Available online at www.sciencedirect.com
ScienceDirect
Comput. Methods Appl. Mech. Engrg. 392 (2022) 114666
Computer methods in applied mechanics and engineering
www.elsevier.com/locate/cma

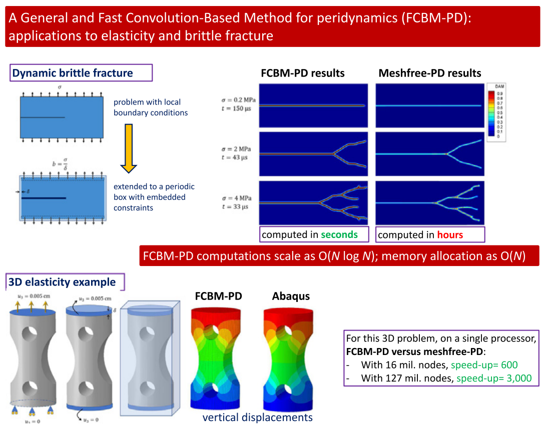
A general and fast convolution-based method for peridynamics: Applications to elasticity and brittle fracture

Siavash Jafarzadeh[a], Farzaneh Mousavi[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 29 May 2021; received in revised form 17 January 2022; accepted 22 January 2022
Available online 17 February 2022

Graphical Abstract


A General and Fast Convolution-Based Method for peridynamics (FCBM-PD): applications to elasticity and brittle fracture
Dynamic brittle fracture
FCBM-PD results
Meshfree-PD results
problem with local boundary conditions
extended to a periodic box with embedded constraints
computed in seconds
computed in hours
FCBM-PD computations scale as O(N log N); memory allocation as O(N)
3D elasticity example
FCBM-PD
Abaqus
vertical displacements

For this 3D problem, on a single processor, **FCBM-PD versus meshfree-PD**:
- With 16 mil. nodes, speed-up= 600
- With 127 mil. nodes, speed-up= 3,000

Abstract

A general and fast convolution-based method (FCBM) for peridynamics (PD) is introduced. Expressing the PD integrals in terms of convolutions and computing them by Fast Fourier Transform (FFT), the computational complexity of PD models drops from O(N^2) to O($N \log_2 N$), with N being the number of discretization nodes. Initial neighbor identification and storing neighbor information is not required, and this means memory allocation scales with O(N) instead of O(N^2), common for existing methods. FCBM is applicable to bounded domains with arbitrary shapes and boundary conditions via an "embedded constraint" (EC) approach. The formulation is shown for certain bond-based and state-based, linear and nonlinear, PD models of elasticity and dynamic brittle fracture, as applications. The method is verified on a 3D elastostatic problem and it is shown that the FCBM-PD reduces the computational time from days to hours and from years to days, compared with the original meshfree discretization for PD models. Large-scale computations of PD models are feasible with the new method,

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

https://doi.org/10.1016/j.cma.2022.114666

and its versatility is demonstrated by simulating, with ease, the difficult problem of cascading crack branching in a brittle plate.

Keywords: Peridynamics; Fast Fourier Transform; Convolution integrals; Elasticity; Dynamic fracture; Crack branching

1. Introduction

Peridynamics (PD) is a nonlocal extension of classical continuum mechanics used with great effect in modeling of fracture and damage [1,2]. In contrast with classical (local) models described by partial differential equations (PDEs), governing equations in PD are integro-differential equations. In PD, nonlocal interactions are considered between points within a certain distance, replacing the usual spatial derivatives with integral operators. Because of this, the usual requirements of continuity and smoothness for displacements are eliminated, allowing one to model the emergence and evolution of discontinuities, such as cracks and damage, as natural parts of the solution to the governing equations [3,4].

However, when usual discretization methods are used, nonlocality increases the cost for a PD model, relative to the cost of numerically approximating local models. The meshfree method with one-point Gaussian quadrature [5] (from now on, in this study we refer to it as "meshfree PD") and finite element methods (FEM) [6–9] have been used to compute numerical solutions to PD equations. For such equations, for a fixed finite range of nonlocal interaction (the horizon size), these methods scale as (N^2), where N is the total number of nodes in the discretization. For large scale problems in 3D, the cost is prohibitive, even on massively parallel computers.

Various attempts have been made to reduce the cost of peridynamic simulations. Coupling the local theory with PD is one approach that uses a local model for parts of the domain, and uses the PD model only at locations near cracks/damage as necessary [10,11]. This approach does not work well for problems in which damage/cracks are (or become) widely distributed throughout the domain, such as in problems like impact fragmentation, etc. [12,13].

The natural *convolutional structure* of PD formulations can be exploited using Fourier transforms as recently shown in [14,15]. Using Fast Fourier Transform (FFT) algorithms, the new scaling for PD computations drops to $O(N \log_2 N)$ [16,17]. These methods, however, are restricted to periodic domains. Fourier spectral methods have been used for nonlocal Allen–Cahn equation [18], fractional-in space reaction diffusion [19], and peridynamic diffusion and wave operators [14,20–22], all in periodic domains. Another class of $O(N \log_2 N)$ methods was also introduced for 1D and 2D problems in [23–25], but these are still restricted to simple geometries and certain horizon shapes. Additionally, extension of these methods to 3D is not straightforward [23–25].

In [26,27], we have introduced a fast convolution-based method (FCBM) for peridynamic diffusion problems. Two different techniques have been proposed in order to adapt the approach to problem sets on arbitrary domains and with general boundary conditions: the *volume penalization (VP)* [26] and the *embedded constraint (EC)* [27] techniques. The method has been validated in 1D, 2D, and 3D against exact nonlocal (manufactured) solutions and against local FEM-based solutions for problems in bounded domains with Dirichlet and Neumann boundary conditions [26,27]. A diffusion problem in a 3D domain with an insulated cutout with billions of degrees of freedom was solved, over tens of thousands of time steps, in a matter of hours on a single processor. The same problem, if solved using the meshfree PD method, would have required years [27].

In this study, we extend the FCBM to the PD equation of motion with examples on elastic deformations and fracture. We describe the class of constitutive models that have convolutional structures and can benefit from the remarkable efficiency of the method. Moreover, we demonstrate how to setup nonlinear problems like dynamic fracture in a brittle material as a convolution structure amenable to the FCBM treatment. This work also aims to serve as guide to demonstrate how one can construct material models with convolutional structures, and how to discretize different types of PD governing equation with this new method. As numerical examples, we formulate the FCBM-EC for linearized bond-based and linearized state-based PD isotropic solids, a nonlinear rotation-invariant model example, and a dynamic fracture problem. For brevity, the FCBM for PD correspondence models is presented in a separate study [28]. We validate the proposed damage model by comparing our simulation results against the published fracture PD simulations that use a popular critical bond strain criterion [5] for bond failure and the meshfree PD method for discretization.

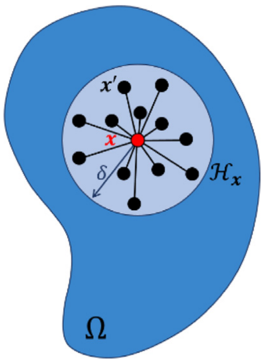


Fig. 1. A generic point $\boldsymbol{x}$, its horizon region $\mathcal{H}_x$, and its family points $\boldsymbol{x}'$ in a peridynamic body Ω .

This article is organized as follows: first a brief review of peridynamics equations of motion is provided, as well as concepts and notations related to PD constitutive modeling. Next, we discuss the convolutional from of PD material models through several examples. We describe the Fourier-based method for solving PD problems in periodic domain first, and then its extension to general bounded domains with arbitrary volume constraints/boundary conditions, using the EC method. We demonstrate the efficiency and validity of the introduced method by a 3D static and a 2D dynamic brittle fracture problems.

2. Peridynamics

We briefly describe the peridynamic formulation for elasticity and brittle fracture. Peridynamics is a nonlocal extension of continuum mechanics that unifies the governing equations for continuous media, media with discontinuities, and discrete media [1]. In this theory, each spatial point can interact with other points within its neighborhood up to a finite size distance. The shape of this neighborhood is arbitrary. In applications, however, for simplicity, for a point denoted by its position vector $\boldsymbol{x}$, the finite size neighborhood is taken as a sphere in 3D (a disk in 2D, or a segment in 1D) centered at $\boldsymbol{x}$ of radius δ [2]. This neighborhood, denoted by $\mathcal{H}_x$, is referred to as the *horizon region of* $\boldsymbol{x}$ and δ is called the *horizon size*. Spatial points located inside $\mathcal{H}_x$ are referred to as the *family* of $\boldsymbol{x}$ and are denoted by $\boldsymbol{x}'$. The *bond vector* for each family pair is defined as $\boldsymbol{\xi} = \boldsymbol{x}' - \boldsymbol{x}$. Fig. 1 shows the schematic of a peridynamic body and the horizon of a generic point in the body.

In this work, we employ the following notation to distinguish between different types of quantities: plain (non-bold) italic letters denote scalars, boldface italic letters denote vectors, and boldface non-italic letters denote tensors. Peridynamic states, discussed in Section 2.2.2, are denoted by underlined letters.

2.1. Peridynamic equation of motion

In continuum mechanics, the equation of motion can be expressed as:

$$\rho \frac{\partial^2 \boldsymbol{u}(\boldsymbol{x}, t)}{\partial t^2} = \boldsymbol{L}(\boldsymbol{x}, t) + \boldsymbol{b}(\boldsymbol{x}, t) \tag{1}$$

where ρ is mass density, $\boldsymbol{u}(\boldsymbol{x}, t)$ is the displacement field at point $\boldsymbol{x}$ and time t, $\boldsymbol{L}(\boldsymbol{x}, t)$ is the *internal force density*, and $\boldsymbol{b}$ is the *external body force density*. In the classical (local) theory, $\boldsymbol{L} = \boldsymbol{L}^{\mathbf{C}}(\boldsymbol{x}, t) = \nabla \cdot \boldsymbol{\sigma}$ (the divergence of the stress tensor) and the stress tensor dependency on strains/displacements is defined via a constitutive relationship. The superscript C denotes the classical definition of $\boldsymbol{L}$.

In peridynamics, $\boldsymbol{L}(\boldsymbol{x}, t)$ is expressed with a nonlocal term instead: an integral which sums up all of the pair-wise forces between $\boldsymbol{x}$ and its family points $\boldsymbol{x}'$:

$$\boldsymbol{L}^{\mathbf{PD}}(\boldsymbol{x}, t) = \int_{\mathcal{H}_x} \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right) \mathrm{d}V_{\boldsymbol{x}'} \tag{2}$$

The function $\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right)$ is called the *dual force density* and describes the net force between the unit volume at $\boldsymbol{x}$ and the unit volume at $\boldsymbol{x}'$ [1]. The specific form of $\boldsymbol{f}$ is determined by the chosen peridynamic constitutive model.

The superscript PD denotes the peridynamics definition of $\boldsymbol{L}$. In the rest of the manuscript, we drop the superscript PD and use $\boldsymbol{L}$ to denote the PD internal force density.

As noted in the introduction, contrary to the local formulation, the PD operator $\boldsymbol{L}(\boldsymbol{x}, t)$ has no continuity/smoothness requirements on the displacement field, allowing for cracks (as discontinuities in the displacement field) to naturally emerge and propagate in peridynamic solutions.

2.2. Peridynamic elastic and brittle fracture constitutive models in bond-based and state-based formulations

The original PD model was a "bond-based" model ([3]), which was later generalized to the state-based formulation of PD ([29]). Next, we review these formulations for describing elasticity and brittle failure.

2.2.1. Bond-based models

In bond-based materials, the dual force density between two family points depends *only* on deformation of their bond, and is independent of the deformation of other bonds in the same family. For example, in an elastic bond-based solid, dual force density depends only on the displacements of the family pair, i.e. $\boldsymbol{f} = \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{u}, \boldsymbol{u}', t\right)$ where $\boldsymbol{u}' = \boldsymbol{u}(\boldsymbol{x}', t)$. Bond-based models suffer from certain restrictions on the material responses they produce. For example, Poisson ratio in bond-based isotropic elastic solids is restricted to a fixed number. To overcome such limitations, PD theory was extended to the state-based formulation by introducing the notion of *peridynamic states* [29].

2.2.2. State-based models

Peridynamics states are nonlinear (in general) operators, defined at a point and time, that map bonds into scalars or vectors, resulting in scalar-states or vector-states, respectively. A vector state is a generalization of the concept of a second-order tensor, which are linear mappings. Here, scalar-states are denoted by underlined italic lowercase letters, and vector-states are denoted by underlined boldface uppercase letters. The square bracket, $[\cdot]$, in front of a state encloses the point and the time at which the state is defined. The angle bracket $\langle\cdot\rangle$, if used, encloses the bond on which the state is operating. A state with the angle brackets in front of it then refers to the particular value associated with the enclosed bond. Here are some of the frequently used states in PD models:

$\underline{\mathbf{X}}[\boldsymbol{x}]$, the *identity state* at point $\boldsymbol{x}$ which returns back the bond vector for any given bond:

$$\underline{\mathbf{X}}[\boldsymbol{x}]\langle\boldsymbol{\xi}\rangle = \boldsymbol{\xi} \tag{3}$$

$\underline{\mathbf{Y}}[\boldsymbol{x}, t]$, the *deformation state* and returns the deformed bond vector at time t:

$$\underline{\mathbf{Y}}[\boldsymbol{x}, t]\langle\boldsymbol{\xi}\rangle = \left(\boldsymbol{x}' + \boldsymbol{u}'\right) - (\boldsymbol{x} + \boldsymbol{u}) = \boldsymbol{\xi} + \boldsymbol{\eta} \tag{4}$$

where $\boldsymbol{\eta} = \boldsymbol{u}' - \boldsymbol{u}$ is the relative displacement.

The *extension state* ($\underline{e}$) is a scalar state defining bond elongation:

$$\underline{e} = \left|\underline{\mathbf{Y}}\right| - \left|\underline{\mathbf{X}}\right|, \textit{i.e.}\ \underline{e}[\boldsymbol{x}, t]\langle\boldsymbol{\xi}\rangle = |\boldsymbol{\xi} + \boldsymbol{\eta}| - |\boldsymbol{\xi}| \tag{5}$$

The *identity scalar-state* ($\underline{x}$) and the *deformation scalar-state* ($\underline{y}$) are used to define the length of the bonds in reference and deformed configurations, respectively:

$$\underline{x} = \left|\underline{\mathbf{X}}\right|, \text{ and } \underline{y} = \left|\underline{\mathbf{Y}}\right|, \textit{i.e.}\ \underline{x}[\boldsymbol{x}]\langle\boldsymbol{\xi}\rangle = |\boldsymbol{\xi}|, \text{ and } \underline{y}[\boldsymbol{x}, t]\langle\boldsymbol{\xi}\rangle = |\boldsymbol{\xi} + \boldsymbol{\eta}| \tag{6}$$

$\underline{\mathbf{T}}[\boldsymbol{x}, t]$ is the *force state* and returns the *bond force* $\boldsymbol{t}$ for a given bond:

$$\underline{\mathbf{T}}[\boldsymbol{x}, t]\langle\boldsymbol{\xi}\rangle = \boldsymbol{t} \tag{7}$$

where $\boldsymbol{t}$ is the force vector that the unit volume at $\boldsymbol{x}'$ exerts on a unit volume at $\boldsymbol{x}$.

The dual force density $\boldsymbol{f}$ in state-based PD is defined as follows:

$$\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right) = \underline{\mathbf{T}}[\boldsymbol{x}, t]\langle\boldsymbol{\xi}\rangle - \underline{\mathbf{T}}\left[\boldsymbol{x}', t\right]\langle-\boldsymbol{\xi}\rangle \tag{8}$$

According to Eq. (8), dual force density is equal to the force density that the unit volume at $\boldsymbol{x}'$ exerts on $\boldsymbol{x}$, minus the force density that unit volume at $\boldsymbol{x}$ exerts on $\boldsymbol{x}'$, i.e. it is the net force between the two unit volumes.

In classical continuum mechanics, constitutive models are tensorial equations, connecting a stress tensor to the strain tensor or the deformation gradient. In state-based PD, constitutive models similarly define the relationships between force states and deformation states. For example, an elastic state-based constitutive model is defined by $\underline{\mathbf{T}} = \underline{\mathbf{T}}\left(\underline{\mathbf{Y}}\right)$, meaning that the force state is a function of the deformation state only. In state-based PD, the force density of a bond depends on the deformation of all other bonds inside the horizons of its two end-nodes [30]. Bond-based PD can be viewed as a special case of the state-based theory.

A frequent mathematical operation on PD states is the *inner product of the states*. Let $\underline{a}$ and $\underline{b}$ to denote two generic scalar states, and $\underline{\mathbf{A}}$ and $\underline{\mathbf{B}}$ to be two generic vector states. The inner product for scalar and vector states then are defined as follows [1]:

$$\underline{a} \bullet \underline{b} = \int_{\mathcal{H}_x} \underline{a}\left\langle \boldsymbol{\xi} \right\rangle \underline{b}\left\langle \boldsymbol{\xi} \right\rangle \mathrm{d}V_{\boldsymbol{x}'}, \qquad \underline{\mathbf{A}} \bullet \underline{\mathbf{B}} = \int_{\mathcal{H}_x} \underline{\mathbf{A}}\left\langle \boldsymbol{\xi} \right\rangle \cdot \underline{\mathbf{B}}\left\langle \boldsymbol{\xi} \right\rangle \mathrm{d}V_{\boldsymbol{x}'} \tag{9}$$

where ($\bullet$) denotes the inner product operation on PD states, and ($\cdot$) denotes the dot product of two vectors.

2.2.3. Modeling of damage and fracture in peridynamics

One common way to describe fracture and damage in a peridynamic model is using the "bond breaking" approach [5]. In this approach, a history dependent binary scalar function μ is inserted in the PD constitutive model:

$$\boldsymbol{L}\left(\boldsymbol{x}, t\right) = \int_{\mathcal{H}_x} \mu\left(\boldsymbol{x}, \boldsymbol{x}', t\right) \boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right) \mathrm{d}V_{\boldsymbol{x}'}, \tag{10}$$

where

$$\mu\left(\boldsymbol{x}, \boldsymbol{x}', t\right) = \begin{cases} 1 & \left(\boldsymbol{x}' - \boldsymbol{x}\right) \text{bond is intact at time } t \\ 0 & \left(\boldsymbol{x}' - \boldsymbol{x}\right) \text{bond broken at time } t \end{cases}. \tag{11}$$

Being "broken" or "intact" is determined by a certain failure criterion. The *critical bond strain* is one of such criterion [2,5], which sets μ to 0 if the bond strain exceeds a critical value. Such critical values can be defined in several different ways [31]. In these models, cracks emerge as outcomes of cascading bond breaking events under loading. One way to monitor the evolution of damage in these models is the *damage index*, defined at a point:

$$d\left(\boldsymbol{x}, t\right) = 1 - \frac{\int_{\mathcal{H}_x} \mu\left(\boldsymbol{x}, \boldsymbol{x}', t\right) \mathrm{d}V_{\boldsymbol{x}'}}{\int_{\mathcal{H}_x} \mathrm{d}V_{\boldsymbol{x}'}} \tag{12}$$

which counts the "number" of broken bonds relative to the total number of bonds for that point at a given time instant. Note that the damage index is not the "definition" of damage in PD, it is merely one way to represent damage in PD. Damage in these bond-breaking PD models is determined by the individual bond failure events, which is a much richer quantity than, for example, the scalar or even tensor variables defined in continuum damage mechanics (see [32]). In some sense, PD damage is a mapping, not necessarily continuous, from the vector space of bonds at a point $\boldsymbol{x}$ to a vector space of dimension equal to the number of bonds at a node. In its discretized version, PD damage can be therefore considered a vector state. The damage index in Eq. (12) is a scalar representation that does not carry the complete information of PD damage. For example, consider these two cases: (1) a point has lost 50% of its bonds uniformly and symmetrically in all directions; and (2) a point has lost 50% of its bonds, but all in one side. In both cases $d = 0.5$, but in the first case the solid is uniformly degraded (distributed damage), and the second case is a split in solid (a through crack).

3. Obtaining convolutional structures for peridynamic models

Many linearized peridynamic models, featuring integral operators, can naturally be expressed via convolutions. Certain nonlinear models are also shown to have a convolutional structure [27,33]. We observe that in general, the following form of a possibly nonlinear PD integrand (e.g. the dual force density in equation of motion):

$$f\left(\boldsymbol{x}, \boldsymbol{x}', t\right) = \sum_{n=1}^{p} a_n\left(\boldsymbol{x}, t\right) b_n\left(\boldsymbol{x}', t\right) c_n\left(\boldsymbol{x} - \boldsymbol{x}', t\right) \tag{13}$$

where a_n, b_n and c_n are functions, and p is arbitrary positive integer, leads to a PD model that possesses a convolutional structure. Indeed, we can express:

$$\begin{aligned}\int_{\mathcal{H}_x} f\left(\boldsymbol{x}, \boldsymbol{x}', t\right)\, dV_{\boldsymbol{x}'} &= \int_{\mathcal{H}_x}\left[\sum_{n=1}^{p} a_n\left(\boldsymbol{x}, t\right) b_n\left(\boldsymbol{x}', t\right) c_n\left(\boldsymbol{x}-\boldsymbol{x}', t\right)\right] dV_{\boldsymbol{x}'} \\ &= \sum_{n=1}^{p} a_n\left(\boldsymbol{x}, t\right) \int_{\mathcal{H}_x} c_n\left(\boldsymbol{x}-\boldsymbol{x}', t\right) b_n\left(\boldsymbol{x}', t\right)\, dV_{\boldsymbol{x}'} = \sum_{n=1}^{p} a_n\left(\boldsymbol{x}, t\right)\left[b_n * c_n\right]\left(\boldsymbol{x}, t\right)\end{aligned} \tag{14}$$

The operator $(*)$ denotes the convolution integral.

This observation allows for the development of efficient numerical techniques that utilize Fourier transforms. In this section we discuss setting up bond-based and state-based PD models in convolutional structures, for linear elastic materials. Moreover, we show how to obtain a convolutional structure for important nonlinear elastic models, and even for fracture problems. Obtaining the convolutional form for PD correspondence models, with applications to plasticity, is discussed in [28]. For the fracture case, we introduce a new damage model as an alternative to models that use the critical bond strain criterion for describing bond failure and does not have a convolutional structure.

3.1. Linear elastic bond-based PD model

The force density for an elastic bond-based PD material can be expressed in the form [3]:

$$\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right) = f\left(\left|\boldsymbol{\xi}+\boldsymbol{\eta}\right|, \boldsymbol{\xi}\right) \frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{\left|\boldsymbol{\xi}+\boldsymbol{\eta}\right|} \tag{15}$$

Consider the well-known homogeneous isotropic elastic bond-based material [2] as an example:

$$\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right) = \alpha\omega\left(\left|\boldsymbol{\xi}\right|\right)\left(\left|\boldsymbol{\xi}+\boldsymbol{\eta}\right| - \left|\boldsymbol{\xi}\right|\right) \frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{\left|\boldsymbol{\xi}+\boldsymbol{\eta}\right|} \tag{16}$$

where α is a scalar and ω is a radially symmetric function of the bond, called the *influence function,* and is zero outside of the horizon: $\omega = 0$ for $|\boldsymbol{\xi}| > \delta$. Although this model is linear in terms of the bond extension, it is nonlinear in terms of displacements. The model in Eq. (16) can be linearized in terms of displacement if needed. According to the linearization carried out in [34], one can rewrite Eq. (16) for small displacements as:

$$\boldsymbol{f}\left(\boldsymbol{x}, \boldsymbol{x}', t\right) = \alpha\omega\left(\left|\boldsymbol{\xi}\right|\right) \frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{\left|\boldsymbol{\xi}\right|^2}\boldsymbol{\eta} = \mathbf{C}\left(\boldsymbol{\xi}\right)\boldsymbol{\eta} \tag{17}$$

where $(\otimes)$ denotes the tensor product operator, and $\mathbf{C}$ is a tensor-valued symmetric function of the bond.

To obtain the convolutional form in Eq. (14), we opt to proceed with the indicial notation (including Einstein summation convention) for expressing vector and tensor quantities.

We substitute Eq. (17) into Eq. (2) to obtain the convolutional form for this model:

$$\begin{aligned} L_i\left(\boldsymbol{x}, t\right) &= \int_{\mathcal{H}_x} f_i\left(\boldsymbol{x}, \boldsymbol{x}', t\right) \mathrm{d}V_{\boldsymbol{x}'} = \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\left[u_j\left(\boldsymbol{x}', t\right) - u_j\left(\boldsymbol{x}, t\right)\right] \mathrm{d}V_{\boldsymbol{x}'} \\ &= \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right) u_j\left(\boldsymbol{x}', t\right) \mathrm{d}V_{\boldsymbol{x}'} - \left[\int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right) \mathrm{d}V_{\boldsymbol{x}'}\right] u_j\left(\boldsymbol{x}, t\right) \\ &= \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}-\boldsymbol{x}'\right) u_j\left(\boldsymbol{x}', t\right) \mathrm{d}V_{\boldsymbol{x}'} - \mathrm{P}_{ij} u_j\left(\boldsymbol{x}, t\right) \\ &= \left[\mathrm{C}_{ij} * u_j\right]\left(\boldsymbol{x}, t\right) - \mathrm{P}_{ij} u_j\left(\boldsymbol{x}, t\right) \end{aligned} \tag{18}$$

for $i, j = 1, 2, 3$ in 3D. Note that Eq. (18) is consistent with the general form in Eq. (14), because one can write $\mathrm{P}_{ij} = \mathrm{C}_{ij} * 1$. In this model, since we investigate homogeneous materials, $\mathbf{C}$ is only a function of $\boldsymbol{\xi}$ and is independent of $\boldsymbol{x}$. As a result $\mathbf{P}$ is a constant tensor. From a computational point of view, $\mathbf{P}$ can be computed once and stored as a preprocessing step. The linear term $\mathrm{P}_{ij} u_j\left(\boldsymbol{x}, t\right)$ is then just a multiplication at each node which is much cheaper than computing a volume integral.

This linearized model in Eq. (17) is based on the "small displacements" assumption, and therefore, cannot be used for large displacements, including rigid body rotations. In general, the nonlinear model in Eq. (16) is used more frequently [35–37] because it is valid for large rotations due to the geometrical nonlinearity of the model. We could not obtain the convolutional form for the nonlinear model in Eq. (16) directly. However, in the next section we introduce an alternative nonlinear bond-based model which has a convolutional structure, and is invariant under arbitrary rigid body rotations.

3.2. Nonlinear elastic bond-based PD model

Here, we introduce a particular nonlinear elastic bond-based model and show how to express it using multiple convolutions. Consider the bond-based constitutive model:

$$\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{x}',t\right)=\alpha\omega\left(|\boldsymbol{\xi}|\right)|\boldsymbol{\xi}|\left(s+\frac{3}{2}s^2+\frac{1}{2}s^3\right)\frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|} \tag{19}$$

where

$$s=\frac{|\boldsymbol{\xi}+\boldsymbol{\eta}|-|\boldsymbol{\xi}|}{|\boldsymbol{\xi}|} \tag{20}$$

is the *bond strain*. This nonlinear model is invariant to rigid body motions and approximates the model in Eq. (16) for $s \ll 1$. Therefore, it can be used for linear elastic materials undergoing small strains and arbitrarily large rotations. Examples for such situations include brittle or quasi brittle fracture of slender beams undergoing deformations that feature large rotations but small strains.

To express $\boldsymbol{L}\left(\boldsymbol{x},t\right)$ in its convolutional form, we first simplify $\boldsymbol{f}$ in terms of displacements and bond vectors:

$$\begin{aligned}\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{x}',t\right)&=\alpha\omega\left(|\boldsymbol{\xi}|\right)|\boldsymbol{\xi}|\left(s+\frac{3}{2}s^2+\frac{1}{2}s^3\right)\frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|}=\\&=\frac{\alpha\omega\left(|\boldsymbol{\xi}|\right)|\boldsymbol{\xi}|}{2}\left[(s+1)^2-1\right](s+1)\frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|}\\&=\frac{\alpha\omega\left(|\boldsymbol{\xi}|\right)|\boldsymbol{\xi}|}{2}\left[\left(\frac{|\boldsymbol{\xi}+\boldsymbol{\eta}|}{|\boldsymbol{\xi}|}\right)^2-1\right]\frac{|\boldsymbol{\xi}+\boldsymbol{\eta}|}{|\boldsymbol{\xi}|}\frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|}\\&=\frac{\alpha\omega\left(|\boldsymbol{\xi}|\right)}{2\,|\boldsymbol{\xi}|^2}\left(|\boldsymbol{\xi}+\boldsymbol{\eta}|^2-|\boldsymbol{\xi}|^2\right)(\boldsymbol{\xi}+\boldsymbol{\eta})=\frac{\alpha\omega\left(|\boldsymbol{\xi}|\right)}{2\,|\boldsymbol{\xi}|^2}\left(2\boldsymbol{\xi}\cdot\boldsymbol{\eta}+\boldsymbol{\eta}\cdot\boldsymbol{\eta}\right)(\boldsymbol{\xi}+\boldsymbol{\eta})\end{aligned} \tag{21}$$

Again, we use the indicial notation for $\boldsymbol{f}$ in Eq. (21):

$$\begin{aligned}f_i\left(\boldsymbol{x},\boldsymbol{x}',t\right)&=\frac{\alpha\omega\left(\sqrt{\xi_p\xi_p}\right)}{2\xi_q\xi_q}\left(2\xi_j\eta_j+\eta_j\eta_j\right)(\xi_i+\eta_i)\\&=\frac{\alpha\omega\left(\sqrt{\xi_p\xi_p}\right)}{2\xi_q\xi_q}\left(2\xi_ju'_j-2\xi_ju_j+u'_ju'_j-2u'_ju_j+u_ju_j\right)\left(\xi_i+u'_i-u_i\right)\\&=\frac{\alpha\omega\left(\sqrt{\xi_p\xi_p}\right)}{2\xi_q\xi_q}\left(2\xi_i\xi_ju'_j-2\xi_i\xi_ju_j+\xi_iu'_ju'_j-2\xi_iu'_ju_j+\xi_iu_ju_j\right.\\&+2\xi_ju'_ju'_i-2\xi_ju_ju'_i+u'_ju'_ju'_i-2u'_ju_ju'_i+u_ju_ju'_i-2\xi_ju'_ju_i\\&\left.+2\xi_ju_ju_i-u'_ju'_ju_i+2u'_ju_ju_i-u_ju_ju_i\right)\end{aligned} \tag{22}$$

Substituting $\xi_i=-\left(x_i-x'_i\right)$, one can write $\boldsymbol{L}\left(\boldsymbol{x},t\right)$ as:

$$\begin{aligned}L_i=\int_{\mathcal{H}_x}f_i\mathrm{d}V_{\boldsymbol{x}'}=\int_{\mathcal{H}_x}&\frac{\alpha\omega\left(\sqrt{(x_p-x'_p)(x_p-x'_p)}\right)}{2(x_q-x'_q)(x_q-x'_q)}\left[2\left(x_i-x'_i\right)\left(x_j-x'_j\right)u'_j\right.\\&-2\left(x_i-x'_i\right)\left(x_j-x'_j\right)u_j-\left(x_i-x'_i\right)u'_ju'_j+2\left(x_i-x'_i\right)u'_ju_j\\&-\left(x_i-x'_i\right)u_ju_j-2\left(x_j-x'_j\right)u'_ju'_i+2\left(x_j-x'_j\right)u_ju'_i+2\left(x_j-x'_j\right)u'_ju_i\\&-2\left(x_j-x'_j\right)u_ju_i+u'_ju'_ju'_i-2u'_ju_ju'_i+u_ju_ju'_i-u'_ju'_ju_i+2u'_ju_ju_i\\&\left.-u_ju_ju_i\right]\mathrm{d}V_{\boldsymbol{x}'}\end{aligned} \tag{23}$$

with $i, j, p, q = 1, 2, 3$ in 3D. Let the tensor $\mathbf{C}$, the vector $\boldsymbol{a}$, and the scalar c denote the following functions of $\boldsymbol{\xi}$:

$$\mathbf{C}(\boldsymbol{\xi}) = \frac{\alpha\omega(|\boldsymbol{\xi}|)}{|\boldsymbol{\xi}|^2}\boldsymbol{\xi}\otimes\boldsymbol{\xi}, \qquad \boldsymbol{a}(\boldsymbol{\xi}) = \frac{\alpha\omega(|\boldsymbol{\xi}|)}{|\boldsymbol{\xi}|^2}\boldsymbol{\xi}, \qquad c(\boldsymbol{\xi}) = \frac{\alpha\omega(|\boldsymbol{\xi}|)}{|\boldsymbol{\xi}|^2} \tag{24}$$

and let $\mathbf{P}$, $\boldsymbol{q}$, and p to be the integrals of $\mathbf{C}$, $\boldsymbol{a}$, and c over the horizon respectively. Since $\mathbf{C}$, $\boldsymbol{a}$, and c are functions of $\boldsymbol{\xi}$ and independent of $\boldsymbol{x}$, their integrals can be computed at any $\boldsymbol{x}$. We can then compute the integrals once at $\boldsymbol{x} = 0$ for example, where $\boldsymbol{\xi} = \boldsymbol{x}'$:

$$\mathbf{P} = \int_{\mathcal{H}_{x=0}} \mathbf{C}(\boldsymbol{x}')\,\mathrm{d}V_{\boldsymbol{x}'}, \qquad \boldsymbol{q} = \int_{\mathcal{H}_{x=0}} \boldsymbol{a}(\boldsymbol{x}')\,\mathrm{d}V_{\boldsymbol{x}'}, \qquad p = \int_{\mathcal{H}_{x=0}} c(\boldsymbol{x}')\,\mathrm{d}V_{\boldsymbol{x}'} \tag{25}$$

Using the definitions in Eqs. (24) and (25), Eq. (23) can be reorganized as follows:

$$\begin{aligned} L_i &= \mathrm{C}_{ij} * u_j - \left(\int_{\mathcal{H}_x} \mathrm{C}_{ij}\mathrm{d}V_{\boldsymbol{x}'}\right) u_j - a_i * (u_j u_j) + (a_i * u_j)\,u_j - \frac{1}{2}\left(\int_{\mathcal{H}_x} a_i \mathrm{d}V_{\boldsymbol{x}'}\right) u_j u_j - a_j \\ &\quad * (u_j u_i) + u_j (a_j * u_i) + (a_j * u_j)\,u_i - \left(\int_{\mathcal{H}_x} a_i \mathrm{d}V_{\boldsymbol{x}'}\right) u_i u_j + \frac{1}{2} c * (u_j u_j u_i) \\ &\quad - \left[c * (u_i u_j)\right] u_j + \frac{1}{2}(c * u_i)\,u_j u_j - \frac{1}{2}\left[c * (u_j u_j)\right] u_i + (c * u_j)\,u_j u_i \\ &\quad - \frac{1}{2} u_j u_j u_i \left(\int_{\mathcal{H}_x} c\mathrm{d}V_{\boldsymbol{x}'}\right) \\ &= \mathrm{C}_{ij} * u_j - \mathrm{P}_{ij} u_j - a_i * (u_j u_j) + (a_i * u_j)\,u_j - \frac{1}{2} q_i u_j u_j - a_j * (u_j u_i) \\ &\quad + u_j (a_j * u_i) + (a_j * u_j)\,u_i - q_i u_i u_j + \frac{1}{2} c * (u_j u_j u_i) - \left[c * (u_i u_j)\right] u_j \\ &\quad + \frac{1}{2}(c * u_i)\,u_j u_j - \frac{1}{2}\left[c * (u_j u_j)\right] u_i + (c * u_j)\,u_j u_i - \frac{1}{2} u_j u_j u_i p \end{aligned} \tag{26}$$

Eq. (26) shows the convolutional form of the $\boldsymbol{L}$ integral for the nonlinear model (rigid body rotation invariant) introduced in Eq. (19).

3.3. State-based linear elastic and isotropic PD solid

In this example we discuss a well-known state-based linear elastic model, known as the *linear isotropic peridynamic solid* [34]. The force state for this material is given by:

$$\underline{\mathbf{T}} = \left(\frac{3\kappa - 5\mu}{m}\right)\underline{\omega}\,\underline{x}\,\vartheta\,\underline{\mathbf{M}} + \frac{15\mu}{m}\underline{\omega}\,\underline{e}\,\underline{\mathbf{M}} \tag{27}$$

where κ and μ are the *bulk* and the *shear moduli*, ϑ is the *nonlocal dilatation* defined at each point by:

$$\vartheta(\boldsymbol{x}, t) = \frac{3}{m}\underline{\omega}\,\underline{x} \bullet \underline{e} = \frac{3}{m}\int_{\mathcal{H}_x} \underline{\omega}\langle\boldsymbol{\xi}\rangle\,\underline{x}\langle\boldsymbol{\xi}\rangle\,\underline{e}\langle\boldsymbol{\xi}\rangle\,\mathrm{d}V_{\boldsymbol{x}'}, \tag{28}$$

m is a normalization factor:

$$m(\boldsymbol{x}, t) = \underline{\omega}\,\underline{x} \bullet \underline{x} = \int_{\mathcal{H}_x} \underline{\omega}\langle\boldsymbol{\xi}\rangle\,\underline{x}\langle\boldsymbol{\xi}\rangle\,\underline{x}\langle\boldsymbol{\xi}\rangle\,\mathrm{d}V_{\boldsymbol{x}'}, \tag{29}$$

$\underline{\omega}$ is the influence function scalar state:

$$\underline{\omega}\langle\boldsymbol{\xi}\rangle = \omega(|\boldsymbol{\xi}|), \tag{30}$$

$\underline{\mathbf{M}}$ is a unit vector state, giving the direction of the deformed bond:

$$\underline{\mathbf{M}}\langle\boldsymbol{\xi}\rangle = \frac{\underline{\mathbf{Y}}\langle\boldsymbol{\xi}\rangle}{|\underline{\mathbf{Y}}\langle\boldsymbol{\xi}\rangle|} = \frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|}, \tag{31}$$

and $\underline{e}$ is the extension state defined in Eq. (5). Note that we can express Eq. (27) in terms of $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ as follows:

$$\underline{\mathbf{T}}\langle\boldsymbol{\xi}\rangle = \left(\frac{3\kappa - 5\mu}{m}\right)\omega(|\boldsymbol{\xi}|)\,|\boldsymbol{\xi}|\,\vartheta\,\frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|} + \frac{15\mu}{m}\omega(|\boldsymbol{\xi}|)\,(|\boldsymbol{\xi}+\boldsymbol{\eta}| - |\boldsymbol{\xi}|)\,\frac{\boldsymbol{\xi}+\boldsymbol{\eta}}{|\boldsymbol{\xi}+\boldsymbol{\eta}|}. \tag{32}$$

The linearized version of this material for small displacements is [34]:

$$\underline{\mathbf{T}}\langle\boldsymbol{\xi}\rangle=\left(\frac{3\kappa-5\mu}{m}\right)\omega(|\boldsymbol{\xi}|)\,\vartheta\boldsymbol{\xi}+\frac{15\mu}{m}\omega(|\boldsymbol{\xi}|)\frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{|\boldsymbol{\xi}|^2}\boldsymbol{\eta} \tag{33}$$

with

$$\vartheta=\frac{3}{m}\int_{\mathcal{H}_x}\omega(|\boldsymbol{\xi}|)\,\boldsymbol{\xi}\cdot\boldsymbol{\eta}\,\mathrm{d}V_{\boldsymbol{x}'} \tag{34}$$

and

$$m=\int_{\mathcal{H}_x}\omega(|\boldsymbol{\xi}|)\,|\boldsymbol{\xi}|^2\,\mathrm{d}V_{\boldsymbol{x}'} \tag{35}$$

Based on this constitutive model, the internal force density is:

$$\begin{aligned}\boldsymbol{L}(\boldsymbol{x},t)&=\int_{\mathcal{H}_x}\boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{x}',t\right)\mathrm{d}V_{\boldsymbol{x}'}=\int_{\mathcal{H}_x}\left\{\underline{\mathbf{T}}[\boldsymbol{x},t]\langle\boldsymbol{\xi}\rangle-\underline{\mathbf{T}}\left[\boldsymbol{x}',t\right]\langle-\boldsymbol{\xi}\rangle\right\}\mathrm{d}V_{\boldsymbol{x}'}\\&=\int_{\mathcal{H}_x}\left\{\left(\frac{3\kappa-5\mu}{m}\right)\omega(|\boldsymbol{\xi}|)\,\vartheta\boldsymbol{\xi}+\frac{15\mu}{m}\omega(|\boldsymbol{\xi}|)\frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{|\boldsymbol{\xi}|^2}\boldsymbol{\eta}\right.\\&\left.-\left(\frac{3\kappa-5\mu}{m}\right)\omega(|-\boldsymbol{\xi}|)\,\vartheta'(-\boldsymbol{\xi})+\frac{15\mu}{m}\omega(|-\boldsymbol{\xi}|)\frac{(-\boldsymbol{\xi})\otimes(-\boldsymbol{\xi})}{|-\boldsymbol{\xi}|^2}(-\boldsymbol{\eta})\right\}\mathrm{d}V_{\boldsymbol{x}'}\\&=\int_{\mathcal{H}_x}\left\{\left(\frac{3\kappa-5\mu}{m}\right)\omega(|\boldsymbol{\xi}|)\left(\vartheta+\vartheta'\right)\boldsymbol{\xi}+2\frac{15\mu}{m}\omega(|\boldsymbol{\xi}|)\frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{|\boldsymbol{\xi}|^2}\boldsymbol{\eta}\right\}\mathrm{d}V_{\boldsymbol{x}'}\end{aligned} \tag{36}$$

where $\vartheta'=\vartheta\left(\boldsymbol{x}',t\right)$. Let

$$\mathbf{C}(\boldsymbol{\xi})=\frac{30\mu}{m}\omega(|\boldsymbol{\xi}|)\frac{\boldsymbol{\xi}\otimes\boldsymbol{\xi}}{|\boldsymbol{\xi}|^2},\quad \boldsymbol{a}(\boldsymbol{\xi})=\left(\frac{3\kappa-5\mu}{m}\right)\omega(|\boldsymbol{\xi}|)\,\boldsymbol{\xi}. \tag{37}$$

Replacing $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ in Eq. (36) with $\boldsymbol{x}'-\boldsymbol{x}$ and $\boldsymbol{u}'-\boldsymbol{u}$ respectively, one gets:

$$\begin{aligned}L_i=\int_{\mathcal{H}_x}&a_i\left(\boldsymbol{x}'-\boldsymbol{x}\right)\left(\vartheta+\vartheta'\right)\mathrm{d}V_{\boldsymbol{x}'}+\int_{\mathcal{H}_x}\mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\left(u_j'-u_j\right)\mathrm{d}V_{\boldsymbol{x}'}\\&=\vartheta\int_{\mathcal{H}_x}a_i\left(\boldsymbol{x}'-\boldsymbol{x}\right)\mathrm{d}V_{\boldsymbol{x}'}-\int_{\mathcal{H}_x}a_i\left(\boldsymbol{x}-\boldsymbol{x}'\right)\vartheta'\mathrm{d}V_{\boldsymbol{x}'}+\int_{\mathcal{H}_x}\mathrm{C}_{ij}\left(\boldsymbol{x}-\boldsymbol{x}'\right)u_j'\mathrm{d}V_{\boldsymbol{x}'}\\&+\left(\int_{\mathcal{H}_x}\mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\mathrm{d}V_{\boldsymbol{x}'}\right)u_j\\&=\vartheta\left(\int_{\mathcal{H}_x}a_i\left(\boldsymbol{x}'-\boldsymbol{x}\right)\mathrm{d}V_{\boldsymbol{x}'}\right)-(a_i*\vartheta)+\mathrm{C}_{ij}*u_j\\&+\left(\int_{\mathcal{H}_x}\mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\mathrm{d}V_{\boldsymbol{x}'}\right)u_j\\&=\vartheta\left(\int a_i\mathrm{d}V_{\boldsymbol{x}}\right)-(a_i*\vartheta)+\mathrm{C}_{ij}*u_j+\left(\int\mathrm{C}_{ij}\mathrm{d}V_{\boldsymbol{x}}\right)u_j\\&=\vartheta q_i-(a_i*\vartheta)+\mathrm{C}_{ij}*u_j+\mathrm{P}_{ij}u_j\end{aligned} \tag{38}$$

To use the numerical method presented in this work, all volume integrals involved in the PD model should have convolutional structure, otherwise the FFT cannot be used to compute them and the proposed method's efficiency is lost. Therefore, $\vartheta(\boldsymbol{x},t)$ needs to be written as a convolution as well:

$$\begin{aligned}\vartheta&=\frac{3}{m}\int_{\mathcal{H}_x}\omega(|\boldsymbol{\xi}|)\,\xi_i\eta_i\mathrm{d}V_{\boldsymbol{x}'}=\frac{-3}{m}\int_{\mathcal{H}_x}a_i\left(\boldsymbol{x}-\boldsymbol{x}'\right)\left(u_i'-u_i\right)\mathrm{d}V_{\boldsymbol{x}'}\\&=\frac{-3}{m}\int_{\mathcal{H}_x}a_i\left(\boldsymbol{x}-\boldsymbol{x}'\right)u_i'\mathrm{d}V_{\boldsymbol{x}'}-\frac{3}{m}\left(\int_{\mathcal{H}_x}a_i\left(\boldsymbol{x}'-\boldsymbol{x}\right)\mathrm{d}V_{\boldsymbol{x}'}\right)u_i\\&=\frac{-3}{m}\left[(a_i*u_i)+q_iu_i\right]\end{aligned} \tag{39}$$

Eqs. (38) and (39), provide the convolutional form for the linearized PD state-based elastic model described in Eq. (33).

3.4. Peridynamic correspondence models

PD state-based models can be divided into two types: *native* PD models and PD *correspondence* models. In native models, the constitutive relationship between the force state and the deformation state is directly derived/defined in the nonlocal setting, while PD correspondence models attempt to *directly* import existing classical (local) constitutive models into a PD formulation. While PD correspondence models may seem convenient, there are several important drawbacks of involving deformation gradients (therefore spatial derivatives that native PD models do not use) from the classical constitutive models: (1) instabilities appear due to existence of zero energy modes [38,39]; (2) relatively higher computational cost compared to native models; and (3) damage models loose the generality available in native PD formulations. A number of solutions have been proposed to reduce/eliminate zero energy modes from PD correspondence models. For example see [38,40].

The derivation of the convolutional structure needed for the use of the FCBM in the correspondence models introduced in [29] is presented in [28,41].

3.5. An energy-based failure model that leads to a convolution structure for fracture problems

The failure model in PD based on the critical bond strain does *not* have a convolution structure. To be able to use the fast convolution-based method in fracture problems, we introduce a new damage model based on critical strain energy density at a point.

3.5.1. Damage model based on critical strain energy density at a point

The strain energy density for the model in Eq. (16) is [34]:

$$W(\boldsymbol{x},t)=\frac{1}{2}\int_{\mathcal{H}_x}\alpha\omega(|\boldsymbol{\xi}|)(|\boldsymbol{\xi}+\boldsymbol{\eta}|-|\boldsymbol{\xi}|)^2\,\mathrm{d}V_{\boldsymbol{x}'} \tag{40}$$

and for the linearized model in Eq. (17) is [42]:

$$W(\boldsymbol{x},t)=\frac{1}{2}\int_{\mathcal{H}_x}\boldsymbol{\eta}\cdot[\mathbf{C}(|\boldsymbol{\xi}|)\,\boldsymbol{\eta}]\,\mathrm{d}V_{\boldsymbol{x}'} \tag{41}$$

The formula for W depends, obviously, on the particular constitutive model chosen. We define a critical strain energy density W_c at a point by calibrating it to Griffith's critical fracture energy for brittle fracture (see Section 3.5.3). Once W at a point reaches W_c, that point loses all of its bonds irreversibly, and it is completely detached from the body. Translating this into the bond damage function defined in Eq. (11), we find:

$$\mu\left(\boldsymbol{x},\boldsymbol{x}',t\right)=\begin{cases}1, & \text{if } W\left(\boldsymbol{x},\tilde{t}\right) \text{ and } W\left(\boldsymbol{x}',\tilde{t}\right)\le W_c\\ 0, & \text{if } W\left(\boldsymbol{x},\tilde{t}\right) \text{or} W\left(\boldsymbol{x}',\tilde{t}\right)> W_c\end{cases},\ \text{for all } 0\le\tilde{t}\le t \tag{42}$$

where $W\left(\boldsymbol{x},\tilde{t}\right)$ denotes the strain energy density at point $\boldsymbol{x}$ and time $\tilde{t}$.

Eq. (42) states that a bond is broken and no longer carries load if the strain energy density at either of its end points exceeds W_c.

To obtain the convolutional form for this brittle failure model, we define the *integrity index* $\lambda(\boldsymbol{x},t)$ at each point as:

$$\lambda(\boldsymbol{x},t)=\begin{cases}1 & W\left(\boldsymbol{x},\tilde{t}\right)\le W_c\\ 0 & W\left(\boldsymbol{x},\tilde{t}\right)> W_c,\end{cases}\ \text{for all } 0\le\tilde{t}\le t \tag{43}$$

With this definition, we observe that we can write Eq. (42) in the equivalent form:

$$\mu\left(\boldsymbol{x},\boldsymbol{x}',t\right)=\lambda(\boldsymbol{x},t)\,\lambda\left(\boldsymbol{x}',t\right)=\lambda\lambda' \tag{44}$$

Next, we show how this new damage model leads to a convolutional structure.

3.5.2. Convolutional structure of the new damage model

In general, we show that a PD damage model with $\boldsymbol{L}(\boldsymbol{x},t) = \int_{\mathcal{H}_x} \mu \boldsymbol{f} \mathrm{d}V_{\boldsymbol{x}'}$ has a convolutional structure if $\boldsymbol{f}$ and μ are both in the form described by Eq. (13):

$$\begin{aligned}
\boldsymbol{L}(\boldsymbol{x},t) &= \int_{\mathcal{H}_x} \mu\left(\boldsymbol{x},\boldsymbol{x}',t\right) \boldsymbol{f}\left(\boldsymbol{x},\boldsymbol{x}',t\right)\, dV_{\boldsymbol{x}'} \\
&= \int_{\mathcal{H}_x} \left[\sum_{n=1}^{p} a_n(\boldsymbol{x},t)\, b_n\left(\boldsymbol{x}',t\right) c_n\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\right]\left[\sum_{m=1}^{q} y_m(\boldsymbol{x},t)\, w_m\left(\boldsymbol{x}',t\right) z_m\left(\boldsymbol{x}\right.\right. \\
&\left.\left. -\boldsymbol{x}',t\right)\right] dV_{\boldsymbol{x}'} = \int_{\mathcal{H}_x}\left[\sum_{n=1}^{p}\sum_{m=1}^{q} a_n y_m(\boldsymbol{x},t)\, b_n w_m\left(\boldsymbol{x}',t\right) c_n z_m\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\right] dV_{\boldsymbol{x}'} \\
&= \sum_{n=1}^{p}\sum_{m=1}^{q} a_n y_m(\boldsymbol{x},t) \int_{\mathcal{H}_x} b_n w_m\left(\boldsymbol{x}',t\right) c_n z_m\left(\boldsymbol{x}-\boldsymbol{x}',t\right)\, dV_{\boldsymbol{x}'} \\
&= \sum_{n=1}^{p}\sum_{m=1}^{q} a_n y_m(\boldsymbol{x},t)\left[(b_n w_m) * (c_n z_m)\right](\boldsymbol{x},t)
\end{aligned} \tag{45}$$

The new $\mu = \lambda\lambda'$ is in the form of Eq. (13). As a result, it can be used with any of the described constitutive models in Section 3. For the linearized bond-based model described in Section 3.1 for example, according to Eq. (10), Eqs. (18) and (44) we can write:

$$\begin{aligned}
L_i = \int_{\mathcal{H}_x} \lambda\lambda' f_i \mathrm{d}V_{\boldsymbol{x}'} &= \int_{\mathcal{H}_x} \lambda(\boldsymbol{x},t)\, \lambda\left(\boldsymbol{x}',t\right) f_i\left(\boldsymbol{x},\boldsymbol{x}',t\right) \mathrm{d}V_{\boldsymbol{x}'} \\
&= \lambda(\boldsymbol{x},t) \int_{\mathcal{H}_x} \lambda\left(\boldsymbol{x}',t\right) \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\left[u_j\left(\boldsymbol{x}',t\right) - u_j(\boldsymbol{x},t)\right] \mathrm{d}V_{\boldsymbol{x}'} \\
&= \lambda \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}-\boldsymbol{x}'\right) \lambda' u_j' \mathrm{d}V_{\boldsymbol{x}'} - \lambda u_j \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}-\boldsymbol{x}'\right)\lambda' \mathrm{d}V_{\boldsymbol{x}'} \\
&= \lambda\left[\mathrm{C}_{ij} * \left(\lambda u_j\right)\right] - \lambda u_j\left[\mathrm{C}_{ij} * \lambda\right]
\end{aligned} \tag{46}$$

The strain energy density for this material model can be written as:

$$\begin{aligned}
W(\boldsymbol{x},t) &= \frac{1}{2}\int_{\mathcal{H}_x} \lambda\lambda' \boldsymbol{\eta}\cdot\left[\mathbf{C}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\boldsymbol{\eta}\right] \mathrm{d}V_{\boldsymbol{x}'} = \frac{1}{2}\int_{\mathcal{H}_x} \lambda\lambda' \eta_i \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\eta_j \mathrm{d}V_{\boldsymbol{x}'} \\
&= \frac{1}{2}\int_{\mathcal{H}_x} \lambda\lambda' \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\left(u_i'-u_i\right)\left(u_j'-u_j\right) \mathrm{d}V_{\boldsymbol{x}'} \\
&= \frac{1}{2}\int_{\mathcal{H}_x} \lambda\lambda' \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\left(u_i'u_j' - u_i'u_j - u_iu_j' + u_iu_j\right) \mathrm{d}V_{\boldsymbol{x}'} \\
&= \frac{1}{2}\lambda\left[\int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\lambda' u_i' u_j' \mathrm{d}V_{\boldsymbol{x}'} - u_j \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\lambda' u_i' \mathrm{d}V_{\boldsymbol{x}'}\right. \\
&\left. - u_i \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\lambda' u_j' \mathrm{d}V_{\boldsymbol{x}'} + u_iu_j \int_{\mathcal{H}_x} \mathrm{C}_{ij}\left(\boldsymbol{x}'-\boldsymbol{x}\right)\lambda' \mathrm{d}V_{\boldsymbol{x}'}\right] \\
&= \frac{1}{2}\lambda\left\{\left[\mathrm{C}_{ij} * \left(\lambda u_iu_j\right)\right] - u_j\left[\mathrm{C}_{ij} * (\lambda u_i)\right] - u_i\left[\mathrm{C}_{ij} * \left(\lambda u_j\right)\right] + u_iu_j\left[\mathrm{C}_{ij} * \lambda\right]\right\}
\end{aligned} \tag{47}$$

The damage index in Eq. (12) can also be written in terms of convolutions if needed, as shown below. By defining a *characteristic influence function* with its support equal to the horizon region:

$$\omega_0(\boldsymbol{\xi}) = \begin{cases} 1 & |\boldsymbol{\xi}| \le \delta \\ 0 & |\boldsymbol{\xi}| > \delta \end{cases}, \tag{48}$$

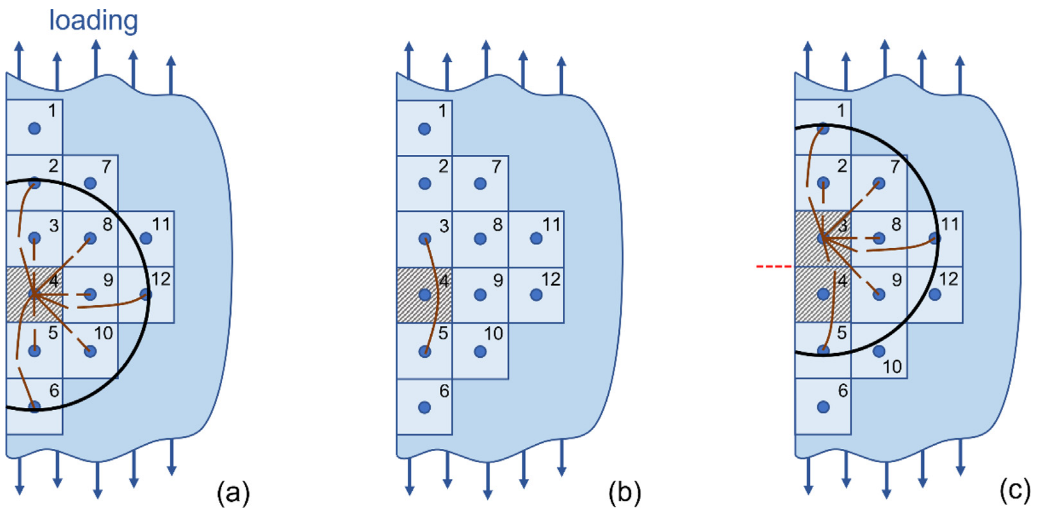


Fig. 2. Schematics of how through cracks form based on the new damage model in 2D, for a case with $\delta = 2\Delta x$. Consecutive nodal failure results in a δ-thick layer of nodes with $d = 1$ (all of their bonds are broken). The black arcs in (a) and (c), show the horizons for nodes 4 and 3, respectively.

one can write:

$$d(\boldsymbol{x}, t) = 1 - \frac{\int_{H_x} \mu \mathrm{d}V_{x'}}{\int_{H_x} \mathrm{d}V_{x'}} = 1 - \frac{\int_{\mathbb{R}^n} \lambda\lambda' \omega_0 \left(\left|\boldsymbol{x}' - \boldsymbol{x}\right|\right) \mathrm{d}V_{x'}}{\int_{\mathbb{R}^n} \omega_0 \left(\left|\boldsymbol{x}' - \boldsymbol{x}\right|\right) \mathrm{d}V_{x'}} = 1 - \frac{\int_{\mathbb{R}^n} \lambda\lambda' \omega_0 \left(\left|\boldsymbol{x} - \boldsymbol{x}'\right|\right) \mathrm{d}V_{x'}}{\beta_0}$$
$$1 - \frac{\lambda \int_{\mathbb{R}^n} \lambda' \omega_0 \left(\left|\boldsymbol{x}' - \boldsymbol{x}\right|\right) \mathrm{d}V_{x'}}{\beta_0} = 1 - \frac{\lambda \left(\lambda * \omega_0\right)}{\beta_0} \tag{49}$$

3.5.3. Calibrating the failure model to Griffith's critical energy release rate

To use this model for predicting fracture, the failure threshold W_c in Eq. (42) needs to be calibrated to a measurable fracture property of the material like Griffith's critical fracture energy. Inspired by the calibration method for the critical bond strain criterion [5], we perform the calibration by considering a "through" planar crack in a body and find a relationship between the PD failure threshold (here W_c) and the critical energy release rate G_0, which is a measurable quantity.

A through crack is one that splits the body into two completely separated parts. In peridynamics, this implies that no bridging bond should exist across the crack surface. For the pointwise criterion given by Eqs. (42) and (43), *a through crack contains a δ-thick layer with points that lost all of their bonds*. In Fig. 2 we show schematically why this is the case.

To understand the evolution of nodal damage and how it leads to crack formation, consider the emergence and propagation of a crack along the red horizontal line in Fig. 2. For clarity, we show the case with $\delta = 2\Delta x$, which has fewer bonds to plot.

Let node No. 4 be the first node where the strain energy density reaches W_c (Fig. 2a). As a consequence of the new point-wise damage criterion, the node loses all of its bonds. After failure of node 4, forces carried by its bonds will be redistributed to the bonds of nearby nodes. These bonds, especially the ones that are bridging over the forming crack (e.g., bond 3–5 in Fig. 2b), will likely be stretched sufficiently to cause the strain energy density W at one or more adjacent nodes to also become critical (e.g., node 3 in Fig. 2c). For the horizon size shown in Fig. 2, once nodes 3 and 4 fail (as shown in Fig. 2c), no bridging bonds will exist across the red dashed line and a through crack will start forming, along with a δ-thick layer of fully damaged nodes ($d = 1$). Note that with this damage model, if a region of thickness δ is not totally damaged along a crack path, bridging bonds will continue to exist.

In general, a through crack with the new model has a damage region of thickness 3δ: a δ-thick layer in the middle with the damage index $d = 1$ and two δ-thick layers on the sides with partially damaged nodes, $0 < d < 0.5$. This

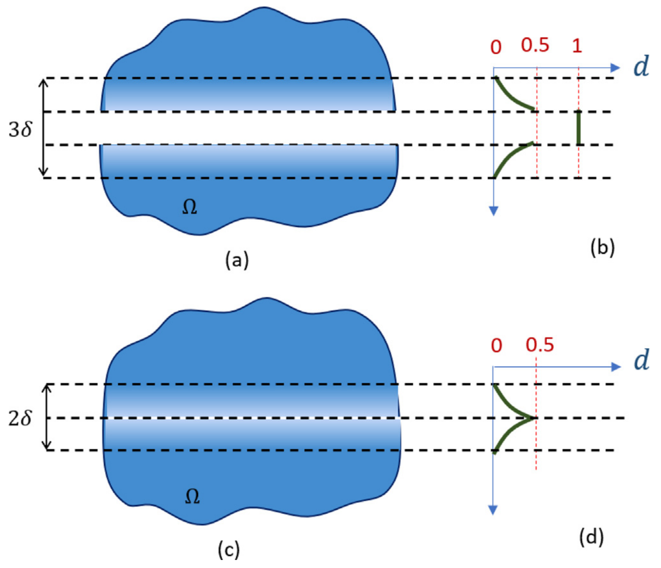


Fig. 3. Schematic comparison between the damage corresponding to a through crack in a peridynamic body in 2D for the new critical strain energy density damage model, and the critical bond strain model commonly-used with the meshfree PD model.

is different from the case for the critical bond strain criterion where a through crack has a 2δ-thick damaged region of partially damaged nodes (with $0 < d < 0.5$). Fig. 3 schematically compares a straight through crack in the new damage model with the one in the critical bond strain damage model in terms of thickness and damage profile across the crack.

Note that the blank channel in Fig. 3a is not because of separation of the split parts, but shows the δ-thick fully damaged layer along the crack. One important observation in the model is the mass loss in this middle layer with $d = 1$. It is reasonable to argue that as long as δ is small relative to the geometry, the mass loss effect is negligible. In the numerical examples in Section 5.2, the performance of the new model is tested against the critical bond strain meshfree PD model in a dynamic fracture problem that involves multiple crack branching events.

To find a relationship between W_c and the critical energy release rate G_0 for model calibration, we consider that all points inside the middle δ-thick layer of a through crack have reached the critical strain energy W_c. G_0 is the energy released for the crack to advance a unit crack area. Let A to denote the crack surface area. We enforce the equality between the strain energy required to grow the crack and create the new crack surface area A, and the corresponding critical energy release rate:

$$A\delta W_c = AG_0 \tag{50}$$

This results in:

$$W_c = \frac{G_0}{\delta} \tag{51}$$

We observe that this calibration is valid for brittle failure and is independent from the deformation model. Moreover, the calibrated threshold is identical for 2D and 3D.

We verify the proposed damage model in Section 5.2 by comparing the model's predictions in dynamic crack branching against published results obtained with the meshfree PD damage model, based on critical bond-strain.

Note that the discussion following Fig. 2 explains how a potential through crack (no bridging bonds) forms. This does not mean that in the new model damage can evolve only as through cracks. Different loading conditions may lead to distributed/diffuse damage. There could be instances where individual nodes, scattered through the domain, are fully damaged, causing their neighboring nodes to experience partial damage, but without leading to the growth of a through crack.

For damage growing as depicted in Fig. 2 in mode I opening, it is likely that nodes connected by bridging bonds will eventually reach W_c, and lose all of their bonds as a consequence; in special cases (e.g. mixed-mode fracture), however, crack opening may be small and temporary (see [43]) and a necessary through crack might not form with the new damage model, as some bonds may continue to bridge the crack. This is not surprising, since even the original PD damage model based on critical bond strain has the same limitation (see [43] for a possible workaround this problem).

Similar to the critical bond-strain criterion, the new damage model does not need to define cracks per se, instead, damage self-organizes (as part of the solution to the problem) into cracks or more diffuse damage, depending on the particular material system and loading conditions. As with the critical bond-strain PD damage model, in the new model one post-processes the damage data (failed bonds and nodes) to visualize the emergent cracks, fragments, etc. An example for tracking cracks in PD is found in [44].

4. The fast convolution-based method for peridynamic models in elasticity and fracture

In this section, we first discuss the convolution-based Fourier method for peridynamic models in elasticity and fracture in 3D periodic domains. Then we introduce the Embedded Constraint formulation to extend the method to problems on bounded domains of arbitrary shape and general nonlocal boundary conditions.

4.1. Fourier-based method for periodic domains

Let $\boldsymbol{x} = \{x_1, x_2, x_3\}$ be the position vector in 3D and the box $\mathbb{T} = [0, L_1] \times [0, L_2] \times [0, L_3]$, be a periodic 3D domain with 0 being "identified" with L_1, L_2, and L_3 in all three directions due to periodicity of the box. We first uniformly discretize the domain, using N_1, N_2, and N_3 number of nodes in the Cartesian coordinate directions 1, 2, and 3 respectively:

$$\begin{gathered}\boldsymbol{x}_{nmp} = \{(n-1)\,\Delta x_1, (m-1)\,\Delta x_2, (p-1)\,\Delta x_3\}\,, \text{where}\,\Delta x_1 = \frac{L_1}{N_1};\, \Delta x_2 = \frac{L_2}{N_2};\, \Delta x_3 = \frac{L_3}{N_3} \\ \text{where}\, n = \{1, 2, \ldots, N_1\}\,;\, m = \{1, 2, \ldots, N_2\}\,;\, p = \{1, 2, \ldots, N_3\}\,.\end{gathered} \tag{52}$$

The total number of nodes is then $N = N_1 N_2 N_3$.

We approximate the vector-valued displacement field $\boldsymbol{u}(\boldsymbol{x}, t)$ with the following $N/2$ -degree trigonometric approximation, also known as the *discrete Fourier series*. Given the discretized periodic domain [45,46], we write:

$$\begin{gathered}u_i^N\left(\boldsymbol{x}_{nmp}, t\right) = \frac{1}{N_3 N_2 N_1} \sum_{k_3=-\frac{N_3}{2}+1}^{\frac{N_3}{2}} \sum_{k_2=-\frac{N_2}{2}+1}^{\frac{N_2}{2}} \sum_{k_1=-\frac{N_1}{2}+1}^{\frac{N_1}{2}} \hat{u}_i\left(\boldsymbol{k}, t\right) e^{2\pi\zeta\left(\frac{k_1 x_1}{L_1}+\frac{k_2 x_2}{L_2}+\frac{k_3 x_3}{L_3}\right)}; \\ \text{and } i = 1, 2, 3\end{gathered} \tag{53}$$

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

$$\hat{u}_i\left(\boldsymbol{k}, t\right) = \sum_{p=1}^{N_3} \sum_{m=1}^{N_2} \sum_{n=1}^{N_1} u_i^N\left(\boldsymbol{x}_{nmp}, t\right) e^{-2\pi\zeta\left(\frac{k_1 n}{N_1}+\frac{k_2 m}{N_2}+\frac{k_3 p}{N_3}\right)};\; i = 1, 2, 3 \tag{54}$$

are the *discrete Fourier coefficients* of u_i^N. The operation on u_i^N in Eq. (54) is called the *discrete Fourier transform* (DFT), and the inverse operation on $\hat{u}_i$ in Eq. (53) is called the *inverse discrete Fourier transform* (iDFT) [45,46]. Note that this definition of DFT maps one-to-one between N values of u_i^N, and N values of $\hat{u}_i$, meaning that the number of Fourier basis functions used in the trigonometric approximation in Eq. (53) is equal to the total number of spatial nodes.

From the computational point of view, the DFT and iDFT operations are carried out via efficient Fast Fourier Transform (FFT) algorithms, and the inverse (iFFT), which have complexity $N \log_2 N$ [16,17].

Note that if the spatial discretization is non-uniform, non-uniform DFT and iDFT [47] can be used instead of Eqs. (53) and (54). Non-uniform fast Fourier transform (NUFFT) algorithms exist (see, e.g., [47–50]) for computing the non-uniform versions of DFT and iDFT at the same complexity of the uniform FFT. In this study, however, we have only employed uniform discretizations.

4.1.1. Fourier-based discretization of peridynamics operators

To use the fast convolution-based method for PD equations, we first approximate the PD integrals via, for example, the one-point Gaussian quadrature rule (mid-point integration) on the periodic domain using the discrete nodes given by Eq. (52). Note that computing this quadrature directly, as done in the meshfree PD method [5], leads to a $O(N^2)$ complexity for each time step. Instead, for PD operators that can be expressed with the convolutional form of Eq. (14), we reorganize the quadrature summation and express it as convolution sums. To compute each convolution, we discretize the convolving functions by using their trigonometric series approximation (see Eq. (53)); we use the FFT to obtain the discrete Fourier coefficients of the convolving functions; we multiply the Fourier coefficients of these functions, and transform back the product into physical space by using the iFFT. Adding up these computed convolution terms returns the value of the PD integral.

Here we show the procedure for the PD operators discussed in Section 3:

(1) The linearized bond-based elastic PD model

As described above, we first approximate the integral in Eq. (18) with Gaussian quadrature, and reorganize it to reach the convolutional structure. The convolutional forms in this section are identical to the continuous versions in Section 3, but in a discrete version. For conciseness, we denote the triple sum $\sum_{q=1}^{N_3}\sum_{s=1}^{N_2}\sum_{r=1}^{N_1}$ by $\sum_{q,s,r=1}^{N_3,N_2,N_1}$, and let $\Delta V = \Delta x_1 \Delta x_2 \Delta x_3$:

$$\begin{aligned} L_i(\boldsymbol{x},t) \cong L_i^N(\boldsymbol{x}_{nmp},t) &= \sum_{q,s,r=1}^{N_3,N_2,N_1} \mathrm{C}_{ij}^N(\boldsymbol{x}_{rsq}-\boldsymbol{x}_{nmp})\left[u_j^N(\boldsymbol{x}_{rsq},t)-u_j^N(\boldsymbol{x}_{nmp},t)\right]\Delta V \\ &= \sum_{q,s,r=1}^{N_3,N_2,N_1} \mathrm{C}_{ij}^N(\boldsymbol{x}_{nmp}-\boldsymbol{x}_{rsq})\,u_j^N(\boldsymbol{x}_{rsq},t)\,\Delta V \\ &\quad - \left[\sum_{q,s,r=1}^{N_3,N_2,N_1} \mathrm{C}_{ij}^N(\boldsymbol{x}_{nmp}-\boldsymbol{x}_{rsq})\,\Delta V\right] u_j^N(\boldsymbol{x}_{nmp},t) \\ &= \Delta V\left[\mathrm{C}_{ij}^N \circledast_N u_j^N\right](\boldsymbol{x}_{nmp},t) - \mathrm{P}_{ij}^N u_j^N(\boldsymbol{x}_{nmp},t) \end{aligned} \tag{55}$$

where C_{ij}^N is the discrete Fourier series approximation of C_{ij} in Eq. (17), and $\circledast_N$ denotes the circular convolution sum (convolution sum on a periodic domain, see, e.g., [51]). P_{ij}^N is a constant matrix, computable by the numerical integration of C_{ij}^N using the 1-point Gaussian quadrature rule on the same discretized domain:

$$\mathrm{P}_{ij}^N = \sum_{p,m,n=1}^{N_3,N_2,N_1} \mathrm{C}_{ij}^N(\boldsymbol{x}_{nmp})\,\Delta V \tag{56}$$

Note that when we use square grids and a spherical horizon, there are nodes near the horizon edge whose volumes are only partially covered by the horizon [52,53]. Using a fixed ΔV for all nodes in a family introduces some quadrature error as it also leaves out contributions to the integral coming from nodes just outside the horizon region but whose volumes are partially covered by it. This error can be reduced by using small grid size (large $m = \delta/\Delta x$), partial volume correction algorithms, or using decaying influence functions [53]. Partial volume correction algorithms like [52,54], replace ΔV in Eq. (55) with a $\Delta V(|\boldsymbol{\xi}|)$ which in computations can be viewed as a part of $\mathrm{C}_{ij}^N(|\boldsymbol{\xi}|)$, and therefore would preserve the convolutional structure. In the examples shown below, we use the constant ΔV algorithm, while being aware of its quadrature error.

To reduce notational complexity, we drop the superscript N for discretized quantities and eliminate the argument brackets $(\boldsymbol{x}_{nmp},t)$ by replacing with the superscript (nmp,t). We use $\mathbf{F}$ and $\mathbf{F}^{-1}$ to denote FFT and iFFT operations. Using FFT and iFFT, the PD integral operator can be computed as:

$$L_i^N(\boldsymbol{x}_{nmp},t) = L_i^{nmp,t} = \mathbf{F}^{-1}\left[\mathbf{F}(\mathrm{C}_{ij})\,\mathbf{F}(u_j)\right]\Big|^{nmp,t}\Delta V - \mathrm{P}_{ij}u_j^{nmp,t} \tag{57}$$

Note that j on the righthand side is a dummy index and denotes summations over $j = 1,2,3$ for each i. For demonstration, we expand Eq. (57) for $i = 1$:

$$\begin{aligned} L_1^{nmp,t} &= \mathbf{F}^{-1}\left[\mathbf{F}(\mathrm{C}_{11})\,\mathbf{F}(u_1) + \mathbf{F}(\mathrm{C}_{12})\,\mathbf{F}(u_2) + \mathbf{F}(\mathrm{C}_{13})\,\mathbf{F}(u_3)\right]\Big|^{nmp,t}\Delta V - \mathrm{P}_{11}u_1^{nmp,t} \\ &\quad - \mathrm{P}_{12}u_2^{nmp,t} - \mathrm{P}_{13}u_3^{nmp,t} \end{aligned} \tag{58}$$

Similar relationships are obtained for $L_2^{nmp,t}$, and $L_3^{nmp,t}$ by expanding Eq. (57) for $i=2$ and 3, and summing over the dummy index j.

(2) The nonlinear bond-based elastic material

We follow the same discretization procedure for the PD nonlinear operator provided in Eq. (23), which yields:

$$\begin{aligned} L_i^{nmp,t} &= \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_j\right)\right]\Big|^{nmp,t}\Delta V - \mathrm{P}_{ij}u_j^{nmp,t} - \mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(u_ju_j\right)\right]\Big|^{nmp,t}\Delta V \\ &+ \mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(u_j\right)\right]\Big|^{nmp,t}\Delta V u_j^{nmp,t} - \frac{1}{2}q_i^{nmp,t}u_j^{nmp,t}u_j^{nmp,t} \\ &- \mathbf{F}^{-1}\left[\mathbf{F}\left(a_j\right)\mathbf{F}\left(u_ju_i\right)\right]\Big|^{nmp,t}\Delta V + u_j^{nmp,t}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(a_j\right)\mathbf{F}\left(u_i\right)\right]\Big|^{nmp,t}\Delta V \\ &+ \mathbf{F}^{-1}\left[\mathbf{F}\left(a_j\right)\mathbf{F}\left(u_j\right)\right]\Big|^{nmp,t}\Delta V - q_i^{nmp,t}u_i^{nmp,t}u_j^{nmp,t} \\ &+ \frac{1}{2}\mathbf{F}^{-1}\left[\mathbf{F}\left(c\right)\mathbf{F}\left(u_ju_ju_i\right)\right]\Big|^{nmp,t}\Delta V - \mathbf{F}^{-1}\left[\mathbf{F}\left(c\right)\mathbf{F}\left(u_iu_j\right)\right]\Big|^{nmp,t}\Delta V u_j^{nmp,t} \\ &+ \frac{1}{2}\mathbf{F}^{-1}\left[\mathbf{F}\left(c\right)\mathbf{F}\left(u_i\right)\right]\Big|^{nmp,t}\Delta V u_j^{nmp,t}u_j^{nmp,t} \\ &- \frac{1}{2}\mathbf{F}^{-1}\left[\mathbf{F}\left(c\right)\mathbf{F}\left(u_ju_j\right)\right]\Big|^{nmp,t}\Delta V u_i^{nmp,t} \\ &+ \mathbf{F}^{-1}\left[\mathbf{F}\left(c\right)\mathbf{F}\left(u_j\right)\right]\Big|^{nmp,t}\Delta V u_j^{nmp,t}u_i^{nmp,t} - \frac{1}{2}u_j^{nmp,t}u_j^{nmp,t}u_i^{nmp,t}p \end{aligned} \tag{59}$$

where C_{ij}, a_j, and c are discrete Fourier series approximations of the functions given in Eq. (24). P_{ij}, q_i, and p are the numerical integrations of C_{ij}, a_j, and c respectively, similar to the quadrature in Eq. (56).

(3) The linearized state-based elastic solid

According to the convolutional form derived in Eq. (38), the discrete version of $L_i(\boldsymbol{x},t)$ for this material is:

$$L_i^{nmp,t} = \vartheta^{nmp,t}q^{nmp,t} - \mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(\vartheta\right)\right]\Big|^{nmp,t}\Delta V + \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_j\right)\right]\Big|^{nmp,t}\Delta V - \mathrm{P}_{ij}u_j^{nmp,t} \tag{60}$$

Where C_{ij} and a_j are discrete Fourier series approximations of the functions given in Eq. (37). P_{ij} and q_i are numerical integrations of C_{ij} and a_j respectively. The discrete version of ϑ given in Eq. (39) is:

$$\vartheta^{nmp,t} = \frac{-3}{m^{nmp,t}}\left\{\mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(u_i\right)\right]\Big|^{nmp,t}\Delta V + q_iu_i^{nmp,t}\right\} \tag{61}$$

(4) The damage model:

Here we show the discretization for the damage model introduced in Section 3.5. According to Eq. (46) one can write:

$$L_i^{nmp,t} = \lambda^{nmp,t}\left\{\mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\lambda u_j\right)\right]\Big|^{nmp,t} - \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\lambda\right)\right]\Big|^{nmp,t}u_j^{nmp,t}\right\}\Delta V \tag{62}$$

where

$$\lambda^{nmp,t} = \begin{cases} 1 & W^{nmp,t-\Delta t} \le \dfrac{G_0}{\delta} \\ 0 & W^{nmp,t-\Delta t} > \dfrac{G_0}{\delta} \end{cases} \tag{63}$$

And $W^{nmp,t-\Delta t}$ is the nodal strain energy density at node $\boldsymbol{x}_{nmp}$ and the previous time step $t-\Delta t$. From Eq. (47):

$$\begin{aligned} W^{nmp,t} = \frac{1}{2}\lambda^{nmp,t}\Delta V\Big\{&\mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_iu_j\lambda\right)\right]\Big|^{nmp,t} - u_j^{nmp,t}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_i\lambda\right)\right]\Big|^{nmp,t} \\ &- u_i^{nmp,t}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_j\lambda\right)\right]\Big|^{nmp,t} \\ &+ u_i^{nmp,t}u_j^{nmp,t}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\lambda\right)\right]\Big|^{nmp,t}\Big\} \end{aligned} \tag{64}$$

According to Eq. (49), damage is computed by:

$$d^{nmp,t} = 1 - \frac{\lambda^{nmp,t}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(\omega_0\right)\mathbf{F}\left(\lambda\right)\right]\Big|^{nmp,t}}{\sum_{p,m,n=1}^{N_3,N_2,N_1}\omega_0^{nmp}} \tag{65}$$

By obtaining the PD operators in discrete form, we proceed to the full discretization of the PD equation of motion for dynamic problems, and the PD equilibrium equation for static problems.

4.1.2. Discretization of peridynamic equation of motion and equilibrium

Once the PD integral operator is obtained in discrete form via the FFT-based quadrature (examples shown in Section 4.1.1), one can write the PD integro-differential equation of motion (see Eq. (1)) as a system of second order ODEs:

$$\rho \left. \frac{\mathrm{d}^2 u_i}{\mathrm{d}t^2} \right|^{nmp,t} = L_i^{nmp,t} + b_i^{nmp,t}; \text{ for } i = 1, 2, 3 \tag{66}$$

where $b_i^{nmp,t}$ is the body force density evaluated at the node nmp at time t. The ODE system can be solved via standard second-order ODE solvers. In the case of velocity-Verlet time integration [13] for example:

$$\begin{aligned}
v_i^{nmp,t+\frac{\Delta t}{2}} &= v_i^{nmp,t} + \frac{\Delta t}{2\rho}\left(L_i^{nmp,t} + b_i^{nmp,t}\right) \\
u_i^{nmp,t+\frac{\Delta t}{2}} &= u_i^{nmp,t} + \Delta t \left(v_i^{nmp,t+\frac{\Delta t}{2}}\right) \\
v_i^{nmp,t+\Delta t} &= v_i^{nmp,t+\frac{\Delta t}{2}} + \frac{\Delta t}{2\rho}\left(L_i^{nmp,t+\Delta t} + b_i^{nmp,t+\Delta t}\right)
\end{aligned} \tag{67}$$

where $\boldsymbol{v}^{nmp,t} = \left\{v_1^{nmpt}, v_2^{nmp,t}, v_3^{nmp,t}\right\}$ denotes the discrete velocity field, and Δt is the time-step.

The described Fourier-based discretization in space and velocity-Verlet time integration can be used to solve PD equation of motion for dynamic problems in period domains. In Section 4.2 we extend the method for arbitrary domains and boundary conditions.

For static problems in periodic domains we need to solve the equilibrium equation:

$$\boldsymbol{L}(\boldsymbol{x}) + \boldsymbol{b}(\boldsymbol{x}) = 0 \tag{68}$$

which is a special case of Eq. (1), without the time dependency. Repeating the same Fourier-based discretization process given by Eqs. (52) to (66), and discarding the time variable t results in:

$$L_i^{nmp} + b_i^{nmp} = 0; \text{ for } i = 1, 2, 3 \tag{69}$$

Eq. (69), in general, can be a linear or nonlinear system of equations in terms of the unknown $\boldsymbol{u}^N$. We define the vector-valued *residual function* $\boldsymbol{R}\left(\boldsymbol{u}^N\right)$:

$$R_i\left(\boldsymbol{u}^N\right) = L_i^{nmp} + b_i^{nmp}; \text{for} i = 1, 2, 3 \tag{70}$$

One can use an iterative solver that finds $\boldsymbol{u}^N$ such that $\boldsymbol{R}\left(\boldsymbol{u}^N\right) = \boldsymbol{0}$. In this framework, when using an iterative solver that involves matrix–vector products, one should not compute such products directly in order to maintain the complexity of $O\left(N \log_2 N\right)$. The matrix–vector products in these solvers are in fact, the PD integrals operating on quantities represented by vectors. Therefore, one can use the FFT-based description of the system ($\boldsymbol{L}$ formulas given in Section 4.1.1) to compute the matrix–vector products at the cost of $\left(N \log_2 N\right)$.

The FFT-based method described in this section so far, is only applicable to the PD problems defined over periodic domains. Next, we discuss the extension of this method to bounded domains with arbitrary shapes and volume constraints using the embedded constraint approach introduced in [27] for scalar problems (diffusion), which is extended here to vector problems (elasticity).

4.2. Embedded constraints for enforcing boundary conditions

Before we describe the embedded constraint method we briefly discuss boundary conditions in PD nonlocal settings.

4.2.1. Boundary conditions in peridynamics

In the classical (local) theory, boundary conditions (BC) restrict the solution for boundary-value problems in on the boundary around the domain. These boundaries are lower dimensional manifolds compared to the domain's dimension. For example, a 3D domain has a 2D boundary, and a 2D domain has a 1D boundary.

In PD, the nonlocality requires the constraints to be defined over a "thick boundary", which has the same dimension as the domain. The constraints are defined over a chunk of the domain with the thickness of at most δ (see Fig. 4) [55].

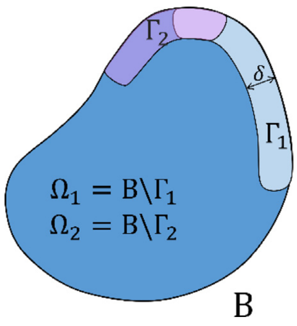

Γ_2
δ
Γ_1
$\Omega_1 = B\backslash\Gamma_1$
$\Omega_2 = B\backslash\Gamma_2$
B

Fig. 4. A 2D peridynamic body (B), consisting of the domain Ω_1 and Ω_2, where u_1 and u_2 are unknown and the boundary layer/constrained volume (Γ_1 and Γ_2) where u_1 and u_2 are independently prescribed.

In many applications it is usually desired/needed to apply local boundary conditions. The reason is that empirical measurements mostly provide the data on surfaces, rather than on a chunk of the domain. One approach to apply a local BC in a PD problem, is to extend the domain by a “fictitious region” to specify certain volume constraints such that the desired local BC is effectively enforced. These types of methods are usually referred to as *the fictitious nodes methods* (FNM) and are extensively discussed in [56]. There are other types of methods for applying local BC to PD problems as well (see, e.g., [57–59]).

For mechanical problems in 2D and 3D, displacement constraints may be assigned over different portions of the boundary for different components. In this study we regard the domain and all constrained volume as the PD body and denote it by B. For each component u_i, Ω_i is the domain where u_i is unknown and Γ_i is where u_i is given. In the 3D case, one can write $\mathrm{B} = \Omega_1 \cup \Gamma_1 = \Omega_2 \cup \Gamma_2 = \Omega_3 \cup \Gamma_3$. Fig. 4 shows a generic 2D PD body consisting of domains and constrained volumes. Note that in this study we only consider cases where u_1, u_2, u_3 are independently prescribed over their corresponding constrained volumes: Γ_1, Γ_2, Γ_3 respectively. The interdependent type constraints where a region requires certain relationship to hold between u_1, u_2, u_3 is not studied here.

In this study, when we intent to use FNM to apply local BCs, the fictitious regions are regarded as Γ_i and therefore are viewed as part of the PD body.

Dynamic problems within PD theory are often formulated as *initial-value volume-constrained PD problems* [60]:

$$\begin{cases} \rho \dfrac{\partial^2 u_i(\boldsymbol{x},t)}{\partial t^2} = L_i(\boldsymbol{x},t) + b_i(\boldsymbol{x},t) & \boldsymbol{x} \in \Omega_i, t > 0 \\ u_i(\boldsymbol{x},0) = u_i^0;\ v_i(\boldsymbol{x},0) = v_i^0 \text{ (initial conditions)} & \boldsymbol{x} \in \Omega_i \\ G_i(u_i) = 0 \text{ (volume constraints)} & \boldsymbol{x} \in \Gamma_i, t \geq 0 \end{cases} ; \text{ and } i = 1,2,3 \tag{71}$$

Static problems are then defined as *volume-constrained PD problems*:

$$\begin{cases} L_i(\boldsymbol{x},t) + b_i(\boldsymbol{x},t) = 0 & \boldsymbol{x} \in \Omega_i, t > 0 \\ G_i(u_i) = 0 \quad \text{(volume constraints)} & \boldsymbol{x} \in \Gamma_i, t \geq 0 \end{cases} ; \text{ and } i = 1,2,3 \tag{72}$$

G_i in Eqs. (71) and (72), denotes known (given) functions of u_i defined on Γ_i, as generic volume constraints. Neumann-type volume constraints are handled, as usual in PD models, as body forces. We use such conditions in the dynamic crack branching example in Section 5.2.

4.2.2. Embedded constraint method

In order to solve general (initial value) volume-constrained PD problems via the fast convolution-based method, the first step is to enclose the whole PD body (B) within a periodic box ($\mathbb{T}^{\mathrm{d}}$) where the superscript “d” denotes the spatial dimension. Fig. 5 shows an enclosed generic body within $\mathbb{T}^2$ (compare with Fig. 4).

Note that B needs to be at least at δ-distance from the edges of $\mathbb{T}^{\mathrm{d}}$ to avoid any undesired nonlocal interactions between the body and the periodic extensions.

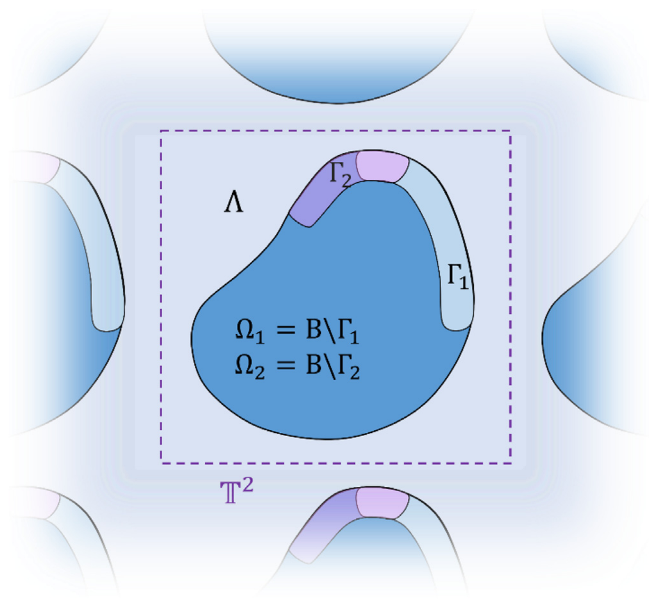


Fig. 5. Extension of a bounded peridynamic body to a 2D periodic box.

We define the following characteristic functions:

$$\chi_{\mathrm{B}}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \mathrm{B} \\ 0 & \boldsymbol{x} \in \mathbb{T}^{\mathrm{d}} \backslash \mathrm{B} = \Lambda \end{cases} \tag{73}$$

and

$$\chi_{\Omega_i}(\boldsymbol{x}) = \begin{cases} 1 & \boldsymbol{x} \in \Omega_i \\ 0 & \boldsymbol{x} \in \mathbb{T}^{\mathrm{d}} \backslash \Omega_i = \Gamma_i \cup \Lambda \end{cases} \tag{74}$$

$\chi_{\mathrm{B}}(\boldsymbol{x})$ allows us to modify the $\boldsymbol{L}(\boldsymbol{x}, t)$ formula for Eq. (2) and "cut" all the bonds that connect B to Λ, i.e. eliminating any interaction between the PD body and the rest of the box $\mathbb{T}^{\mathrm{d}}$:

$$L_i(\boldsymbol{x}, t) = \int_{\mathcal{H}_x} \chi_{\mathrm{B}}(\boldsymbol{x})\, \chi_{\mathrm{B}}(\boldsymbol{x}')\, f_i(\boldsymbol{x}, \boldsymbol{x}', t)\, \mathrm{d}V_{\boldsymbol{x}'} = \int_{\mathcal{H}_x} \chi_{\mathrm{B}} \chi'_{\mathrm{B}} f_i \mathrm{d}V_{\boldsymbol{x}'} \tag{75}$$

Note that $\chi_{\mathrm{B}} \chi'_{\mathrm{B}}$ is always zero for bonds that have one end in B and one end in Λ. With this modification, any nonlocal interaction between the point inside and outside the body (B) is filtered by $\chi_{\mathrm{B}} \chi'_{\mathrm{B}} = 0$. This effectively means that the body is mechanically cut out from the rest of $\mathbb{T}^{\mathrm{d}}$.

Similar to the case for $\mu = \lambda\lambda'$ discussed in Section 3.5.2, one can show that convolutional structure of $L_i(\boldsymbol{x}, t)$ is preserved after it is modified with $\chi_{\mathrm{B}} \chi'_{\mathrm{B}}$ in Eq. (75).

Note that the uniformity/nonuniformity of the grids does not affect the embedded constraint approach, since χ_{B} is defined based on the nodal coordinates, regardless of the nature of the grid spacing.

4.2.3. FCBM-EC for dynamic problems

In the case of PD dynamic problems consider the following initial value volume-constrained problem where $u_i(\boldsymbol{x}, t)$ are explicitly specified on Γ_i:

$$\begin{cases} \rho \dfrac{\partial^2 u_i(\boldsymbol{x}, t)}{\partial t^2} = L_i(\boldsymbol{x}, t) + b_i(\boldsymbol{x}, t) & \boldsymbol{x} \in \Omega_i, t > 0 \\ u_i(\boldsymbol{x}, 0) = u_i^0;\ v_i(\boldsymbol{x}, 0) = v_i^0 \text{ (initial conditions)} & \boldsymbol{x} \in \Omega_i \\ u_i(\boldsymbol{x}, t) = g_i(\boldsymbol{x}, t) \text{ (volume constraints)} & \boldsymbol{x} \in \Gamma_i, t \geq 0 \end{cases} ;\ \text{and } i = 1, 2, 3 \tag{76}$$

We use $\chi_{\Omega_i}(\boldsymbol{x})$ and $\chi_{\mathrm{B}}(\boldsymbol{x})$ to replace Eq. (76), which is a problem on the bounded domain B, with the following equivalent problem on the periodic domain $\mathbb{T}^{\mathrm{d}}$:

$$\begin{cases} \rho \dfrac{\partial^2 u_i}{\partial t^2} = \chi_{\Omega_i}\left(\displaystyle\int_{\mathcal{H}_x} \chi_{\mathrm{B}}\chi'_{\mathrm{B}} f_i \mathrm{d}V_{\boldsymbol{x}'} + b_i\right) + \left(1-\chi_{\Omega_i}\right)\rho \dfrac{\partial^2 w_i}{\partial t^2} & \boldsymbol{x} \in \mathbb{T}^{\mathrm{d}},\ t>0 \\ u_i(\boldsymbol{x},0) = \chi_{\Omega_i}\left(u_i^0\right) + \left(1-\chi_{\Omega_i}\right) w_i(\boldsymbol{x},0) & \boldsymbol{x} \in \mathbb{T}^{\mathrm{d}} \\ v_i(\boldsymbol{x},0) = \chi_{\Omega_i}\left(v_i^0\right) + \left(1-\chi_{\Omega_i}\right) \dfrac{\partial w_i(\boldsymbol{x},0)}{\partial t} & \boldsymbol{x} \in \mathbb{T}^{\mathrm{d}} \end{cases} \tag{77}$$

where $w_i(\boldsymbol{x},t)$, is a function which is equal to volume constraints on Γ_i, and zero elsewhere:

$$w_i(\boldsymbol{x},t) = \begin{cases} g_i(\boldsymbol{x},t) & \boldsymbol{x} \in \Gamma_i \\ 0 & \boldsymbol{x} \in \mathbb{T}^{\mathrm{d}} \backslash \Gamma_i \end{cases} \tag{78}$$

The values of w_i on Ω_i do not participate in the computation as they are masked out from the solution (see Eq. (77)), and therefore can be chosen arbitrarily. In our examples, we chose the value zero. The values of w_i on Λ, do participate in the computation, but cannot influence the results because of the presence of the $\chi_{\mathrm{B}}\chi'_{\mathrm{B}}$ factor in the integral operators which renders all of the contributions from Λ zero. Consequently, w_i on Λ can be chosen arbitrarily, and we have set those values to be zero. With these choices, however, the field variable $u_i(\boldsymbol{x},t)$ becomes discontinuous at the $\Omega_i - \Lambda$ interface. In the numerical examples shown below we did not encounter any Gibbs-type phenomenon associated with the FFT operations on u_i. Nevertheless, the influence of the extension's (w_i's) smoothness for the field variable u_i on convergence rates is worthy of investigation, and this is planned for the future.

Since the problem in Eq. (77) is defined over a periodic domain, we can use the Fourier-based quadrature for computing $L_i^{nmp,t}$ and write:

$$\rho \left.\frac{\mathrm{d}^2 u_i}{\mathrm{d}t^2}\right|^{nmp,t} = \chi_{\Omega_i}^{nmp}\left(L_i^{nmp,t} + b_i^{nmp,t}\right) + \left(1-\chi_{\Omega_i}^{nmp}\right)\rho \left.\frac{\mathrm{d}^2 w_i}{\mathrm{d}t^2}\right|^{nmp,t};\ \text{for } i=1,2,3 \tag{79}$$

According to the definition of χ_{Ω_i} in Eq. (74), one can rewrite Eq. (79) as:

$$\rho \left.\frac{\mathrm{d}^2 u_i}{\mathrm{d}t^2}\right|^{nmp,t} = \begin{cases} L_i^{nmp,t} + b_i^{nmp,t}, & \boldsymbol{x}_{nmp} \in \Omega_i \\ \rho \left.\dfrac{\mathrm{d}^2 w_i}{\mathrm{d}t^2}\right|^{nmp,t}, & \boldsymbol{x}_{nmp} \in \mathbb{T}\backslash\Omega_i \end{cases};\ \text{for } i=1,2,3 \tag{80}$$

We use the velocity-Verlet method for temporal integration (see Section 4.1.2) of the upper branch of Eq. (80) (on Ω_i), as there is no need for temporal integration of the lower branch, since the $u_i^{nmp,t}$ values are already known (on $\mathbb{T}\backslash\Omega_i$). Substituting the first relation in Eq. (67) into the second relation in the same equation, and applying that to Eq. (80), we find:

$$u_i^{nmp,t+\Delta t} = \begin{cases} u_i^{nmp,t} + \Delta t\left[v_i^{nmp,t} + \dfrac{\Delta t}{2\rho}\left(L_i^{nmp,t} + b_i^{nmp,t}\right)\right] & ;\boldsymbol{x}_{nmp} \in \Omega_i \\ w_i^{nmp,t+\Delta t} & ;\boldsymbol{x}_{nmp} \in \mathbb{T}^{\mathrm{d}}\backslash\Omega_i \end{cases} \tag{81}$$

Applying the third relation from Eq. (67) to Eq. (80) gives:

$$v_i^{nmp,t+\Delta t} = \begin{cases} v_i^{nmp,t} + \dfrac{\Delta t}{2\rho}\left[\left(L_i^{nmp,t} + b_i^{nmp,t}\right) + \left(L_i^{nmp,t+\Delta t} + b_i^{nmp,t+\Delta t}\right)\right] & ;\boldsymbol{x}_{nmp} \in \Omega_i \\ 0\,(\text{value not used}) & ;\boldsymbol{x}_{nmp} \in \mathbb{T}^{\mathrm{d}}\backslash\Omega_i \end{cases} \tag{82}$$

Using the characteristic function χ_{Ω_i}, Eqs. (81) and (82) can be re-written as:

$$u_i^{nmp,t+\Delta t} = \chi_{\Omega_i}^{nmp}\left\{u_i^{nmp,t} + \Delta t\left[v_i^{nmp,t} + \frac{\Delta t}{2\rho}\left(L_i^{nmp,t} + b_i^{nmp,t}\right)\right]\right\} + \left(1-\chi_{\Omega_i}^{nmp}\right) w_i^{nmp,t+\Delta t} \tag{83}$$

and

$$v_i^{nmp,t+\Delta t} = \chi_{\Omega_i}^{nmp}\left\{v_i^{nmp,t} + \frac{\Delta t}{2\rho}\left[\left(L_i^{nmp,t} + b_i^{nmp,t}\right) + \left(L_i^{nmp,t+\Delta t} + b_i^{nmp,t+\Delta t}\right)\right]\right\} \tag{84}$$

4.3. FCBM-EC for static and quasi-static problems

For static and quasi-static problems, consider the following PD volume-constrained problem where $\boldsymbol{u}(\boldsymbol{x})$ is explicitly specified on the constrained volume:

$$\begin{cases} L_i(\boldsymbol{x}) + b_i(\boldsymbol{x}) = 0 & , \boldsymbol{x} \in \Omega_i \\ u_i(\boldsymbol{x}) = g_i(\boldsymbol{x}) \text{ (volume constraints)} & , \boldsymbol{x} \in \Gamma_i \end{cases} ; \text{ and } i = 1, 2, 3 \tag{85}$$

Similar to the dynamic problem, we replace this description which is defined on the bounded domain B, with the following equivalent problem on the periodic domain $\mathbb{T}^{\mathrm{d}}$:

$$\chi_{\Omega_i}(\boldsymbol{x}) \left[\int_{\mathcal{H}_x} \chi_{\mathrm{B}}(\boldsymbol{x}) \chi_{\mathrm{B}}(\boldsymbol{x}') f_i(\boldsymbol{x}, \boldsymbol{x}') \mathrm{d}V_{\boldsymbol{x}'} + b_i(\boldsymbol{x}) \right] + \left[1 - \chi_{\Omega_i}(\boldsymbol{x})\right] \left[u_i(\boldsymbol{x}) - w_i(\boldsymbol{x})\right] = 0 \tag{86}$$

where

$$w_i(\boldsymbol{x}) = \begin{cases} g_i(\boldsymbol{x}) & \boldsymbol{x} \in \Gamma_i \\ 0 & \boldsymbol{x} \in \mathbb{T} \backslash \Gamma_i \end{cases} \tag{87}$$

Discretization of Eq. (86) leads to:

$$\chi_{\Omega_i}^{nmp} \left(L_i^{nmp} + b_i^{nmp}\right) + \left(1 - \chi_{\Omega_i}^{nmp}\right) \left(u_i^{nmp} - w_i^{nmp}\right) = 0; \text{ for } i = 1, 2, 3 \tag{88}$$

Where L_i^{nmp} is allowed to be computed using the Fourier-based quadrature due to the periodic reconstruction of the problem. Note that the second term in Eq. (88) is already satisfied by setting $u_i^N = w_i^N$. As a result, we define the following residual functions:

$$R_i^{nmp}\left(\boldsymbol{u}^N\right) = \chi_{\Omega_i}^{nmp} \left(L_i^{nmp} + b_i^{nmp}\right); \text{for} i = 1, 2, 3 \tag{89}$$

One can choose an appropriate iterative solver to solve $\boldsymbol{R}\left(\boldsymbol{u}^N\right) = \boldsymbol{0}$ for $\boldsymbol{u}^N$. For the static example in Section 5.1, we use a conjugate gradient method [61] to solve for the displacement field.

4.4. Discussion on the accuracy and the spectral description of the method

The fast convolution based method introduced in this study for elasticity and fracture, and in [27] for diffusion problems, can be viewed as a meshfree discretization of peridynamics equations with FFT-accelerated quadrature. FCBM can also be viewed as a Fourier spectral method where the inner products with the Fourier basis functions are computed by quadrature approximation. In the Fourier spectral description of FCBM, one can assume the Fourier series expression of the convolving functions in Eq. (14) and write:

$$\begin{aligned} \int_{\mathcal{H}_x} f\left(\boldsymbol{x}, \boldsymbol{x}', t\right) \, dV_{\boldsymbol{x}'} &= \sum_{n=1}^{p} a_n(\boldsymbol{x}, t) \left[b_n * c_n\right](\boldsymbol{x}, t) \\ &= \sum_{n=1}^{p} a_n(\boldsymbol{x}, t) \mathcal{F}^{-\mathbf{1}} \left[L_1 L_2 L_3 \mathcal{F}(b_n) \mathcal{F}(c_n)\right](\boldsymbol{x}, t) \end{aligned} \tag{90}$$

where $\mathcal{F}$ and $\mathcal{F}^{-\mathbf{1}}$ are the exact Fourier transform and its inverse defined as:

$$\begin{aligned} \mathcal{F}(M) &= \frac{1}{L_3 L_2 L_1} \int_{\mathbb{T}} M(\boldsymbol{x}, t) e^{-2\pi\zeta\left(\frac{k_1 x_1}{L_1} + \frac{k_2 x_2}{L_2} + \frac{k_3 x_3}{L_3}\right)} \mathrm{d}x_1 \mathrm{d}x_2 \mathrm{d}x_3 \\ \mathcal{F}^{-\mathbf{1}}\left(\hat{M}\right) &= \sum_{k_3, k_2 k_1 = -\infty}^{+\infty} \hat{M}(\boldsymbol{k}, t) e^{2\pi\zeta\left(\frac{k_1 x_1}{L_1} + \frac{k_2 x_2}{L_2} + \frac{k_3 x_3}{L_3}\right)} \end{aligned} \tag{91}$$

In Eq. (90), if we truncate the Fourier series at N_1, N_2, N_3 modes in the three Cartesian directions, and approximate the $\mathcal{F}$ integrals via midpoint quadrature, using the uniform grid spacing described by Eq. (52), we recover the equations given in Section 4.1.1, having FFT-based quadrature operations.

Recall that, in general, Fourier spectral discretizations show spectral accuracy (exponential convergence rate) if the solution is smooth on $\mathbb{T}$, and if the Fourier multipliers [14] (Fourier transform of the kernels of PD operator, i.e. $\mathcal{F}(c_n)$ in Eq. (90)) are computed exactly. For example, the Fourier multipliers for the PD Laplacian operator

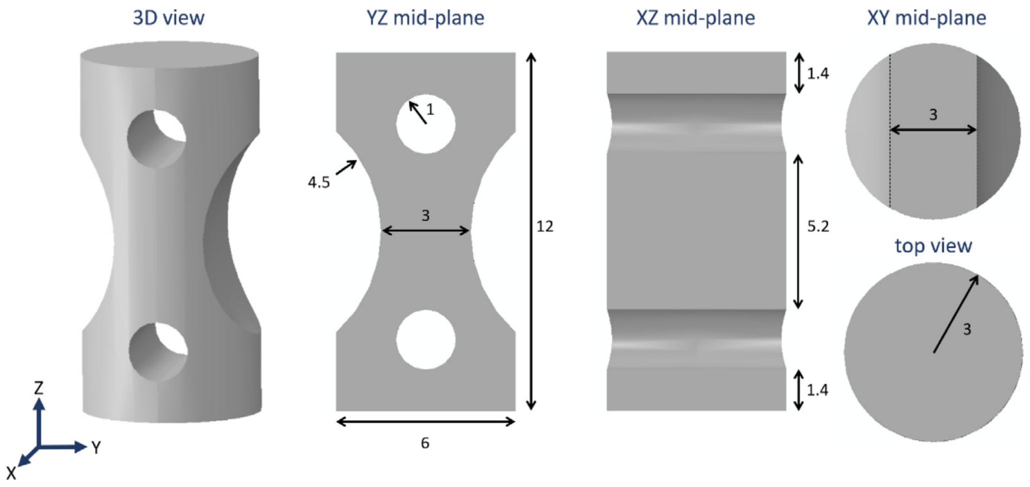


Fig. 6. The specimen's 3D configuration, YZ and XY mid-planes' cross-sectional views, and the top/bottom view. Dimensions are given in centimeters.

can be expressed in terms of hypergeometric functions [14,20]. While such analytical formulas for the Fourier multipliers lead to spectral accuracy, they depend on the kernels' forms and may not be always easy to find. Note also that evaluating these analytical relationships can be challenging (see, e.g., [20]). The quadrature approximation used in this study bounds the FCBM accuracy to that of the quadrature, but it is a general approach and can be easily used for any given kernel. Moreover, the EC method we use for incorporating boundary conditions leads to working with Fourier transforms of non-smooth functions (e.g. characteristic functions), which automatically would drop the exponential convergence rate even when analytical formulas for the Fourier multipliers are known.

The convergence rate of FCBM in terms of spatial discretization size is expected to be polynomial, bounded by quadrature's accuracy and the rate of convergence for the Fourier series approximation of transformed functions. In case of certain diffusion problems for example, the FCBM spatial rate of convergence has been shown to be quadratic [27]. We show that FCBM's spatial rate of convergence for a 3D linear elasticity example in Section 5.1 is superlinear. Rigorous error estimates and numerical analysis for FCBM is of significant interest and will be studied in the future.

Another concern with spectral methods is that aliasing errors might reduce the accuracy when a Fourier transform of a product operation is involved. We did not encounter noticeable aliasing errors in the examples shown in this study (see the convergence study in Section 5.1). However, if these errors become significant in a particular problem, one can use de-aliasing techniques such the 2/3rd rule [62] to remove them.

5. Numerical examples

In this section we solve two example problems via FCBM-EC. First, we solve a 3D example for an elastic deformation of a relatively complex geometry under static loading, using the state-based PD model. This allows us to verify our FCBM-EC static formulation, and also the convolutional form of the state-based model. Then, we solve a 2D brittle fracture problem where we verify our new damage model, as well as the FCBM-EC formulation for PD dynamic problems, with the linearized bond-based model.

5.1. Elastostatic deformations in a 3D body with a complex shape

In this example we compute the elastic deformation of a 3D semi-cylindrical dog bone specimen with two lateral thorough holes subjected to a static uniaxial tension. Fig. 6 shows the specimen's 3D configuration, mid-plane cross-sections and the top view, with the dimensions. The Young modulus and Poisson ratio considered are $E = 60$ GPa and $\nu = 0.4$.

We select the horizon size to be $\delta = 0.3$ (sufficiently small relative to the size of the drilled holes, the smallest geometrical feature of the sample, see [63]) and extend the specimen's geometry by δ from the top and the bottom

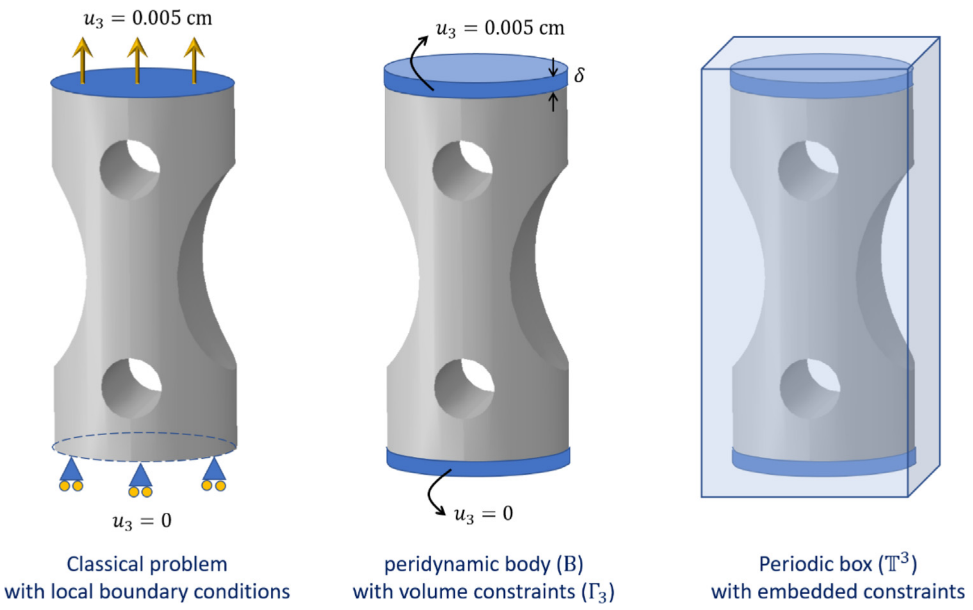

$u_3 = 0.005$ cm
$u_3 = 0$
Classical problem with local boundary conditions
$u_3 = 0.005$ cm
δ
$u_3 = 0$
peridynamic body (B) with volume constraints (Γ_3)
Periodic box ($\mathbb{T}^3$) with embedded constraints

Fig. 7. Left: Specimen's configuration with local boundary conditions; Middle: PD body B with Γ_3 extensions for applying volume constraints/nonlocal BC; Right: further extension to the periodic box $\mathbb{T}^3$ for discretization with FCBM-EC.

as the fictitious domain for enforcing volume constraints. These extensions are referred to as Γ_3, because u_3 is specified on them. Then, we further extend the PD body B (including Γ_3) in all directions by δ to form a box of size $6.6 \times 6.6 \times 14.2\text{cm}^3$ as the periodic domain $\mathbb{T}^3$ (see Fig. 7).

We perform two PD simulations using two different types of nonlocal BC for each on the top and bottom boundary layers. In one simulation we set $u_3 = 0.005$ cm for all the points in Γ_3 on the top, and $u_3 = 0$ for all the points in the bottom layer. This approach is referred to as the naïve fictitious nodes method (naïve FNM) [56], and is one of the most convenient methods to determine volume constraints for approximating local BCs. In the second simulation we use the mirror-based FNM [56,64] to define the volume constraints on Γ_3. In mirror-based FNM volume constraints on each point of Γ_3 are determined from the desired local BC value and the solution value on an interior Ω point, referred to as the "mirror" point. The description of mirror-based FNM is provided in the appendix. This approach is a better approximation of local BCs compared with the naïve FNM. However, its implementation is not as easy as the naïve approach, since the mirror-based FNM requires updating the volume constraints at each time step/iteration according to the computed solution at the previous time step/iteration. Mirror-based FNM are also more difficult to implement for curved geometries since one needs to know the normal to the surface along the boundaries. For more details please see [56].

The rest of the body's surface, which is traction-free, is treated the same in all PD simulations here, which is by eliminating nonlocal interactions between B and the rest of the box using the χ_B characteristic function (see Section 4.2.2 and Eq. (73)). This is identical to naïve FNM for traction-free conditions [56].

Considering the state-based linear elastic model in Eq. (38) and the modification in Eq. (75) to remove the nonlocal interactions between the body B with the rest of the box (Λ), the PD operator is expressed as:

$$\begin{aligned} L_i = \int_{\mathcal{H}_x} \chi_B \chi_B' f_i \mathrm{d}V_{x'} &= \int_{\mathcal{H}_x} \chi_B \chi_B' \left[a_i \left(\boldsymbol{x}' - \boldsymbol{x}\right)\left(\vartheta + \vartheta'\right) + \mathrm{C}_{ij}\left(\boldsymbol{x}' - \boldsymbol{x}\right)\left(u_j' - u_j\right)\right] \mathrm{d}V_{x'} \\ &= \chi_B \left\{ \vartheta \int_{\mathcal{H}_x} -a_i \left(\boldsymbol{x} - \boldsymbol{x}'\right) \chi_B' \mathrm{d}V_{x'} - \int_{\mathcal{H}_x} a_i \left(\boldsymbol{x} - \boldsymbol{x}'\right) \chi_B' \vartheta' \mathrm{d}V_{x'} \right. \\ &\quad \left. + \int_{\mathcal{H}_x} \mathrm{C}_{ij} \left(\boldsymbol{x} - \boldsymbol{x}'\right) \chi_B' u_j' \mathrm{d}V_{x'} + \left(\int_{\mathcal{H}_x} \mathrm{C}_{ij} \left(\boldsymbol{x}' - \boldsymbol{x}\right) \chi_B' \mathrm{d}V_{x'} \right) u_j \right\} \\ &= \chi_B \left[-\vartheta \left(a_i * \chi_B\right) - \left(a_i * \chi_B \vartheta\right) + \left(\mathrm{C}_{ij} * \chi_B u_j\right) + \left(\mathrm{C}_{ij} * \chi_B\right) u_j \right] \end{aligned} \tag{92}$$

Note that the disconnection between B and Λ via χ_{B} must be considered in computing ϑ and m as well, for consistency. Therefore, we modify Eq. (39) as follows:

$$\begin{aligned}\vartheta &= \frac{3}{m}\int_{\mathcal{H}_x} \chi_{\mathrm{B}}\chi'_{\mathrm{B}}\omega\left(|\boldsymbol{\xi}|\right)\xi_i\eta_i \mathrm{d}V_{\boldsymbol{x}'} = \frac{-3\chi_{\mathrm{B}}}{m}\int_{\mathcal{H}_x} a_i\left(\boldsymbol{x}-\boldsymbol{x}'\right)\chi'_{\mathrm{B}}\left(u'_i-u_i\right)\mathrm{d}V_{\boldsymbol{x}'} \\ &= \frac{-3\chi_{\mathrm{B}}}{m}\int_{\mathcal{H}_x} a_i\left(\boldsymbol{x}-\boldsymbol{x}'\right)\chi'_{\mathrm{B}}u'_i \mathrm{d}V_{\boldsymbol{x}'} - \frac{3}{m}\left(\int_{\mathcal{H}_x} a_i\left(\boldsymbol{x}'-\boldsymbol{x}\right)\chi'_{\mathrm{B}}\mathrm{d}V_{\boldsymbol{x}'}\right)u_i \\ &= \frac{-3\chi_{\mathrm{B}}}{m}\left[\left(a_i * \chi_{\mathrm{B}}u_i\right)+\left(a_i * \chi_{\mathrm{B}}\right)u_i\right]\end{aligned} \tag{93}$$

where

$$m = \int_{\mathcal{H}_x} \chi_{\mathrm{B}}\chi'_{\mathrm{B}}\omega\left(|\boldsymbol{\xi}|\right)|\boldsymbol{\xi}|^2 \mathrm{d}V_{\boldsymbol{x}'} = \chi_{\mathrm{B}}\int_{\mathcal{H}_x}\omega\left(\boldsymbol{x}-\boldsymbol{x}'\right)\left|\boldsymbol{x}-\boldsymbol{x}'\right|^2\chi'_{\mathrm{B}}\mathrm{d}V_{\boldsymbol{x}'} = \chi_{\mathrm{B}}\left(\omega\left|\boldsymbol{x}\right|^2 * \chi_{\mathrm{B}}\right) \tag{94}$$

Similar to L_i, including $\chi_{\mathrm{B}}\chi'_{\mathrm{B}}$ in calculation of ϑ and m preserves the convolutional structure of the integrals and allows one to compute them by FFT at the cost of $O\left(N\log_2 N\right)$.

To perform the simulations, we need to choose an influence function ω. Here, we use $\omega = \frac{1}{|\xi|}$ as one of the most used options [2,65].

We discretized the domain considering $N_1 = N_2 = 2^7$ and $N_3 = 2^8$, in the three Cartesian directions. The m-factor (ratio of horizon to grid size) for this example happens to be the same for all directions: $\delta/\Delta x_1 = \delta/\Delta x_2 = \delta/\Delta x_3 = 5.82$. Note that the m-factor can be different along different directions, if grid spacing is not identical in all directions (see the example in Section 5.2).

Following the FCBM discretization described in Section 4.1.1, we can write the discretized version of the equations above as:

$$\begin{aligned}L_i^{nmp} = \chi_{\mathrm{B}}^{nmp}\Big\{&-\vartheta^{nmp}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(\chi_{\mathrm{B}}\right)\right]\Big|^{nmp} - \mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(\chi_{\mathrm{B}}\vartheta\right)\right]\Big|^{nmp} \\ &+ \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\chi_{\mathrm{B}}u_j\right)\right]\Big|^{nmp} + \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\chi_{\mathrm{B}}\right)\right]\Big|^{nmp}u_j^{nmp}\Big\}\Delta V\end{aligned} \tag{95}$$

where

$$\vartheta^{nmp} = -\frac{3\chi_{\mathrm{B}}^{nmp}}{m^{nmp}}\left\{\mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(\chi_{\mathrm{B}}u_i\right)\right]\Big|^{nmp} + \mathbf{F}^{-1}\left[\mathbf{F}\left(a_i\right)\mathbf{F}\left(\chi_{\mathrm{B}}\right)\right]\Big|^{nmp}u_i^{nmp}\right\}\Delta V \tag{96}$$

and

$$m^{nmp} = \chi_{\mathrm{B}}^{nmp}\,\mathbf{F}^{-1}\left[\mathbf{F}\left(\omega\left(x_1^2+x_2^2+x_3^2\right)\right)\mathbf{F}\left(\chi_{\mathrm{B}}\right)\right]\Big|^{nmp}\Delta V \tag{97}$$

In order to use the FCBM-EC formulation for static problems given by in Eq. (89), we construct χ_{Ω_i} from Eq. (74), knowing that $\Omega_1 = \Omega_2 = \mathrm{B}$, and $\Omega_3 = \mathrm{B}\backslash\Gamma_3$. In this study, we use a conjugate gradient algorithm to solve Eq. (89).

We also solve the local version of this problem using a commercial finite element package. To this aim, we use Abaqus/Standard 6.19-1 solver with over 1×10^7 D3D4 linear tetrahedral elements and about 2×10^6 nodes, which is about the same number of degrees of freedom in Ω_{B} in the FCBM-EC simulation.

All the simulations in this study are performed on a Dell-Precision T7910 workstation PC, Intel(R) Xeon(R) CPU E5-2643 W v4 @3.40 GHz logical processors, and 128 GB of installed memory.

Fig. 8 shows the displacement field obtained by FCBM-EC (using the Naïve FNM boundary conditions) and by Abaqus in 3D. A rotating view of the vertical displacement field (u_3) obtained by FCBM is shown in Video 1.

As we observe from the displacement contours, the FCBM-EC solution of the PD model and FEM solution of the classical model are visually very close. To investigate the difference quantitatively, we plot the absolute relative difference between the two solutions in the cross-sectional views in Fig. 9. The plotted differences are computed from:

$$E_i^{nmp} = \frac{\left|\left(u_i^{nmp}\right)_{\mathrm{FCBM}} - \left(u_i^{nmp}\right)_{\mathrm{Abaqus}}\right|}{\max_{\text{all nodes}}\left(u_i^{nmp}\right)_{\mathrm{Abaqus}}} \tag{98}$$

where E_i^{nmp} is the absolute relative difference of u_i^{nmp}. We intentionally use "difference" instead of "error" for referring to E_i^{nmp}, because the governing equations in Abaqus are different from PD (one local and one nonlocal).

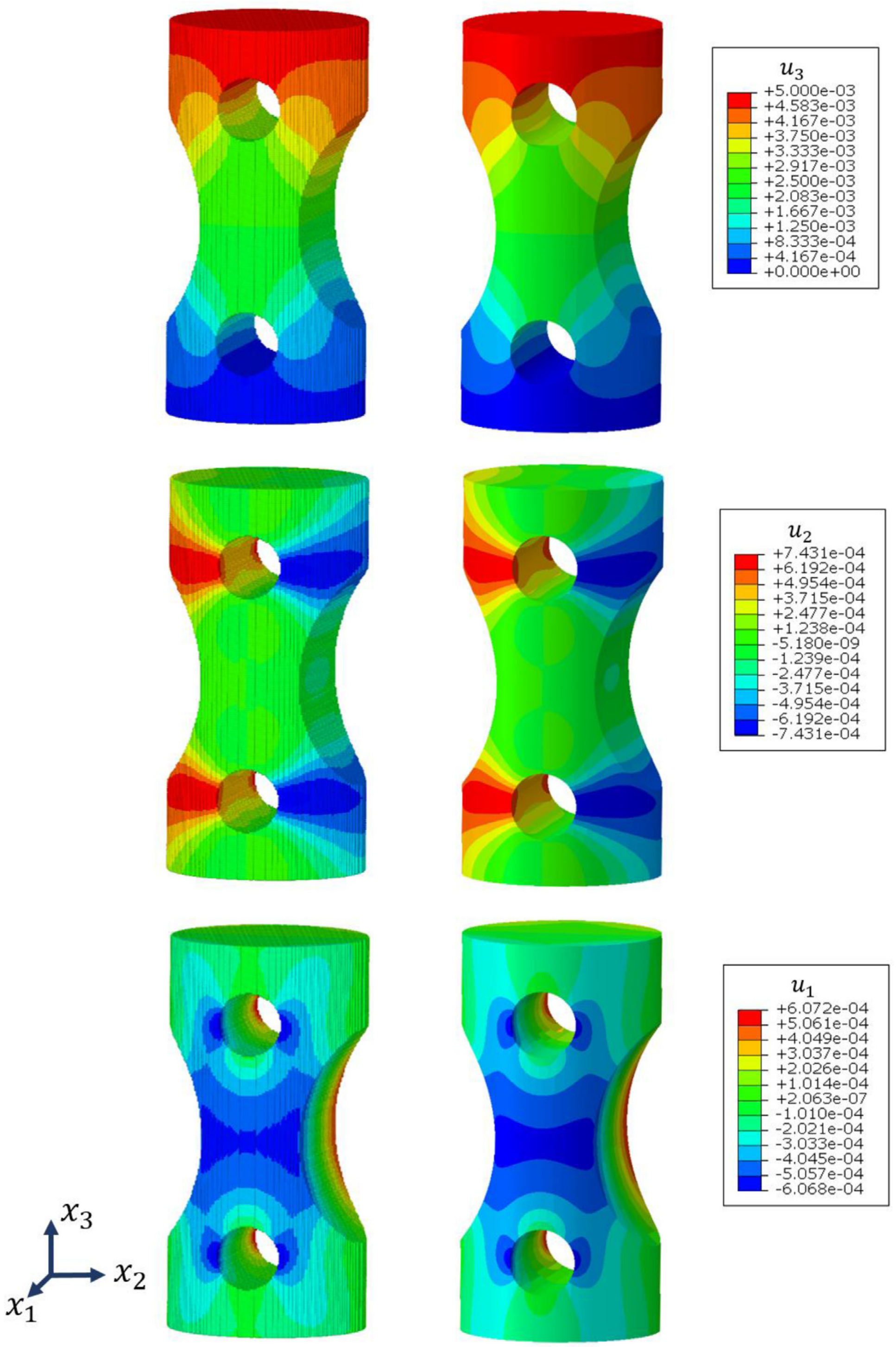


Fig. 8. Displacement field (u_1, u_2, u_3) in 3D, obtained from simulations of the static tension problem using FCBM-EC discretization of peridynamics (left) and finite element solution of the classical equilibrium by Abaqus static analysis (right).

Therefore, the difference in the results is not only attributed to the numerical methods' error, but it also originates from the nature of the governing equations.

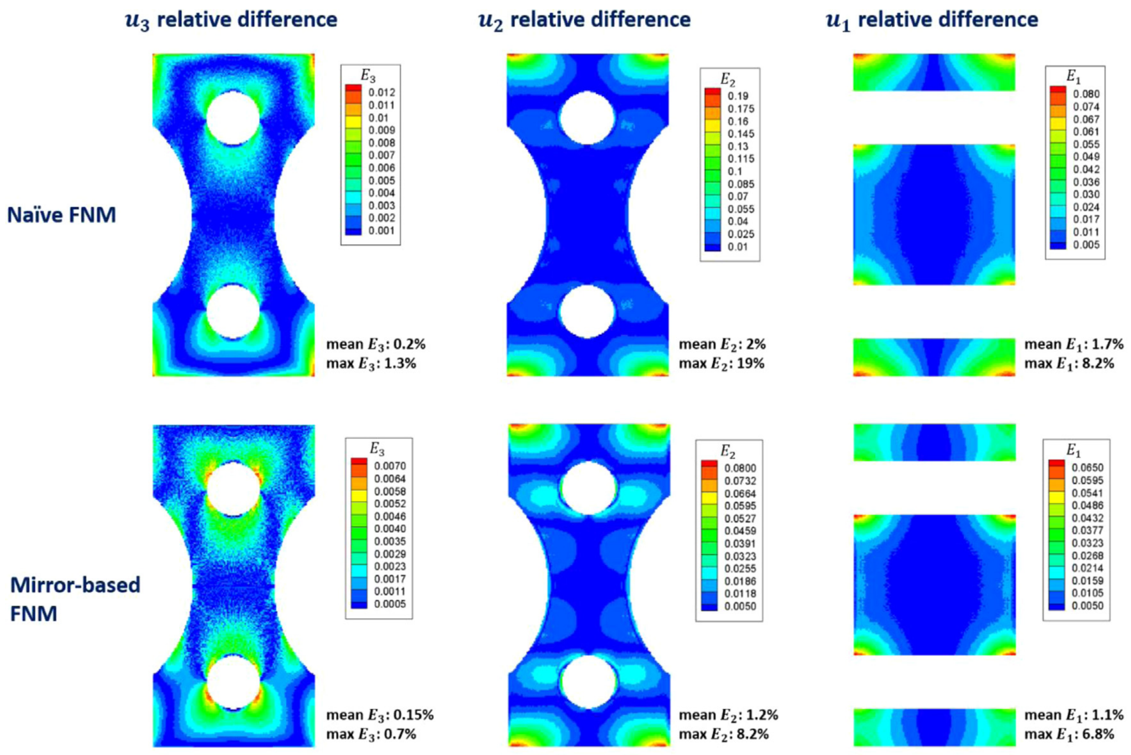


Fig. 9. Absolute relative difference plotted in cross-sectional views for displacement fields obtained via FCBM-EC in comparison with the solutions from the Abaqus static analysis (FEM solution of classical equilibrium). Top row: FCBM solver used Naïve fictitious nodes method (FNM) for boundary conditions; Bottom row: mirror-based FNM was used for the top and the bottom boundaries. Differences for u_3 and u_2 are plotted in YZ mid-plane and u_1 is plotted in XZ mid-plane cross-section (see Fig. 6).

Note that the rectangular gaps present in the right column of Fig. 9 correspond to the holes in the sample. For those figures we used a different cross-sectional view than for the other two sets of plots in Fig. 9, namely the XZ mid-plane (see Fig. 6). The reason for this selection is that along the YZ mid-plane, which is used to plot the displacement differences for u_2 and u_3, the u_1 displacement is zero.

The maximum difference between PD and Abaqus solutions, as expected, is observed at the corners, and this is the result of the differences between local and nonlocal BCs. Using mirror-based FNM on the top and bottom boundaries has reduced the maximum difference between the local and the nonlocal solutions associated with BCs at the corners. Mean differences for FCBM PD solutions with either FNMs, remain below 2% compared the FEM (Abaqus) solution for the classical model.

Next, we compare the computational time between the FCBM-EC and the meshfree PD method (that uses direct quadrature) for the same PD problem, using various discretization sizes. To this aim, we set $N = 2^{19}, 2^{22}, 2^{25}, 2^{28}$ for the FCBM simulation. Note that the N here is the total number of nodes within the box $\mathbb{T}$. The number of nodes within the body B with these numbers are $N_{\mathrm{B}} = 0.25, 2, 15.9, 127.2$ million nodes, respectively. To have a fair comparison, we use N_{B} nodes for the meshfree PD simulations, since $N - N_{\mathrm{B}}$ number of nodes locate within the gap region in FCBM-EC, which are not needed in the meshfree PD method. We use naïve FNM for enforcing BCs on both the meshfree and the FCBM-EC simulations.

Table 1 compares the computational time required to perform the simulations using FCBM-EC and the meshfree PD method. We have also taken advantage of MATLAB's built-in multi-threaded and GPU FFT functions. We performed our FCBM simulations on 1CPU, 8CPUs (using multi-threaded FFT), and on a GPU (using GPU enabled FFT) in separate tests. Multithreaded computation on more than 8 CPUs did not lead to further improvements.

Although the extra space in the FCBM-EC computational box has almost twice the number of nodes used by the meshfree PD for the actual domain, FCBM-EC simulations are significantly faster. Due to the difference between

Table 1
Computational time for solving the 3D example, using FCBM-EC and the meshfree PD with direct quadrature (DQ).

m-factor	∼3	∼6	∼12	∼23
Number of nodes (N_B)	∼250 × 10^3	∼2 × 10^6	∼16 × 10^6	∼127 × 10^6
Meshfree PD with DQ (1 CPU)	1 h	2.8 days[a]	5.9 months[a]	31.1 yrs[a]
FCBM-EC (1 CPU)	5.5 min	53 min	7.5 h	3.9 days
FCBM-EC (8 CPUs)	2 min	16.9 min	2.5 h	1.4 days
FCBM-EC (GPU)	2.7 min	9.9 min	out of memory	out of memory

[a]Time is estimated, using the time for $m = 3$ and knowing that the computational time scales in O(N^2) for the method with direct quadrature [27].

Table 2
Memory allocation required for solving the 3D example, using FCBM-EC and the meshfree PD with direct quadrature (DQ).

m-factor	∼3	∼6	∼12	∼23
Number of nodes in the body	∼250 × 10^3	∼2 × 10^6	∼16 × 10^6	∼127 × 10^6
Meshfree PD with DQ	5.1 GB	327 GB[a]	20.4 TB[a]	1,308 TB[a]
FCBM-EC	235 MB	1.8 GB	14.7 GB	117.5 GB

[a]Estimated, using the memory required for the case with $m = 3$, and knowing that memory allocation scales as O(N^2) for the method with direct quadrature [23]. The memory of the computer system here was 128 GB.

the computational complexity of the methods, as the number of nodes increases, the efficiency gain of FCBM-EC becomes higher. The new method reduces PD computations from hours to minutes and from years to days. From Table 1, computations that were impossible to conduct using a single CPU with the meshfree PD method, are now easily achievable via FCBM-EC. Bulit-in Matlab's FFT functions allowed us to benefit from parallel computations by adding a few lines of codes in the FCBM solver. In this example, using more than 8 CPUs does not lead to much more efficiency.

Another significant advantage of FCBM-EC compared with the meshfree PD is the memory allocation. In simulations provided in Table 1, the memory allocation required by the meshfree PD only allowed for the lowest resolution ($N_B = 250 \times 10^3$). The computational time for higher spatial resolutions are estimated, knowing that they scale with O(N^2) for the meshfree PD [27]. Most meshfree implementations initially find and store family nodes for each node to improve efficiency during the quadrature [66,67]. Without this initialization, family search is required for each node at each time step/iteration which can potentially add a significant amount of time. However, storing family information requires allocating arrays of $N \times M$ where M is the maximum number of family nodes. Knowing that for a fixed δ, M scales with N, memory allocation in the meshfree PD scales with O(N^2). In contrast, FCBM-EC does not need to search and store family information. The integrals are computed in Fourier space, without explicitly adding individual interactions between family nodes. As a result, the variables in FCBM are arrays of size of $N \times 1$ and their storage scales as O(N). Table 2 shows the memory allocation required by FCBM and the meshfree PD for our 3D example.

We also plot the FCBM convergence of displacement field with respect to the spatial discretization (*m*-convergence, for the horizon mentioned in the problem setup above) for this 3D elasticity problem. Since the exact analytical nonlocal solution for this 3D example is not known, we use the following error measure for the convergence study:

$$\text{relative error} = \frac{|\boldsymbol{U}_{\text{new}} - \boldsymbol{U}_{\text{old}}|_{L_2}}{|\boldsymbol{U}_{\text{new}}|_{L_2}} \tag{99}$$

where $\boldsymbol{U}$ is the nodal displacements vector: a vector containing displacements in the three directions for all nodes (u_i^{nmp} for all i, n, m, p). The "old" and "new" subscripts denote two simulations with $\Delta x_{\text{new}} = \frac{1}{2}\Delta x_{\text{old}}$. We use the coarser grid nodal coordinates (from the "old" simulation) to compute the L_2 norms in Eq. (99). Again, we have used naïve FNM for enforcing BCs in these simulations. Notice that the boundary conditions treatment affects convergence rates, but in this manuscript we did not study this issue.

In Fig. 10 we plot this relative error versus the discretization size (Δx) in log–log scale.

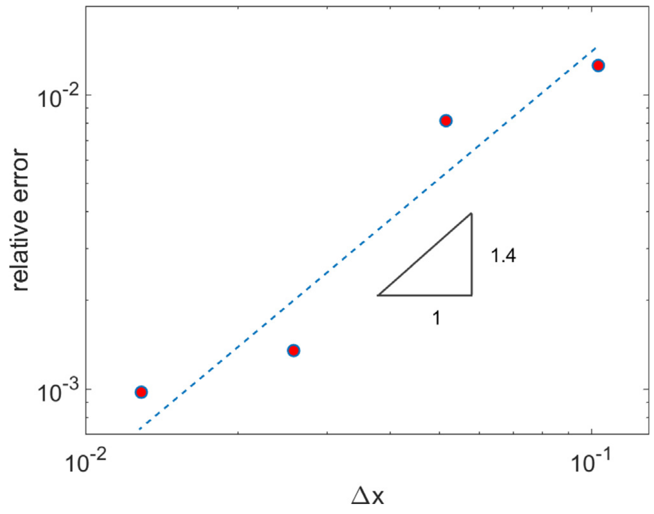


Fig. 10. Convergence of FCBM-EC solution for displacements with respect to spatial discretization size for the 3D elasticity example. The dashed line is a linear fit of the data.

We observe a superlinear rate of convergence for this 3D elasticity example. This rate is similar to what was reported in [68], where an alternate version of this method (FCBM with volume penalization [26]) was used for a PD wave equation with discontinuous initial conditions in 2D. As discussed in Section 4.4, the accuracy of the FCBM in the present form is bounded by the quadrature and the finite Fourier series approximation errors.

In the next section, we solve a 2D fracture example and compare the results with the published meshfree PD solution to show the new method's efficiency and versatility in modeling dynamics brittle fracture problems, including those that involve multiple crack branching events.

5.2. Dynamic brittle fracture and crack branching in 2D

In this section we test the damage model introduced in Section 3.5 and its FCBM-EC implementation and verify it against published dynamic brittle fracture simulations obtained by the meshfree PD model that uses critical bond-strain for bond breaking [36].

The problem description is as follows: A thin plate of 10×4 cm ^{2} soda-lime glass with a precrack of length 5 cm is subjected to sudden stresses that are distributed uniformly on the top and bottom boundaries and remain constant during the simulations. The sample geometry and loading conditions are shown in Fig. 11.

(b) Peridynamics problem

In the peridynamic version of the problem (Fig. 11b), we impose the body force density $b = \sigma/\delta$ on the δ-thick top and bottom layers of the body to enforce the desired load-controlled BCs. As noted earlier the method to impose/approximate local BCs in PD models is not unique and one may choose other approaches [56,57,69]. In this example, we used the body force approach, in order to have a similar loading as in [36], but our method can handle other approaches as well. Because of the through crack definition in the new damage model (see Section 3.5.3), the pre-crack in our PD model is thicker than the one used in [36] with the meshfree PD. The influence of the crack width difference in the two models is discussed after presenting the results.

The glass has the Young modulus $E = 72\,\text{GPa}$, density $\rho = 2440\,\text{Kg/m}^3$, and critical fracture energy (critical energy release rate) $G_0 = 3.8\,\text{J/m}^2$ [36]. We conduct three simulations with three different stress values: $\sigma = 0.2, 2, 4\,\text{MPa}$. These three tests are shown to result in different fracture patterns and branching behavior in [36]. We choose $\delta = 0.1$ cm as in [36].

In order to use FCBM-EC, we first extend the domain by δ in all directions to construct the periodic box (see Fig. 12).

Note that unlike the example in Section 5.1, there are no fictitious domain in this simulations since we apply the load as a body force directly on the top and the bottom layers of the plate. Considering the damage model in

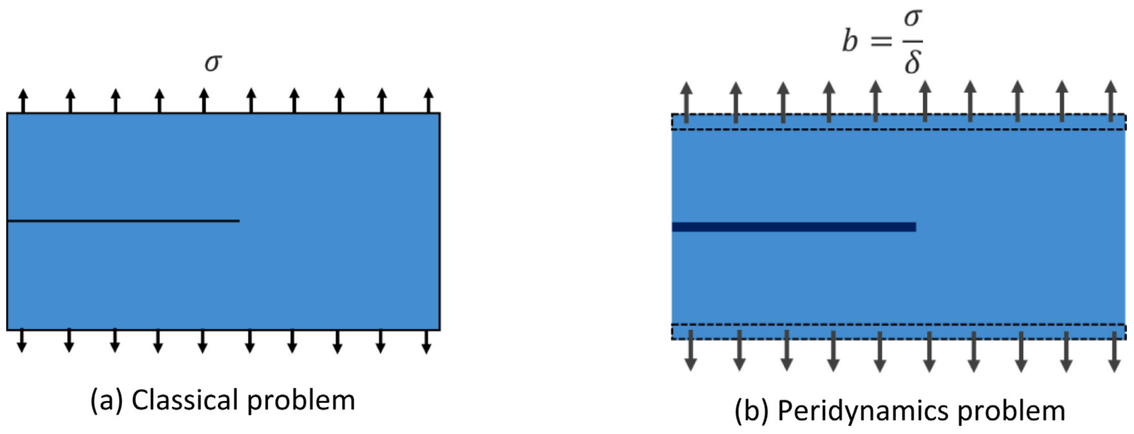


Fig. 11. Classical problem description (a), and the corresponding peridynamics version (b).

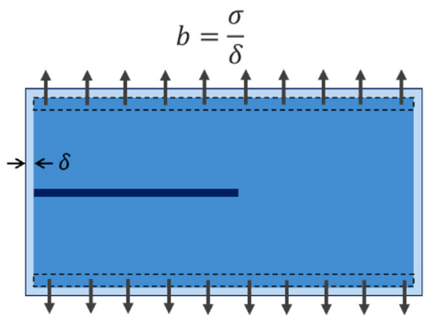


Fig. 12. Domain extension to be used in the FCBM-EC PD solver.

Eq. (46) and the modification in Eq. (75) to disconnect the body from the rest of the gap region, the PD operator can be expressed as:

$$L_i = \int_{\mathcal{H}_x} \chi_B \chi'_B \lambda \lambda' f_i \mathrm{d}V_{x'} = \chi_B \lambda \int_{\mathcal{H}_x} \chi'_B \lambda' \mathrm{C}_{ij} \left(\boldsymbol{x} - \boldsymbol{x}'\right) \left(u'_j - u_j\right) \mathrm{d}V_{x'}$$
$$= \chi_B \lambda \left\{\left[\mathrm{C}_{ij} * \left(\chi_B \lambda u_j\right)\right] - u_j \left[\mathrm{C}_{ij} * \chi_B \lambda\right]\right\} \tag{100}$$

To compute C_{ij}, we need to choose an influence function ω and calibrate the constant α to the classical elasticity constants. We use a common influence function $\omega = \frac{1}{|\xi|}$, which results in $\alpha = \frac{9E}{\pi\delta^3}$ for plane stress (Poisson ratio is restricted to $\nu = \frac{1}{3}$) [2].

We discretized the domain considering $N_1 = 2^9$ and $N_2 = 2^8$. The m-factor (ratio of horizon to grid size) is then $\delta/\Delta x_1 = 5.02$ and $\delta/\Delta x_2 = 6.10$ in x_1 and x_2 directions respectively. By comparing Eq. (100) with Eq. (46), one can write the discretized version by replacing λ with $\chi_B\lambda$ in Eq. (62):

$$L_i^{nmp,t} = \chi_B^{nmp} \lambda^{nmp,t} \left\{ \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\chi_B \lambda u_j\right)\right]\Big|^{nmp,t} - \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\chi_B\lambda\right)\right]\Big|^{nmp,t} u_j^{nmp,t} \right\} \Delta V \tag{101}$$

To conduct the simulations, we use the FCBM-EC formulation for dynamic problems given by Eqs. (83) and (84). We choose a time step of $\Delta t = 5 \times 10^{-8}\, s$ which is the same as the one used by the meshfree PD simulations in [36]. In general, one can choose any Δt that satisfies the stability condition given in [5]. At each time step $\lambda^{nmp,t+\Delta t}$ is updated using Eq. (63). This requires computing $W^{nmp,t}$. Similar to the process for obtaining the internal force

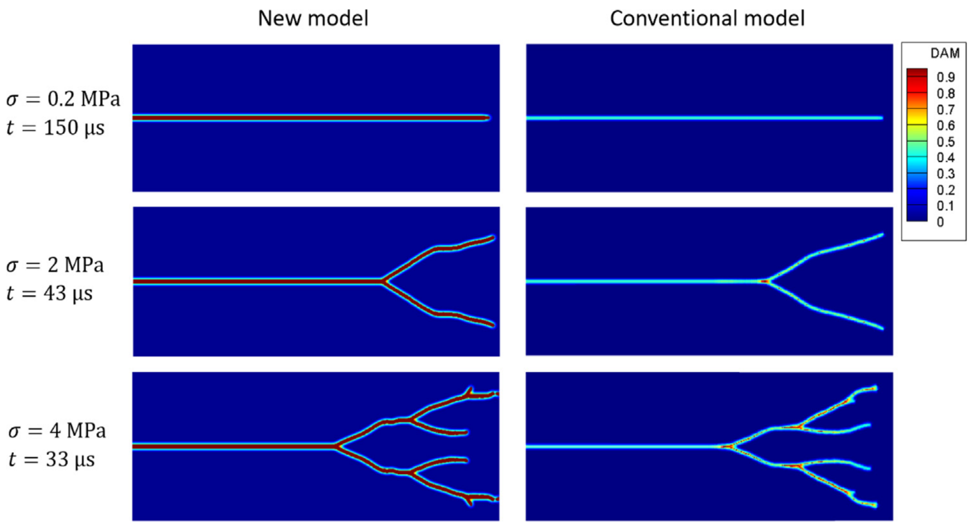


Fig. 13. Damage maps from PD simulations obtained from (left column) the new damage model (pointwise energy-based bond-breaking) solved via FCBM-EC, and from (right column) the meshfree PD solution with the critical bond-strain criterion [36].

density PD operator in Eq. (100), $W^{nmp,t}$ formula is found by replacing λ with $\chi_{\mathrm{B}}\lambda$ in Eq. (64):

$$\begin{aligned} W^{nmp,t} = & \frac{1}{2}\chi_{\mathrm{B}}^{nmp}\lambda^{nmp,t}\Delta V\,\Big\{\mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_i u_j \chi_{\mathrm{B}}\lambda\right)\right]\Big|^{nmp,t} \\ & - u_j^{nmp,t}\ \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_i \chi_{\mathrm{B}}\lambda\right)\right]\Big|^{nmp,t} \\ & - u_i^{nmp,t}\ \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(u_j \chi_{\mathrm{B}}\lambda\right)\right]\Big|^{nmp,t} \\ & + u_i^{nmp,t} u_j^{nmp,t}\ \mathbf{F}^{-1}\left[\mathbf{F}\left(\mathrm{C}_{ij}\right)\mathbf{F}\left(\chi_{\mathrm{B}}\lambda\right)\right]\Big|^{nmp,t}\Big\} \end{aligned} \tag{102}$$

Damage is computed from Eq. (65). Fig. 13 shows the crack patterns predicted by the new FCBM-EC PD model in comparison with the published results [36] obtained via the meshfree PD discretization of the bond-based model in Eq. (16) that uses a critical bond strain criterion, for three different loadings, at times when the cracks are about the fully split the sample.

The new damage model and solution method match, in each case, the crack patterns obtained by the meshfree PD model, including the multiple crack branching scenario. It takes the FCBM-EC model seconds to solve a problem that requires hours with the meshfree PD model.

The slight differences observed between the two simulations are expected because the two models are slightly different: the FCBM-EC discretizes the linearized bond-based PD model, while the meshfree PD solution is for the regular bond-based model; in addition, there is some difference between the damage models used. The meshfree PD simulations use the critical bond strain criterion while the FCBM solutions employ the new energy-based criterion. As discussed in Section 3.5.3, the new model shows cracks with "thickness" 3δ, with the middle δ-thick layer having fully failed nodes ($d = 1$), whereas the meshfree PD model leads to 2δ-thick cracks (see Fig. 3). This difference, however, does not seem to have a significant effect on the predicted fracture patterns. From Fig. 13 it is noticed that the meshfree PD results show a slight crack thickening just before crack branching while the FCBM results show constant thickness for all crack paths. However, a closer look at the fracture kinetics at the crack tip at the time of branching in the FCBM simulation, reveals that the "physics of branching" are similar between the two methods/models. Fig. 14 shows zoom-in views of several snapshot during the branching process in the test with $\sigma = 2$ MPa for the FCBM-EC PD model.

We observe that the three-phase process of branching reported in [36] is observed here as well. Phase I: propagation (Fig. 14a and b); Phase II: wave pile-up and thickening of damage (Fig. 14c and d); Phase III: damage migration and branching (Fig. 14e and f). Before branching (Fig. 14a and b), while the crack grows in a straight line, damage progression happens as follows: nodes near the crack tip get damaged, which leads to the removal of

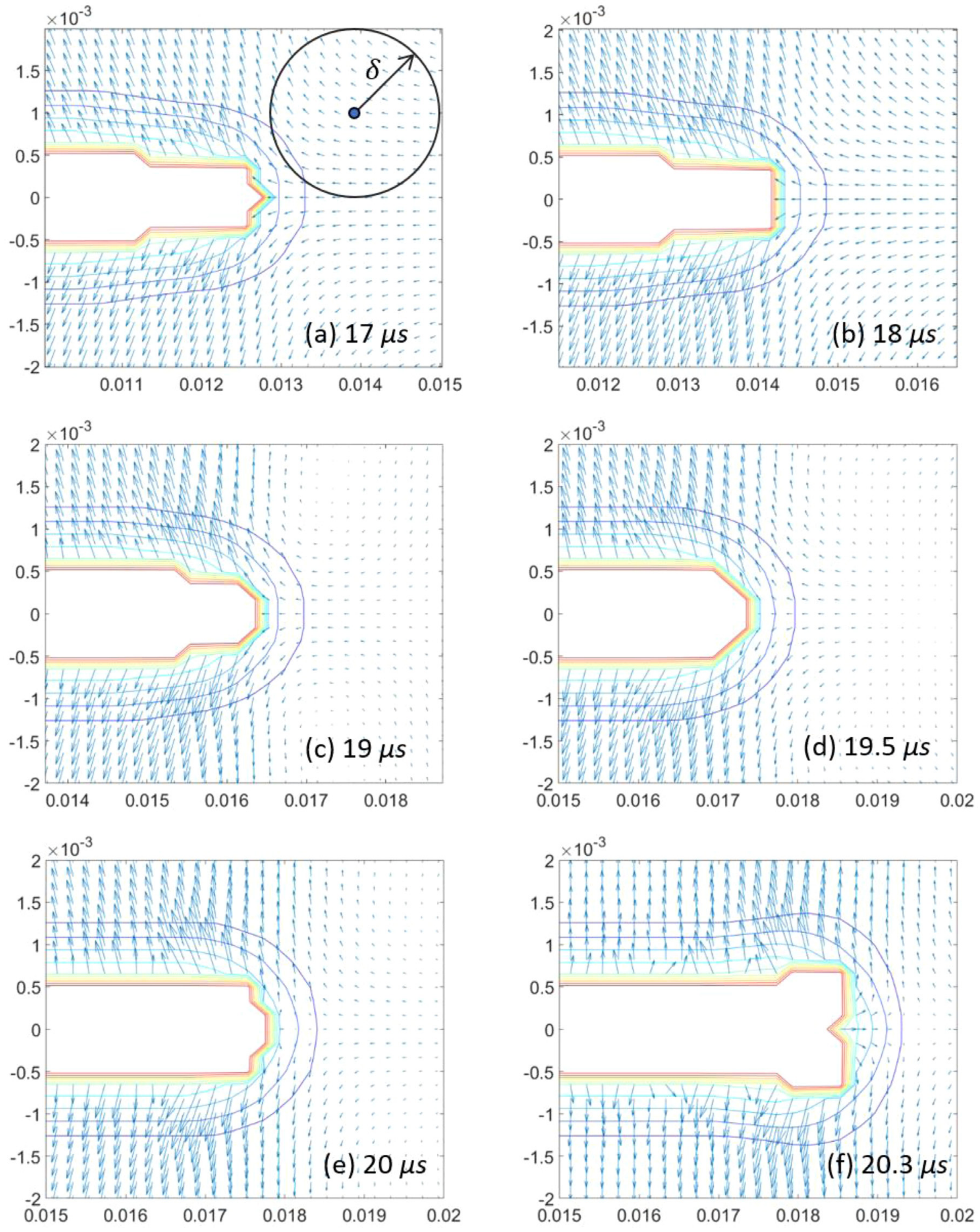


Fig. 14. Dynamics of crack branching around the crack tip with the new energy-based nodal damage model. Arrows show the velocity vector field while the contours show the damage index value. The damage contour colors are associated with the legend in Fig. 13. The PD horizon used, is plotted, to scale, in the top left picture.

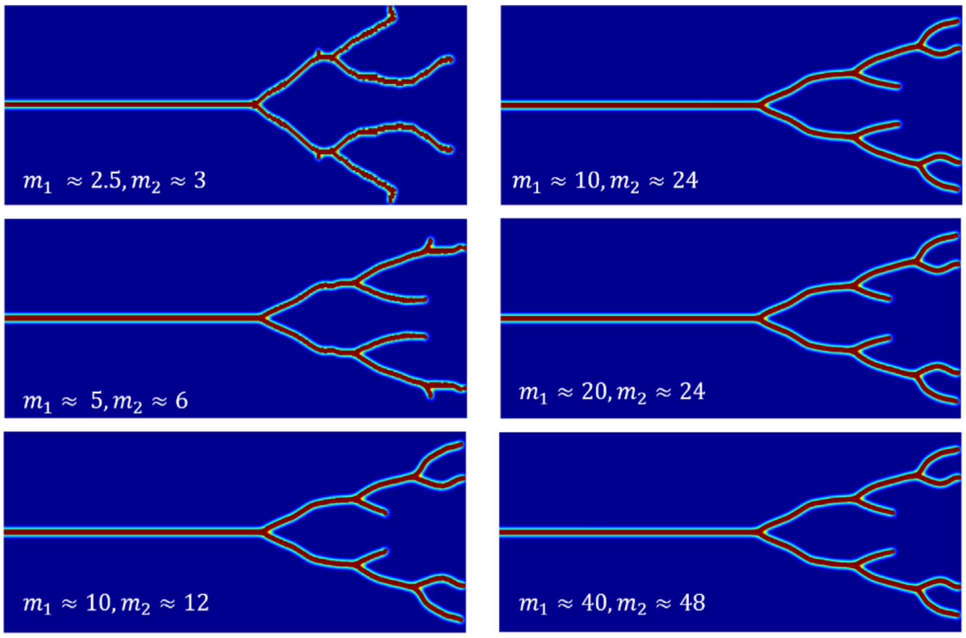


Fig. 15. An m-convergence study in terms of fracture patterns for the example with suddenly applied load $\sigma = 4\,\text{MPa}$.

all of their bonds. As the crack advances, bonds for nodes near the crack tip may still bridge the developing crack, and therefore increase the strain energy for these nodes. The strain energy density for nodes on the top and bottom crack surfaces with bridging bonds reach now the critical value and they also get removed, reaching damage index equal to one. This is why there is a widening of the crack past a horizon-length of the "process zone". This can continue as a steady state crack growth, but when the energy delivered at the crack tip is larger, then (Fig. 14c and d) the material in front of the crack tip moves more forcefully towards the advancing crack and leads to a pile-up of strain energy on the banks of the process zone, meaning that nodes on these banks reach critical strain energy not due to increased straining of their bridging bonds, but of all of their bonds: consequently, they fail sooner. This corresponds to the thickening of damage just before branching of a crack that is observed experimentally [70–72]. When this thickening reaches the tip of the process zone, (Fig. 14e and f), damage has to migrate from the center line and create the pathways for the crack to branch, since the strain energy concentration is no longer along the symmetry line, but at the corners of the thicker damage zone.

What we see, is that the FCBM PD model also captures the thickening of the crack just before branching, but this is partly obscured by the model's limitation to simulating cracks that are thinner than a thickness of 3δ, compared with the meshfree PD model in which the limit is 2δ (see Fig. 3).

One of the important advantages of FCBM, is that one can now easily perform m-convergence studies (refining the grid size with a fixed horizon size) up to high values of m, which was previously not practical, or even impossible, with the meshfree PD method. Note that m-convergence studies for fracture problems are particularly important since studies have shown that a grid independent crack path is obtained only when using larger m values [73] than what normally have been employed in most applications.

Here we performed an m-convergence study for the 2D fracture problem described above with $\sigma = 4\,\text{MPa}$. We kept the horizon fixed $\delta = 0.1$ cm, and changed the number of nodes in the x_1 and x_2 directions. Six simulations were conducted with $(N_1, N_2) = \left(2^7, 2^6\right)$, $\left(2^8, 2^7\right)$, $\left(2^{10}, 2^9\right)$, $\left(2^{10}, 2^{10}\right)$, $\left(2^{11}, 2^{10}\right)$, and $\left(2^{12}, 2^{11}\right)$, leading to the m-factors: $(m_1, m_2) \approx (2.5, 3)$, $(5, 6)$, $(10, 12)$, $(10, 24)$, $(20, 24)$, and $(40, 48)$ respectively. The case with $(m_1, m_2) \approx (10, 24)$ is chosen to test whether discretization anisotropy influences the results. Fig. 15 shows the m-convergence results.

We observe that fracture patterns converge at a value of $m \approx 10$. This is consistent with a study that concluded m-values higher than 7 are needed to have a correct crack path [73]. For m-values higher than 10, we also notice that grid spacing anisotropy does not alter the results.

In Fig. 16, we show how the simulations' run-time on a single CPU scales with the problem size.

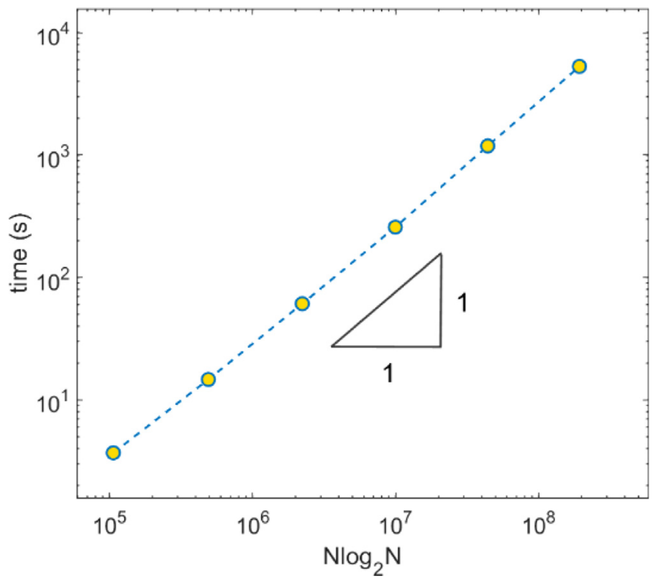


Fig. 16. Computational complexity of FCBM for this 2D fracture problem. Simulations' run-time is plotted versus $N\log_2 N$.

As expected, the simulations' time scales with $N\log_2 N$. The largest simulation here with $(m_1, m_2) \approx (40, 48)$ took only 88 min on a single CPU, while the meshfree PD solver would have needed several months to complete the same simulation (knowing that its run-time scales with $O\left(N^2\right)$). This is an important finding: with the new model, we can now efficiently solve problems that were not accessible/feasible before: dynamic brittle fracture problems in which the fracture patterns require high m-values.

Evolution of displacement and velocity fields, as well as strain energy density and damage index during fracture are provided in Video 2 for the converged case with $m \approx 10$. The velocity components (v_1 and v_2) in this video, show some wave emissions created, in spurts, along the newly formed crack surfaces as the cracks grow. We investigated the discretization effect on the frequency and amplitude of these wavelets and found that these features are independent of the discretization. They are dependent then only on the damage models used, including the horizon size. Effectively, these spurts are a result of the kinetics of bond-breaking in the damage model we used here: in a dynamic process, crack propagation happens one/a few nodes at the time; i.e. one or a few nodes at the crack tip suddenly lose all of their bonds when reaching their critical strain energy density, and instantly change the force balance for their family nodes, for the next time step. This node-by-node failure process, releases kinetic energy in small packets at the crack tip, causing these small fluctuations in the velocity field.

Experimental evidence confirms that crack growth in dynamic brittle fracture happens through consecutive bursts and not in a completely smooth fashion. This is seen in fracture experiments often, as Hull states in [74], page 262: "crack growth often occurs in a series of jumps".

Similar wavelets along growing cracks, can also be observed in the videos from simulation in [36], where the meshfree PD method with the critical strain criterion was used. These wavelets are much "shaper" and better "organized" in the current FCBM PD simulations compared to the "noisier" ones from [36], and that is mainly because of the difference in the damage models: in FCBM PD, one removes a node and all of its bonds suddenly, once the node's strain energy density reaches the critical value; in the meshfree PD model, individual bonds are removed once they reach their critical value.

6. Conclusions

A general and fast convolution-based method (FCBM) for peridynamics was introduced. In this method, one tries to write the PD integrals in the form of convolutions. When this is possible, one then uses the one-point Gaussian quadrature over the domain and transforms the discretized convolution-based PD equations into products of discrete Fourier coefficients. Following these transformations, the PD integrals are computed with the FFT and its inverse operation at the cost of $O(N \log_2 N)$, instead of $O(N^2)$ needed by commonly-used meshfree or FEM discretization methods of PD models. For time-dependent problems, a time-marching scheme is used. Since the integrals involved are now computed in the Fourier space, neighbor identifications and storing neighbor information is no longer needed. As a result, the memory allocation becomes of $O(N)$ in FCBM while other methods need $O(N^2)$ storage. To extend the applicability of the method to arbitrary domains and boundary conditions, the embedded constraint (EC) approach, previously used for PD diffusion problems, was extended here to the general setting that includes the PD equations of motion.

The method introduced here is general and can be used for any nonlocal model as long as a convolutional structure can be identified for it. In this study, the procedure was applied to the PD equations of motion and problems in elasticity and dynamic fracture were solved. A convolutional structure for several material models was obtained: linearized bond- and state-based elastic materials, and a nonlinear elastic bond-based model. The convolutional form for PD correspondence models can also be obtained and details and examples are provided in a separate study. To exploit the FCBM efficiency for fracture problems, a new energy-based damage model was introduced so that it led to a convolutional structure for such problems. The method was verified on a 3D elastostatic problem over a complex geometry and a 2D dynamic brittle fracture problem with crack branching cascades.

Comparisons between the computational efficiency of the new FCBM-EC method for PD models with that of the original meshfree discretization of PD formulations showed that problems requiring years of computations (on a single processor) with the latter method can be executed in a matter of days (on the same processor) with the former. Memory allocation was also two orders of magnitude less than what was required by the meshfree PD. One can now easily reach crack paths independent of the grid used because choosing a large number of nodes inside the PD horizon is no longer a major computational obstacle. Fast simulation of complex fracture problems with high accuracy are now possible via the FCBM-EC method for PD models, as the efficiency gains compared with the original meshfree discretization method can reach a factor of 10^3–10^4 or more.

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

This work has been supported by the National Science Foundation, USA under CMMI CDS&E Award No. 1953346, and by a Nebraska System Science award from the Nebraska Research Initiative. This work was completed utilizing the Holland Computing Center of the University of Nebraska, which receives support from the Nebraska Research Initiative.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.cma.2022.114666.

References

[1] S.A. Silling, R. Lehoucq, Peridynamic theory of solid mechanics, in: Advances in Applied Mechanics, Elsevier, 2010, pp. 73–168.
[2] F. Bobaru, J.T. Foster, P.H. Geubelle, S.A. Silling, Handbook of Peridynamic Modeling, CRC Press, 2016.
[3] S.A. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, J. Mech. Phys. Solids 48 (2000) 175–209.
[4] Z. Chen, F. Bobaru, Peridynamic modeling of pitting corrosion damage, J. Mech. Phys. Solids 78 (2015) 352–381.
[5] S.A. Silling, E. Askari, A meshfree method based on the peridynamic model of solid mechanics, Comput. Struct. 83 (2005) 1526–1535.
[6] 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.
[7] X. Tian, Q. Du, Asymptotically compatible schemes and applications to robust discretization of nonlocal models, SIAM J. Numer. Anal. 52 (2014) 1641–1665.
[8] X. Tian, Q. Du, Nonconforming discontinuous Galerkin methods for nonlocal variational problems, SIAM J. Numer. Anal. 53 (2015) 762–781.
[9] 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.
[10] 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.

[11] M. D'Elia, X. Li, P. Seleson, X. Tian, Y. Yu, A review of local-to-nonlocal coupling methods in nonlocal diffusion and nonlocal mechanics, J. Peridynamics Nonlocal Model. (2021) http://dx.doi.org/10.1007/s42102-020-00038-7.
[12] W. Hu, Y. Wang, J. Yu, C.-F. Yen, F. Bobaru, Impact damage on a thin glass plate with a thin polycarbonate backing, Int. J. Impact Eng. 62 (2013) 152–165.
[13] 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.
[14] B. Alali, N. Albin, Fourier spectral methods for nonlocal models, J. Peridynamics Nonlocal Model. 2 (2020) 317–335.
[15] M. D'Elia, Q. Du, C. Glusa, M. Gunzburger, X. Tian, Z. Zhou, Numerical methods for nonlocal and fractional models, Acta Numer. 29 (2020) 1–124.
[16] J.W. Cooley, J.W. Tukey, An algorithm for the machine calculation of complex Fourier series, Math. Comp. 19 (1965) 297–301.
[17] M. Frigo, S.G. Johnson, The design and implementation of FFTW3, Proc. IEEE 93 (2005) 216–231.
[18] Q. Du, J. Yang, Asymptotically compatible Fourier spectral approximations of nonlocal Allen–Cahn equations, SIAM J. Numer. Anal. 54 (2016) 1899–1919.
[19] A. Bueno-Orovio, D. Kay, K. Burrage, Fourier spectral methods for fractional-in-space reaction–diffusion equations, BIT Numer. Math. 54 (2014) 937–954.
[20] 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.
[21] 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.
[22] B. Alali, N. Albin, Fourier multipliers for nonlocal Laplace operators, Appl. Anal. (2019) http://dx.doi.org/10.1080/00036811.2019.1692134.
[23] 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.
[24] 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.
[25] N. Du, X. Guo, H. Wang, A fast state-based peridynamic numerical model, Commun. Comput. Phys. 27 (2020) 274–291.
[26] 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.
[27] S. Jafarzadeh, L. Wang, A. Larios, F. Bobaru, A fast convolution-based method for peridynamic transient diffusion in arbitrary domains, Comput. Methods Appl. Mech. Engrg. 375 (2021) 113633.
[28] F. Mousavi, S. Jafarzadeh, A. Larios, F. Bobaru, The FCBM for peridynamic correspondence models: applications to plasticity and ductile fracture, in preparetion.
[29] S.A. Silling, M. Epton, O. Weckner, J. Xu, E. Askari, Peridynamic states and constitutive modeling, J. Elasticity 88 (2007) 151–184.
[30] G. Sarego, Q.V. Le, F. Bobaru, M. Zaccariotto, U. Galvanetto, Linearized state-based peridynamics for 2-D problems, Internat. J. Numer. Methods Engrg. 108 (2016) 1174–1197.
[31] D. Dipasquale, G. Sarego, M. Zaccariotto, U. Galvanetto, A discussion on failure criteria for ordinary state-based peridynamics, Eng. Fract. Mech. 186 (2017) 378–398.
[32] S. Murakami, Continuum Damage Mechanics: A Continuum Mechanics Approach to the Analysis of Damage and Fracture, Springer Science & Business Media, 2012.
[33] L. Lopez, S.F. Pellegrino, A spectral method with volume penalization for a nonlinear peridynamic model, Internat. J. Numer. Methods Engrg. 122 (2021) 707–725.
[34] S.A. Silling, Linearized theory of peridynamic states, J. Elasticity 99 (2010) 85–111.
[35] Y.D. Ha, F. Bobaru, Studies of dynamic crack propagation and crack branching with peridynamics, Int. J. Fract. 162 (2010) 229–244.
[36] F. Bobaru, G. Zhang, Why do cracks branch? A peridynamic investigation of dynamic brittle fracture, Int. J. Fract. 196 (2015) 59–98.
[37] F. Mousavi, S. Jafarzadeh, F. Bobaru, An ordinary state-based peridynamic elastoplastic 2D model consistent with J2 plasticity, Int. J. Solids Struct. 229 (2021) 111146.
[38] S.A. Silling, Stability of peridynamic correspondence material models and their particle discretizations, Comput. Methods Appl. Mech. Engrg. 322 (2017) 42–57.
[39] M. Behzadinasab, J.T. Foster, On the stability of the generalized, finite deformation correspondence model of peridynamics, Int. J. Solids Struct. 182 (2020) 64–76.
[40] M. Behzadinasab, J.T. Foster, A semi-Lagrangian constitutive correspondence framework for peridynamics, J. Mech. Phys. Solids (2020) 103862.
[41] S. Jafarzadeh, Novel and Fast Peridynamic Models for Material Degradation and Failure, The University of Nebraska-Lincoln, 2021.
[42] J. Trageser, P. Seleson, Bond-based peridynamics: A tale of two Poisson's ratios, J. Peridynamics Nonlocal Model. 2 (2020) 278–288.
[43] Z. Xu, G. Zhang, Z. Chen, F. Bobaru, Elastic vortices and thermally-driven cracks in brittle materials with peridynamics, Int. J. Fract. 209 (2018) 203–222.
[44] N. Zhu, C. Kochan, E. Oterkus, S. Oterkus, Fatigue analysis of polycrystalline materials using peridynamic theory with a novel crack tip detection algorithm, Ocean Eng. 222 (2021) 108572.
[45] C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, Spectral Methods, Springer, 2006.
[46] D.A. Kopriva, Implementing Spectral Methods for Partial Differential Equations: Algorithms for Scientists and Engineers, Springer Science & Business Media, 2009.
[47] G. Plonka, D. Potts, G. Steidl, M. Tasche, Numerical Fourier Analysis, Springer, 2018.

[48] J.P. Boyd, A fast algorithm for Chebyshev, fourier, and sinc interpolation onto an irregular grid, J. Comput. Phys. 103 (1992) 243–257.
[49] A. Dutt, V. Rokhlin, Fast Fourier transforms for nonequispaced data, SIAM J. Sci. Comput. 14 (1993) 1368–1393.
[50] D. Potts, G. Steidl, Fast summation at nonequispaced knots by NFFT, SIAM J. Sci. Comput. 24 (2003) 2013–2037.
[51] M.C. Jeruchim, P. Balaban, K.S. Shanmugan, Simulation of communication systems: modeling, in: Methodology and Techniques, Springer Science & Business Media, 2006.
[52] F. Bobaru, Y.D. Ha, Adaptive refinement and multiscale modeling in 2D peridynamics, Int. J. Multiscale Comput. Eng. 9 (2011) 635–660.
[53] P. Seleson, D.J. Littlewood, Convergence studies in meshfree peridynamic simulations, Comput. Math. Appl. 71 (2016) 2432–2448.
[54] M.L. Parks, R.B. Lehoucq, S.J. Plimpton, S.A. Silling, Implementing peridynamics within a molecular dynamics code, Comput. Phys. Comm. 179 (2008) 777–783.
[55] 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.
[56] J. Zhao, S. Jafarzadeh, Z. Chen, F. Bobaru, An algorithm for imposing local boundary conditions in peridynamic models of diffusion on arbitrary domains, 2020, http://dx.doi.org/10.31224/osf.io/7z8qr, engrXiv Preprints.
[57] B. Aksoylu, F. Celiker, O. Kilicer, Nonlocal operators with local boundary conditions in higher dimensions, Adv. Comput. Math. 45 (2019) 453–492.
[58] B. Aksoylu, F. Celiker, O. Kilicer, Nonlocal operators with local boundary conditions: An overview, in: G.Z. Voyiadjis (Ed.), Handbook of Nonlocal Continuum Mechanics for Materials and Structures, Springer International Publishing, 2018, pp. 1293–1330.
[59] Y. Yu, H. You, N. Trask, An asymptotically compatible treatment of traction loading in linearly elastic peridynamic fracture, Comput. Methods Appl. Mech. Engrg. 377 (2021) 113691.
[60] Q. Du, M. Gunzburger, R. Lehoucq, K. Zhou, Analysis of the volume-constrained peridynamic Navier equation of linear elasticity, J. Elasticity 113 (2013) 193–217.
[61] U.M. Ascher, C. Greif, A First Course on Numerical Methods, SIAM, 2011.
[62] S.A. Orszag, On the elimination of aliasing in finite-difference schemes by filtering high-wavenumber components, J. Atmos. Sci. 28 (1971) 1074.
[63] 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.
[64] S. Oterkus, E. Madenci, A. Agwai, Peridynamic thermal diffusion, J. Comput. Phys. 265 (2014) 71–96.
[65] P. Seleson, M. Parks, On the role of the influence function in the peridynamic theory, Int. J. Multiscale Comput. Eng. 9 (2011) 689–706.
[66] D.J. Littlewood, Roadmap for Peridynamic Software Implementation, SAND Report, Sandia National Laboratories, Albuquerque, NM and Livermore, CA, 2015.
[67] B. Vazic, C. Diyaroglu, E. Oterkus, S. Oterkus, Family member search algorithms for peridynamic analysis, J. Peridynamics Nonlocal Model. 2 (2020) 59–84.
[68] L. Lopez, S.F. Pellegrino, A space–time discretization of a nonlinear peridynamic model on a 2D lamina, 2021, arXiv preprint arXiv:2102.06485.
[69] Q. Le, F. Bobaru, Surface corrections for peridynamic models in elasticity and fracture, Comput. Mech. 61 (2018) 499–518.
[70] K. Ravi-Chandar, W. Knauss, An experimental investigation into dynamic fracture: III, on steady-state crack propagation and crack branching, Int. J. Fract. 26 (1984) 141–154.
[71] E. Johnson, Process region changes for rapidly propagating cracks, Int. J. Fract. 55 (1992) 47–63.
[72] K. Ravi-Chandar, Dynamic Fracture, Elsevier, 2004.
[73] 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.
[74] D. Hull, Fractography: observing, Measuring and Interpreting Fracture Surface Topography, Cambridge University Press, 1999.