Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

A preconditioned fast collocation method for a linear bond-based peridynamic model

Xuhao Zhang[1], Xiao Li[2], Aijie Cheng[2*] and Hong Wang[3]

*Correspondence: aijie@sdu.edu.cn
[2]School of Mathematics, Shandong University, Jinan, China
Full list of author information is available at the end of the article

Abstract

We develop a fast collocation method for a static bond-based peridynamic model. Based on the analysis of the structure of the stiffness matrix, a fast matrix-vector multiplication technique was found, which can be used in the Krylov subspace iteration method. In this paper, we also present an effective preconditioner to accelerate the convergence of the Krylov subspace iteration method. Using the block-Toeplitz–Toeplitz-block (BTTB)-type structure of the stiffness matrix, we give a block-circulant-circulant-block (BCCB)-type preconditioner. The numerical experiments show the utility of the preconditioned fast collocation method.

Keywords: Nonlocal models; Peridynamic model; Preconditioner; Fast collocation method

1 Introduction

In the last decades, the peridynamic model has been applied in many research fields, such as failure and damage in composite laminates, crack propagation and branching, crack nucleation, phase transformations in solids, impact damage, damage in concrete, and so on [1–7]. In contrast to the classical theory of solid or fluid mechanics, which are usually modeled by partial differential equations [8–10], the peridynamic model does not explicitly involve any spatial derivatives of the displacement. Thus it provides a more natural description for problems with spontaneous formation of discontinuities or other singularities.

To date, several numerical methods have been developed and analyzed for the nonlocal diffusion model and peridynamic model as well as other related nonlocal models, such as finite element discretizations, finite difference method, finite volume method, collocation method, and the meshfree method [1, 11–15]. However, mathematically speaking, because of the nonlocality of these models, the stiffness matrices resulting from finite element method or collocation method are usually full or dense diagonal. The widely used Krylov subspace iteration method requires $O(N^2)$ of memory to store the stiffness matrix and $O(N^2)$ computational work in each iteration to solve the associated linear system where N is the number of spatial nodes. Therefore, the simulation of peridynamic model is usually time-consuming especially for large-scale problems (e.g.,multi-dimensional peridynamic models).

Several research works have been devoted to reducing the memory requirement and computations of the corresponding numerical schemes for peridynamic model [15–18]. These works are based on the analysis of Toeplitz or Toeplitz-type structure of the stiffness matrices and on the fast matrix-vector multiplication technique which can be used in the Krylov subspace iteration method. In [18], a fast collocation method for a two-dimensional bond-based linear peridynamic model was developed by carefully exploring the BTTB-type structure of the stiffness matrix $\mathbf{A}_{2N}$. However, from the numerical experiments we can find that the convergence is slow, especially when the singularity of the kernel function is stronger.

In this paper, we follow the work started in [18] and develop a preconditioned fast collocation method for a linear peridynamic model. We provide an effective preconditioner to accelerate the convergence of the Krylov subspace iteration method. The rest of the paper is organized as follows. In Sect. 2, we recall the bilinear collocation method developed in [18]. However, the stiffness matrix is rewritten by reordering the unknowns differently from the unknowns' arrangement in [18]. In Sect. 3, we analyze that each block matrix of the stiffness matrix is a BTTB matrix. Based on this analysis, the operations in each Krylov subspace iteration and the storage of the stiffness matrix can be reduced. In Sect. 4, to accelerate the convergence of the Krylov subspace iteration method, a BCCB-type preconditioner is provided. In Sect. 5, we carry out several numerical experiments to investigate the performance of this preconditioner. In Sect. 6, we draw concluding remarks.

2 The bilinear collocation method

We consider the following two-dimensional bond-based linear peridynamic model:

$$\begin{aligned} &\int_{B_\delta(\mathbf{x})} \mathbf{C}\big(\mathbf{x}',\mathbf{x}\big)\big(\mathbf{u}\big(\mathbf{x}'\big)-\mathbf{u}(\mathbf{x})\big)\,d\mathbf{x}' = \mathbf{f}(\mathbf{x}), \quad \mathbf{x}\in\Omega,\\ &\mathbf{u}(\mathbf{x}) = \mathbf{g}(\mathbf{x}), \quad \mathbf{x}\in\Omega_c. \end{aligned} \tag{1}$$

Here $\mathbf{C}$ is the micromodulus function

$$\mathbf{C}\big(\mathbf{x}',\mathbf{x}\big) = \sigma\big(\big|\mathbf{x}'-\mathbf{x}\big|\big)\left[\frac{(\mathbf{x}'-\mathbf{x})}{|\mathbf{x}'-\mathbf{x}|}\otimes\frac{(\mathbf{x}'-\mathbf{x})}{|\mathbf{x}'-\mathbf{x}|}\right]; \tag{2}$$

$\Omega := (0,x_R)\times(0,y_R)$ represents a rectangular domain; $\mathbf{x} := (x,y)^T$ and $\mathbf{x}' := (x',y')^T$ are positions of the particles in the reference configuration; $\mathbf{u}(\mathbf{x}) := [v(\mathbf{x}),w(\mathbf{x})]^T$ and $\mathbf{u}(\mathbf{x}') := [v(\mathbf{x}'),w(\mathbf{x}')]^T$ represent the displacements of particles $\mathbf{x}$ and $\mathbf{x}'$ with respect to the reference configuration, respectively. $\sigma(\cdot)$ is the kernel function; $\delta>0$ is the horizon of the material; $B_\delta(\mathbf{x})$ is assumed to be an open disk that is centered at $\mathbf{x}$ with δ being the radius; Ω_c denotes a boundary zone surrounding Ω with width δ; $\mathbf{f}(\mathbf{x}) := [f^v(\mathbf{x}),f^w(\mathbf{x})]^T$ represents the external force density; $\mathbf{g}(\mathbf{x}) := [g^v(\mathbf{x}),g^w(\mathbf{x})]^T$ is the prescribed nonlocal boundary data imposed on the boundary zone Ω_c. For clarity, we show the domains Ω, Ω_c, and $B_\delta(\mathbf{x})$ in Fig. 1.

Given two n-dimensional vectors

$$\mathbf{V} = \begin{pmatrix} v_1\\ v_2\\ \vdots\\ v_n \end{pmatrix}, \qquad \mathbf{W} = \begin{pmatrix} w_1\\ w_2\\ \vdots\\ w_n \end{pmatrix},$$

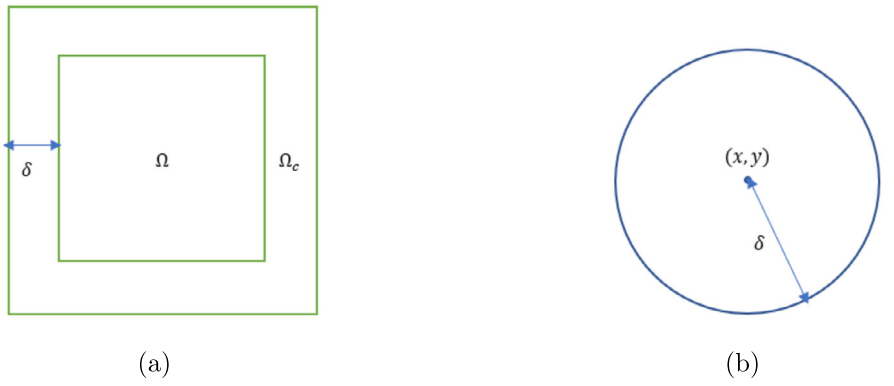

Figure 1 The left figure (**a**) shows the relationship between the domain Ω and the boundary zone Ω_c with width δ; The right figure (**b**) represents the integral domain $B_\delta(\mathbf{x})$ in (1), which is an open disk centered at $\mathbf{x}$ with a radius δ

the tensor product '$\otimes$' in (2) is defined as

$$\mathbf{V}\otimes\mathbf{W}=\begin{pmatrix} v_1w_1 & v_1w_2 & \cdots & v_1w_n \\ v_2w_1 & v_2w_2 & \ddots & v_2w_n \\ \vdots & \ddots & \ddots & \vdots \\ v_nw_1 & v_nw_2 & \cdots & v_nw_n \end{pmatrix}.$$

For the convenience of the following chapter, we also give the definition of tensor product in the matrix case. Given two matrices $\mathbf{A}=(a_{i,j})\in\mathbb{R}^{p,q}$ and $\mathbf{B}\in\mathbb{R}^{r,s}$, the tensor product of $\mathbf{A}$ and $\mathbf{B}$ is defined as follows:

$$\mathbf{A}\otimes\mathbf{B}=\begin{pmatrix} a_{1,1}\mathbf{B} & a_{1,2}\mathbf{B} & \cdots & a_{1,q}\mathbf{B} \\ a_{2,1}\mathbf{B} & a_{2,2}\mathbf{B} & \ddots & a_{2,q}\mathbf{B} \\ \vdots & \ddots & \ddots & \vdots \\ a_{p,1}\mathbf{B} & a_{p,1}\mathbf{B} & \cdots & a_{p,q}\mathbf{B} \end{pmatrix}.$$

Let N_x and N_y be positive integers. We define a spatial partition on $\bar{\Omega}$ by $x_i := ih_x$ for $i = 0,1,\ldots,N_x$ and $y_j := jh_y$ for $j = 0,1,\ldots,N_y$, where $h_x := x_R/N_x$ and $h_y := y_R/N_y$ are the mesh sizes in the x and y directions, respectively. To handle the discretization on the boundary zone Ω_c, we extend the partition to (x_i,y_j) for $i=-K+1,\ldots,-1,0,1,\ldots,N_x,N_x+1,\ldots,N_x+K-1$ and $j=-L+1,\ldots,-1,0,1,\ldots,N_y,N_y+1,\ldots,N_y+L-1$. Here,

$$K:=\lceil\delta/h_x\rceil,\qquad L:=\lceil\delta/h_y\rceil \tag{3}$$

are the ceilings of δ/h_x and δ/h_y, respectively.

Let $\psi(\xi)=1-|\xi|$ for $\xi\in[-1,1]$ and zero otherwise. The two-dimensional pyramid functions $\phi_{i,j}(x,y)$ centered at (x_i,y_j) can be expressed as

$$\phi_{ij}(x,y)=\psi\left(\frac{x-x_i}{h_x}\right)\psi\left(\frac{y-y_j}{h_y}\right),\quad 0\le i\le N_x, 0\le j\le N_y. \tag{4}$$

Then the trial functions v and w in the displacement vector $\mathbf{u}$ can be written as

$$\begin{aligned} v(x,y) &= \sum_{i'=-K+1}^{N_x+K-1} \sum_{j'=-L+1}^{N_y+L-1} v_{i',j'} \phi_{i',j'}(x,y), \\ w(x,y) &= \sum_{i'=-K+1}^{N_x+K-1} \sum_{j'=-L+1}^{N_y+L-1} w_{i',j'} \phi_{i',j'}(x,y). \end{aligned} \tag{5}$$

We choose (x_i, y_j) for $i = 1, \dots, N_x - 1$ and $j = 1, \dots, N_y - 1$ as our collocation points. By substituting (5) into (1), we obtain the following collocation scheme:

$$\begin{aligned} &\int_{B_\delta(x_i,y_j)} \frac{\sigma(|(x'-x_i, y'-y_j)|)}{(x'-x_i)^2 + (y'-y_j)^2} \Bigg\{ (x'-x_i)^2 \Bigg[v_{i,j} - \sum_{i'=-K+1}^{N_x+K-1} \sum_{j'=-L+1}^{N_y+L-1} v_{i',j'} \phi_{i',j'}(x', y') \Bigg] \\ &\quad + (x'-x_i)(y'-y_j) \Bigg[w_{i,j} - \sum_{i'=-K+1}^{N_x+K-1} \sum_{j'=-L+1}^{N_y+L-1} w_{i',j'} \phi_{i',j'}(x', y') \Bigg] \Bigg\} dx'\, dy' = f^v(x_i, y_j), \\ &\int_{B_\delta(x_i,y_j)} \frac{\sigma(|(x'-x_i, y'-y_j)|)}{(x'-x_i)^2 + (y'-y_j)^2} \Bigg\{ (y'-y_j)^2 \Bigg[w_{i,j} - \sum_{i'=-K+1}^{N_x+K-1} \sum_{j'=-L+1}^{N_y+L-1} w_{i',j'} \phi_{i',j'}(x', y') \Bigg] \\ &\quad + (x'-x_i)(y'-y_j) \Bigg[v_{i,j} - \sum_{i'=-K+1}^{N_x+K-1} \sum_{j'=-L+1}^{N_y+L-1} v_{i',j'} \phi_{i',j'}(x', y') \Bigg] \Bigg\} dx'\, dy' = f^w(x_i, y_j), \\ &\quad 1 \le i \le N_x - 1, 1 \le j \le N_y - 1. \end{aligned} \tag{6}$$

Let $N = (N_x - 1)(N_y - 1)$ be the number of unknowns. If we define the vector of unknowns $\mathbf{u}_{2N}$ and the vector of right-hand side as follows:

$$\begin{aligned} \mathbf{u}_{2N} := [&v_{1,1}, v_{2,1}, \dots, v_{N_x-1,1}, v_{1,2}, \dots, v_{N_x-1,2}, \\ &\cdots, v_{1,N_y-1}, \dots, v_{N_x-1,N_y-1}, \\ &w_{1,1}, w_{2,1}, \dots, w_{N_x-1,1}, w_{1,2}, \dots, w_{N_x-1,2}, \\ &\cdots, w_{1,N_y-1}, \dots, w_{N_x-1,N_y-1}]^T, \\ \mathbf{f}_{2N} := [&f^v_{1,1}, f^v_{2,1}, \dots, f^v_{N_x-1,1}, f^v_{1,2}, \dots, f^v_{N_x-1,2}, \\ &\cdots, f^v_{1,N_y-1}, \dots, f^v_{N_x-1,N_y-1}, \\ &f^w_{1,1}, f^w_{2,1}, \dots, f^w_{N_x-1,1}, f^w_{1,2}, \dots, f^w_{N_x-1,2}, \\ &\cdots, f^w_{1,N_y-1}, \dots, f^w_{N_x-1,N_y-1}]^T, \end{aligned} \tag{7}$$

then the collocation scheme can be written as the following matrix form:

$$\mathbf{A}_{2N} \mathbf{u}_{2N} = \mathbf{f}_{2N}, \tag{8}$$

where the $2N$-by-$2N$ stiffness matrix $\mathbf{A}_{2N}$ can be expressed as the following form in which each matrix block is of N order:

$$\mathbf{A}_{2N} = \begin{pmatrix} \mathbf{A}_N^{v,v} & \mathbf{A}_N^{v,w} \\ \mathbf{A}_N^{w,v} & \mathbf{A}_N^{w,w} \end{pmatrix}. \tag{9}$$

In (9), the entries in the submatrix $\mathbf{A}_N^{v,v}$ are defined as

$$A_{m,n}^{v,v} := \int_{B_\delta(x_i,y_j)} \frac{\sigma(|(x'-x_i,y'-y_j)|)}{(x'-x_i)^2+(y'-y_j)^2}\big(x'-x_i\big)^2\big(\delta_{m,n}-\phi_{i',j'}\big(x',y'\big)\big)\,dx'\,dy'. \tag{10}$$

Similarly, the entries in the submatrices $\mathbf{A}_N^{v,w}$ and $\mathbf{A}_N^{w,w}$ are given by

$$A_{m,n}^{v,w} := \int_{B_\delta(x_i,y_j)} \frac{\sigma(|(x'-x_i,y'-y_j)|)}{(x'-x_i)^2+(y'-y_j)^2}\big(x'-x_i\big)\big(y'-y_j\big)\big(\delta_{m,n}-\phi_{i',j'}\big(x',y'\big)\big)\,dx'\,dy' \tag{11}$$

and

$$A_{m,n}^{w,w} := \int_{B_\delta(x_i,y_j)} \frac{\sigma(|(x'-x_i,y'-y_j)|)}{(x'-x_i)^2+(y'-y_j)^2}\big(y'-y_j\big)^2\big(\delta_{m,n}-\phi_{i',j'}\big(x',y'\big)\big)\,dx'\,dy', \tag{12}$$

respectively. $\mathbf{A}_N^{v,w} = \mathbf{A}_N^{w,v}$. The function $\phi_{i,j}(x,y)$ denotes the two-dimensional bilinear basis function at the grid point (x_i,y_j). The global indices m and n are related to the indices (i,j) and (i',j') by

$$\begin{aligned} &m = (j-1)\times(N_x-1)+i, \quad 1\le i\le N_x-1, 1\le j\le N_y-1,\\ &n = \big(j'-1\big)\times(N_x-1)+i', \quad 1\le i'\le N_x-1, 1\le j'\le N_y-1. \end{aligned} \tag{13}$$

In (7) $f_{i,j}^v$ and $f_{i,j}^w$ are defined as follows:

$$\begin{aligned} f_{i,j}^v =& f^v(x_i,y_j) + \sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\int_{B_\delta(x_i,y_j)}\frac{\sigma(|(x'-x_i,y'-y_j)|)}{((x'-x_i)^2+(y'-y_j)^2)}\\ &\times\big(x'-x_i\big)^2 g^v(x_{i''},y_{j''})\phi_{i'',j''}\big(x',y'\big)\,dx'\,dy'\\ &+\sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\int_{B_\delta(x_i,y_j)}\frac{\sigma(|(x'-x_i,y'-y_j)|)}{((x'-x_i)^2+(y'-y_j)^2)}\\ &\times\big(x'-x_i\big)\big(y'-y_j\big)g^w(x_{i''},y_{j''})\phi_{i'',j''}\big(x',y'\big)\,dx'\,dy',\\ f_{i,j}^w =& f^w(x_i,y_j) + \sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\int_{B_\delta(x_i,y_j)}\frac{\sigma(|(x'-x_i,y'-y_j)|)}{((x'-x_i)^2+(y'-y_j)^2)}\\ &\times\big(x'-x_i\big)\big(y'-y_j\big)g^v(x_{i''},y_{j''})\phi_{i'',j''}\big(x',y'\big)\,dx'\,dy'\\ &+\sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\int_{B_\delta(x_i,y_j)}\frac{\sigma(|(x'-x_i,y'-y_j)|)}{((x'-x_i)^2+(y'-y_j)^2)}\\ &\times\big(y'-y_j\big)^2 g^w(x_{i''},y_{j''})\phi_{i'',j''}\big(x',y'\big)\,dx'\,dy'. \end{aligned} \tag{14}$$

3 The structure of stiffness matrix

Theorem 1 *Each of the submatrices* $\mathbf{A}_N^{v,v}$, $\mathbf{A}_N^{v,w}$, *and* $\mathbf{A}_N^{w,w}$ *has a BTTB structure. More precisely, each matrix can be expressed as a* (N_y-1)*-by-*(N_y-1) *block-banded Toeplitz matrix with a block bandwidth* $2L+1$,

$$\mathbf{A}_N^I = \begin{pmatrix} \mathbf{T}_0^I & \cdots & \mathbf{T}_L^I & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{T}_{-L}^I & \ddots & \mathbf{T}_0^I & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{T}_0^I & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{T}_0^I & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{T}_0^I & \ddots & \mathbf{T}_L^I \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{T}_{-L}^I & \cdots & \mathbf{T}_0^I \end{pmatrix}, \tag{15}$$

where the superscript $I=(v,v),(v,w)$, *or* (w,w). *Furthermore, each matrix block* $\mathbf{T}_j^I$, *with* $-L\le j\le L$, *is an* (N_x-1)*-by-*(N_x-1) *banded Toeplitz matrix with a bandwidth* $2K+1$

$$\mathbf{T}_j^I = \begin{pmatrix} t_{0,j}^I & \cdots & t_{K,j}^I & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ t_{-K,j}^I & \ddots & t_{0,j}^I & \ddots & \ddots & 0 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & t_{0,j}^I & \ddots & \ddots & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & \ddots & \ddots & t_{0,j}^I & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & \ddots & \ddots & t_{0,j}^I & \ddots & t_{K,j}^I \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & t_{-K,j}^I & \cdots & t_{0,j}^I \end{pmatrix}. \tag{16}$$

Proof We only investigate the structure of $\mathbf{A}_N^{v,v}$. The analysis of $\mathbf{A}_N^{v,w}$ and $\mathbf{A}_N^{w,w}$ is similar. By expanding the matrix $\mathbf{A}_N^{v,v}$, we can easily find that $\mathbf{A}_N^{v,v}$ can be written as the following form:

$$\mathbf{A}_N^{v,v} = \begin{pmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,1} & \cdots & \mathbf{B}_{1,N_y-1} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \ddots & \mathbf{B}_{2,N_y-1} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{B}_{N_y-1,1} & \mathbf{B}_{N_y-1,2} & \cdots & \mathbf{B}_{N_y-1,N_y-1} \end{pmatrix}. \tag{17}$$

Here, each block matrix $\mathbf{B}_{j,j'}$ of order N_x-1 denotes the interaction of row j and row j' in the discrete system for $1\le j\le N_y-1$ and $1\le j'\le N_y-1$. From (11) and (13), we can find

that the entry $A_{m,n}^{v,v} \neq 0$ if and only if

$$\operatorname{supp}(\phi_{i',j'}) \cap B_\delta(x_i, y_j) \neq \emptyset. \tag{18}$$

Therefore all the matrix blocks $\mathbf{B}_{j,j'}$ with $|j - j'| > L$ in (17) vanish. Then $\mathbf{A}_N^{v,v}$ is a $2L + 1$ block-banded matrix and can be expressed as the following form:

$$\mathbf{A}_N^{v,v} = \begin{pmatrix} \mathbf{B}_{1,1} & \mathbf{B}_{1,2} & \cdots & \mathbf{B}_{1,L+1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{B}_{2,1} & \mathbf{B}_{2,2} & \ddots & \cdots & \ddots & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{B}_{L+1,1} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{B}_{N_y-L-1,N_y-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_{N_y-1,N_y-L-1} & \cdots & \mathbf{B}_{N_y-1,N_y-1} \end{pmatrix}. \tag{19}$$

Furthermore, in each matrix block $\mathbf{B}_{j,j'}$, we have $A_{m,n}^{v,v} = 0$ for $|i - i'| > K$. That is, $\mathbf{B}_{j,j'}$ is a $2K + 1$ banded matrix expressed as

$$\mathbf{B}_{j,j'} = \begin{pmatrix} b_{j,j'}^{1,1} & b_{j,j'}^{1,2} & \cdots & b_{j,j'}^{1,K+1} & 0 & \cdots & 0 \\ b_{j,j'}^{2,1} & b_{j,j'}^{2,2} & \ddots & \cdots & \ddots & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ b_{j,j'}^{K+1,1} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & b_{j,j'}^{N_y-K-1,N_y-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \mathbf{0} & \cdots & \mathbf{0} & b_{j,j'}^{N_y-1,N_y-K-1} & \cdots & b_{j,j'}^{N_y-1,N_y-1} \end{pmatrix}. \tag{20}$$

By introducing the following translations:

$$\xi_1 = x' - x_i, \qquad \xi_2 = y' - y_j, \tag{21}$$

the entries $\mathbf{A}_{m,n}^{v,v}$ given in (11) can be reduced to

$$A_{m,n}^{v,v} = \int_{B_\delta(0,0)} \frac{\xi_1^2 \sigma(\|(\xi_1, \xi_2)\|)}{\xi_1^2 + \xi_2^2} \left(\delta_{m,n} - \psi\left(\frac{\xi_1 - x_{i'-i}}{h_x} \right) \psi\left(\frac{\xi_2 - y_{j'-j}}{h_y} \right) \right) d\xi_1 \, d\xi_2. \tag{22}$$

Let $j_1' - j_1 = j_2' - j_2 = l, -L \le l \le L$, and let

$$\begin{aligned}
&m_1 = (j_1 - 1)(N_x - 1) + i, \quad 1 \le i \le N_x - 1, 1 \le j_1 \le N_y - 1,\\
&n_1 = \big(j_1' - 1\big)(N_x - 1) + i', \quad 1 \le i' \le N_x - 1, 1 \le j_1' \le N_y - 1,\\
&m_2 = (j_2 - 1)(N_x - 1) + i, \quad 1 \le i \le N_x - 1, 1 \le j_2 \le N_y - 1,\\
&n_2 = \big(j_2' - 1\big)(N_x - 1) + i', \quad 1 \le i' \le N_x - 1, 1 \le j_2' \le N_y - 1.
\end{aligned} \tag{23}$$

Then we observe that, for $1 \le i, i' \le N_x - 1$,

$$\begin{aligned}
b_{j_1,j_1'}^{i,i'} &= A_{m_1,n_1}^{v,v}\\
&= \int_{B_\delta(0,0)} \frac{\xi_1^2 \sigma(\|(\xi_1,\xi_2)\|)}{\xi_1^2 + \xi_2^2} \left(\delta_{m_1,n_1} - \psi\left(\frac{\xi_1 - x_{i'-i}}{h_x}\right)\psi\left(\frac{\xi_2 - y_{j_1'-j_1}}{h_y}\right)\right) d\xi_1\, d\xi_2\\
&= \int_{B_\delta(0,0)} \frac{\xi_1^2 \sigma(\|(\xi_1,\xi_2)\|)}{\xi_1^2 + \xi_2^2} \left(\delta_{m_2,n_2} - \psi\left(\frac{\xi_1 - x_{i'-i}}{h_x}\right)\psi\left(\frac{\xi_2 - y_{j_2'-j_2}}{h_y}\right)\right) d\xi_1\, d\xi_2\\
&= A_{m_2,n_2}^{v,v} = b_{j_2,j_2'}^{i,i'}.
\end{aligned} \tag{24}$$

According to (24), we have proved $\mathbf{B}_{j_1,j_1'} = \mathbf{B}_{j_2,j_2'}$ if the block matrices $\mathbf{B}_{j_1,j_1'}$ and $\mathbf{B}_{j_2,j_2'}$ are on the same diagonal.

Let $i_3' - i_3 = i_4' - i_4 = k, -K \le k \le K$, and let

$$\begin{aligned}
&m_3 = (j - 1)(N_x - 1) + i_3, \quad 1 \le i_3 \le N_x - 1, 1 \le j \le N_y - 1,\\
&n_3 = \big(j' - 1\big)(N_x - 1) + i_3', \quad 1 \le i_3' \le N_x - 1, 1 \le j' \le N_y - 1,\\
&m_4 = (j - 1)(N_x - 1) + i_4, \quad 1 \le i_4 \le N_x - 1, 1 \le j \le N_y - 1,\\
&n_4 = \big(j' - 1\big)(N_x - 1) + i_4', \quad 1 \le i_4' \le N_x - 1, 1 \le j' \le N_y - 1.
\end{aligned} \tag{25}$$

Then we observe that

$$\begin{aligned}
b_{j,j'}^{i_3,i_3'} &= A_{m_3,n_3}^{v,v}\\
&= \int_{B_\delta(0,0)} \frac{\xi_1^2 \sigma(\|(\xi_1,\xi_2)\|)}{\xi_1^2 + \xi_2^2} \left(\delta_{m_3,n_3} - \psi\left(\frac{\xi_1 - x_{i_3'-i_3}}{h_x}\right)\psi\left(\frac{\xi_2 - y_{j'-j}}{h_y}\right)\right) d\xi_1\, d\xi_2\\
&= \int_{B_\delta(0,0)} \frac{\xi_1^2 \sigma(\|(\xi_1,\xi_2)\|)}{\xi_1^2 + \xi_2^2} \left(\delta_{m_4,n_4} - \psi\left(\frac{\xi_1 - x_{i_4'-i_4}}{h_x}\right)\psi\left(\frac{\xi_2 - y_{j'-j}}{h_y}\right)\right) d\xi_1\, d\xi_2\\
&= A_{m_4,n_4}^{v,v} = b_{j,j'}^{i_4,i_4'}.
\end{aligned} \tag{26}$$

According to (26), we conclude that each block matrix $\mathbf{B}_{j,j'}$ is a banded Toeplitz matrix. Combining (24) and (26), we complete the proof. □

Corollary 1 *The stiffness matrix* $\mathbf{A}_{2N}$ *can be stored in* $O(N)$ *memories.*

Proof From the structure of $\mathbf{A}_{2N}$, we only prove that the block matrix $\mathbf{A}_N^{v,v}$ can be stored in $O(N)$ memories. From Theorem 1 we can find that the matrix $\mathbf{A}_N^{v,v}$ can be stored only

by storing the following $(2K+1)$-by-$(2L+1)$ entries:

$$\mathbf{G} = \begin{pmatrix} t^{\nu,\nu}_{-K,-L} & \cdots & t^{\nu,\nu}_{-K,0} & \cdots & t^{\nu,\nu}_{-K,L} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ t^{\nu,\nu}_{0,-L} & \cdots & t^{\nu,\nu}_{0,0} & \cdots & t^{\nu,\nu}_{0,L} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ t^{\nu,\nu}_{K,-L} & \cdots & t^{\nu,\nu}_{K,0} & \cdots & t^{\nu,\nu}_{K,j} \end{pmatrix}. \tag{27}$$

Hence, $\mathbf{A}_{2N}$ can be stored in $O(4*(2K+1)*(2L+1)) = O(N)$ memories. □

Corollary 2 *For any vector $u \in \mathbb{R}^{2N}$, the matrix-vector multiplication $\mathbf{A}_{2N}\mathbf{u}$ can be carried out in $O(N\log N)$ operations.*

Proof We divide the vector $\mathbf{u}$ in half. That is,

$$\mathbf{u} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix}, \tag{28}$$

in which $\mathbf{u}_1, \mathbf{u}_1 \in \mathbb{R}^N$. Therefore

$$\mathbf{A}_{2N}u = \begin{pmatrix} \mathbf{A}^{\nu,\nu}_N & \mathbf{A}^{\nu,w}_N \\ \mathbf{A}^{\nu,w}_N & \mathbf{A}^{w,w}_N \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{A}^{\nu,\nu}_N\mathbf{u}_1 + \mathbf{A}^{\nu,w}_N\mathbf{u}_2 \\ \mathbf{A}^{\nu,w}_N\mathbf{u}_1 + \mathbf{A}^{w,w}_N\mathbf{u}_2 \end{pmatrix}. \tag{29}$$

Because of the BTTB structure of the matrices $\mathbf{A}^{\nu,\nu}_N$, $\mathbf{A}^{\nu,w}_N$, and $\mathbf{A}^{w,w}_N$, the matrix-vector multiplications $\mathbf{A}^{\nu,\nu}_N\mathbf{u}_1$, $\mathbf{A}^{\nu,w}_N\mathbf{u}_2$, $\mathbf{A}^{\nu,w}_N\mathbf{u}_1$, and $\mathbf{A}^{w,w}_N\mathbf{u}_2$ can be computed in $O(N\log N)$ operations [19]. Accordingly, the total matrix-vector multiplication $\mathbf{A}_{2N}\mathbf{u}$ can be evaluated in $O(4N\log N) = O(N\log N)$ operations. □

4 A preconditioned fast Krylov subspace method

In the Krylov subspace iteration method, each iteration consists of matrix-vector multiplications and the related vector operations. Thus a fast Krylov subspace iteration method can be developed due to the fast matrix-vector multiplication proved in the previous section. It can reduce the computational work from $O(N^2)$ to $O(N\log N)$ per Krylov subspace iteration. Furthermore, from Corollary 2, the memory requirement can also be reduced from $O(N^2)$ to $O(N)$. However, the number of iterations may still be large due to the singularity of kernel function σ. Accordingly, we need to introduce an effective preconditioner to reduce the number of iterations and the overall computational cost.

For a BTTB-type matrix $\mathbf{A_{2N}}$, a straightforward preconditioner can be chosen as

$$\mathbf{C}_{F,1} = \begin{pmatrix} \mathbf{C}^{\nu,\nu}_{F,1} & \mathbf{C}^{\nu,w}_{F,1} \\ \mathbf{C}^{w,\nu}_{F,1} & \mathbf{C}^{w,w}_{F,1} \end{pmatrix} \in \mathbb{R}^{2N}, \tag{30}$$

where $\mathbf{C}_{F,1}^{v,w} = \mathbf{C}_{F,1}^{w,v}$ and each matrix $\mathbf{C}_{F,1}^{I} \in \mathbb{R}^{N\times N}$, with $I = (v,v)$, (v,w) or (w,w), is written as [19–21]

$$\mathbf{C}_{F,1}^{I} = \begin{pmatrix} \mathbf{C}_0^I & \cdots & \mathbf{C}_L^I & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{C}_{-L}^I & \ddots & \mathbf{C}_0^I & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{C}_0^I & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{C}_0^I & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{C}_0^I & \ddots & \mathbf{C}_L^I \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{C}_{-L}^I & \cdots & \mathbf{C}_0^I \end{pmatrix}, \tag{31}$$

in which the matrix block $\mathbf{C}_l^I$, with $-L \le l \le L$, is the Chan's circulant matrix of the Toeplitz matrix $\mathbf{T}_l^I$:

$$\mathbf{C}_l^I = \begin{pmatrix} c_{0,l}^I & c_{1,l}^I & \cdots & c_{N_x-3,l}^I & c_{N_x-2,l}^I \\ c_{-1,l}^I & c_{0,l}^I & \ddots & \ddots & c_{N_x-3,l}^I \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ c_{3-N_x,l}^I & \ddots & \ddots & c_{0,l}^I & c_{1,l}^I \\ c_{2-N_x,l}^I & c_{3-N_x,l}^I & \cdots & c_{-1,l}^I & c_{0,l}^I \end{pmatrix}. \tag{32}$$

In (32), the diagonals of $\mathbf{C}_l^I$ are given by

$$c_{-k,l}^I = \begin{cases} \frac{(N_x-1-k)t_{-k,l}^I + kt_{N_x-1-k,l}^I}{N_x-1}, & 0 \le k \le N_x-2, \\ c_{1-N_x-k,l}^I, & 2-N_x \le k < 0. \end{cases} \tag{33}$$

However, this preconditioner is not easy to invert. So we consider the following preconditioner called BCCB-type preconditioner in this paper:

$$\mathbf{C}_{F,2} = \begin{pmatrix} \mathbf{C}_{F,2}^{v,v} & \mathbf{C}_{F,2}^{v,w} \\ \mathbf{C}_{F,2}^{w,v} & \mathbf{C}_{F,2}^{w,w} \end{pmatrix}, \tag{34}$$

where for $I = (v,v)$, (v,w),or (w,w), $\mathbf{C}_{F,2}^I$ represents the BCCB matrix of $\mathbf{A}_N^I$, and it can be generated by $\mathbf{C}_{F,1}^I$ defined in (31). More precisely, $\mathbf{C}_{F,2}^I$ can be expressed as

$$\mathbf{C}_{F,2}^I = \begin{pmatrix} \tilde{\mathbf{C}}_0^I & \tilde{\mathbf{C}}_1^I & \cdots & \tilde{\mathbf{C}}_{N_y-3}^I & \tilde{\mathbf{C}}_{N_y-2}^I \\ \tilde{\mathbf{C}}_{-1}^I & \tilde{\mathbf{C}}_0^I & \ddots & \ddots & \tilde{\mathbf{C}}_{N_y-3}^I \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \tilde{\mathbf{C}}_{3-N_y}^I & \ddots & \ddots & \tilde{\mathbf{C}}_0^I & \tilde{\mathbf{C}}_1^I \\ \tilde{\mathbf{C}}_{2-N_y}^I & \tilde{\mathbf{C}}_{3-N_y}^I & \cdots & \tilde{\mathbf{C}}_{-1}^I & \tilde{\mathbf{C}}_0^I \end{pmatrix}, \tag{35}$$

where each circulant matrix block is defined by

$$\tilde{\mathbf{C}}^I_{-k} = \begin{cases} \frac{(N_y-1-k)\mathbf{C}^I_{-k,l}+k\mathbf{C}^I_{N_y-1-k,l}}{N_y-1}, & 0 \le k \le N_y-2, \\ \mathbf{C}^I_{1-N_y-k,l}, & 2-N_y \le k < 0. \end{cases} \tag{36}$$

Let $\mathbf{F}_n$ be the n-order discrete Fourier transform matrix and $\mathbf{I}_2$ be the 2-order identity matrix

$$\mathbf{I}_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Let $\mathbf{F}_2 = \mathbf{F}_{N_y-1} \otimes \mathbf{F}_{N_x-1}$ be the two-dimensional discrete Fourier transform matrix and $\mathbf{F}_2^* = (\mathbf{F}_{N_y-1} \otimes \mathbf{F}_{N_x-1})^*$ be the two-dimensional inverse Fourier transform matrix. Then the matrix $\mathbf{C}_{F,2}$ can be decomposed into the following form:

$$\begin{aligned} \mathbf{C}_{F,2} &= (\mathbf{I}_2 \otimes \mathbf{F}_2)\boldsymbol{\Lambda}\left(\mathbf{I}_2 \otimes \mathbf{F}_2^*\right) \\ &= (\mathbf{I}_2 \otimes \mathbf{F}_2)\begin{pmatrix} \boldsymbol{\Lambda}^{v,v} & \boldsymbol{\Lambda}^{v,w} \\ \boldsymbol{\Lambda}^{w,v} & \boldsymbol{\Lambda}^{w,w} \end{pmatrix}\left(\mathbf{I}_2 \otimes \mathbf{F}_2^*\right). \end{aligned} \tag{37}$$

Here $\boldsymbol{\Lambda}^I$ is a diagonal matrix in which the entries on the main diagonal are the eigenvalues of the matrix $\mathbf{C}^I_{F,2}$ for $I = (v,v)$, (v,w), or (w,w). That is, $\boldsymbol{\Lambda}^I$ can be written as

$$\boldsymbol{\Lambda}^I = \begin{pmatrix} \lambda^I_{1,1} & 0 & \cdots & 0 \\ 0 & \lambda^I_{2,2} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda^I_{N,N} \end{pmatrix}. \tag{38}$$

To invert the matrix $\mathbf{C}_{F,2}$ fast, we consider the computational cost of solving the linear equation

$$\mathbf{C}_{F,2}\mathbf{y} = \mathbf{d} \tag{39}$$

for some $\mathbf{d} \in \mathbb{R}^{2N}$.

Theorem 2 *For any vector* $\mathbf{d} \in \mathbb{R}^{2N}$, *linear system* (39) *can be solved in* $O(N \log N)$ *operations.*

Proof Recall that the eigenvalues of $\mathbf{C}^I_{F,2}$ can be computed in $O(N \log N)$ operations by two-dimensional fast Fourier transform (2DFFT). Thus, the computation of $\boldsymbol{\Lambda}$ requires $O(N \log N)$ operations. From (37), to inverse $\mathbf{C}_{F,2}$ efficiently, we need to compute the inverse of $\boldsymbol{\Lambda}$ efficiently.

We introduce a permutation matrix $\mathbf{P}$ such that

$$\mathbf{P}^*\boldsymbol{\Lambda}\mathbf{P} = \begin{pmatrix} \tilde{\boldsymbol{\Lambda}}_{1,1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \tilde{\boldsymbol{\Lambda}}_{2,2} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \tilde{\boldsymbol{\Lambda}}_{N,N} \end{pmatrix}, \tag{40}$$

where for $1 \le k \le N$,

$$\tilde{\boldsymbol{\Lambda}}_{k,k} = \begin{pmatrix} \lambda_{k,k}^{v,v} & \lambda_{k,k}^{v,w} \\ \lambda_{k,k}^{w,v} & \lambda_{k,k}^{w,w} \end{pmatrix}. \tag{41}$$

Therefore, each matrix block $\tilde{\boldsymbol{\Lambda}}_{k,k}$ with $1 \le k \le N$ is a 2-by-2 matrix. It requires $O(1)$ operations to inverse $\tilde{\boldsymbol{\Lambda}}_{k,k}$. Accordingly, the overall computational work to inverse the matrix $\boldsymbol{\Lambda}$ is $O(N)$ operations. To inverse the matrix $\mathbf{C}_{F,2}$, we also have to compute $(\mathbf{I}_2 \otimes \mathbf{F}_2)\mathbf{u}$ and $(\mathbf{I}_2 \otimes \mathbf{F}_2^*)\mathbf{u}$ for some $\mathbf{u} \in \mathbb{R}^{2N}$. It requires $O(N \log N)$ operations by using 2DFFT and two-dimensional inverse fast Fourier transform (2DIFFM).

More precisely, we give the following procedures to solve the linear system (39):

- Evaluate the matrix $\boldsymbol{\Lambda}$ by 2DFFT. Note that it needs to be computed only once before the startup of the Krylov subspace iterative method. It requires $O(N \log N)$ operations.
- Divide the vector $\mathbf{d}$ in half. That is, $\mathbf{d} = [\mathbf{d}_1, \mathbf{d}_2]^T$, where $\mathbf{d}_1, \mathbf{d}_2 \in \mathbb{R}^N$. Carry out the 2DFFTs $\mathbf{z}_1 = \mathbf{F}_2\mathbf{d}_1$ and $\mathbf{z}_2 = \mathbf{F}_2\mathbf{d}_2$. It requires $O(N \log N)$ operations.
- For $1 \le i \le N$, we solve the matrix equation

$$\tilde{\boldsymbol{\Lambda}}_{k,k} \begin{pmatrix} \mathbf{w}_1(i) \\ \mathbf{w}_2(i) \end{pmatrix} = \begin{pmatrix} \mathbf{z}_1(i) \\ \mathbf{z}_2(i) \end{pmatrix}, \tag{42}$$

  where $\mathbf{w}_1(i)$ and $\mathbf{w}_2(i)$ are the ith elements of the vectors $\mathbf{w}_1$ and $\mathbf{w}_2$, respectively. It requires $O(N)$ operations.
- Carry out the two-dimensional inverse fast Fourier transform(2DIFFM) $\mathbf{y}_1 = \mathbf{F}_2^*\mathbf{w}_1$ and $\mathbf{y}_2 = \mathbf{F}_2^*\mathbf{w}_2$. Then we obtain the solution

$$\mathbf{y} := [\mathbf{y}_1, \mathbf{y}_2]^T. \tag{43}$$

□

It requires $O(N \log N)$ operations.

5 Numerical experiments

In this section, we carry out some numerical experiments to investigate the performance of the preconditioned fast collocation method with the BCCB-type preconditioner discussed in Sect. 3. In the numerical experiments, we consider the peridynamic model (1) with the kernel function [15]

$$\sigma\big(|(x,y)|\big) = \frac{1}{(x^2+y^2)^{1+s}}. \tag{44}$$

Let $\Omega = (0,1) \times (0,1)$. The radius of the horizon $\delta = 1/8$. The exact solution $\mathbf{u}(x,y) = (v(x,y), w(x,y))$ is chosen to be

$$v(x,y) = w(x,y) = x(1-x)y(1-y), \quad (x,y) \in \Omega.$$

We also use v and w to define g^v and g^w on the boundary zone Ω_c. The right-hand side external forcing term $\mathbf{f}(x,y)$ is computed by polar coordinates transformation as fol-

lows:

$$\begin{aligned} f^{v}(x,y) &= \frac{3\pi\delta^{2-2s}}{8-8s}\left(y-y^{2}\right)+\frac{\pi\delta^{2-2s}}{8-8s}\left(3x+2y-4xy-x^{2}-1\right)-\frac{\pi\delta^{4-2s}}{32-16s}, \\ f^{w}(x,y) &= \frac{3\pi\delta^{2-2s}}{8-8s}\left(x-x^{2}\right)+\frac{\pi\delta^{2-2s}}{8-8s}\left(2x+3y-4xy-y^{2}-1\right)-\frac{\pi\delta^{4-2s}}{32-16s}. \end{aligned} \tag{45}$$

For simplicity, we choose $h_x = h_y = h$.

We solve linear equation (8) by the conjugate gradient squared method (CGS), the fast conjugate gradient squared method (FCGS) without any preconditioners, and the fast conjugate gradient squared method with BCCB-type preconditioner (BCCB-FCGS). The following numerical experiments are performed on a computer with 8-GB memory. The accuracy requirement for iteration termination is 10^{-8}.

Example 1 In this numerical example, we choose $s = 3/8$ in (44). We present the L^2 errors of the numerical solutions generated by the CGS solver, the FCGS solver, and the BCCB-FCGS solver for the mesh size from $1/2^4$ to $1/2^9$ in Table 1. We also present the CPU time consumed by these solvers and the number of iterations in the iterative methods in Table 1. In addition to the advantages of fast methods (FCGS and BCCB-FCGS) over traditional method (CGS) in memory requirement and CPU time, we observe from Table 1 that the BCCB-type preconditioner can significantly reduce the number of iterations and CPU time in contrast to FCGS without any preconditioners. For example, while h is equal to 2^9, the FCGS solver takes 455 iterations and nearly 3 hours of CPU time, but the BCCB solver needs only 58 iterations and nearly 8 minutes of CPU time without any loss of accuracy. However, while we carried out the numerical test with $h = 1/2^{10}$, we found a small increase in L^2 error. This small increase may be caused by singular integral when we calculate the main-diagonal entries of $\mathbf{A}^{v,v}$, $\mathbf{A}^{v,w}$, and $\mathbf{A}^{w,w}$.

In this numerical experiment, we also use a linear regression to fit the convergence rate α and the associated constant C_α in the error estimate

$$\|\mathbf{u}-\mathbf{u_h}\|_{L^2(\Omega)} \le C_\alpha h^\alpha. \tag{46}$$

We also present these results in Table 1. We see that the convergence rate α is sublinear.

Example 2 We choose $s = 0$ in (44). We present the corresponding numerical results in Table 2 as in Example 1. We have the similar observations as in Example 1. However, in comparison with the results showed in Table 1, an improved accuracy of the numerical solutions and a reduced number of iterations in these three iteration methods are observed in Table 2.

6 Conclusions

Firstly, a fast bilinear collocation method for a static bond-based peridynamic model was developed by reordering the unknowns, which is different from [18], and by carefully analyzing the structure of the stiffness matrix. This work significantly reduces the computational work to solve the resulting linear system from $O(N^2)$ to $O(N\log N)$ per Krylov

Table 1 Performance of CGS, FCGS, and BCCB-FCGS in Example 1

	h	$\|\mathbf{u}_h - \mathbf{u}\|_{L^2}$	# of iter.	CPUs
CGS	$1/2^4$	3.2905×10^{-2}	46	0.1 s
	$1/2^5$	2.3835×10^{-2}	85	1.28 s
	$1/2^6$	1.4337×10^{-2}	140	33.57 s
	$1/2^7$	7.4180×10^{-3}	215	>10 h
	$1/2^8$		out of memory	
	$1/2^9$		out of memory	
		$C_\alpha = 0.26, \alpha = 0.60$		
FCGS	$1/2^4$	3.2905×10^{-2}	47	0.95 s
	$1/2^5$	2.3835×10^{-2}	85	4.5 s
	$1/2^6$	1.4337×10^{-2}	140	22.2 s
	$1/2^7$	7.4180×10^{-3}	215	2 m 38 s
	$1/2^8$	5.8471×10^{-3}	302	13 m 50 s
	$1/2^9$	2.4611×10^{-3}	455	2 h 58 m 13 s
		$C_\alpha = 0.28, \alpha = 0.73$		
BCCB-FCGS	$1/2^4$	3.2899×10^{-2}	20	1.08 s
	$1/2^5$	2.3816×10^{-2}	28	2.2 s
	$1/2^6$	1.4306×10^{-2}	36	9.8 s
	$1/2^7$	7.3757×10^{-3}	44	32.8 s
	$1/2^8$	5.8005×10^{-3}	51	2 m 3 s
	$1/2^9$	2.4611×10^{-3}	58	7 m 48 s
	$1/2^{10}$	2.7433×10^{-3}	76	1 h 13 m 16 s
		$C_\alpha = 0.21, \alpha = 0.66$		

Table 2 Performance of CGS, FCGS, and BCCB-FCGS in Example 2

	h	$\|\mathbf{u}_h - \mathbf{u}\|_{L^2}$	# of iter .	CPUs
CGS	$1/2^4$	1.6387×10^{-2}	35	0.04 s
	$1/2^5$	6.8624×10^{-3}	48	0.93 s
	$1/2^6$	2.3722×10^{-3}	59	18.05 s
	$1/2^7$	9.4000×10^{-4}	78	>10 h
	$1/2^8$		out of memory	
	$1/2^9$		out of memory	
		$C_\alpha = 0.81, \alpha = 1.39$		
FCGS	$1/2^4$	1.6387×10^{-2}	35	0.8 s
	$1/2^5$	6.8624×10^{-3}	49	3.4 s
	$1/2^6$	2.3722×10^{-3}	59	15.9 s
	$1/2^7$	9.4000×10^{-4}	78	1 m 10 s
	$1/2^8$	7.2921×10^{-4}	99	4 m 11 s
	$1/2^9$	2.2423×10^{-4}	113	38 m 58 s
		$C_\alpha = 0.41, \alpha = 1.20$		
BCCB-FCGS	$1/2^4$	1.6374×10^{-2}	19	1 s
	$1/2^5$	6.8154×10^{-3}	23	2.4 s
	$1/2^6$	2.3057×10^{-3}	26	8.8 s
	$1/2^7$	8.6494×10^{-4}	29	22.9 s
	$1/2^8$	6.9507×10^{-4}	34	1 m 18 s
	$1/2^9$	1.8282×10^{-4}	31	8 m 24 s
	$1/2^{10}$	8.6238×10^{-4}	36	1 h 2 m 57 s
		$C_\alpha = 0.13, \alpha = 0.89$		

subspace iteration. It also reduces storage of the stiffness matrix from $O(N^2)$ to $O(N)$. However, the number of iterations in the Krylov subspace iteration method seems to be large especially for the peridynamic model in which the kernel function has stronger singularity. In this paper, using the structure of the stiffness matrix, we construct an efficient BCCB-type preconditioner to improve the performance of the convergence behavior of

the fast collocation method. The numerical experiments show the efficiency of this preconditioner.

Acknowledgements
The authors are grateful to the editor and the anonymous referees for their valuable comments and suggestions on the paper.

Funding
This work was supported in part by the National Natural Science Foundation of China under Grants 91630207, 11971272.

Availability of data and materials
All of the authors declare that all the data can be accessed in our manuscript in the numerical simulation section.

Competing interests
The authors declare that there is no conflict of interests.

Authors' contributions
All authors read and approved the final manuscript.

Author details
[1]School of Mathematics and Statistics, Qilu University of Technology, Jinan, China. [2]School of Mathematics, Shandong University, Jinan, China. [3]Department of Mathematics, University of South Carolina, Columbia, USA.

Publisher's Note

Received: 3 April 2020 Accepted: 17 May 2020 Published online: 27 May 2020

References

1. Oterkus, E., Madenci, E., Weckner, O., Silling, S., Bogert, P.: Combined finite element and peridynamic analysis for predicting failure in a stiffened composite curved panel with a central slot. Compos. Struct. **94**, 839–850 (2012)
2. Kilic, B., Agwai, A., Madenci, E.: Peridynamic theory for progressive damage prediction in centercracked composite laminates. Compos. Struct. **90**, 141–151 (2009)
3. Ha, Y.D., Bobaru, F.: Characteristics of dynamic brittle fracture captured with peridynamics. Eng. Fract. Mech. **78**, 1156–1168 (2011)
4. Silling, S., Weckner, O., Askari, E., Bobaru, F.: Crack nucleation in a peridynamic solid. Int. J. Fract. **162**, 219–227 (2010)
5. Dayal, K., Bhattacharya, K.: Kinetics of phase transformations in the peridynamic formulation of continuum mechanics. J. Mech. Phys. Solids **54**, 1811–1842 (2006)
6. Bobaru, F., Ha, Y.D., Hu, W.: Damage progression from impact in layered glass modeled with peridynamics. Cent. Eur. J. Eng. **2**, 551–561 (2012)
7. Gerstle, W., Sau, N., Silling, S.: Peridynamic modeling of concrete structures. Nucl. Eng. Des. **237**, 1250–1258 (2007)
8. Li, T., Pintus, N., Viglialoro, G.: Properties of solutions to porous medium problems with different sources and boundary conditions. Z. Angew. Math. Phys. **70**, 1–18 (2019)
9. Shah, R., Li, T.: The thermal and laminar boundary layer flow over prolate and oblate spheroids. Int. J. Heat Mass Transf. **121**, 607–619 (2018)
10. Jiang, C., Zada, A., Şenel, M.T., Li, T.: Synchronization of bidirectional *N*-coupled fractional-order chaotic systems with ring connection based on antisymmetric structure. Adv. Differ. Equ. **2019**, 456 (2019)
11. Chen, X., Gunzburger, M.: Continuous and discontinuous finite element methods for a peridynamics model of mechanics. Comput. Methods Appl. Mech. Eng. **200**, 1237–1250 (2011)
12. Du, Q., Ju, L., Tian, L., Zhou, K.: A posteriori error analysis of finite element methods linear nonlocal diffusion and peridynamic models. Math. Comput. **82**, 1889–1922 (2013)
13. Seleson, P.: Improved one-point quadrature algorithms for two-dimensional peridynamic models based on analytical calculations. Comput. Methods Appl. Mech. Eng. **282**, 184–217 (2014)
14. Seleson, P., Littlewood, D.: Convergence studies in meshfree peridynamic simulations. Comput. Math. Appl. **71**, 2432–2448 (2016)
15. Wang, H., Tian, H.: A fast and faithful collocation method with efficient matrix assembly for a two-dimensional nonlocal diffusion model. Comput. Methods Appl. Mech. Eng. **273**, 19–36 (2014)
16. Wang, H., Tian, H.: A fast Galerkin method with efficient matrix assembly and storage for a peridynamic model. J. Comput. Phys. **231**, 7730–7738 (2012)
17. Zhang, X., Gunzburger, M., Ju, L.: Nodal-type collocation methods for hypersingular integral equations and nonlocal diffusion problems. Comput. Methods Appl. Mech. Eng. **299**, 401–420 (2016)
18. Zhang, X., Wang, H.: A fast collocation method for a static bond-based linear peridynamic model. Comput. Methods Appl. Mech. Eng. **311**, 280–303 (2016)
19. Ng, M.K.: Iterative Methods for Toeplitz Systems. Oxford University Press, Oxford (2004)
20. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM, Rhode Island (2003)
21. Fu, H., Wang, H.: A preconditioned fast finite difference method for space-time fractional partial differential equations. Fract. Calc. Appl. Anal. **20**, 88–116 (2017)