26], and PD diffusion and wave equations with linear operators [27–31], over periodic domains. A certain class of PD models discretized with the FEM or the meshfree method, are amenable to fast solutions using the Fourier-based approach introduced in [32,33].

To apply Fourier methods to non-periodic domains and arbitrary boundary conditions, we recently introduced a fast convolution-based method (FCBM) for 1D linear peridynamic diffusion problems in [34], using a *volume penalization (VP)* technique previously used in the context of PDEs in, e.g., [35]. We obtained significant efficiency gains with the FCBM-VP compared with meshfree discretization for PD diffusion in 1D, motivating the present study on formulating such approaches in higher dimensions and for nonlinear problems.

In this paper we replace the volume penalization technique used in [34] with a better scheme we call *Embedded Constraint (EC)*, and extend the formulation to diffusion problems in two and three-dimensions, and also to nonlinear PD diffusion equations. We explain the advantages of the new EC versus the VP approach for a 1D case, then we verify the new method with a nonlinear 2D example. We also use the FCBM-EC to compute a 3D diffusion example in a bounded domain with an insulated penny-shaped cutout subjected to Neumann and Dirichlet BCs. We determine computational efficiency gains of the newly introduced method relative to the meshfree discretization in 3D.

2. Peridynamic transient diffusion

The theory of peridynamics was originally developed for mechanics and fracture [36], and has been extensively applied modeling fracture and failure [37–40]. This nonlocal theory has been extended to diffusion-type problems [41,42] and several other fields [43,44].

In peridynamics, a spatial point, denoted by its position vector $\boldsymbol{x} \in \mathbb{R}^3$, interacts with other points within its finite size neighborhood, usually a disk in 2D (sphere in 3D) centered at $\boldsymbol{x}$. The neighborhood is called the *horizon region* of $\boldsymbol{x}$ denoted by $\mathcal{H}_{\boldsymbol{x}}$, and its radius is called *horizon size* (or simply, the *horizon*) denoted by δ. Points located inside $\mathcal{H}_{\boldsymbol{x}}$ are denoted by $\boldsymbol{x}'$ and are referred to as the *family* of $\boldsymbol{x}$ [12,13]. Fig. 1 schematically shows a peridynamic body with a generic point $\boldsymbol{x}$, its family and its horizon.

Objects that carry the pairwise nonlocal interactions between $\boldsymbol{x}$ and $\boldsymbol{x}'$ are called *bonds*.

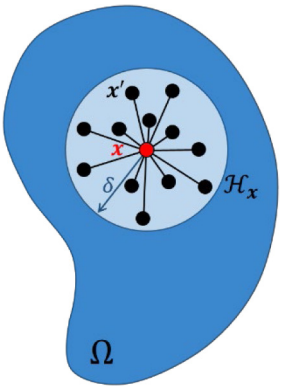


Fig. 1. A peridynamic body with a generic point $\boldsymbol{x}$ and its horizon $\mathcal{H}_{\boldsymbol{x}}$.

2.1. The peridynamic diffusion equation

Consider the general nonlinear anisotropic inhomogeneous case of the classical diffusion equation:

$$\frac{\partial u\left(\boldsymbol{x},t\right)}{\partial t}=\nabla\cdot\left[\mathbf{C}\left(\boldsymbol{x},t\right)\cdot\nabla u\left(\boldsymbol{x},t\right)\right]+r\left(\boldsymbol{x},t\right), \tag{1}$$

where $u\left(\boldsymbol{x},t\right)$ is the function describing the diffusing quantity (e.g. concentration in mass transport, temperature in heat transfer, etc.) at point $\boldsymbol{x}$ and time t, r is a source term, ∇ is the gradient operator, $(\cdot)$ denotes inner (dot) product, and $\mathbf{C}$ is the second-order diffusivity tensor that may depend on position and time. Nonlinearity in Eq. (1) arises if $\mathbf{C}$ depends on u (e.g. concentration-dependent diffusivity, or temperature-dependent conductivity). If $\mathbf{C}$ is independent of u the equation is linear. In this case, for an isotropic and homogeneous diffusion defined by ν as the scalar diffusivity constant, Eq. (1) becomes:

$$\frac{\partial u\left(\boldsymbol{x},t\right)}{\partial t}=\nu\nabla^{2}u\left(\boldsymbol{x},t\right)+r\left(\boldsymbol{x},t\right), \tag{2}$$

where ∇^2 denotes the classical Laplacian operator.

A nonlocal version of Eq. (1) is the general PD diffusion equation [41,42]:

$$\frac{\partial u\left(\boldsymbol{x},t\right)}{\partial t}=\mathcal{L}_{\gamma}u(\boldsymbol{x},t)+r\left(\boldsymbol{x},t\right), \tag{3}$$

where the classical diffusion operator $\nabla\cdot(\mathbf{C}\cdot\nabla)$ in Eq. (1) is replaced with the PD diffusion operator $\mathcal{L}_{\gamma}$:

$$\mathcal{L}_{\gamma}u\left(\boldsymbol{x},t\right)=\int_{\mathcal{H}_{\boldsymbol{x}}}\gamma(\boldsymbol{x},\boldsymbol{x}',t)[u\left(\boldsymbol{x}',t\right)-u(\boldsymbol{x},t)]\mathrm{d}\boldsymbol{x}' \tag{4}$$

In Eq. (4), the *kernel* γ is assumed to be nonnegative and symmetric in the first two variables: $\gamma\left(\boldsymbol{x},\boldsymbol{x}',t\right)=\gamma(\boldsymbol{x}',\boldsymbol{x},t)$. It defines the nonlocal interaction between $\boldsymbol{x}$ and $\boldsymbol{x}'$ [41,42]. PD diffusion is nonlinear if γ depends on $u(\boldsymbol{x},t)$ and/or $u\left(\boldsymbol{x}',t\right)$.

For isotropic homogeneous diffusion with constant diffusivity, the peridynamic diffusion equation becomes [41, 42,45]:

$$\frac{\partial u\left(\boldsymbol{x},t\right)}{\partial t}=\nu\mathcal{L}_{\mu}u\left(\boldsymbol{x},t\right)+r\left(\boldsymbol{x},t\right), \tag{5}$$

where $\mathcal{L}_{\mu}$ is the PD Laplacian operator, a nonlocal version of the Laplacian in Eq. (2). The PD Laplacian can be expressed as:

$$\mathcal{L}_{\mu}u\left(\boldsymbol{x},t\right)=\int_{\mathcal{H}_{\boldsymbol{x}}}\mu(\boldsymbol{x}'-\boldsymbol{x})[u\left(\boldsymbol{x}',t\right)-u(\boldsymbol{x},t)]\mathrm{d}\boldsymbol{x}', \tag{6}$$

where the kernel function μ is nonnegative and symmetric, i.e. $\mu\left(\boldsymbol{x}\right)=\mu(-\boldsymbol{x})$.

In order to take advantage of the speedup provided by Fourier-based methods for solving PD diffusion problems, we need to express the PD operator in terms of convolutions. For example, the PD Laplacian given in Eq. (6) can be decomposed into a convolution integral term and a linear term:

$$\mathcal{L}_\mu u\left(\boldsymbol{x},t\right)=\int_{\mathcal{H}_{\boldsymbol{x}}}\mu(\boldsymbol{x}-\boldsymbol{x}')u\left(\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'-\left(\int_{\mathcal{H}_{\boldsymbol{x}}}\mu\left(\boldsymbol{x}'-\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}'\right)u\left(\boldsymbol{x},t\right), \tag{7}$$

Since $\mu(\boldsymbol{x}-\boldsymbol{x}')$ is zero outside of $\mathcal{H}_{\boldsymbol{x}}$:

$$\mathcal{L}_\mu u\left(\boldsymbol{x},t\right)=\left[\mu*u\right]\left(\boldsymbol{x},t\right)-\beta u\left(\boldsymbol{x},t\right) \tag{8}$$

where $(*)$ denotes the convolution operator:

$$\left[\mu*u\right]\left(\boldsymbol{x},t\right)=\int_{\mathcal{R}^3}\mu\left(\boldsymbol{x}-\boldsymbol{x}'\right)u\left(\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}', \tag{9}$$

and:

$$\beta=\int_{\mathcal{H}_0}\mu\left(\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}. \tag{10}$$

Note that besides this linear case, one can perform a similar decomposition for certain classes of nonlinear problems for which such convolutional expressions exist. For example, for kernels of the form:

$$\gamma(\boldsymbol{x},\boldsymbol{x}',t)=\sigma\left(\boldsymbol{x},\boldsymbol{x}',u,u',t\right)\omega(\boldsymbol{x}'-\boldsymbol{x},t) \tag{11}$$

with $\omega\left(\boldsymbol{x}'-\boldsymbol{x},t\right)=\omega(\boldsymbol{x}-\boldsymbol{x}',t)$, $\sigma\left(\boldsymbol{x},\boldsymbol{x}',u,u',t\right)=\sigma\left(\boldsymbol{x}',\boldsymbol{x},u',u,t\right)$, and σ being a linear combination of functions (or products of such functions) that depend on $(\boldsymbol{x},u,t)$ or $\left(\boldsymbol{x}',u',t\right)$. The decomposition into convolutions for a specific example of such PD nonlinear diffusion operators is given below.

Let the kernel γ in Eq. (4) be defined as:

$$\gamma(\boldsymbol{x},\boldsymbol{x}',t)=\left[\frac{f\left(\boldsymbol{x},u,t\right)h\left(\boldsymbol{x}',u',t\right)+f\left(\boldsymbol{x}',u',t\right)h\left(\boldsymbol{x},u,t\right)}{2}\right]\omega(\boldsymbol{x}'-\boldsymbol{x},t) \tag{12}$$

where u' denotes $u\left(\boldsymbol{x}',t\right)$ for notation simplicity. Similarly, by denoting f' and h' for $f\left(\boldsymbol{x}',u',t\right)$ and $h\left(\boldsymbol{x}',u',t\right)$ respectively, the nonlinear diffusion PD operator becomes:

$$\begin{aligned}\mathcal{L}_\gamma u\left(\boldsymbol{x},t\right)=&\int_{\mathcal{H}_{\boldsymbol{x}}}\left\{\left[\frac{f\left(\boldsymbol{x},u,t\right)h\left(\boldsymbol{x}',u',t\right)+f\left(\boldsymbol{x}',u',t\right)h\left(\boldsymbol{x},u,t\right)}{2}\right]\omega(\boldsymbol{x}'-\boldsymbol{x},t)\right\}\left[u\left(\boldsymbol{x}',t\right)\right.\\&\left.-u(\boldsymbol{x},t)\right]\mathrm{d}\boldsymbol{x}'=\frac{1}{2}\int_{\mathcal{H}_{\boldsymbol{x}}}\left(fh'+f'h\right)\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\left(u'-u\right)\mathrm{d}\boldsymbol{x}'\\&=\frac{1}{2}\int_{\mathcal{H}_{\boldsymbol{x}}}\left(fh'u'-fh'u+f'hu'-f'hu\right)\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'\\&=\frac{1}{2}\left[\int_{\mathcal{H}_{\boldsymbol{x}}}fh'u'\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'-\int_{\mathcal{H}_{\boldsymbol{x}}}fh'u\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'\right.\\&\left.+\int_{\mathcal{H}_{\boldsymbol{x}}}f'hu'\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'-\int_{\mathcal{H}_{\boldsymbol{x}}}f'hu\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'\right]\\&=\frac{1}{2}\left[f\int_{\mathcal{H}_{\boldsymbol{x}}}h'u'\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'-fu\int_{\mathcal{H}_{\boldsymbol{x}}}h'\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'\right.\\&\left.+h\int_{\mathcal{H}_{\boldsymbol{x}}}f'u'\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'-hu\int_{\mathcal{H}_{\boldsymbol{x}}}f'\omega\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}'\right]\\&=\frac{1}{2}\left\{f\left[\omega*(hu)\right]-fu\left(\omega*h\right)+h\left[\omega*(fu)\right]-hu\left(\omega*f\right)\right\}\end{aligned} \tag{13}$$

Note that Eq. (8), the linear case, is recovered from Eq. (13) when $f=h=1$, and $\omega=\mu$.

The proposed method is general and can be applied to *any integral operator with a convolutional structure*. For example, in a separate work [46], we have obtained results that show how to apply the method for PD elasticity models (with small and large rotations) and for dynamic brittle fracture problems.

In Section 3 we use the convolution form given in Eq. (8) for the linear case to construct the fast convolution-based method for peridynamic models of diffusion, and then we repeat those steps for the nonlinear case given in Eq. (13).

3. The fast convolution-based method for 3D peridynamic transient diffusion

In this section we discuss the FCBM for linear (general) and nonlinear (particular case in Eq. (13)) PD diffusion in 3D periodic domains.

Consider the isotropic homogeneous PD diffusion given by Eq. (5). In order to apply Fourier-based methods, we need to assume periodicity of the 3D domain in the Cartesian directions. Let $u(\boldsymbol{x},t)$ be a complex-valued scalar function defined over the periodic domain$\mathbb{T}=[0,L_1]\times[0,L_2]\times[0,L_3]$, with 0 being "identified" with L_1, L_2, and L_3 in all three directions to create the domain periodicity (in Section 4 we will show how to extend this to arbitrary domains). Then $u(\boldsymbol{x},t)$ can be expressed with the infinite Fourier series in space [47,48]:

$$u(\boldsymbol{x},t)=\sum_{k_3=-\infty}^{+\infty}\sum_{k_2=-\infty}^{+\infty}\sum_{k_1=-\infty}^{+\infty}\hat{u}(\boldsymbol{k},t)\,e^{2\pi\zeta\left(\frac{k_1x_1}{L_1}+\frac{k_2x_2}{L_2}+\frac{k_3x_3}{L_3}\right)} \tag{14}$$

where $\boldsymbol{k}=\{k_1,k_2,k_3\}$ is the integer vector of Fourier modes, $\zeta=\sqrt{-1}$, and:

$$\hat{u}(\boldsymbol{k},t)=\frac{1}{L_3L_2L_1}\int_0^{L_3}\int_0^{L_2}\int_0^{L_1}u(\boldsymbol{x},t)\,e^{-2\pi\zeta\left(\frac{k_1x_1}{L_1}+\frac{k_2x_2}{L_2}+\frac{k_3x_3}{L_3}\right)}\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3 \tag{15}$$

are the Fourier coefficients of u for different values of $\boldsymbol{k}$. Eq. (15) is also called the Fourier transform of u while Eq. (14) is the inverse Fourier transform of $\hat{u}$. A similar Fourier series representation is possible for the kernel function μ (see Eq. (6)) if it is assumed to be a periodic complex-valued scalar function defined over T:

$$\mu(\boldsymbol{x})=\sum_{k_3=-\infty}^{+\infty}\sum_{k_2=-\infty}^{+\infty}\sum_{k_1=-\infty}^{+\infty}\hat{\mu}(\boldsymbol{k})\,e^{2\pi\zeta\left(\frac{k_1x_1}{L_1}+\frac{k_2x_2}{L_2}+\frac{k_3x_3}{L_3}\right)} \tag{16}$$

$$\hat{\mu}(\boldsymbol{k})=\frac{1}{L_3L_2L_1}\int_0^{L_3}\int_0^{L_2}\int_0^{L_1}\mu(\boldsymbol{x})\,e^{-2\pi\zeta\left(\frac{k_1x_1}{L_1}+\frac{k_2x_2}{L_2}+\frac{k_3x_3}{L_3}\right)}\mathrm{d}x_1\mathrm{d}x_2\mathrm{d}x_3 \tag{17}$$

In this case, the PD Laplacian operator of u becomes:

$$\mathcal{L}_\mu u(\boldsymbol{x},t)=[\mu *_{\mathbb{T}} u](\boldsymbol{x},t)-\beta u(\boldsymbol{x},t) \tag{18}$$

where $(*_{\mathrm{T}})$ denotes the 3D circular (also called cyclic or periodic) convolution on T [49]:

$$[\mu *_{\mathbb{T}} u](\boldsymbol{x},t)=\int_{\mathbb{T}}\mu\left(\boldsymbol{x}-\boldsymbol{x}'\right)u\left(\boldsymbol{x}',t\right)\mathrm{d}\boldsymbol{x}' \tag{19}$$

In the Fourier space, the circular convolution is transformed into multiplication of the convoluted functions [49]:

$$\boldsymbol{\mathcal{F}}\{[\mu *_{\mathbb{T}} u](\boldsymbol{x},t)\}=L_3L_2L_1\boldsymbol{\mathcal{F}}(\mu)\,\boldsymbol{\mathcal{F}}(u)=L_3L_2L_1\hat{\mu}(\boldsymbol{k})\,\hat{u}(\boldsymbol{k},t)\,, \tag{20}$$

where $\boldsymbol{\mathcal{F}}$ denotes the Fourier transform operation.

To arrive at the solution using our Fourier-based method, we need to use the *Discrete Fourier Transform* (DFT) [47]. To this aim, we first approximate u and μ with their *truncated (finite) Fourier series*. Let N_1, N_2, and N_3 be three integer numbers up to which the Fourier series are truncated in the three directions:

$$u^N(\boldsymbol{x},t)=\sum_{k_3=-N_3/2}^{+N_3/2}\sum_{k_2=-N_2/2}^{+N_2/2}\sum_{k_1=-N_1/2}^{+N_1/2}\hat{u}(\boldsymbol{k},t)\,e^{2\pi\zeta\left(\frac{k_1x_1}{L_1}+\frac{k_2x_2}{L_2}+\frac{k_3x_3}{L_3}\right)} \tag{21}$$

$$\mu^N(\boldsymbol{x})=\sum_{k_3=-N_3/2}^{+N_3/2}\sum_{k_2=-N_2/2}^{+N_2/2}\sum_{k_1=-N_1/2}^{+N_1/2}\hat{\mu}(\boldsymbol{k})\,e^{2\pi\zeta\left(\frac{k_1x_1}{L_1}+\frac{k_2x_2}{L_2}+\frac{k_3x_3}{L_3}\right)} \tag{22}$$

We then select a uniform discretization of the spatial domain, with the same number of nodes as the Fourier modes in the truncations above [47] in each direction:

$$\begin{aligned}&\boldsymbol{x}_{ijl}=(i\,\Delta x_1,\,j\,\Delta x_2,\,l\,\Delta x_3),\ \text{where}\ \Delta x_1=\frac{L_1}{N_1};\,\Delta x_2=\frac{L_2}{N_2};\,\Delta x_3=\frac{L_3}{N_3}\\&\text{and}\ \ =\{0,\ldots,N_1-1\};\,j=\{0,\ldots,N_2-1\};\ \ l=\{0,\ldots,N_3-1\}.\end{aligned} \tag{23}$$

Thus, the *discrete Fourier transforms* (DFT) for u and μ are:

$$\tilde{u}\left(\boldsymbol{k},t\right)=\sum_{l=0}^{N_3-1}\sum_{j=0}^{N_2-1}\sum_{i=0}^{N_1-1}u^N\left(\boldsymbol{x}_{ijl},t\right)e^{-2\pi\zeta\left(\frac{k_1 i}{N_1}+\frac{k_2 j}{N_2}+\frac{k_3 l}{N_3}\right)} \tag{24}$$

$$\tilde{\mu}\left(\boldsymbol{k}\right)=\sum_{l=0}^{N_3-1}\sum_{j=0}^{N_2-1}\sum_{i=0}^{N_1-1}\mu^N\left(\boldsymbol{x}_{ijl}\right)e^{-2\pi\zeta\left(\frac{k_1 i}{N_1}+\frac{k_2 j}{N_2}+\frac{k_3 l}{N_3}\right)}, \tag{25}$$

and the *inverse discrete Fourier transforms* (iDFT) for $\tilde{u}$ and $\tilde{\mu}$ are:

$$u^N\left(\boldsymbol{x}_{ijl},t\right)=\frac{1}{N_3N_2N_1}\sum_{k_3=-\frac{N_3}{2}}^{\frac{N_3}{2}-1}\sum_{k_2=-\frac{N_2}{2}}^{\frac{N_2}{2}-1}\sum_{k_1=-\frac{N_1}{2}}^{\frac{N_1}{2}-1}\tilde{u}\left(\boldsymbol{k},t\right)e^{2\pi\zeta\left(\frac{k_1 i}{N_1}+\frac{k_2 j}{N_2}+\frac{k_3 l}{N_3}\right)} \tag{26}$$

$$\mu^N\left(\boldsymbol{x}_{ijl}\right)=\frac{1}{N_3N_2N_1}\sum_{k_3=-N_3/2}^{+N_3/2}\sum_{k_2=-N_2/2}^{+N_2/2}\sum_{k_1=-N_1/2}^{+N_1/2}\tilde{\mu}\left(\boldsymbol{k}\right)e^{2\pi\zeta\left(\frac{k_1 i}{N_1}+\frac{k_2 j}{N_2}+\frac{k_3 l}{N_3}\right)} \tag{27}$$

The PD Laplacian of u^N on the discretized domain can be expressed via one-point Gaussian quadrature:

$$\begin{aligned}\mathcal{L}_\mu u^N\left(\boldsymbol{x}_{ijl},t\right)&=\sum_{m=0}^{N_3-1}\sum_{q=0}^{N_2-1}\sum_{p=0}^{N_1-1}\mu^N\left(\boldsymbol{x}_{ijl}-\boldsymbol{x}_{pqm}\right)\left[u^N\left(\boldsymbol{x}_{pqm},t\right)-u^N\left(\boldsymbol{x}_{ijl},t\right)\right]\Delta x_1\Delta x_2\Delta x_3\\&=\sum_{m=0}^{N_3-1}\sum_{q=0}^{N_2-1}\sum_{p=0}^{N_1-1}\mu^N\left(\boldsymbol{x}_{ijl}-\boldsymbol{x}_{pqm}\right)u^N\left(\boldsymbol{x}_{pqm},t\right)\Delta x_1\Delta x_2\Delta x_3\\&-\left\{\sum_{m=0}^{N_3-1}\sum_{q=0}^{N_2-1}\sum_{p=0}^{N_1-1}\mu^N\left(\boldsymbol{x}_{ijl}-\boldsymbol{x}_{pqm}\right)\Delta x_1\Delta x_2\Delta x_3\right\}u^N\left(\boldsymbol{x}_{ijl},t\right)\\&=\left[\mu^N*_{\mathbb{T}}u^N\right]\left(\boldsymbol{x}_{ijl},t\right)-\beta^N u^N\left(\boldsymbol{x}_{ijl},t\right)\end{aligned} \tag{28}$$

With these definitions for the truncated Fourier series representations and DFT operations, one can write [50]:

$$\begin{aligned}&\left[\mu^N*_{\mathbb{T}}u^N\right]\left(\boldsymbol{x}_{ijl},t\right)=\\&\mathbf{iDFT}\left\{\mathbf{DFT}\left\{\left[\mu^N*_{\mathbb{T}}u^N\right]\left(\boldsymbol{x}_{ijl},t\right)\right\}\right\}=\mathbf{iDFT}\left\{\Delta x_1\Delta x_2\Delta x_3\mathbf{DFT}\left[\sum_{m=0}^{N_3-1}\sum_{q=0}^{N_2-1}\sum_{p=0}^{N_1-1}\mu^N\left(\boldsymbol{x}_{ijl}-\boldsymbol{x}_{pqm}\right)u^N\left(\boldsymbol{x}_{pqm},t\right)\right]\right\}=\\&\mathbf{iDFT}\left[\Delta x_1\Delta x_2\Delta x_3\tilde{\mu}\left(\boldsymbol{k}\right)\tilde{u}\left(\boldsymbol{k},t\right)\right]\end{aligned} \tag{29}$$

Using the Fast Fourier Transform (FFT) algorithms [23,24], the cost for the DFT operation and its inverse is $O(N\log 2N)$, where $N=N_1N_2N_3$ is the total number of nodes. To simplify notation, let $u^N\left(\boldsymbol{x}_{ijl},t\right)=u^N_{ijl,t}$ and $\mu^N\left(\boldsymbol{x}_{ijl}\right)=\mu^N_{ijl}$. The linear nonlocal operator in discrete form is:

$$\mathcal{L}_\mu u^N\big|_{ijl,t}=\mathbf{FFT}^{-1}\left[\Delta x_1\Delta x_2\Delta x_3\mathbf{FFT}\left(\mu^N_{ijl}\right)\mathbf{FFT}\left(u^N_{ijl,t}\right)\right]-\beta^N u^N_{ijl,t}, \tag{30}$$

where $\mathbf{FFT}$ and $\mathbf{FFT}^{-1}$refer to the FFT and inverse FFT operations, and:

$$\beta^N=\sum_{l=0}^{N_3-1}\sum_{j=0}^{N_2-1}\sum_{i=0}^{N_1-1}\mu^N\left(\boldsymbol{x}_{ijl}\right)\Delta x_1\Delta x_2\Delta x_3, \tag{31}$$

For singular kernels ($\lim_{\boldsymbol{x}\to 0}\mu\left(\boldsymbol{x}\right)=\infty$, e.g. the kernel in Section 5.3), $\mu^N*_{\mathbb{T}}u^N$ and β^N become singular which make $\mathcal{L}_\mu u^N\big|_{ijl,t}$ in Eq. (28) to seem undefined, but in fact it is defined. The canceling pairs of terms lead to canceling singularities in $\mu^N*_{\mathbb{T}}u^N$ and β^N, and lead to a value of zero for $\mathcal{L}_\mu u^N\big|_{ijl,t}$ at the singularity point ($ijl=pqm$ in Eq. (28)). In the actual implementation, to overcome numerical issues we set $\mu^N\left(\boldsymbol{x}_{ijl}\right)\big|_{\boldsymbol{x}_{ijl}=0}=0$.

For integrable kernels, β can be calculated analytically at the continuum level formulation in Eq. (18). When using the discretized formulation, however, the β integral should also be computed using the same discretization as

the convolution term $\mu^N *_{\mathbb{T}} u^N$. Otherwise, discretization inconsistency between $\mu^N *_{\mathbb{T}} u^N$ and β^N can undermine the accuracy of the computed $\mathcal{L}_\mu u^N$.

Let $r_{ijl,t}$ denote $r\left(\boldsymbol{x}_{ijl}, t\right)$, then the spatially-discretized PD diffusion equation with the FCBM then reads:

$$\frac{\mathrm{d}u^N_{ijl,t}}{\mathrm{d}t} = \nu \mathbf{FFT}^{-1}\left[\Delta x_3 \Delta x_2 \Delta x_1 \mathbf{FFT}\left(\mu^N_{ijl}\right) \mathbf{FFT}\left(u^N_{ijl,t}\right)\right] - \nu \beta^N u^N_{ijl,t} + r_{ijl,t} \tag{32}$$

Using, for example, the forward Euler method in time for approximating the solution of the ordinary differential equation (ODE) in Eq. (32), yields:

$$u^N_{ijl,t+\Delta t} = u^N_{ijl,t} + \Delta t \left\{\nu \mathbf{FFT}^{-1}\left[\Delta x_3 \Delta x_2 \Delta x_1 \tilde{\mu}_{k_1 k_2 k_3} \mathbf{FFT}\left(u^N_{ijl,t}\right)\right] - \nu \beta^N u^N_{ijl,t} + r_{ijl,t}\right\} \tag{33}$$

Note that $\tilde{\mu}_{k_1 k_2 k_3} = \mathbf{FFT}\left(\mu^N_{ijl}\right)$ is precomputed only once, before the time integration loop, since it does not depend on time in our case.

The steps shown above for the isotropic homogeneous linear PD diffusion given in Eq. (5) can be easily extended to nonlinear cases in which the nonlinear PD operator is decomposable into convolutional terms. For example, in the case of the nonlinear diffusion with the time dependent properties given by Eqs. (4) and (13), the FCBM leads to the following discretization:

$$\begin{aligned} u^N_{ijl,t+\Delta t} = u^N_{ijl,t} &\\ + \Delta t \Bigg\{ &\frac{1}{2}\Delta x_3 \Delta x_2 \Delta x_1 \left\{ f^N_{ijl,t} \mathbf{FFT}^{-1}\left[\mathbf{FFT}\left(\omega^N_{ijl,t}\right) \mathbf{FFT}\left(h^N_{ijl,t} u^N_{ijl,t}\right)\right]\right. \\ - &f^N_{ijl,t} u^N_{ijl,t} \mathbf{FFT}^{-1}\left[\mathbf{FFT}\left(\omega^N_{ijl,t}\right) \mathbf{FFT}\left(h^N_{ijl,t}\right)\right] \\ + &h^N_{ijl,t} \mathbf{FFT}^{-1}\left[\mathbf{FFT}\left(\omega^N_{ijl,t}\right) \mathbf{FFT}\left(f^N_{ijl,t} u^N_{ijl,t}\right)\right] \\ - &\left. h^N_{ijl,t} u^N_{ijl,t} \mathbf{FFT}^{-1}\left[\mathbf{FFT}\left(\omega^N_{ijl,t}\right) \mathbf{FFT}\left(f^N_{ijl,t}\right)\right]\right\} + r_{ijl,t} \Bigg\} \end{aligned} \tag{34}$$

The only difference between the linear and nonlinear cases is the discretized diffusion operator, because the nonlinear case requires more FFT and FFT^{-1} calculations (in the example above the nonlinear case would cost four times more than the linear case). Notably, the complexity for the nonlinear case remains $O(N \log 2N)$.

4. The FCBM for 3D peridynamic diffusion on arbitrary domains

In this section, we first discuss volume constraints (nonlocal boundary conditions) in peridynamic problems. Then we introduce a new approach to apply the FCBM to problems in non-periodic domains with arbitrary volume constraints.

4.1. Boundary conditions in peridynamics

Boundary conditions in PD nonlocal approach are in the form of volume constraints, defined over a δ thick volume layer on the domain exterior. Fig. 2 schematically shows a PD domain Ω and its nonlocal boundary layer Γ. More details on PD volume constraints are discussed in [51].

According to nonlocal vector calculus [51], *volume-constrained peridynamic problems* are defined analogous to *boundary value problems* with PDEs in the local theory. Volume constrained peridynamic transient diffusion problem can be generally expressed as [42]:

$$\begin{cases} \dfrac{\partial u\left(\boldsymbol{x}, t\right)}{\partial t} = \mathcal{L}_\gamma u\left(\boldsymbol{x}, t\right) + r\left(\boldsymbol{x}, t\right) & \boldsymbol{x} \in \Omega, \ t > 0 \\ u\left(\boldsymbol{x}, 0\right) = u_0 \left(\text{initial condition}\right) & \boldsymbol{x} \in \Omega \\ G\left(u\left(\boldsymbol{x}, t\right)\right) = 0 \ \left(\text{volume constraints}\right) & \boldsymbol{x} \in \Gamma, \quad t \geq 0 \end{cases} \tag{35}$$

where G is a function that prescribes the constraints of u on Γ. In Section 5.2 we present the solution for a problem with such volume constraints.

In most engineering problems, measurements are taken at the surfaces of a body, not through a layer near the surface. The natural representation of such measurements is via local boundary conditions. Methods for imposing

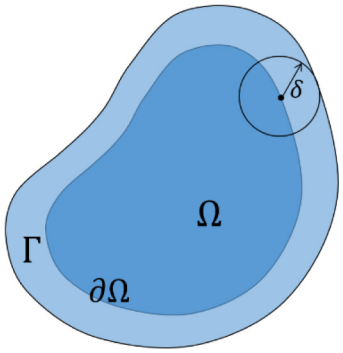


Fig. 2. A peridynamic domain Ω and its nonlocal boundary layer or constrained volume Γ and their interface $\partial\Omega$.

local boundary conditions on peridynamic bodies have been studied in [52–55]. One such method is the *fictitious layer/nodes* method (e.g. [52,54]) where certain volume constraint is considered on Γ, leading to desired boundary conditions on $\partial\Omega$ (see Fig. 2). This method is exact in 1D and reasonably accurate in 2D and 3D (except at regions with high gradients, corners and sharp curvatures, see [54]). In Sections 5.1 and 5.3, we use this method to enforce local BCs in two PD problems, by determining the corresponding volume constraint values on the nonlocal boundary layer Γ. Details for imposing Dirichlet and Neumann BCs to peridynamic models are provided in Appendix A.

4.2. Enforcing boundary conditions with a new "embedded constraint" approach

In this section, we introduce a new way to extend the FCBM described in Section 3, to bounded domains with arbitrary volume constraints (boundary conditions).

As mentioned earlier, while domain periodicity is a requirement for Fourier methods, modifications have been introduced for local theories to solve problems with non-periodic boundary conditions (e.g. see [35,56–59]). In [34], we introduced a volume penalization technique to solve problems with non-periodic volume constraints in 1D peridynamic diffusion using a Fourier-based method. Here, we introduce a simpler, more general, and more efficient approach to impose non-periodic boundary conditions in the solution of PD Fourier-based models.

Consider the following nonlocal transient diffusion volume-constrained problem defined over an arbitrary domain Ω and its nonlocal boundary layer Γ:

$$\begin{cases} \dfrac{\partial u(\boldsymbol{x},t)}{\partial t} = \mathcal{L}_\gamma u(\boldsymbol{x},t) + r(\boldsymbol{x},t) & \boldsymbol{x}\in\Omega,\ t>0 \\ u(\boldsymbol{x},0) = u_0 & \boldsymbol{x}\in\Omega \\ u(\boldsymbol{x},t) = g(\boldsymbol{x},t) & \boldsymbol{x}\in\Gamma,\ t\geq 0 \end{cases} \tag{36}$$

We emphasize that the nonlocal formulation in Eq. (36) can "represent" both classical (local) Dirichlet and Neumann BCs. Indeed, one can specify a certain profile for $g(\boldsymbol{x},t)$ over Γ, such that the solution of Eq. (36) *effectively* matches a given classical Dirichlet or Neumann BC on $\partial\Omega$. Formulas for how to specify g for Dirichlet or Neumann BCs are provided in Appendix A.

To make the FCBM work for this problem, we first place $\Omega\cup\Gamma$ inside a periodic box ($\mathbb{T}$) (see Fig. 3 for the 2D case). $\mathbb{T}\backslash(\Omega\cup\Gamma)$ is denoted by Λ and is referred to as the *gap region*.

We then replace the volume-constrained problem in Eq. (36) with the following periodic problem:

$$\begin{cases} \dfrac{\partial u(\boldsymbol{x},t)}{\partial t} = \chi(\boldsymbol{x},t)\left[\mathcal{L}_\gamma u(\boldsymbol{x},t) + r(\boldsymbol{x},t)\right] + [1-\chi(\boldsymbol{x},t)]\dfrac{\partial w(\boldsymbol{x},t)}{\partial t} & \boldsymbol{x}\in\mathbb{T},\ t>0 \\ u(\boldsymbol{x},0) = u_0 & \boldsymbol{x}\in\mathbb{T} \end{cases} \tag{37}$$

where χ denotes the *mask function*

$$\chi(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x}\in\Omega \\ 0 & \boldsymbol{x}\in\mathbb{T}\backslash\Omega = \Gamma\cup\Lambda \end{cases} \tag{38}$$

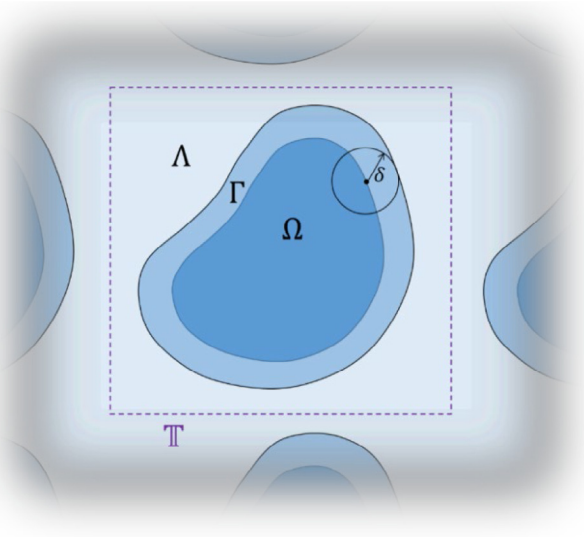


Fig. 3. Extension of the bounded-domain Ω and its nonlocal boundary layer Γ by periodicity.

and

$$w(\boldsymbol{x},t)=\begin{cases}0 & \boldsymbol{x}\in\Omega\\ g(\boldsymbol{x},t) & \boldsymbol{x}\in\Gamma\\ z(\boldsymbol{x},t) & \boldsymbol{x}\in\Lambda\end{cases} \tag{39}$$

where $g(\boldsymbol{x},t)$ is the volume constraint in Eq. (36), and $z(\boldsymbol{x},t)$ is a function defining *gap values*. These gap values can be defined arbitrarily, for instance taken as zero. The reason is that, unlike the volume constraint function g, the domain of definition of the gap function z is a horizon size (δ) away from Ω. Therefore, there is no "direct interaction" between the values of the z (over Λ) and u (over Ω), see Fig. 3. An arbitrary selection for the z function values may mean a jump discontinuity at the Γ–Λ interface for the unknown function u (over $\mathbb{T}$). We have numerically studied to see whether this lack of smoothness leads to associated errors (Gibbs-type phenomenon). We have not yet noticed any undesired oscillations. We have tested this for the elastic problem as well (see [46]) and no Gibbs-type phenomenon was observed there either. It is possible that the reason behind these observations is the smoothing effect introduced by convolution operations, noted previously in the literature (see [60–62]). One can, of course, employ smooth extensions to $\mathbb{T}$ using a technique similar to those available in the literature for local models (see [47,57]).

Now that the original volume-constrained problem is replaced with a periodic problem. We apply the Fourier collocation method described in Section 3 to spatially discretize the periodic problem. Let the domain $\mathbb{T}$ be discretized according to Eq. (23); then problem (37) becomes the following initial value problem for a system of ODEs:

$$\begin{cases}\dfrac{\mathrm{d}u^N_{ijl,t}}{\mathrm{d}t}=\chi_{ijl}\left(\mathcal{L}_\gamma u\big|_{ijl,t}+r_{ijl,t}\right)+\left(1-\chi_{ijl}\right)\dfrac{\mathrm{d}w_{ijl,t}}{\mathrm{d}t} & \boldsymbol{x}\in\mathbb{T},\ t>0\\ u_{ijl,t=0}=u_{0,ijl} & \boldsymbol{x}\in\mathbb{T}\end{cases} \tag{40}$$

where

$$\chi_{ijl}=\begin{cases}1 & \boldsymbol{x}_{ijl}\in\Omega\\ 0 & \boldsymbol{x}_{ijl}\in\Gamma\cup\Lambda\end{cases} \tag{41}$$

and

$$w_{ijl,t} = \begin{cases} 0 & \boldsymbol{x}_{ijl} \in \Omega \\ g_{ijl,t} & \boldsymbol{x}_{ijl} \in \Gamma \\ z_{ijl,t} & \boldsymbol{x}_{ijl} \in \Lambda \end{cases} \tag{42}$$

$\mathcal{L}_\gamma u\big|_{ijl,t}$ is the discrete peridynamic Laplacian operator calculated via Fourier transform according to Section 3. The particular formula for $\mathcal{L}_\gamma u\big|_{ijl,t}$ depends on the kernel inside the operator (see Eqs. (30) and (34) as two examples).

According to Eq. (41), one re-writes the system of ODE in Eq. (40):

$$\frac{\mathrm{d}u^N_{ijl,t}}{\mathrm{d}t} = \begin{cases} \mathcal{L}_\gamma u\big|_{ijl,t} + r_{ijl,t} & \boldsymbol{x}_{ijl} \in \Omega \\ \dfrac{\mathrm{d}w_{ijl,t}}{\mathrm{d}t} & \boldsymbol{x}_{ijl} \in \Gamma \cup \Lambda \end{cases} \tag{43}$$

To solve (43) in time, we use an ODE solver (here we select the forward Euler method). Knowing $u = w$ on $\mathbb{T}\backslash\Omega$, one can write:

$$u^N_{ijl,t+\Delta t} = \begin{cases} u^N_{ijl,t} + \Delta t \left(\mathcal{L}_\gamma u\big|_{ijl,t} + r_{ijl,t} \right) & \boldsymbol{x}_{ijl} \in \Omega \\ w_{ijl,t+\Delta t} & \boldsymbol{x}_{ijl} \in \Gamma \cup \Lambda \end{cases} \tag{44}$$

We use Eq. (41) to re-write Eq. (44):

$$u^N_{ijl,t+\Delta t} = \chi_{ijl} \left[u^N_{ijl,t} + \Delta t \left(\mathcal{L}_\gamma u\big|_{ijl,t} + r_{ijl,t} \right) \right] + \left(1 - \chi_{ijl}\right) w_{ijl,t+\Delta t} \tag{45}$$

We call this way of imposing volume constraints the *"embedded constraint"* method, since the constraints are embedded into the governing integro-differential equation.

According to the stability analysis for FCBM-VP in [34], one can easily show that the stability condition in the absence of a penalization term for linear diffusion results in:

$$\Delta t \leq \frac{1}{\nu\beta} \tag{46}$$

This condition implies that stability does not depend on the spatial discretization resolution, but depends on the horizon size and the kernel function. This condition is similar to the CFL condition in the meshfree method [52]. In the volume penalization method introduced in [34] (see Appendix B for a brief review), higher accuracy requires a potentially larger penalization parameter which may shrink the maximum allowed time step, and consequently increase the computational cost. One advantage of the new embedded constraint method compared with the volume penalization is that *the maximum time step size does not depend on a penalization parameter*.

Another major advantage of EC over VP is that EC can be applied directly to other PD equations, e.g. PD equations of motion, while the penalization method in [34] cannot without modifications. The specifics of the VP method depend on whether one solves PDEs with first-order or second-order partial derivatives in time (see [34]).

5. Examples and discussion

In this section, we first numerically evaluate the accuracy and convergence of the newly introduced FCBM-EC for a 1D peridynamic linear diffusion example with non-periodic boundary conditions and compare its performance versus the penalization method in [34]. Then, we solve a 2D volume-constrained peridynamic nonlinear diffusion problem and perform spatial and temporal convergence studies to *the exact nonlocal solution* (obtained via the method of manufactured solutions). Finally, we show a 3D diffusion problem with non-periodic local boundary conditions using the FCM-EC and discuss the computational efficiency of the method relative to the meshfree discretization and FEM-based Abaqus solutions.

5.1. Performance of the FCBM with embedded constraint (EC) in 1D

To evaluate the performance of the FCBM-EC in terms of accuracy and convergence, we test the method for solving a PD problem for which we have the analytical solution (based on the method of manufactured solutions,

see e.g. [63]). Consider the following 1D PD linear diffusion problem on the bounded domain $\Omega = \left[-\frac{L}{2}, \frac{L}{2}\right]$:

$$\frac{\partial u\,(x,t)}{\partial t} = \nu \int_{\mathcal{H}_x} \mu(x'-x)[u\,(x') - u(x)]\mathrm{d}x' + r\,(x,t) \tag{47}$$

where:

$$\mu\,(x) = \begin{cases} \dfrac{12}{\delta^3}\left(1 - \dfrac{|x|}{\delta}\right); & |x| \le \delta \\ 0; & |x| > \delta \end{cases} \tag{48}$$

and

$$r\,(x,t) = \nu\left\{\frac{6L^2}{\delta^4\pi^2}\left[\cos\left(\frac{2\pi\delta}{L}\right) - 1\right] + \frac{12}{\delta^2} - 1\right\} e^{-\nu t}\sin\left(\frac{2\pi x}{L}\right), \tag{49}$$

with the initial condition:

$$u\,(x,0) = \frac{2x}{L} + \sin\left(\frac{2\pi x}{L}\right), \tag{50}$$

and subjected to Dirichlet boundary conditions:

$$u\left(-\frac{L}{2}, t\right) = -1 \tag{51}$$

$$u\left(\frac{L}{2}, t\right) = 1 \tag{52}$$

This problem was used in [34] to verify the FCBM with volume penalization. The exact analytical solution to the above problem is:

$$u\,(x,t) = \frac{2x}{L} + e^{-\nu t}\sin\left(\frac{2\pi x}{L}\right) \tag{53}$$

In order to apply the FCBM we first extend the bounded domain $\Omega = [-\frac{L}{2}, \frac{L}{2}]$ by a length δ on both ends to create the boundary volumes (Γ). In the extended domain $\mathbb{T} = \left[-\frac{L}{2} - \delta, \frac{L}{2} + \delta\right)$, volume constraints should be specified on $\Gamma_1 = \left[-\frac{L}{2} - \delta, -\frac{L}{2}\right)$ and $\Gamma_2 = \left(\frac{L}{2}, \frac{L}{2} + \delta\right]$, as the left and right boundary layers. In this example, we consider $\mathbb{T} = \Omega \cup \Gamma$ to be periodic; therefore no gap region is included ($\Lambda = \varnothing$). There are two alternatives for applying the local BC given in Eqs. (51) and (52): (1) use the exact volume constraints (because we employ the exact, manufactured solution for the nonlocal problem), and (2) use a technique (e.g. Appendix A and [34]) for constructing volume constraints based on the given local BCs. In order to compare with the results in [34], here we use the second approach. This example will also demonstrate how one can apply local BCs to a general peridynamic problem, even when the nonlocal analytical solution is not known.

Let $L = 2$, $\nu = 0.2$, $\delta = 0.2$, and the total diffusion time $t_{\max} = 15$. Accordingly, $\mathbb{T} = [-1.2, 1.2)$ is the periodic extended domain with -1.2 identified with 1.2. Let $N = 2^9$ be the total number of nodes and $\Delta t = 1.67 \times 10^{-2}$ as the maximum time step allowed by Eq. (46). Fig. 4 shows the numerical results from the FCBM-EC versus the exact solution. The corresponding code is found in the Supplementary materials.

As observed, boundary conditions are correctly enforced using the introduced IC method.

We performed convergence tests on this example to study the influence of spatial and temporal resolutions on the maximum relative error: $\max_{0<t<t_{\max}} \frac{\left\|u_{i,t} - u^N_{i,t}\right\|_\infty}{\left\|u_{i,t=0}\right\|_\infty}$. In peridynamics the ratio $m = \frac{\delta}{\Delta x}$ is usually referred to as "m factor" [64] which is a measure for family node density. For convergence tests in terms of spatial discretization, we refine the spatial discretization while keeping the horizon size fixed. This is called "m-convergence" [64]. We varied $N = 2^6$ to 2^{13} (m varied from 5.3 to 682.7 accordingly). Results are presented in Fig. 5a. We performed a similar convergence study using the previously introduced VP method (see Appendix B and [34]), and the results are shown in Fig. 5b. The code for the FCBM-VP is found in the Supplementary materials.

By comparing results in Fig. 5a and Fig. 5b, we observe that the EC method is more accurate than the VP approach for the same time step size. Note that the temporal resolution in the VP method is tied to penalization parameter (ε) magnitude. For each time step, we chose the smallest possible ε to test the optimum performance of

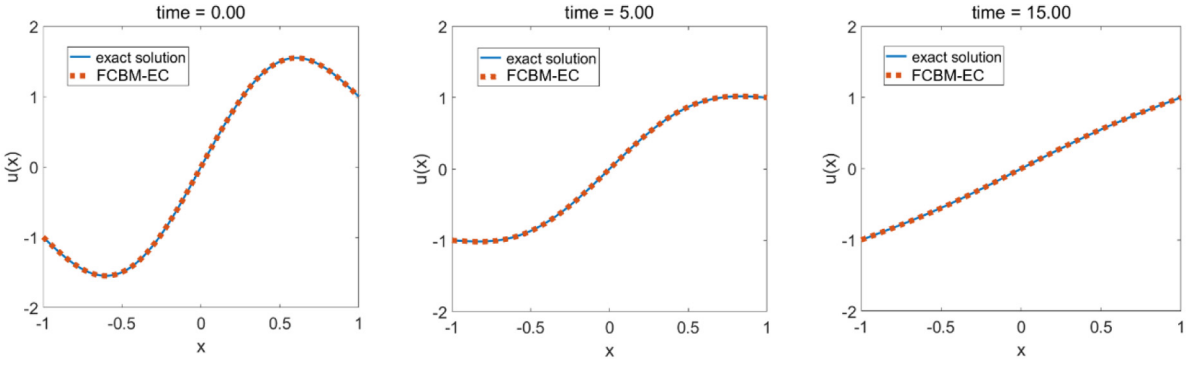


Fig. 4. Comparison of the numerical solution obtain with the FCBM-EC and the exact solution for a 1D nonlocal diffusion problem with two Dirichlet (non-periodic) boundary conditions.

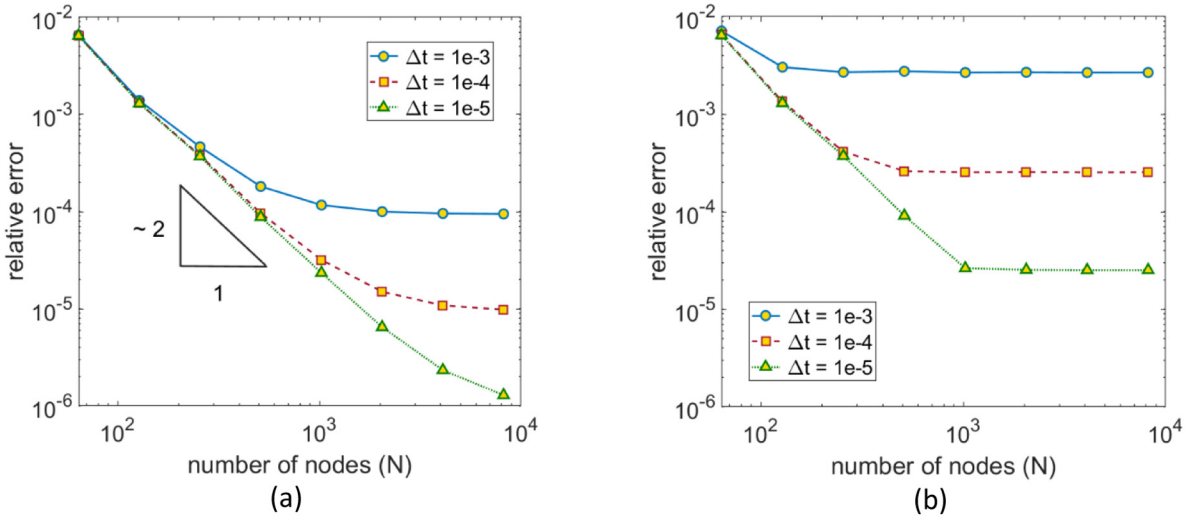


Fig. 5. Studies in terms of spatial resolution for three time step sizes: a) FCBM-EC; b) FCBM-VP, with the penalization parameter resolved for each time step size for optimality.

VP versus EC (see Appendix B). For both the FCBM-EC and FCBM-VP, the rate of convergence is $O(\Delta x^2)$ for sufficiently small Δt.

While rigorous error estimates and convergence analysis have not yet been carried out for this method, the observed quadratic rate of convergence is likely due to the quadratic rate of the Gaussian quadrature used in Eq. (28).

Although Fig. 5a already shows the error for three different time steps, we perform a separate temporal convergence test for FCBM-EC and show the results in Fig. 6. For this test the spatial resolution is fixed: $N = 2^{13} (m = 682.7)$.

We observe a linear rate of convergence, $O(\Delta t)$, which is expected from the Forward Euler time integrator.

One can use methods with higher order accuracy (e.g. explicit 4th order Runge–Kutta) for temporal integration. In that case, the stability restriction on time step size for explicit methods would be different from that shown in Eq. (46).

To check the computational complexity for the two methods (the meshfree method using direct quadrature, and the FCBM-EC using the FFT), we solved the 1D problem above, considering various spatial discretization size. We ran the meshfree method using $N = 2^9$ to 2^{20} and the FCBM-EC with $N = 2^{16}$ to 2^{28}. The time step in all these tests was chosen to be $\Delta t = 1.67 \times 10^{-2}$. Simulations were conducted using a Dell-Precision T7810 workstation PC, Intel(R) Xeon(R) CPU E5-2687 W v3 @3.10 GHz logical processors, and 64 GB of installed memory. We

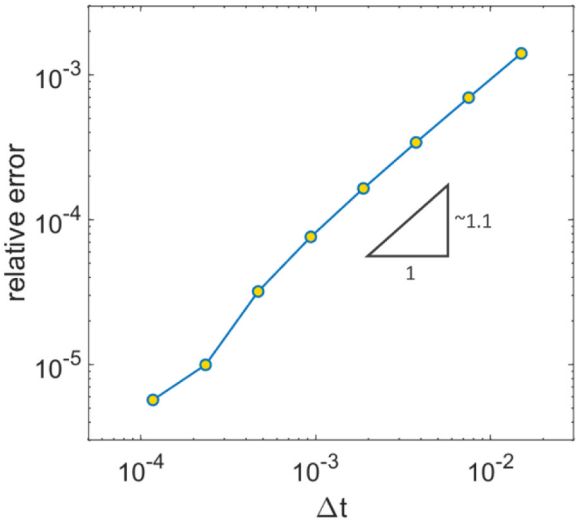


Fig. 6. Convergence study for the FCBM-EC in terms of temporal resolution for a 1D example with non-periodic boundary conditions.

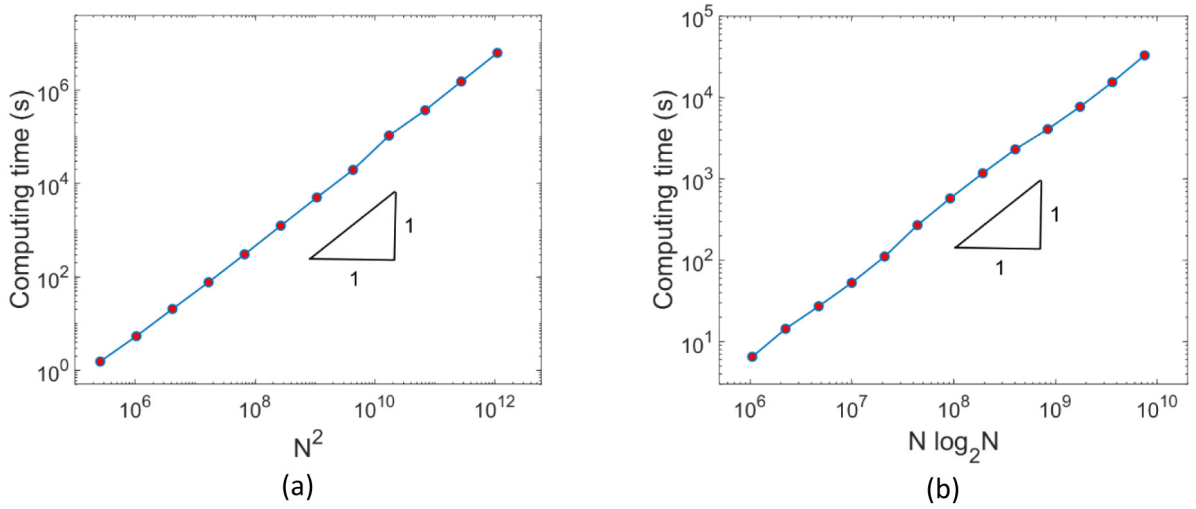


Fig. 7. Complexity of the meshfree method (direct quadrature) (a) versus the FCBM-EC (FFT-based) (b).

utilized only one CPU for these computations. Fig. 7 shows the computational run times for each method, relative to the problem size.

As expected, the complexity of meshfree method with direct quadrature is $O(N^2)$, while that of the FCBM-EC is $O(N\log_2 N)$.

5.2. A 2D nonlinear volume-constrained problem

In this part, we examine the new FCBM-EC for solving nonlinear and non-periodic PD diffusion problems in 2D. We use again the *method of manufactured solutions* [63] to construct a 2D nonlinear volume-constrained peridynamic transient diffusion problem. Consider the following function:

$$u(\boldsymbol{x}, t) = e^{-0.2t}\ (1 - x_1^2)(1 - x_2^2), \boldsymbol{x} = \{x_1, x_2\} \in \mathbb{R}^2 \text{ and } t \geq 0 \tag{54}$$

Eq. (54) is the analytical solution to the following nonlinear problem:

$$\begin{cases} \dfrac{\partial u\left(\boldsymbol{x},t\right)}{\partial t} = \mathcal{L}_\gamma u\left(\boldsymbol{x},t\right) + r\left(\boldsymbol{x},t\right) & \boldsymbol{x} \in \Omega = [-1,1]^2\,,\ t > 0 \\ u\left(\boldsymbol{x},0\right) = \left(1-x_1^2\right)\left(1-x_2^2\right) & \boldsymbol{x} \in \Omega = [-1,1]^2 \\ u\left(\boldsymbol{x},t\right) = e^{-0.2t}\ \left(1-x_1^2\right)\left(1-x_2^2\right) & \boldsymbol{x} \in \Gamma = [-1.2,1.2]^2 \setminus [-1,1]^2\,,\ t \geq 0 \end{cases} \tag{55}$$

where

$$\mathcal{L}_\gamma u\left(\boldsymbol{x},t\right) = \int_{\mathcal{H}_{\boldsymbol{x}}} \gamma\left(\boldsymbol{x}',\boldsymbol{x}\right)\left[u\left(\boldsymbol{x}',t\right) - u\left(\boldsymbol{x},t\right)\right] \mathrm{d}\boldsymbol{x}' \tag{56}$$

with

$$\gamma(\boldsymbol{x}',\boldsymbol{x}) = u\left(\boldsymbol{x},t\right) u\left(\boldsymbol{x}',t\right) \omega(\boldsymbol{x}'-\boldsymbol{x}) \tag{57}$$

and

$$\omega\left(\boldsymbol{x}\right) = 0.2\left(1 - \frac{\|\boldsymbol{x}\|}{\delta}\right). \tag{58}$$

This is a special case of Eq. (13) where $f = h = u$. If u is temperature for example, with this kernel, Eq. (55) is a nonlinear PD problem with temperature dependent diffusivity. In Eq. (55):

$$\begin{aligned} r\left(\boldsymbol{x},t\right) = \frac{-0.2\pi e^{-0.6t}}{443520}\left(1-x_1^2\right)\left(1-x_2^2\right)\big[&189\delta^{10} + \delta^8\left(4620x_1^2 + 4620x_2^2 - 3080\right) \\ &+ \delta^6\left(7920x_1^4 + 92400x_1^2x_2^2 - 44880x_1^2 + 7920x_2^4 - 44880x_2^2 + 23760\right) \\ &+ \delta^4\left(110880x_1^4x_2^2 - 22176x_1^4 + 110880x_1^2x_2^4 - 443520x_1^2x_2^2\right. \\ &\left. + 155232x_1^2 - 22176x_2^4 + 155232x_2^2 - 44352\right) + 44352e^{0.4t}\big] \end{aligned} \tag{59}$$

Assuming $\delta = 0.4$, we consider $\Omega \cup \Gamma = [-1.4, 1.4]^2$ to be the 2D periodic box $\mathbb{T}$ to solve the problem with the FCBM-EC. Note that for this example $\Lambda = \varnothing$. We discretize the domain with uniform grid spacing in both directions: $N_1 = N_2 = 2^7$ which results in an m factor of 36.6. The numerical solution with the FCBM-EC is calculated from:

$$u^N_{ij,t+\Delta t} = \chi_{ij}\left[u^N_{ij,t} + \Delta t\left(\mathcal{L}_\gamma u\big|_{ij,t} + r_{ijl,t}\right)\right] + \left(1-\chi_{ij}\right) w_{ij,t+\Delta t} \tag{60}$$

where

$$\begin{aligned} \mathcal{L}_\gamma u\big|_{ij,t} = 0.2\Delta x_2 \Delta x_1 \Big\{ &u^N_{ij,t}\,\mathbf{FFT}^{-1}\left[\mathbf{FFT}\left(\omega^N_{ij,t}\right)\mathbf{FFT}\left(\left(u^N_{ij,t}\right)^2\right)\right] \\ &- \left(u^N_{ij,t}\right)^2 \mathbf{FFT}^{-1}\left[\mathbf{FFT}\left(\omega^N_{ij,t}\right)\mathbf{FFT}\left(u^N_{ij,t}\right)\right]\Big\} \end{aligned} \tag{61}$$

Eq. (61) is obtained when substituting f and h with u in Eq. (34). We computed Eq. (61) to solve this nonlinear transient problem up to $t = 15$ with $\Delta t = 0.01$. Simulation results for three snapshots at $t = 1, 5$ and 15 are shown in Fig. 8. The corresponding Matlab code is provided in the Supplementary materials. The relative error is computed from:

$$\text{error}\left(\boldsymbol{x}_{ij},t\right) = \frac{\left|u^N_{ij,t} - u^{\text{exact}}_{ij,t}\right|}{\left\|u_{ij,0}\right\|_\infty} \tag{62}$$

This example shows how the FCBM-EC is capable of solving nonlinear volume-constrained PD diffusion problems in higher dimensions.

We have also performed spatial and temporal convergence studies for this nonlinear 2D example. For spatial convergence, we used a fixed $\Delta t = 10^{-5}$ (for $0 < t \leq 5$) and varied m-factor from 4.6 to 73.1 ($N_1 = N_2$ from 2^4 to 2^8). For temporal convergence, we used a fixed spatial resolution of $N_1 = N_2 = 2^8$ ($m = 73.1$) and varied Δt from 0.1 to 10^{-5} (for $0 < t \leq 5$). The results for these tests are provided in Fig. 9. Here, the relative error is computed by: $\max_{0<t<5} \frac{\left\|u_{ij,t} - u^N_{ij,t}\right\|_\infty}{\left\|u_{ij,t=0}\right\|_\infty}$.

Similar to the 1D example (see Figs. 5a and 6) we observe that convergence rate is quadratic in terms of Δx and linear in terms of Δt (due to the Forward Euler time integrator) for this 2D nonlinear example.

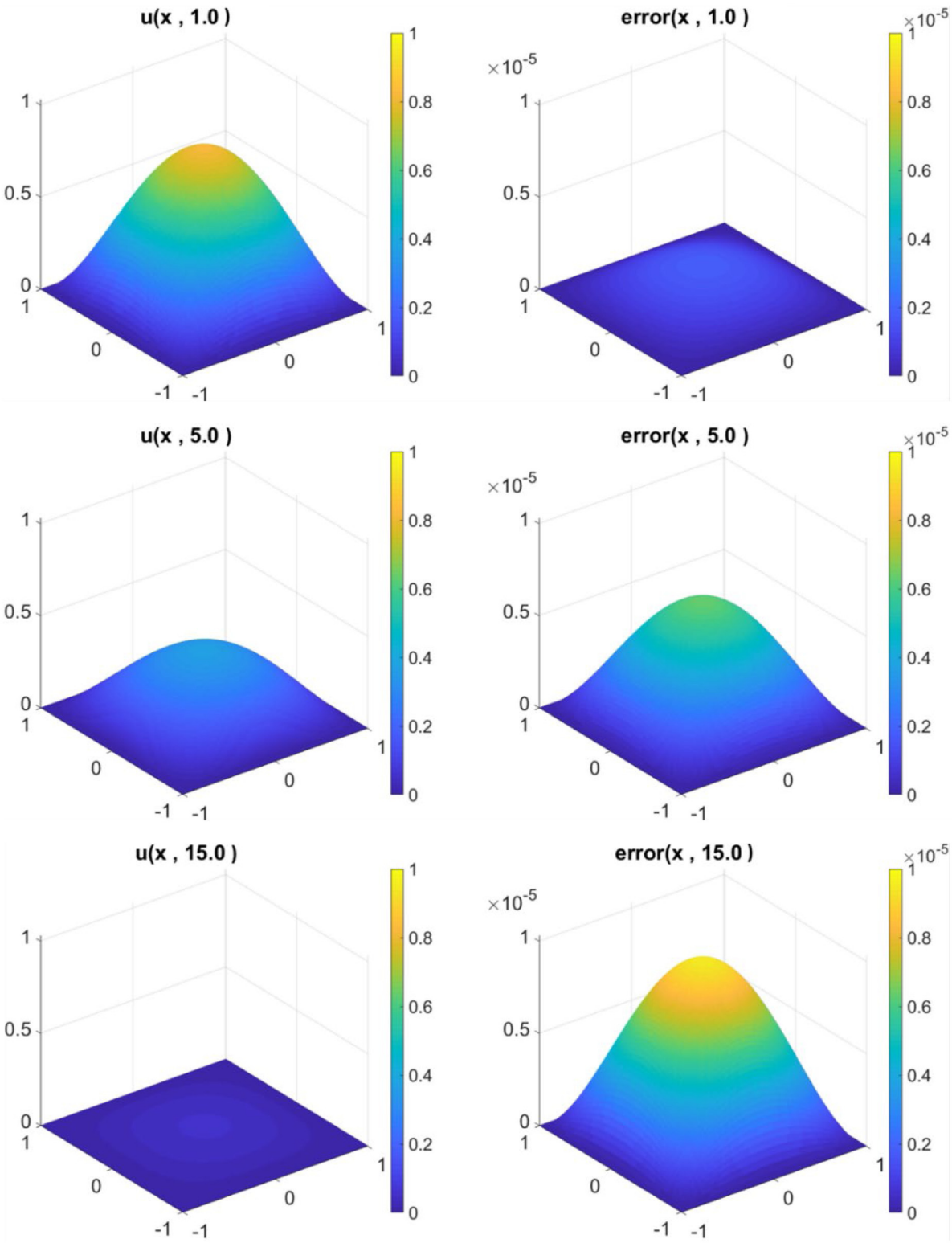


Fig. 8. Solution of a 2D nonlinear volume-constrained peridynamic transient diffusion problem with the FCBM-EC. Left column shows the solution at three times ($t = 1$, 5, and 15) while the right column shows the corresponding relative error distribution compared with the exact nonlocal solution.

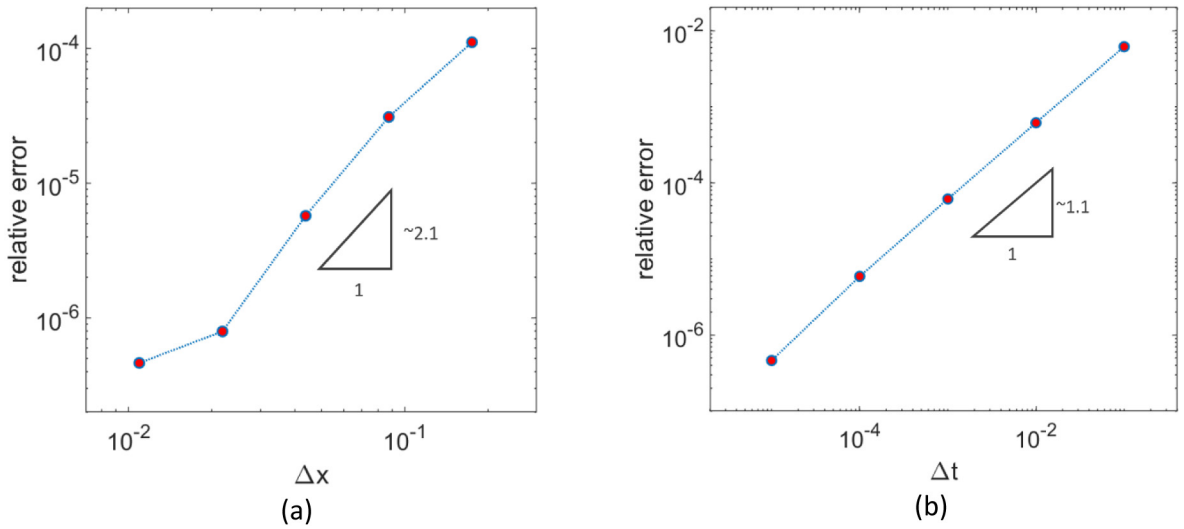


Fig. 9. Convergence studies for FCBM-EC in terms of spatial (a) and temporal (b) resolutions for the nonlinear 2D volume-constrained peridynamic transient diffusion problem.

After verification of the method, we aim to examine the efficiency of FCBM-EC for a 3D problem with complex geometry and boundary conditions.

5.3. Efficiency gains of the FCBM-EC for problems in 3D

In this part, we use the FCBM-EC and the meshfree (one-point Gaussian integration) spatial discretization to solve a 3D PD transient diffusion example with non-periodic local BCs in a domain with a non-trivial geometry. We compare the performance of the two approaches for discretization sizes ranging from 2 million (m-factor of 5.8) to more than 1 billion nodes (m-factor of 46.5), in 3D. Note that in practical problems, there is rarely a reason for using m-factor values larger than 5–10 [65–67]. However, when high accuracy is required, or for problems where nonlocality is relatively large compared to the sample's size, larger m-values (like the ones used here) may be needed. The quadratic convergence rate for the spatial discretization discussed in Sections 5.1 and 5.2 motivates using such large m-values.

The example in this section is a 3D PD diffusion in a $2 \times 2 \times 2$ cube with a cutout, having fixed (but different) u values imposed on the top and bottom surfaces (Dirichlet BCs), and zero flux conditions imposed on all four cube's sides (Neumann BCs). The cube has a horizontally aligned penny-shaped insulated cutout/obstacle at the center (see Fig. 10). The cutout acts as a diffusion barrier with zero flux BCs on all of its surfaces. The mathematical layout for this problem is:

$$\frac{\partial u\left(\boldsymbol{x}, t\right)}{\partial t} = \nu \int_{\mathcal{H}_{\boldsymbol{x}}} \mu(\boldsymbol{x}' - \boldsymbol{x})[u\left(\boldsymbol{x}'\right) - u(\boldsymbol{x})]\mathrm{d}\boldsymbol{x}' \tag{63}$$

on $\Omega = [-1, 1]^3$ with the kernel function [8,68]:

$$\mu\left(\boldsymbol{x}\right) = \begin{cases} \frac{9}{2\pi\delta^3\|\boldsymbol{x}\|^2}; \|\boldsymbol{x}\| \leq \delta \\ 0; \|\boldsymbol{x}\| > \delta, \end{cases} \tag{64}$$

subjected to the initial condition:

$$u\left(\boldsymbol{x}, 0\right) = 0, \tag{65}$$

and the local boundary conditions:

$$u\left(\boldsymbol{x}, t\right) = 1, \boldsymbol{x} = \{x_1, x_2, 1\} \quad \text{(a)} \tag{66}$$

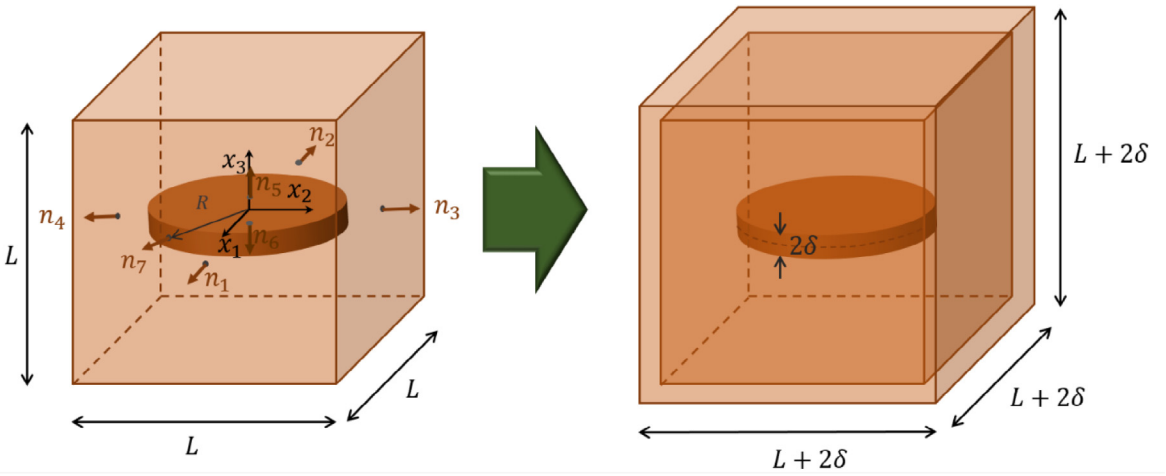


Fig. 10. Schematics of the cube with an insulated cut-out as the original domain (Ω), and the extended domain ($\mathbb{T} = \Omega \cup \Gamma$) with nonlocal boundary layers for imposing boundary conditions in the FCBM-EC.

$$u(\boldsymbol{x}, t) = 0, \boldsymbol{x} = \{x_1, x_2, -1\} \tag{b}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_1 = 0, \boldsymbol{x} = \{1, x_2, x_3\} \text{ and } \boldsymbol{n}_1 = \{1, 0, 0\} \tag{c}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_2 = 0, \boldsymbol{x} = \{-1, x_2, x_3\} \text{ and } \boldsymbol{n}_2 = \{-1, 0, 0\} \tag{d}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_3 = 0, \boldsymbol{x} = \{x_1, 1, x_3\} \text{ and } \boldsymbol{n}_3 = \{0, 1, 0\} \tag{e}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_4 = 0, \boldsymbol{x} = \{x_1, -1, x_3\} \text{ and } \boldsymbol{n}_4 = \{0, -1, 0\} \tag{f}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_5 = 0, \boldsymbol{x} = \{x_1, x_2, 0.1\} \text{ and } \sqrt{x_1^2 + x_2^2} \leq 0.7, \text{ and } \boldsymbol{n}_5 = \{0, 0, 1\} \tag{g}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_6 = 0, \boldsymbol{x} = \{x_1, x_2, -0.1\} \text{ and } \sqrt{x_1^2 + x_2^2} \leq 0.7, \text{ and } \boldsymbol{n}_6 = \{0, 0, -1\} \tag{h}$$

$$\nabla u(\boldsymbol{x}, t) \cdot \boldsymbol{n}_7 = 0, \sqrt{x_1^2 + x_2^2} = 0.7 \text{ and } -0.1 \leq x_3 \leq 0.1, \text{ and } \boldsymbol{n}_7 = \left\{\frac{x_1}{0.7}, \frac{x_2}{0.7}, 0\right\} \tag{i}$$

Relations (66)(a) and (b) are the top and bottom Dirichlet BCs, (c) to (f) are zero flux conditions on cube sides, and (g), (h), and (i) are the zero-flux BCs on the cutout surfaces (see Fig. 10). For this example, the cutout radius is 0.7, and $\delta = 0.1$. Similar to the 1D example in Section 5.1, in order to impose local BCs, we use the fictitious domain and the formulas in Appendix A. For the external surfaces, we extend the cube by δ in all six directions and define volume constraints on these extensions. For the top and bottom (with conditions (66)(g) and (h)) surfaces of the cutout, the extension ends up being the actual cutout volume, of thickness 2δ. Fig. 10 shows the original and the domain extended by the constrained volume: $\Omega \cup \Gamma$.

Since the extended domain ($\Omega \cup \Gamma$) is already a cube, similar to the 1D example in Section 5.1, we consider the periodic box $\mathbb{T}$ to be $\Omega \cup \Gamma$ without any gap ($\Lambda = \varnothing$).

Note that here, we do not enforce the BC (i) in the PD model, because, given the non-convex characteristics of the geometry, that would overlap regions of fictitious nodes associated with BCs (g), (h), and (i). That would create difficulties in the Fourier space because different points in the domain would now be linked to values at the same fictitious node. The reason for choosing a cutout of thickness 2δ was to avoid this complication for the domain extension corresponding to the top and bottom surfaces of the cutout. Note that this issue does not appear when using a different method to enforce the local boundary conditions (see [53–55]). Here, for simplicity we opted for the fictitious nodes method. As we shall see, the error introduced by not enforcing condition (66)(i) is relatively small. We emphasize that this issue is only present when enforcing a local boundary condition in a PD formulation; it does not exist for volume constraints (nonlocal boundary conditions).

For discretization we first use $N_1 = N_2 = N_3 = 2^7$, resulting in m-factor of 5.8 and $N = 2^{21}$ as the total number of nodes. Time step is $\Delta t = 5.5 \times 10^{-5}$ to satisfy the stability condition in Eq. (46). The general algorithm for FCBM-EC for 3D diffusion is provided in Appendix C, and the Matlab code is found in the Supplementary materials.

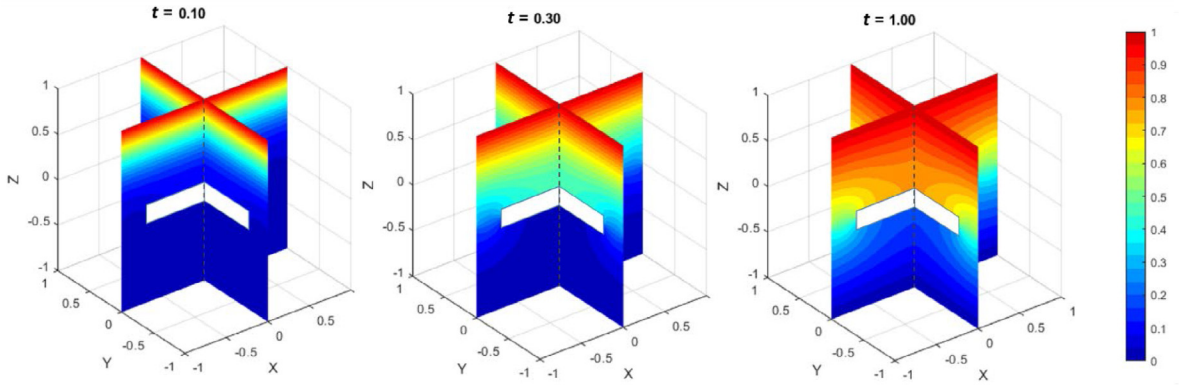


Fig. 11. Peridynamic diffusion simulation results using the FCBM-EC in a cube with an insulated cutout. Temperature/concentration distribution is plotted for two perpendicular cross-sections through the cubic domain.

Fig. 11 shows the simulation results in three snapshots at $t = 0.1$, 0.3 and 1. Two perpendicular cross-sectional views of the 3D domain are plotted to make the insulated discontinuity (cutout) visible in the figure.

From the temperature/concentration distribution we observe that the BCs are enforced and the insulated cutout acts as an obstacle to the diffusive transport process. We compare our FCBM-EC solution for the peridynamic problem with a Finite Element solution for the corresponding classical boundary value diffusion problem. For the finite element analysis, we used *Abaqus/Standard 6.14-2* (note: implicit solver, compared with our explicit solver) with over 10^7 DC3D4 type elements and more than 1.8×10^6 nodes, which is about the same number of degrees of freedom used in the FCBM-EC simulation.

As observed from Fig. 12, the FCBM nonlocal solution is very close to the FE local solution with Abaqus. Although contour levels are set to be the same for two top legends, the contour colors are slightly different since the corresponding colors from Abaqus graphics are slightly different from those in *Tecplot* used in plotting the results obtained with the FCBM-EC. The bottom row in Fig. 12, shows the absolute relative error distribution, along with its mean value. Relative error here is computed as $\frac{\left|u_{ij,t}^{\text{FCBM}}-u_{ij,t}^{\text{Abaqus}}\right|}{\left\|u_{ij,0}^{\text{Abaqus}}\right\|_{\infty}}$ on the displayed cross-sectional plane.

We note that there is a small discrepancy between the two solutions near the sides of the rectangular obstacle. This is due to not enforcing the exact insulated condition on the short lateral sides of the cutout (see earlier discussion). For the Abaqus solution, all BCs in Eq. (66) are enforced.

In order to demonstrate the remarkable efficiency of FCBM-EC, we solved the same example via the conventional meshfree method with one-point Gaussian quadrature, and compared the computational times for these methods for various spatial resolutions. Both methods are implemented in MATLAB R2014b. Simulations are performed on a supercomputer machine in Holland Computing Center of the University of Nebraska-Lincoln, with Intel Xeon E5-2670 2.60 GHz CPUs, up to 512 GB RAM per CPU and a Tesla V100 GPU with 32 GB memory. Matlab's built-in parallel FFT solvers use multithreading and GPU computations (see the 3D code in Supplementary materials). We make use of this in our computational tests. Table 1 shows the computational time for the 3D example above using the meshfree with Gaussian quadrature on a single CPU versus using the FCBM-EC on a single CPU, on 8 or 16 CPUs, and on a GPU, for four spatial discretizations. The time steps are identical for all simulations since the stability condition is independent of the discretization size (it only depends on ν, μ and δ, which are the same for all tests).

As observed the diffusion example with over 1 billion nodes is solved in a few days via the FCBM-EC, while the same computation is intractable with the meshfree method using one-point Gaussian quadrature.

Note that the computational time for the meshfree method in the cases with $m = 11.6$, 23.3, and 46.5 ($N = 2^{24}, 2^{27}$, and 2^{30}) is estimated knowing that it scales as $O\left(N^2\right)$. Although the computational time for the meshfree method in the case with $m = 11.6$ ($N = 2^{24}$) would be reasonable (several days), the available memory however, was not sufficient to perform this test. The FCBM-EC, in contrast, did not run out of memory even for 1 billion nodes. The reason is that peridynamic meshfree solvers store family nodes for each node at the beginning to

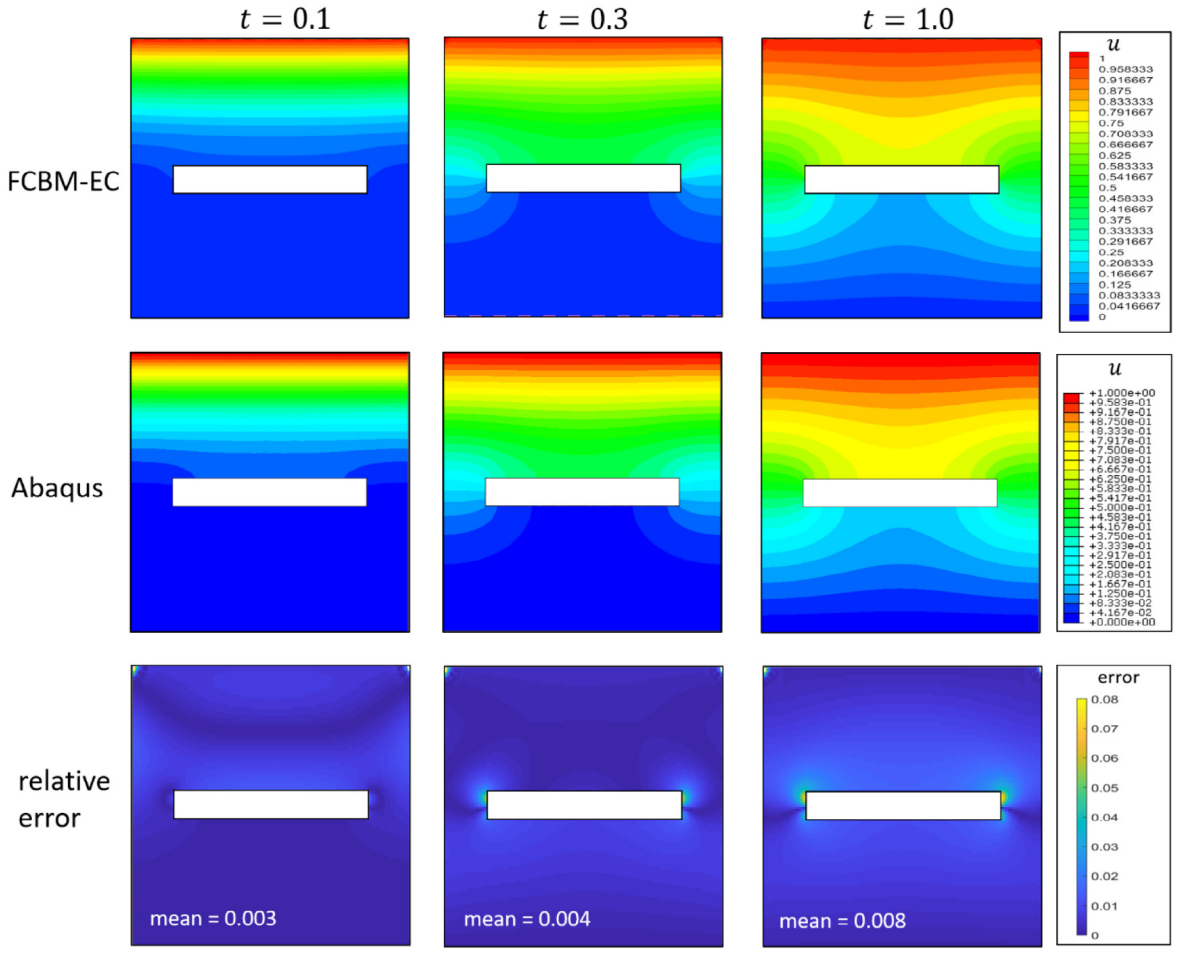


Fig. 12. Comparison of the 3D FCBM-EC solution in a cross-section, against Abaqus solution for the corresponding 3D transient boundary value diffusion problem.

Table 1
Computational time for solving the 3D example up to $t = 1$, using the FCBM-EC and the Gaussian quadrature (GQ)-based meshfree method.

m-factor	5.8	11.6	23.3	46.5
Number of nodes	$2^{21} = 2{,}097{,}152$	$2^{24} = 16{,}777{,}216$	$2^{27} = 134{,}217{,}728$	$2^{30} = 1{,}073{,}741{,}824$
Meshfree with GQ (1 CPU)	4.6 h	12 days	2.1 yrs (!)	1.4 centuries (!)
FCBM-EC (1 CPU)	10.7 min	1.9 h	13.7 h	5.1 days
FCBM-EC (8 CPUs)	6 min	36.6 min	4.5 h	1.4 days
FCBM-EC (16 CPUs)	2.7 min	34.7 min	3.9 h	1.3 days
FCBM-EC (GPU)	29 min	38.3 min	2.1 h	out of memory

increase efficiency when doing the Gaussian quadrature for the convolution integral. For the FCBM-EC, however, there is no need to store family nodes: the convolution integrals are replaced by multiplication of the Fourier coefficients, which do not require the family nodes for each node. Obviously, the meshfree method can also work without storing nodal families, but the computational time increases substantially when one needs to re-compute them at every time step.

The results in Table 1 show important efficiency gains between the FCBM-EC versus the meshfree discretization. Notice also the correlation of these gains with the m-factor value. The reason for the larger gains at higher m-values

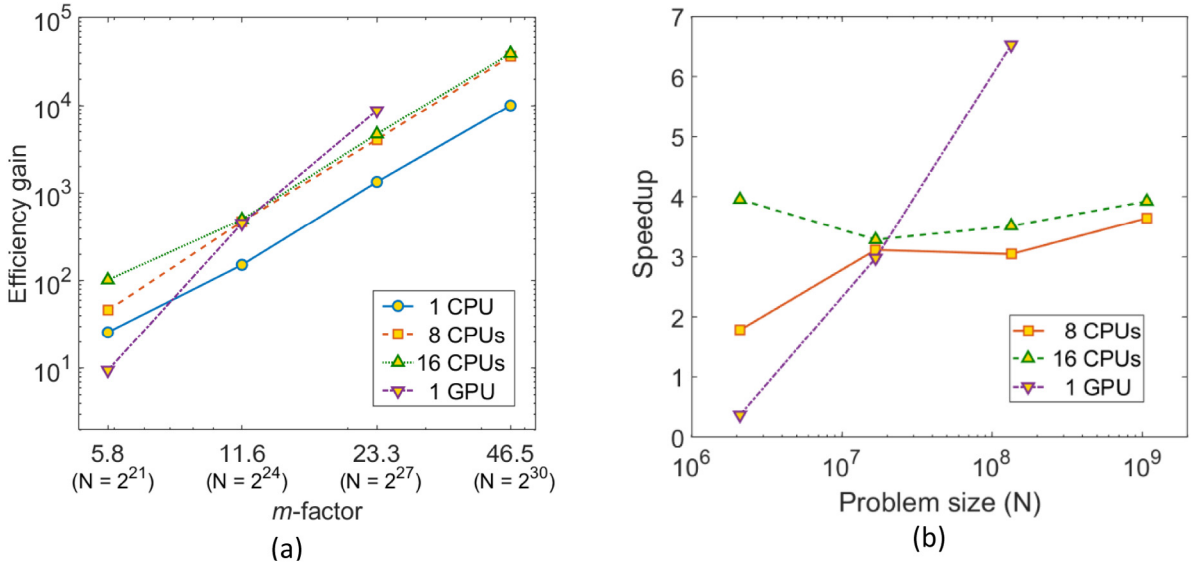


Fig. 13. Efficiency gains relative to the m-factor values for 3D FCBM-EC solution over the meshfree method (Eq. (67)); b) Speedup versus problem size (N) for FCBM-EC parallel computing using Matlab's parallel FFT solvers (from Eq. (68)).

is related to the same point mentioned above: the convolution integrals are replaced by multiplication of the Fourier coefficients, which do not require the family nodes for each node. In other words, one can say that "a nonlocal computation in the physical space gets transformed into a local computation in Fourier space".

Efficiency gains (computed using Eq. (67)) and speedup (using Eq. (68)) are plotted versus the problem size (degrees of freedom) and m-factor values in Fig. 13.

$$\text{Efficiency gain} = \frac{\text{computation time for FCBM}}{\text{computation time for meshfree method with one CPU}} \tag{67}$$

$$\text{Speedup} = \frac{\text{computation time for FCBM with multiple CPUs or GPU}}{\text{computation time for FCBM with one CPU}} \tag{68}$$

These computational experiments (via Matlab's parallel FFT functions) show that little is gained from using more than 8 CPUs with multi-threading. They also show that GPU computations using Matlab's intrinsic functions (see the 3D code in Supplementary materials) is preferred over Matlab's multi-threading for problems with higher spatial resolution (here when the $N > 2^{24} \cong 16.8 \times 10^6$) , as long as GPU computations are feasible from the memory point of view.

The FE and FCM-EC simulations shown in Fig. 12 were performed on the same machine using 16 CPUs, and both have a similar number of degrees of freedom (about 2 million DOFs). The Abaqus simulation took about 2.5 h (implicit solver[1]) to complete while FCM-EC simulation took only 2.7 min. Please note that this comparison is provided here only as a guidance. A general-purpose software like the diffusion solver in Abaqus, is not necessarily optimized for the particular problem setup here. More importantly, we are using an explicit-type solver for the PD results, while Abaqus uses an implicit one. This is not necessarily a disadvantage for Abaqus, since we believe that an implicit solver implemented for our method (currently in the works) will lead to even faster performance. Furthermore, our own implementation for the FCBM-EC, likewise, is not necessarily an optimal one, as we are using Matlab's libraries of functions. Our goal was to write a simple and readable implementation, not the most efficient one (see Appendix C).

One could try comparing Abaqus solution and the FCBM-EC solutions in terms of computational cost for reaching a certain precision. However, this is not easily done here because we are not comparing solutions of the same problem, but of two distinct problems: one nonlocal, one local.

[1] Note that there is no explicit solver in Abaqus for diffusion problems.

6. Conclusions

In this study the *Fast Convolution-based Method with Embedded Constraints* (FCBM-EC) is introduced for peridynamic (PD) linear and nonlinear diffusion problems. This Fourier-based method reduces the computational complexity from $O(N^2)$ in the conventional discretizations of PD problems to $O(N \log 2N)$. The embedded constraint method introduced here allows for imposing *arbitrary* nonlocal boundary conditions (volume constraints) on peridynamic models on irregular domains while maintaining efficiency from using the Fast Fourier Transform (FFT). The arbitrary domain is extended to a periodic box and the volume constraints are "embedded" into the integro-differential PD equation. We verified the new method for linear and some special types of nonlinear diffusion problems with nonperiodic boundary conditions against exact nonlocal solutions (obtained via the manufactured solutions method). Numerical studies for 1D and 2D examples show that the FCBM-EC converges quadratically with respect to the number of nodes used in the spatial discretization.

A 3D example for transient peridynamic diffusion shows that a computation for a problem with a larger number of nodes inside the horizon (an m-factor of about 46) and over 1 billion nodes (computed over 1.8×10^4 time steps), which would be intractable via the commonly used meshfree discretization method for PD, is solved in about a day with the FCBM-EC. The same 3D transient problem, but solved with 2 million nodes, requires 22 h in Abaqus/Standard (implicit solver), but less than 3 min with the new method. The efficiency gains for using FCBM-EC instead of the meshfree solver increase substantially with the number of nodes covered by the nonlocal region. Our numerical tests show further speedup is achieved when using Matlab's FFT functions on GPUs or multiple CPUs.

Declaration of competing interest

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

Acknowledgments

The work of S.J., L.W., and F.B. was supported in part by the National Science Foundation, USA under CMMI CDS&E Award No. 1953346. The research of A.L. was supported by the NSF, USA Grant Nos. 1953346 and 1716801. This work was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

Appendix A. Imposing local Dirichlet and Neumann boundary conditions on peridynamic bodies using a fictitious region method

In order to impose local BC on some generic point $\boldsymbol{x}_b$ on the boundary $\partial\Omega$, we assign volume constraint values $u|_\Gamma$ on $\boldsymbol{x}_\Gamma$ ($\boldsymbol{x} \in \Gamma$), based on the given local BC and the solution values u at $\boldsymbol{x} \in \Omega$ on the interior side of the boundary $\partial\Omega$ (see Fig. A.1)

To impose the local Dirichlet BC:

$$u(\boldsymbol{x}_b, t) = u_b, \tag{A.1}$$

on a generic $\boldsymbol{x}_b \in \partial\Omega_{\mathrm{D}}$ (boundary subjected to Dirichlet BC), we define the following volume constraint on $\boldsymbol{x}_\Gamma$:

$$u|_\Gamma(\boldsymbol{x}_b + d\boldsymbol{n}, t) = 2u_b - u(\boldsymbol{x}_b - d\boldsymbol{n}, t)\,; \text{for } 0 < d \le \delta \tag{A.2}$$

where $\boldsymbol{n}$ is the unit outward normal vector on $\partial\Omega$ at $\boldsymbol{x}_b$ (See Fig. A.1.).

To impose the local Neumann BC:

$$\nabla u(\boldsymbol{x}_b, t) \cdot \boldsymbol{n} = q_b, \tag{A.3}$$

on some $\boldsymbol{x}_b \in \partial\Omega_{\mathrm{N}}$ (boundary subjected to Neumann BC), we define the following volume constraints on $\boldsymbol{x}_\Gamma$:

$$u|_\Gamma(\boldsymbol{x}_b + d\boldsymbol{n}, t) = -2q_b d + u(\boldsymbol{x}_b - d\boldsymbol{n}, t)\,; \text{for } 0 < d \le \delta \tag{A.4}$$

When implementing Eqs. (A.2) and (A.4) to find the constraint values on $\boldsymbol{x}_{ijl}|_\Gamma$ nodes after spatial discretization, we should note that the corresponding interior point: $\boldsymbol{x} = \boldsymbol{x}_b - d\boldsymbol{n}$ (see Fig. A.1), may not locate exactly

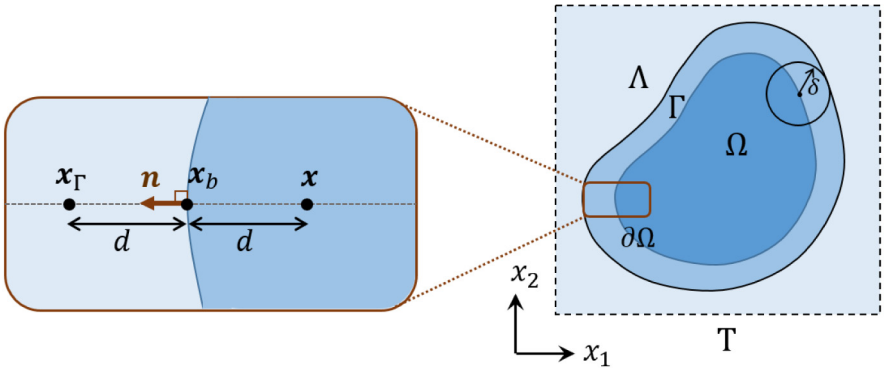


Fig. A.1. Fictitious-node method for imposing local BC on $\partial\Omega$ by assigning certain volume constraints depending on values across the boundary, inside Ω.

on a node. Since u_{ijl} values exist on nodes only in the discretized Ω, one can use an interpolation method to approximate$u\left(\boldsymbol{x}_b - d\boldsymbol{n}, t\right)$ in Eqs. (A.2) and (A.3). For the 1D example in Section 5.1 we used linear interpolation (see 1D codes in Supplementary materials), while for the 3D example in Section 5.3 we used the nearest nodal value (see the 3D code in Supplementary materials) instead of interpolation which is faster but less accurate.

Here we only give the relationships for Dirichlet and Neumann types BCs. One can derive such volume constraints for enforcing other types of local BCs, for example Robin [42], radiation/Sommerfeld [69], absorbing [70], etc.

It should be noted that the implementation of this scheme is not straightforward on domains with curved boundaries or sharp corners, and may not exactly reproduce local BCs at such locations [54].

Appendix B. Fast convolution-based method with volume penalization (FCBM-VP)

FCBM-VP was introduced in [34] for 1D linear diffusion. Here we provide the extension of this formulation to higher dimensions and nonlinear problems.

In order to solve a general volume constrained peridynamic diffusion problem as described by Eq. (35) with the FCBM-VP, we first extend the domain Ω and the constrained volume Γ to a periodic box$\mathbb{T} = \Omega \cup \Gamma \cup \Lambda$ (see Fig. 3). This step is identical to the FCBM-EC introduced in Section 4.2. We then replace the problem description in Eq. (35) by the following periodic diffusion problem extended by a penalization term:

$$\begin{cases} \dfrac{\partial u\left(\boldsymbol{x}, t\right)}{\partial t} = \mathcal{L}_\gamma u\left(\boldsymbol{x}, t\right) + r\left(\boldsymbol{x}, t\right) - \dfrac{1 - \chi\left(\boldsymbol{x}, t\right)}{\varepsilon}\left[u\left(\boldsymbol{x}, t\right) - w\left(\boldsymbol{x}, t\right)\right] & \boldsymbol{x} \in \mathbb{T},\ t > 0 \\ u\left(\boldsymbol{x}, 0\right) = u_0 & \boldsymbol{x} \in \mathbb{T} \end{cases} \tag{B.1}$$

where $\chi\left(\boldsymbol{x}, t\right)$ and $w\left(\boldsymbol{x}, t\right)$ are defined in Eq. (38) and (39) respectively, and $0 < \varepsilon \ll 1$ is the *penalization parameter*. The penalization term, which is zero on Ω but large on $\Gamma \cup \Lambda$, forces $u\left(\boldsymbol{x}, t\right)$ to smoothly get close to $w\left(\boldsymbol{x}, t\right)$ on $\Gamma \cup \Lambda$ [34].

A 3D discretization as described by Eq. (21) to (27) leads to:

$$\begin{cases} \dfrac{\mathrm{d}u^N_{ijl,t}}{\mathrm{d}t} = \mathcal{L}_\gamma u\big|_{ijl,t} + r_{ijl,t} - \dfrac{1 - \chi_{ijl}}{\varepsilon}\left(u^N_{ijl,t} - w_{ijl,t}\right) & \boldsymbol{x} \in \mathbb{T},\ t > 0 \\ u_{ijl,t=0} = u_{0,ijl} & \boldsymbol{x} \in \mathbb{T} \end{cases} \tag{B.2}$$

where χ_{ijl} and $w_{ijl,t}$ are given in Eq. (41) and (42), and $\mathcal{L}_\gamma u\big|_{ijl,t}$ is efficiently computed with the FFT and inverse FFT, for all linear and some nonlinear cases (see Sections 2 and 3). The system of ODEs in Eq. (B.2) is then solved with an ODE solver. For example, with the Forward Euler method we have:

$$u^N_{ijl,t+\Delta t} = u^N_{ijl,t} + \Delta t\left[\mathcal{L}_\gamma u\big|_{ijl,t} + r_{ijl,t} - \frac{1 - \chi_{ijl}}{\varepsilon}\left(u^N_{ijl,t} - w_{ijl,t}\right)\right] \tag{B.3}$$

One important difference between the FCBM-VP and FCBM-EC, is the stability condition for the explicit time integration. In FCBM-VP, stability is restricted by ε, whereas the FCBM-EC has not such restriction.

Appendix C. Algorithm for the FCBM-EC method for 3D problems

A Matlab implementation of the peridynamic FCBM-EC for 3D linear transient diffusion with local boundary conditions, similar to the 3D example in Section 5.3 is as follows:

Given the problem and the inclusive periodic box $\mathbb{T} = \Omega \cup \Gamma \cup \Lambda = L_1 \times L_2 \times L_3$:

- Inputs:
  - Physical parameters: ν, δ, $r(\boldsymbol{x}, t)$, $\mu(\boldsymbol{x})$, β, L_1, L_2, L_3, $t_{\max}$
  - Initial condition: $u_0 = u(\boldsymbol{x}, t = 0)$
  - Boundary conditions: $u_{\mathrm{b}} = u(\cdot, t)|_{\partial\Omega_{\mathrm{D}}}$, $q_{\mathrm{b}} = \nabla u(\cdot, t) \cdot \boldsymbol{n}|_{\partial\Omega_{\mathrm{N}}}$
  - Number of nodes for each dimension: N_1, N_2, N_3 (powers of 2 are preferred)
  - Time step: $\Delta t \left(\leq \frac{1}{\nu\beta}\right)$
- Initialization:
  - Calculate grid spacing: $\Delta x_1 = \frac{L_1}{N_1}$, $\Delta x_2 = \frac{L_2}{N_2}$, $\Delta x_3 = \frac{L_3}{N_3}$
  - Discretize the box T: $\boldsymbol{x}_{ijl} = \{x_i, x_j, x_l\}$ where
    $x_i = x_1^0 + (i-1)\Delta x_1$ and $i = 1, 2, \ldots, N_1$
    $x_j = x_2^0 + (j-1)\Delta x_2$ and $j = 1, 2, \ldots, N_2$
    $x_l = x_3^0 + (l-1)\Delta x_3$ and $l = 1, 2, \ldots, N_3$
  - Discretize the kernel function $\mu_{ijl} = \mu(\boldsymbol{x}_{ijl})$
  - $\mu_{ijl}\big|_{\boldsymbol{x}_{ijl}=0} = 0$ (for singular kernels only)
  - Shift the discretized periodic kernel function (if either $x_1^0, x_2^0, x_3^0 \neq 0$): $\mu^s_{ijl} = \mu\left(x_i - x_1^0, x_j - x_2^0, x_l - x_3^0\right)$
  - Discretize the initial condition and the source term: $u^0_{ijl} = u_0\left(\boldsymbol{x}_{ijl}, 0\right)$; $r^0_{ijl} = r(\boldsymbol{x}_{ijl}, 0)$;
  - Fast Fourier transform μ^s_{ijl} and u^0_{ijl}: $\widetilde{\mu^s}_{k_1k_2k_3} = \mathbf{FFT}\left(\mu^s_{ijl}\right)$ and $\widetilde{u^0}_{k_1k_2k_3} = \mathbf{FFT}\left(u^0_{ijl}\right)$
  - Construct the discretized mask function: $\chi_{ijl} = \begin{cases} 1 & \boldsymbol{x}_{ijl} \in \Omega \\ 0 & \boldsymbol{x}_{ijl} \in \mathbb{T}\backslash\Omega = \Gamma \cup \Lambda \end{cases}$
  - Construct the variable containing volume constraints and gap values (w):
    Allocate $w_{ijl} = 0$
    Calculate and assign volume constraints on Γ based on given BC and Appendix A: $w_{ijl}\big|_{\Gamma,t=0}$
    Calculate and assign values on the gap (if $\Lambda \neq \varnothing$): $w_{ijl}\big|_{\Lambda,t=0}$
  - Initialize step counter: $n = 0$
  - Initialize time: $t^n = 0$
- Solve the transient diffusion: while $t^n < t_{\max}$
  - Update time: $t^{n+1} = t^n + \Delta t$
  - Update solution: $u^{n+1}_{ijl} = \chi_{ijl}\left(u^n_{ijl} + \Delta t\left\{\nu\mathbf{FFT}^{-1}\left[\Delta x_3\Delta x_2\Delta x_1\tilde{\mu}_{k_1k_2k_3}\mathbf{FFT}\left(u^n_{ijl}\right)\right] - \nu\beta^N u^n_{ijl} + r_{ijl}\right\}\right) + \left(1 - \chi_{ijl}\right) w^n_{ijl}$
  - Update the source term: $r^{n+1}_{ijl} = r_{ijl}\left(\boldsymbol{x}_{ijl}, t^{n+1}\right)$
  - Re-apply volume constraints and the gap values by updating w^{n+1}_{ijl}
  - Fast Fourier transform u^{n+1}_{ijl}: $\widetilde{u^{n+1}}_{k_1k_2k_3} = \mathbf{FFT}\left(y^{n+1}_{ijl}\right)$
  - Update step counter: $n = n + 1$

Note that the DFT formulation that governs Matlab FFT solvers is based on domains with zeros as the lower bound coordinates, i.e. $\mathbb{T} = [0, L_1) \times [0, L_2) \times [0, L_3)$. As a result, in order to maintain consistency with FFT solvers,

if the lower bound of the problem domain's coordinates are non-zero, the kernel function needs to be shifted as shown with μ^s_{ijl} in the algorithm above (see also [34]).

For nonlinear problems with kernels in the form of Eq. (11), $\mu(\boldsymbol{x})$ is replaced with $\omega(\boldsymbol{x})$ in the algorithm and the evolution equation is modified according to the kernel specifics. For example, see Eq. (34).

Appendix D. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cma.2020.113633.

References

[1] A. Mogilner, L. Edelstein-Keshet, A non-local model for a swarm, J. Math. Biol. 38 (1999) 534–570.
[2] F. Cucker, S. Smale, Emergent behavior in flocks, IEEE Trans. Automat. Control 52 (2007) 852–862.
[3] J.A. Carrillo, M. Fornasier, J. Rosado, G. Toscani, Asymptotic flocking dynamics for the kinetic Cucker–Smale model, SIAM J. Math. Anal. 42 (2010) 218–236.
[4] R. Shvydkoy, E. Tadmor, Eulerian dynamics with a commutator forcing III. fractional diffusion of order $0 < \alpha < 1$, Physica D 376 (2018) 131–137.
[5] R.M. Colombo, M. Garavello, M. Lécureux-Mercier, A class of nonlocal models for pedestrian traffic, Math. Models Methods Appl. Sci. 22 (2012) 1150023.
[6] S. Gourley, J.-H. So, J. Wu, Nonlocality of reaction–diffusion equations induced by delay: biological modeling and nonlinear dynamics, J. Math. Sci. 124 (2004) 5119–5153.
[7] F. Bobaru, M. Duangpanya, A peridynamic formulation for transient heat conduction in bodies with evolving discontinuities, J. Comput. Phys. 231 (2012) 2764–2785.
[8] Z. Chen, F. Bobaru, Peridynamic modeling of pitting corrosion damage, J. Mech. Phys. Solids 78 (2015) 352–381.
[9] S. Jafarzadeh, Z. Chen, F. Bobaru, Computational modeling of pitting corrosion, Corros. Rev. 37 (2019) 419–439.
[10] S. Jafarzadeh, Z. Chen, S. Li, F. Bobaru, A peridynamic mechano-chemical damage model for stress-assisted corrosion, Electrochim. Acta 323 (2019) 134795.
[11] 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.
[12] S.A. Silling, R. Lehoucq, Peridynamic theory of solid mechanics, in: Advances in Applied Mechanics, Elsevier, 2010, pp. 73–168.
[13] F. Bobaru, J.T. Foster, P.H. Geubelle, S.A. Silling, Handbook of Peridynamic Modeling, CRC Press, 2016.
[14] M. Zaccariotto, T. Mudric, D. Tomasi, A. Shojaei, U. Galvanetto, Coupling of FEM meshes with Peridynamic grids, Comput. Methods Appl. Mech. Engrg. 330 (2018) 471–497.
[15] G. Zhang, G.A. Gazonas, F. Bobaru, Supershear damage propagation and sub-Rayleigh crack growth from edge-on impact: A peridynamic analysis, Int. J. Impact Eng. 113 (2018) 73–87.
[16] S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics, Comput. Struct. 83 (2005) 1526–1535.
[17] F. Bobaru, G. Zhang, Why do cracks branch? A peridynamic investigation of dynamic brittle fracture, Int. J. Fract. 196 (2015) 59–98.
[18] X. Chen, M. Gunzburger, Continuous and discontinuous finite element methods for a peridynamics model of mechanics, Comput. Methods Appl. Mech. Engrg. 200 (2011) 1237–1250.
[19] X. Tian, Q. Du, Asymptotically compatible schemes and applications to robust discretization of nonlocal models, SIAM J. Numer. Anal. 52 (2014) 1641–1665.
[20] X. Tian, Q. Du, Nonconforming discontinuous Galerkin methods for nonlocal variational problems, SIAM J. Numer. Anal. 53 (2015) 762–781.
[21] B. Ren, C. Wu, E. Askari, A 3D discontinuous Galerkin finite element method with the bond-based peridynamics model for dynamic brittle failure analysis, Int. J. Impact Eng. 99 (2017) 14–25.
[22] J. Mehrmashhadi, M. Bahadori, F. Bobaru, Comparison of peridynamic and phase-field models for dynamic brittle fracture in glassy materials, 2020, http://dx.doi.org/10.31224/osf.io/4zc69, engrXiv Preprints.
[23] J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex fourier series, Math. Comput. 19 (1965) 297–301.
[24] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2005) 216–231.
[25] Q. Du, J. Yang, Asymptotically compatible Fourier spectral approximations of nonlocal Allen–Cahn equations, SIAM J. Numer. Anal. 54 (2016) 1899–1919.
[26] A. Bueno-Orovio, D. Kay, K. Burrage, Fourier spectral methods for fractional-in-space reaction–diffusion equations, BIT Numer. Math. 54 (2014) 937–954.
[27] Q. Du, J. Yang, Fast and accurate implementation of Fourier spectral approximations of nonlocal diffusion operators and its applications, J. Comput. Phys. 332 (2017) 118–134.
[28] G.M. Coclite, A. Fanizzi, L. Lopez, F. Maddalena, S.F. Pellegrino, Numerical methods for the nonlocal wave equation of the peridynamics, Appl. Numer. Math. 155 (2020) 119–139.
[29] R.M. Slevinsky, H. Montanelli, Q. Du, A spectral method for nonlocal diffusion operators on the sphere, J. Comput. Phys. 372 (2018) 893–911.
[30] B. Alali, N. Albin, Fourier spectral methods for nonlocal models, 2019, arXiv preprint arXiv:1907.11998.
[31] B. Alali, N. Albin, Fourier multipliers for nonlocal Laplace operators, Appl. Anal. (2019) 1–21.

[32] H. Wang, H. Tian, A fast Galerkin method with efficient matrix assembly and storage for a peridynamic model, J. Comput. Phys. 231 (2012) 7730–7738.
[33] H. Wang, H. Tian, A fast and faithful collocation method with efficient matrix assembly for a two-dimensional nonlocal diffusion model, Comput. Methods Appl. Mech. Engrg. 273 (2014) 19–36.
[34] S. Jafarzadeh, A. Larios, F. Bobaru, Efficient solutions for nonlocal diffusion problems via boundary-adapted spectral methods, J. Peridyn. Nonlocal Model. 2 (2020) 85–110.
[35] N.K.-R. Kevlahan, J.-M. Ghidaglia, Computation of turbulent flow past an array of cylinders using a spectral method with Brinkman penalization, Eur. J. Mech. B Fluids 20 (2001) 333–350.
[36] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, J. Mech. Phys. Solids 48 (2000) 175–209.
[37] Y.D. Ha, F. Bobaru, Characteristics of dynamic brittle fracture captured with peridynamics, Eng. Fract. Mech. 78 (2011) 1156–1168.
[38] Z. Chen, S. Niazi, F. Bobaru, A peridynamic model for brittle damage and fracture in porous materials, Int. J. Rock Mech. Min. Sci. 122 (2019) 104059.
[39] J. Mehrmashhadi, L. Wang, F. Bobaru, Uncovering the dynamic fracture behavior of PMMA with peridynamics: The importance of softening at the crack tip, Eng. Fract. Mech. 219 (2019) 106617.
[40] M. Behzadinasab, J.T. Foster, A semi-Lagrangian constitutive correspondence framework for peridynamics, J. Mech. Phys. Solids (2020) 103862.
[41] F. Bobaru, M. Duangpanya, The peridynamic formulation for transient heat conduction, Int. J. Heat Mass Transfer 53 (2010) 4047–4059.
[42] 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.
[43] E. Madenci, S. Oterkus, Peridynamics for coupled field equations, in: F. Bobaru, J. Foster, P. Geubelle, S. Silling (Eds.), Handbook of Peridynamic Modeling, CRC Press, 2016.
[44] E. Lejeune, C. Linder, Modeling tumor growth with peridynamics, Biomech. Model. Mechanobiol. 16 (2017) 1141–1157.
[45] P. Radu, K. Wells, A doubly nonlocal laplace operator and its connection to the classical Laplacian, J. Integral Equations Appl. 31 (2019) 379–409.
[46] S. Jafarzadeh, F. Mousavi, A. Larios, F. Bobaru, A fast convolution-based method for peridynamic models of deformation and fracture, 2020, in preparation.
[47] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods, Springer, 2006.
[48] D.A. Kopriva, Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers, Springer Science & Business Media, 2009.
[49] M.C. Jeruchim, P. Balaban, K.S. Shanmugan, Simulation of Communication Systems: Modeling, Methodology and Techniques, Springer Science & Business Media, 2006.
[50] J. Cooley, P. Lewis, P. Welch, Application of the fast Fourier transform to computation of Fourier integrals, Fourier series, and convolution integrals, IEEE Trans. Audio Electroacoust. 15 (1967) 79–84.
[51] Q. Du, M. Gunzburger, R.B. Lehoucq, K. Zhou, A nonlocal vector calculus, nonlocal volume-constrained problems, and nonlocal balance laws, Math. Models Methods Appl. Sci. 23 (2013) 493–540.
[52] S. Oterkus, E. Madenci, A. Agwai, Peridynamic thermal diffusion, J. Comput. Phys. 265 (2014) 71–96.
[53] B. Aksoylu, F. Celiker, O. Kilicer, Nonlocal operators with local boundary conditions: An overview, in: Handbook of Nonlocal Continuum Mechanics for Materials and Structures, 2018, pp. 1–38.
[54] Q. Le, F. Bobaru, Surface corrections for peridynamic models in elasticity and fracture, Comput. Mech. 61 (2018) 499–518.
[55] M. D'Elia, X. Tian, Y. Yu, A physically-consistent, flexible and efficient strategy to convert local boundary conditions into nonlocal volume constraints, 2019), arXiv preprint arXiv:1906.04259.
[56] A. Averbuch, M. Israeli, L. Vozovoi, A fast Poisson solver of arbitrary order accuracy in rectangular regions, SIAM J. Sci. Comput. 19 (1998) 933–952.
[57] A. Bueno-Orovio, V.M. Pérez-García, F.H. Fenton, Spectral methods for partial differential equations in irregular domains: the spectral smoothed boundary method, SIAM J. Sci. Comput. 28 (2006) 886–900.
[58] H.-C. Yu, H.-Y. Chen, K. Thornton, Extended smoothed boundary method for solving partial differential equations with general boundary conditions on complex boundaries, Modelling Simulation Mater. Sci. Eng. 20 (2012) 075008.
[59] D. Fortunato, A. Townsend, Fast Poisson solvers for spectral methods, IMA J. Numer. Anal. 40 (2020) 1994–2018.
[60] A. Buades, B. Coll, J.-M. Morel, Image denoising methods. A new nonlocal principle, SIAM Rev. 52 (2010) 113–147.
[61] L. Pizarro, P. Mrázek, S. Didas, S. Grewenig, J. Weickert, Generalised nonlocal image smoothing, Int. J. Comput. Vis. 90 (2010) 62–87.
[62] J.V. Burke, T. Hoheisel, C. Kanzow, Gradient consistency for integral-convolution smoothing functions, Set-Valued Var. Anal. 21 (2013) 359–376.
[63] P.J. Roache, The method of manufactured solutions for code verification, in: Computer Simulation Validation, Springer, 2019, pp. 295–318.
[64] F. Bobaru, M. Yang, L.F. Alves, S.A. Silling, E. Askari, J. Xu, Convergence, adaptive refinement and scaling in 1D peridynamics, Internat. J. Numer. Methods Engrg. 77 (2009) 852–877.
[65] Z. Chen, F. Bobaru, Selecting the kernel in a peridynamic formulation: A study for transient heat diffusion, Comput. Phys. Comm. 197 (2015) 51–60.
[66] D. Dipasquale, G. Sarego, M. Zaccariotto, U. Galvanetto, Dependence of crack paths on the orientation of regular 2D peridynamic grids, Eng. Fract. Mech. 160 (2016) 248–263.
[67] F. Bobaru, J. Mehrmashhadi, Z. Chen, S. Niazi, Intraply fracture in fiber-reinforced composites: a peridynamic analysis, in: ASC 33rd Annual Technical Conference & 18th US-Japan Conference on Composite Materials, Seattle, 2018, p. 9.

[68] J. Zhao, Z. Chen, J. Mehrmashhadi, F. Bobaru, Construction of a peridynamic model for transient advection-diffusion problems, Int. J. Heat Mass Transfer 126 (2018) 1253–1266.

[69] A. Shojaei, U. Galvanetto, T. Rabczuk, A. Jenabi, M. Zaccariotto, A generalized finite difference method based on the Peridynamic differential operator for the solution of problems in bounded and unbounded domains, Comput. Methods Appl. Mech. Engrg. 343 (2019) 100–126.

[70] A. Shojaei, F. Mossaiby, M. Zaccariotto, U. Galvanetto, A local collocation method to construct Dirichlet-type absorbing boundary conditions for transient scalar wave propagation problems, Comput. Methods Appl. Mech. Engrg. 356 (2019) 629–651.