Zhang and Cheng *Advances in Difference Equations* (2020) 2020:479
https://doi.org/10.1186/s13662-020-02931-3

Advances in Difference Equations
a SpringerOpen Journal

RESEARCH Open Access

Fast collocation method for a two-dimensional variable-coefficient linear nonlocal diffusion model

Xuhao Zhang[1*] and Aijie Cheng[2]

*Correspondence: zhanghaotimmy@163.com
[1] School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan, 250100, China
Full list of author information is available at the end of the article

Abstract

In this paper, a fast collocation method is developed for a two-dimensional variable-coefficient linear nonlocal diffusion model. By carefully dealing with the variable coefficient in the integral operator and then analyzing the structure of the coefficient matrix, we can reduce the computational operations in each Krylov subspace iteration from $O(N^2)$ to $O(N \log N)$ and the memory requirement for the coefficient matrix from $O(N^2)$ to $O(N)$. Numerical experiments are carried out to show the utility of the fast collocation method.

Keywords: Nonlocal diffusion model; Variable coefficient; Fast collocation method

1 Introduction

Recently, nonlocal models such as fractional partial differential equations (FPDEs) and nonlocal diffusion and peridynamic models have been applied in many research fields. These models can be regarded as a generalization of classical PDEs [1–4] and have shown the utility in modeling some challenging phenomena including anomalous diffusion and long-range spatial interactions [5–13]. These phenomena cannot be modeled properly by classical integer-order PDEs. Especially speaking, nonlocal diffusion and peridynamic models provide a more natural method to describe physical problems especially near singularities and discontinuities [14]. However, because of the nonlocality, the numerical algorithms usually yield dense or full coefficient matrices, which can cause significantly increased computational complexity and memory storage. For example, widely used direct solvers need $O(N^3)$ operations to solve the linear system and $O(N^2)$ computer memory space to store the coefficient matrix in which N is the number of unknowns.

To date, there have been many papers aimed to develop fast numerical schemes for nonlocal models to reduce the computational complexity and the memory requirement [13, 15–21]. Among these papers, in [18], the authors developed a fast collocation method for a two-dimensional linear nonlocal diffusion model. We developed a fast-method in [19] and a preconditioned fast collocation method in [20] for two-dimensional linear bond-based peridynamic model, respectively. All these papers are based on the Toeplitz-like structure of the coefficient matrix. This structure can help us to reduce the computational work from $O(N^2)$ to $O(N \log N)$ in each Krylov subspace iteration and the memory storage from $O(N^2)$ to $O(N)$. Numerical examples in these papers show the utility of the fast methods.

In [22], a variable-coefficient peridynamic model was developed to account for the heterogeneity of the elastic material. However, because the variable coefficient occurs inside the integral operator, the coefficient matrix resulting from the numerical discretization does not maintain the Toeplitz-like structure as in [18, 19]. Therefore the fast methods developed before do not apply to the variable-coefficient peridynamic model. To overcome this difficulty, [23] developed a fast collocation method to a one-dimensional variable-coefficient nonlocal diffusion model based on a piecewise-constant approximation to the variable coefficient.

In this paper, a fast collocation method is developed for the two-dimensional variable-coefficient linear nonlocal diffusion model based on Taylor expansion of the variable coefficient. The rest of the paper is organized as follows:. In Sect. 2, we present the bilinear collocation scheme for the two-dimensional variable-coefficient linear nonlocal diffusion model. In Sect. 3, we study the structure of the coefficient matrix and prove that the coefficient matrix can be approximated by a sum of the products of diagonal matrix and Toeplitz matrix. Then fast matrix-vector multiplication and the reduced memory requirement can be proved, which can be used in the Krylov subspace iteration method. In Sect. 4, we do numerical experiments to investigate the computational benefits of the fast collocation method. In Sect. 5, we make the concluding remarks.

2 The bilinear collocation approximation for variable-coefficient linear nonlocal diffusion model

To begin with, we consider the following two-dimensional variable-coefficient static linear nonlocal diffusion model[22]:

$$\begin{cases} \int_{B_\delta(x,y)} (\alpha(x',y') + \alpha(x,y))\sigma(x-x',y-y')(u(x,y) - u(x',y'))\,dx'\,dy' \\ \quad = f(x,y), \quad (x,y) \in \Omega, \\ u(x,y) = g(x,y), \quad (x,y) \in \Omega_c. \end{cases} \tag{1}$$

Here $\Omega = (0,x_R) \times (0,y_R)$; δ represents a parameter of the model which determines the range of interactions; Ω_c denotes a boundary zone surrounding Ω with width δ; $B_\delta(x,y)$ is an open disk in which the center is located at (x,y) and the radius is δ. $\sigma(x,y)$ denotes the integral kernel; $f(x,y)$ and $g(x,y)$ represent the source term and prescribed nonlocal boundary data, respectively; $\alpha(x,y)$ is the elasticity coefficient with lower and upper bounds.

Let N_x and N_y represent the number of mesh grids in the x and y directions, respectively. Let $h_x := x_R/N_x$ be the mesh size in the x direction. Let $h_y := y_R/N_y$ be the mesh size in the y direction. We define a spatial partition on $\bar{\Omega}$ by $x_i := ih_x$ for $i = 0,1,\dots,N_x$ and $y_j := jh_y$ for $j = 0,1,\dots,N_y$. To account for the nodes on Ω_c, we need to extend the partition to (x_i,y_j) for $i = -K+1,\dots,-1,0,1,\dots,N_x,N_x+1,\dots,N_x+K-1$ and $j = -L+1,\dots,-1,0,1,\dots,N_y,N_y+1,\dots,N_y+L-1$. Here,

$$K = \left\lceil \frac{\delta}{h_x} \right\rceil, \qquad L = \left\lceil \frac{\delta}{h_y} \right\rceil. \tag{2}$$

Let $\psi(\xi)$ be equal to $1-|\xi|$ when $\xi \in [-1,1]$ and equal to zero otherwise. Then the pyramid function $\phi_{ij}(x,y)$ centered at (x_i,y_j) can be expressed as

$$\phi_{ij}(x,y) = \psi\left(\frac{x-x_i}{h_x}\right)\psi\left(\frac{y-y_j}{h_y}\right), \quad -K \le i \le N_x + K, -L \le j \le N_y + L. \tag{3}$$

Then the trial function u is

$$u(x,y) = \sum_{i'=-K}^{N_x+K}\sum_{j'=-L}^{N_y+L} u_{i',j'}\phi_{i',j'}(x,y). \tag{4}$$

Because for the spatial nodes $(x_i,y_j) \in \Omega_c$, $u(x_i,y_j) = g(x_i,y_j)$ are known, we choose the spatial nodes $(x_i,y_j) \in \Omega$ as the collocation points. Then we obtain a collocation scheme as follows:

$$\begin{aligned}&\int_{B_\delta(x_i,y_j)} \left(\alpha\left(x',y'\right) + \alpha(x_i,y_j)\right)\sigma\left(x_i - x', y_j - y'\right)\left(u(x_i,y_j) - u\left(x',y'\right)\right) dx'\,dy' \\ &\quad = f(x_i,y_j), \quad 1 \le i \le N_x - 1, 1 \le j \le N_y - 1. \end{aligned}\tag{5}$$

We substitute (4) into (5) and rewrite (5) as follows:

$$\begin{aligned}&\alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)} \sigma\left(x_i - x', y_j - y'\right)\left(u_{i,j} - \sum_{i'=-K+1}^{N_x+K-1}\sum_{j'=-L+1}^{N_y+L-1} u_{i',j'}\phi_{i',j'}\left(x',y'\right)\right) dx'\,dy' \\ &\qquad + \int_{B_\delta(x_i,y_j)} \alpha\left(x',y'\right)\sigma\left(x_i - x', y_j - y'\right)\left(u_{i,j} - \sum_{i'=-K+1}^{N_x+K-1}\sum_{j'=-L+1}^{N_y+L-1} u_{i',j'}\phi_{i',j'}\left(x',y'\right)\right) dx'\,dy' \\ &\quad = f_{i,j}, \quad 1 \le i \le N_x - 1, 1 \le j \le N_y - 1, \end{aligned}\tag{6}$$

where $f_{i,j} = f(x_i,y_j)$ and u_{ij} is the collocation approximation to the true solution $u(x_i,y_j)$.

Let $N := (N_x - 1) \times (N_y - 1)$ be the number of collocation points. Let $\mathbf{u}$ and $\mathbf{f}$ be N-dimensional vectors defined by

$$\mathbf{u} := [u_{1,1},\ldots,u_{N_x-1,1},u_{1,2},\ldots,u_{N_x-1,2},u_{1,N_y-1},\ldots,u_{N_x-1,N_y-1}]^T \tag{7}$$

and

$$\mathbf{f} := [f_{1,1},\ldots,f_{N_x-1,1},f_{1,2},\ldots,f_{N_x-1,2},f_{1,N_y-1},\ldots,f_{N_x-1,N_y-1}]^T, \tag{8}$$

respectively.

Then the collocation approximation (6) can be expressed in the following matrix form:

$$(\mathbf{A}_1 + \mathbf{A}_2)\mathbf{u} = \mathbf{f}. \tag{9}$$

Here, $\mathbf{A}_1 \in \mathbb{R}^{N\times N}$ and $\mathbf{A}_2 \in \mathbb{R}^{N\times N}$ are associated with the first term and the second term on the left-hand side of (6), respectively.

For the matrix $\mathbf{A}_1$, each entry $A_1^{m,n}$ for $m = 1,\dots,N$ and $n = 1,\dots,N$ is defined as

$$A_1^{m,n} = \alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)} \sigma\left(x_i - x', y_j - y'\right)\left(\delta_{m,n} - \phi_{i',j'}\left(x',y'\right)\right) dx'\,dy'. \tag{10}$$

For the matrix $\mathbf{A}_2$, each entry $A_2^{m,n}$ for $m = 1,\dots,N$ and $n = 1,\dots,N$ is given by

$$A_2^{m,n} = \int_{B_\delta(x_i,y_j)} \alpha\left(x',y'\right)\sigma\left(x_i - x', y_j - y'\right)\left(\delta_{m,n} - \phi_{i',j'}\left(x',y'\right)\right) dx'\,dy'. \tag{11}$$

In (10) and (11), $\delta_{m,n}$ is equal to 1 for $m = n$ and equal to zero otherwise. The elements $\{f_m, m = 1,\dots,N\}$ of $\mathbf{f}$ on the right hand side in (9) are given by

$$\begin{aligned} f_m = f_{i,j} &+ \sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}} \sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}} \alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)} \\ &\times \sigma\left(x_i - x', y_j - y'\right) g(x_{i''},y_{j''})\phi_{i'',j''}\left(x',y'\right) dx'\,dy' \\ &+ \sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}} \sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}} \int_{B_\delta(x_i,y_j)} \alpha\left(x',y'\right) \\ &\times \sigma\left(x_i - x', y_j - y'\right) g(x_{i''},y_{j''})\phi_{i'',j''}\left(x',y'\right) dx'\,dy'. \end{aligned} \tag{12}$$

In (10), (11), and (12), the global indices (m,n) and the local indices (i,j), (i',j') have the following relationship:

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

3 Structure of the coefficient matrices A_1 and A_2 in (9)

To construct a fast collocation method, we are now in a position to analyse the structure of the coefficient matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ in (9).

Theorem 1 *The matrix* $\mathbf{A}_1$ *can be expressed as*

$$\mathbf{A}_1 = \mathbf{D}_1\mathbf{B}_1. \tag{14}$$

Here $\mathbf{D}_1$ *is a diagonal matrix and the entries on the main diagonal are given by*

$$D_1^{m,m} = \alpha(x_i,y_j), \quad 1\le i\le N_x-1, 1\le j\le N_y-1. \tag{15}$$

$\mathbf{B}_1$ *is a block-Toeplitz–Toeplitz-block* (*BTTB*) *matrix. More precisely,* $\mathbf{B}_1$ *is a* (N_y-1)*-by-*(N_y-1) *banded block-Toeplitz matrix with bandwidth* $(2L+1)$

$$\mathbf{B}_1 = \begin{pmatrix} \mathbf{B}_1^0 & \cdots & \mathbf{B}_1^L & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{B}_1^{-L} & \ddots & \mathbf{B}_1^0 & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{B}_1^0 & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{B}_1^0 & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{B}_1^0 & \ddots & \mathbf{B}_1^L \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{B}_1^{-L} & \cdots & \mathbf{B}_1^0 \end{pmatrix}. \tag{16}$$

In (16), *each block matrix* $\mathbf{B}_1^j, -L \le j \le L$ *is a* (N_x-1)*-by-*(N_x-1) *banded Toeplitz matrix with bandwidth* $2K+1$

$$\mathbf{B}_1^j = \begin{pmatrix} b_1^{0,j} & \cdots & b_1^{K,j} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ b_1^{-K,j} & \ddots & b_1^{0,j} & \ddots & \ddots & 0 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & b_1^{0,j} & \ddots & \ddots & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & \ddots & \ddots & b_1^{0,j} & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & \ddots & \ddots & t_1^{0,j} & \ddots & b_1^{K,j} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & b_1^{-K,j} & \cdots & b_1^{0,j} \end{pmatrix}. \tag{17}$$

Proof By a straightforward calculation and analysis of $\mathbf{A}_1$, we can find that

$$\mathbf{A}_1 = \mathbf{D}_1\mathbf{B}. \tag{18}$$

The entries in $\mathbf{B}$ are given by

$$\mathbf{B}^{m,n} = \int_{B_\delta(x_i,y_j)} \sigma\left(x_i - x', y_j - y'\right)\left(\delta_{m,n} - \phi_{i',j'}\left(x',y'\right)\right) dx'\, dy', \tag{19}$$

where the local indices (i,j), (i',j') and the global indices m, n are related by (13). Following the proof of Theorem 1 in [24], we see that $\mathbf{B}$ is a BTTB matrix and can be expressed as $\mathbf{B}_1$. □

Now we consider the structure of matrix $\mathbf{A}_2$. From (11), we observe that the variable coefficient occurs inside the integral, which brings a global impact to the numerical discretization. If we calculate the matrix $\mathbf{A}_2$ straightforwardly, we find $\mathbf{A}_2$ is also a $2K+1$

block-banded $2L+1$ banded-block matrix but destroys the BTTB structure. Therefore if we use this structure to solve the linear system, the computational complexity and memory requirement will not decrease.

In this paper, we consider the following approximate way. That is, although the original matrix $\mathbf{A}_2$ does not have the BTTB structure, it can be approximated by a sum of products of diagonal matrix and BTTB matrix if the variable coefficient $\alpha(x,y)$ is approximated by Taylor expansion. We denote the sum by $\mathbf{A}_2^a$.

Theorem 2 *The matrix* $\mathbf{A}_2$ *can be approximately decomposed as*

$$\mathbf{A}_2^a = \mathbf{D}_1\mathbf{B}_1 + \mathbf{D}_2\mathbf{B}_2 + \mathbf{D}_3\mathbf{B}_3. \tag{20}$$

Here $\mathbf{D}_1$ *and* $\mathbf{B}_1$ *are given in the Theorem 1.* $\mathbf{D}_2$ *and* $\mathbf{D}_3$ *are diagonal matrices where the main diagonal entries are defined as*

$$D_2^{m,m} = \frac{\partial\alpha(x_i,y_j)}{\partial x}, \quad 1 \le i \le N_x - 1, 1 \le j \le N_y - 1, \tag{21}$$

and

$$D_3^{m,m} = \frac{\partial\alpha(x_i,y_j)}{\partial y}, \quad 1 \le i \le N_x - 1, 1 \le j \le N_y - 1, \tag{22}$$

respectively. $\mathbf{B}_2$ *and* $\mathbf{B}_3$ *have the same structure as* $\mathbf{B}_1$. *The definition of* $\mathbf{B}_2$ *and* $\mathbf{B}_3$ *can be easily obtained by replacing the subscript* 1 *by* 2 *and* 3 *in* (16) *and* (17), *respectively.*

Proof Using Taylor expansion, the variable coefficient $\alpha(x',y')$ can be expanded to the following form at the point (x_i,y_j):

$$\alpha\left(x',y'\right) \approx \alpha(x_i,y_j) + \frac{\partial\alpha(x_i,y_j)}{\partial x}\left(x'-x_i\right) + \frac{\partial\alpha(x_i,y_j)}{\partial y}\left(y'-y_j\right). \tag{23}$$

Substituting (23) into (11), each entry of $\mathbf{A}_1$ can be rewritten as

$$\begin{aligned}
A_2^{m,n} &= \int_{B_\delta(x_i,y_j)} \alpha\left(x',y'\right)\sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy' \\
&\approx \alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)} \sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy' \\
&\quad + \frac{\partial\alpha(x_i,y_j)}{\partial x}\int_{B_\delta(x_i,y_j)} \left(x'-x_i\right)\sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy' \\
&\quad + \frac{\partial\alpha(x_i,y_j)}{\partial y}\int_{B_\delta(x_i,y_j)} \left(y'-y_j\right)\sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy' \\
&= I_1^{m,n} + I_2^{m,n} + I_3^{m,n},
\end{aligned} \tag{24}$$

where

$$
\begin{aligned}
I_1^{m,n} &= \alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)} \sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy',\\
I_2^{m,n} &= \frac{\partial\alpha(x_i,y_j)}{\partial x}\int_{B_\delta(x_i,y_j)} \left(x'-x_i\right)\sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy',\\
I_3^{m,n} &= \frac{\partial\alpha(x_i,y_j)}{\partial y}\int_{B_\delta(x_i,y_j)} \left(y'-y_j\right)\sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy'.
\end{aligned}
\tag{25}
$$

We first consider the term $I_1^{m,n}$. Because $I_1^{m,n}$ is the same as $A_1^{m,n}$, the matrix form of $I_1^{m,n}$ for $1 \le i \le N_x - 1, 1 \le j \le N_y - 1$ is equal to $\mathbf{D}_1\mathbf{B}_1$ given in Theorem 1. Here the global indices (m,n) and the local indices (i,j), (i',j') are related by (13).

Next, we only prove the structure of matrix generated by $I_2^{m,n}$. The structure of matrix generated by $I_3^{m,n}$ is similar. Let the matrix form of $I_2^{m,n}$, with $1 \le i \le N_x-1, 1 \le j \le N_y-1$, be denoted by $\mathbf{A}$. By a direct analysis, we have

$$
\mathbf{A} = \mathbf{D}_2\mathbf{M}, \tag{26}
$$

where $\mathbf{D}_2$ is defined in (21) and each entry of $\mathbf{M}$ is defined as

$$
M^{m,n} = \int_{B_\delta(x_i,y_j)} \left(x'-x_i\right)\sigma\left(x_i-x',y_j-y'\right)\left(\delta_{m,n}-\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy'. \tag{27}
$$

Furthermore, $\mathbf{M}$ is of the form

$$
\mathbf{M} = \begin{pmatrix}
\mathbf{M}^{1,1} & \mathbf{M}^{1,2} & \cdots & \mathbf{M}^{1,N_y-1}\\
\mathbf{M}^{2,1} & \mathbf{M}^{2,2} & \ddots & \mathbf{M}^{2,N_y-1}\\
\vdots & \ddots & \ddots & \vdots\\
\mathbf{M}^{N_y-1,1} & \mathbf{M}^{N_y-1,2} & \cdots & \mathbf{M}^{N_y-1,N_y-1}
\end{pmatrix}. \tag{28}
$$

Here each matrix block $\mathbf{M}^{j,j'}, 1 \le j \le N_y - 1, 1 \le j' \le N_y - 1$ is of order $N_x - 1$ and can be expressed as

$$
\mathbf{M}^{j,j'} = \begin{pmatrix}
M_{1,1}^{j,j'} & M_{1,2}^{j,j'} & \cdots & M_{1,N_x-1}^{j,j'}\\
M_{2,1}^{j,j'} & M_{2,2}^{j,j'} & \ddots & M_{2,N_x-1}^{j,j'}\\
\vdots & \ddots & \ddots & \vdots\\
M_{N_x,1}^{j,j'} & M_{N_x-1,2}^{j,j'} & \cdots & M_{N_x-1,N_x-1}^{j,j'}
\end{pmatrix}. \tag{29}
$$

We now analyze the structure of $\mathbf{M}$. From (13) and the second equation in (25), we have $I_2^{m,n} \neq 0$ if and only if

$$
B_\delta(x_i,y_j) \cap \operatorname{supp}(\phi_{i',j'})) \neq \emptyset. \tag{30}
$$

From (28) and (30), each matrix block $\mathbf{M}^{j,j'}$ satisfying $|j-j'|>L$ becomes zero matrix. Thus $\mathbf{M}$ has a block-banded structure given by

$$\mathbf{M}=\begin{pmatrix} \mathbf{M}^{1,1} & \cdots & \mathbf{M}^{1,L+1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{M}^{L+1,1} & \ddots & \ddots & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \mathbf{M}^{N_y-L-1,N_y-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{M}^{N_y-1,N_y-L-1} & \cdots & \mathbf{M}^{N_y-1,N_y-1} \end{pmatrix}. \tag{31}$$

From (29) and (30), we also observe that all the entries $M^{j,j'}_{i,i'}$ with $|i-i'|>K$ in $\mathbf{M}^{j,j'}$ vanish. That is, $\mathbf{M}^{j,j'}$ satisfying $|j-j'|\le L$ has a banded structure expressed as

$$\mathbf{M}^{j,j'}=\begin{pmatrix} m^{j,j'}_{1,1} & \cdots & m^{j,j'}_{1,K+1} & 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ m^{j,j'}_{K+1,1} & \ddots & \ddots & \ddots & \ddots & 0 & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \ddots & \ddots & 0 & \ddots & \ddots & \ddots & \ddots & m^{j,j'}_{N_x-K-1,N_x-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 & m^{j,j'}_{N_x-1,N_x-K-1} & \cdots & m^{j,j'}_{N_x-1,N_x-1} \end{pmatrix}. \tag{32}$$

By the transformations

$$\xi_1=x'-x_i, \qquad \xi_2=y'-y_j,$$

the pyramid function given in (2) can be written as

$$\begin{aligned}\phi_{i',j'}(x',y') &= \psi\left(\frac{\xi_1-x_{i'-i}}{h_x}\right)\psi\left(\frac{\xi_2-y_{j'-j}}{h_y}\right) \\ &= \phi_{i'-i,j'-j}(\xi_1,\xi_2).\end{aligned}$$

Then from Eq. (27) can be deduced

$$M^{m,n}=\int_{B_\delta(0,0)}\xi_1\sigma(-\xi_1,-\xi_2)\big(\delta_{m,n}-\phi_{i'-i,j'-j}(\xi_1,\xi_2)\big)\,d\xi_1\,d\xi_2. \tag{33}$$

Let $j_1' - j_1 = j_2' - j_2 = l, -L \le l \le L$. That is, the matrix blocks $\mathbf{M}^{j_1,j_1'}$ and $\mathbf{M}^{j_2,j_2'}$ are on the same diagonal. Let

$$\begin{aligned} m_1 &= (j_1-1)(N_x-1)+i, \quad 1\le i\le N_x-1, 1\le j_1\le N_y-1,\\ n_1 &= (j_1'-1)(N_x-1)+i', \quad 1\le i'\le N_x-1, 1\le j_1'\le N_y-1,\\ m_2 &= (j_2-1)(N_x-1)+i, \quad 1\le i\le N_x-1, 1\le j_2\le N_y-1,\\ n_2 &= (j_2'-1)(N_x-1)+i', \quad 1\le i'\le N_x-1, 1\le j_2'\le N_y-1. \end{aligned} \tag{34}$$

Then we have

$$\begin{aligned} m_{i,i'}^{j_1,j_1'} &= M^{m_1,n_1}\\ &= \int_{B_\delta(0,0)} \xi_1\sigma(-\xi_1,-\xi_2)\bigl(\delta_{m_1,n_1} - \phi_{i'-i,j_1'-j_1}(\xi_1,\xi_2)\bigr)\,d\xi_1\,d\xi_2\\ &= \int_{B_\delta(0,0)} \xi_1\sigma(-\xi_1,-\xi_2)\bigl(\delta_{m_2,n_2} - \phi_{i'-i,j_2'-j_2}(\xi_1,\xi_2)\bigr)\,d\xi_1\,d\xi_2\\ &= M^{m_2,n_2} = m_{i,i'}^{j_2,j_2'}, \quad 1\le i,i'\le N_x-1. \end{aligned} \tag{35}$$

According to (35), we have proved $\mathbf{M}^{j_1,j_1'} = \mathbf{M}^{j_2,j_2'}$.

Let $i_3' - i_3 = i_4' - i_4 = k, -K \le k \le K$. That is, the entries $m_{i_3,i_3'}^{j,j'}$ and $m_{i_4,i_4'}^{j,j'}$ in each matrix block $\mathbf{M}^{j,j'}$ are on the same diagonal. Let

$$\begin{aligned} m_3 &= (j-1)(N_x-1)+i_3, \quad 1\le i_3\le N_x-1, 1\le j\le N_y-1,\\ n_3 &= (j'-1)(N_x-1)+i_3', \quad 1\le i_3'\le N_x-1, 1\le j'\le N_y-1,\\ m_4 &= (j-1)(N_x-1)+i_4, \quad 1\le i_4\le N_x-1, 1\le j\le N_y-1,\\ n_4 &= (j'-1)(N_x-1)+i_4', \quad 1\le i_4'\le N_x-1, 1\le j'\le N_y-1. \end{aligned} \tag{36}$$

Then we observe that

$$\begin{aligned} m_{i_3,i_3'}^{j,j'} &= M^{m_3,n_3}\\ &= \int_{B_\delta(0,0)} \xi_1\sigma(-\xi_1,-\xi_2)\bigl(\delta_{m_3,n_3} - \phi_{i_3'-i_3,j'-j}(\xi_1,\xi_2)\bigr)\,d\xi_1\,d\xi_2\\ &= \int_{B_\delta(0,0)} \xi_1\sigma(-\xi_1,-\xi_2)\bigl(\delta_{m_4,n_4} - \phi_{i_4'-i_4,j'-j}(\xi_1,\xi_2)\bigr)\,d\xi_1\,d\xi_2\\ &= M^{m_4,n_4} = m_{i_4,i_4'}^{j,j'}. \end{aligned} \tag{37}$$

Combining (35) and (37), we conclude that the matrix defined in (26) is a BTTB matrix. □

Using the Taylor expansion (23), the collocation scheme (6) can be changed to an approximate form,

$$2\alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)} \sigma\bigl(x_i-x',y_j-y'\bigr)\left(\hat{u}_{i,j} - \sum_{i'=-K}^{N_x+K}\sum_{j'=-L}^{N_y+L} \hat{u}_{i',j'}\phi_{i',j'}\bigl(x',y'\bigr)\right)dx'\,dy'$$

$$
\begin{aligned}
&+\frac{\partial\alpha(x_i,y_j)}{\partial x}\int_{B_\delta(x_i,y_j)}\sigma\left(x_i-x',y_j-y'\right)\left(\hat{u}_{i,j}-\sum_{i'=-K}^{N_x+K}\sum_{j'=-L}^{N_y+L}\hat{u}_{i',j'}\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy'\\
&+\frac{\partial\alpha(x_i,y_j)}{\partial y}\int_{B_\delta(x_i,y_j)}\sigma\left(x_i-x',y_j-y'\right)\left(\hat{u}_{i,j}-\sum_{i'=-K}^{N_x+K}\sum_{j'=-L}^{N_y+L}\hat{u}_{i',j'}\phi_{i',j'}\left(x',y'\right)\right)dx'\,dy'\\
={}&f_{i,j},\quad 1\le i\le N_x-1,1\le j\le N_y-1,
\end{aligned}
\tag{38}
$$

where $\hat{u}_{i,j}$ is the alternative collocation approximation to the true solution $u(x_i,y_j)$. The matrix form of (38) can be written as

$$
\hat{\mathbf{A}}\hat{\mathbf{u}}=(2\mathbf{D}_1\mathbf{B}_1+\mathbf{D}_2\mathbf{B}_2+\mathbf{D}_3\mathbf{B}_3)\hat{\mathbf{u}}=\hat{\mathbf{f}},
\tag{39}
$$

where

$$
\hat{\mathbf{u}}:=[\hat{u}_{1,1},\dots,\hat{u}_{N_x-1,1},\hat{u}_{1,2},\dots,\hat{u}_{N_x-1,2},\hat{u}_{1,N_y-1},\dots,\hat{u}_{N_x-1,N_y-1}]^T
\tag{40}
$$

and

$$
\hat{\mathbf{f}}:=[\hat{f}_{1,1},\dots,\hat{f}_{N_x-1,1},\hat{f}_{1,2},\dots,\hat{f}_{N_x-1,2},\hat{f}_{1,N_y-1},\dots,\hat{f}_{N_x-1,N_y-1}]^T.
\tag{41}
$$

The elements in $\hat{\mathbf{f}}$ are given by

$$
\begin{aligned}
\hat{f}_{i,j}={}&f_{i,j}+\sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\alpha(x_i,y_j)\int_{B_\delta(x_i,y_j)}\\
&\times\sigma\left(x_i-x',y_j-y'\right)g(x_{i''},y_{j''})\phi_{i'',j''}\left(x',y'\right)dx'\,dy'\\
&+\frac{\partial\alpha(x_i,y_j)}{\partial x}\sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\int_{B_\delta(x_i,y_j)}\\
&\times\sigma\left(x_i-x',y_j-y'\right)g(x_{i''},y_{j''})\phi_{i'',j''}\left(x',y'\right)dx'\,dy'\\
&+\frac{\partial\alpha(x_i,y_j)}{\partial y}\sum_{\substack{-K+1\le i''\le 0\\ N_x\le i''\le N_x+K-1}}\sum_{\substack{-L+1\le j''\le 0\\ N_y\le j''\le N_y+L-1}}\int_{B_\delta(x_i,y_j)}\\
&\times\sigma\left(x_i-x',y_j-y'\right)g(x_{i''},y_{j''})\phi_{i'',j''}\left(x',y'\right)dx'\,dy',\\
&1\le i\le N_X-1,1\le j\le N_y-1.
\end{aligned}
\tag{42}
$$

Corollary 1 *The coefficient matrices $\hat{\mathbf{A}}$ can be stored in $O(N)$ memories.*

Proof From (39), to store the matrices $\hat{\mathbf{A}}$, we need to store $\mathbf{D}_1$, $\mathbf{D}_2$, $\mathbf{D}_3$, $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$. Because $\mathbf{D}_1$, $\mathbf{D}_2$, and $\mathbf{D}_3$ are all diagonal matrices, the storage of these three matrices requires $O(N)$ memories.

According to the fact that matrices $\mathbf{B}_1$, $\mathbf{B}_2$, and $\mathbf{B}_3$ have the same structure, we only need to analyse the storage requirement of $\mathbf{B}_1$. Because of the BTTB structure of $\mathbf{B}_1$, to store

$\mathbf{B}_1$, we need to store only $(2K+1)$-by-$(2L+1)$ entries which can be arranged as follows:

$$\mathbf{G} = \begin{pmatrix} b_1^{-K,-L} & \cdots & b_1^{-K,0} & \cdots & b_1^{-K,L} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_1^{0,-L} & \cdots & b_1^{0,0} & \cdots & b_1^{0,L} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ b_1^{K,-L} & \cdots & b_1^{K,0} & \cdots & b_1^{K,j} \end{pmatrix}, \tag{43}$$

Hence, the memory requirement of $\mathbf{B}_1$ is $O(N)$. □

Corollary 2 *The matrix-vector multiplication* $\hat{\mathbf{A}}\mathbf{u}$ *can be calculated in* $O(N\log N)$ *operations for any vector* $\mathbf{u} \in \mathbb{R}^N$.

Proof From (39), we have

$$\hat{\mathbf{A}}\mathbf{u} = 2(\mathbf{D}_1\mathbf{B}_1\mathbf{u}) + \mathbf{D}_2\mathbf{B}_2\mathbf{u} + \mathbf{D}_3\mathbf{B}_3\mathbf{u}. \tag{44}$$

We only consider the first term on the right-hand side in (44). The other two terms are calculated similarly. Because the matrix $\mathbf{B}_1$ is a BTTB matrix, the matrix-vector multiplication $\mathbf{G}_1 = \mathbf{B}_1\mathbf{u}$ can be finished in $O(N\log N)$ operations[25]. Subsequently, $\mathbf{D}_1\mathbf{G}_1$ can be calculated in $O(N)$ operations due to the fact that $\mathbf{D}_1$ is a diagonal matrix. In summary, $\mathbf{D}_1\mathbf{B}_1\mathbf{u}$ can be calculated in $O(N\log N)$ operations. □

In the well-known Krylov subspace iteration method, such as conjugate gradient method (CG) and conjugate gradient squared method (CGS), each iteration contains several matrix-vector multiplications and vector operations. Because of the nonlocal property of the nonlocal diffusion model, the coefficient matrices $\mathbf{A}_1$ and $\mathbf{A}_2$ are usually dense. Hence, the matrix-vector multiplications in each iteration need $O(N^2)$ computational operations. All other computations in each iteration still need $O(N)$ computational operations. Thus, a fast Krylov subspace iteration method can be developed by calculating the matrix-vector multiplications in each iteration using Corollary 2. Moreover, the computer memory storage can be reduced for the coefficient matrix $\hat{\mathbf{A}}$ using Corollary 1.

4 Numerical experiments

In this section, several numerical experiments are done to investigate the performance of the fast collocation method for the two-dimensional variable-coefficient nonlocal diffusion model. In the numerical runs as follows:, the variable-coefficient nonlocal diffusion model (1) with kernel function [26]

$$\sigma(x,y) = \frac{1}{(x^2+y^2)^{1+s}}, \quad s < \frac{1}{2} \tag{45}$$

is considered.

In the following numerical experiments, the domain Ω in (1) is chosen to be $\Omega = (0,1) \times (0,1)$. We divide Ω by uniform square meshes, i.e., $N_x = N_y$. The horizon parameter is fixed to $\delta = 1/8$. The true solution is chosen to be $x(1-x)y(1-y)$. $g(x,y) = u(x,y)$ on the boundary zone Ω_c. The variable coefficient $\alpha(x,y) = 1 + x^2 + y^2$. We use MATLAB to run the following numerical examples on a 8G-memory laptop.

Table 1 The L^2 errors of the collocation schemes (6) and (38)

$N_x = N_y$	$\Vert u_h - u\Vert_{L^2}$	$\Vert \hat{u}_h - u\Vert_{L^2}$
2^4	1.1033×10^{-2}	1.0974×10^{-2}
2^5	4.5320×10^{-3}	4.4807×10^{-3}

Example 1 We investigate the credibility of the collocation method (38) in this numerical example. The value of s in (45) is fixed to be zero. The right-hand term can be analytically computed by polar coordinate transformation and Maple

$$\begin{aligned} f(x,y) = &-\frac{\pi}{64}x^4 - \frac{\pi}{64}y^4 + \frac{\pi}{64}x^3 + \frac{\pi}{64}y^3 - \frac{517\pi}{32{,}768}x^2 - \frac{517\pi}{32{,}768}y^2 + \frac{515\pi}{32{,}768}x \\ &+ \frac{515\pi}{32{,}768}y - \frac{3\pi}{32}x^2y^2 + \frac{\pi}{16}x^2y + \frac{\pi}{16}xy^2 - \frac{\pi}{32}xy - \frac{193\pi}{6{,}291{,}456}. \end{aligned} \tag{46}$$

We run this numerical experiment with the original collocation method (6) and the approximate collocation method (38), respectively. In (6), there are not any approximations to the variable coefficient. However, in (38), a Taylor expansion is used to approximate the variable coefficient. We solve (6) and (38) by CGS and present the L^2 errors and the number of iterations of these two collocation methods in Table 1.

From Table 2, we can observe that the L^2 errors of the approximate collocation method are comparable to that of the original collocation method. Furthermore, the L^2 errors of the approximate collocation method are slightly better than that of the original collocation method. This is because the formation of matrix $\mathbf{A_2}$ requires lots of numerical integrations (precisely speaking, we need $4(2K + 1)^2$ numerical integrations to calculate each non-zero entry), but the formation of matrix $\hat{\mathbf{A}}$ needs only a total of $3(2K + 1)^2$ numerical integrations. The accumulation of truncation errors of the latter is better than that of the former. When $N_x = N_y = 2^6$, because of the massive numerical integrations, the formation of matrix $\mathbf{A_2}$ is very time-consuming(precisely speaking, it took us more than 2 days). Hence, when $N_x = N_y$ is larger than 2^6, we terminated the numerical experiment.

Example 2 In this numerical example, we investigate the convergence behavior of the fast approximate collocation method by choosing different kernel functions. The value of s in (45) is chosen to be 3/8, 0 and −1/2, respectively. When s is equal to 3/8, −1/2, the corresponding right-hand terms are given by

$$\begin{aligned} f(x,y) = &-\frac{2^{1/4}\pi}{10}x^4 - \frac{2^{1/4}\pi}{10}y^4 + \frac{2^{1/4}\pi}{10}x^3 + \frac{2^{1/4}\pi}{10}y^3 - \frac{2^{1/4}3353\pi}{33{,}280}x^2 \\ &- \frac{2^{1/4}3353\pi}{33{,}280}y^2 + \frac{2^{1/4}3343\pi}{33{,}280}x + \frac{2^{1/4}3343\pi}{33{,}280}y - \frac{2^{1/4}3\pi}{5}x^2y^2 \\ &+ \frac{2^{1/4}2\pi}{5}x^2y + \frac{2^{1/4}2\pi}{5}xy^2 - \frac{2^{1/4}\pi}{5}xy - \frac{2^{1/4}2701\pi}{17{,}891{,}328} \end{aligned} \tag{47}$$

and

$$\begin{aligned} f(x,y) = &-\frac{\pi}{768}x^4 - \frac{\pi}{768}y^4 + \frac{\pi}{768}x^3 + \frac{\pi}{768}y^3 - \frac{259\pi}{196{,}608}x^2 \\ &- \frac{259\pi}{196{,}608}y^2 + \frac{1289\pi}{983{,}040}x + \frac{1289\pi}{983{,}040}y - \frac{\pi}{128}x^2y^2 \end{aligned}$$

Table 2 Performance of FCG, CG, and Gauss while $s = 3/8$

	$N_x = N_y$	$\|\mathbf{u}_h - \mathbf{u}\|_{L^2}$	# of iter.	CPUs
Gauss	2^4	2.2848×10^{-2}		0.73 s
	2^5	1.6317×10^{-2}		20 s
	2^6	9.8298×10^{-3}		19 m 2 s
	2^7			> 10 h
	2^8		out of memory	
		$C_\alpha = 0.13, \alpha = 0.61$		
CGS	2^4	2.2848×10^{-2}	33	0.53 s
	2^5	1.6317×10^{-2}	104	8.1 s
	2^6	9.8298×10^{-3}	165	3 m 49 s
	2^7			> 10 h
	2^8		out of memory	
		$C_\alpha = 0.13, \alpha = 0.61$		
FCGS	2^4	2.2848×10^{-2}	32	0.1 s
	2^5	1.6317×10^{-2}	104	1 s
	2^6	9.8298×10^{-3}	165	15.82 s
	2^7	4.9156×10^{-3}	303	1 m 19 s
	2^8	2.2000×10^{-3}	625	11 m 59 s
		$C_\alpha = 0.28, \alpha = 0.85$		

Table 3 Performance of FCG, CG, and Gauss while $s = 0$

	$N_x = N_y$	$\|\mathbf{u}_h - \mathbf{u}\|_{L^2}$	# of iter.	CPUs
Gauss	2^4	1.0974×10^{-2}		0.59 s
	2^5	4.4807×10^{-3}		14.6 s
	2^6	1.4500×10^{-3}		19 m 1 s
	2^7			> 10 h
	2^8		out of memory	
		$C_\alpha = 0.65, \alpha = 1.46$		
CGS	2^4	1.0974×10^{-2}	22	0.27 s
	2^5	4.4807×10^{-3}	34	4.2 s
	2^6	1.4500×10^{-3}	44	2 m 41 s
	2^7	3.7331×10^{-4}	53	5 h 28 m 14 s
	2^8		out of memory	
		$C_\alpha = 1.12, \alpha = 1.63$		
FCGS	2^4	1.0974×10^{-2}	22	0.19 s
	2^5	4.4807×10^{-3}	34	0.44 s
	2^6	1.4500×10^{-3}	44	5.33 s
	2^7	3.7331×10^{-4}	53	18.9 s
	2^8	2.2977×10^{-5}	62	1 m 40 s
		$C_\alpha = 6.6, \alpha = 2.14$		

$$+\frac{\pi}{192}x^2y + \frac{\pi}{192}xy^2 - \frac{\pi}{384}xy - \frac{901\pi}{293{,}601{,}280}, \tag{48}$$

respectively.

In this numerical experiment, we solve the resulting approximate collocation method (38) by Gauss elimination method (Gauss), CGS, and the fast CGS (FCGS), respectively. We present the L^2 errors of the numerical solutions, the number of iterations in the corresponding iteration methods, and the CPU time spent in the above solvers in Tables 2–4. Furthermore, we use least squares method to estimate the constant C_α and the convergence rate α in the error estimate

$$\|u - u_h\|_{L^2} \le C_\alpha h^\alpha. \tag{49}$$

Table 4 Performance of FCG, CG, and Gauss while $s = -1/2$

	$N_x = N_y$	$\|\mathbf{u}_h - \mathbf{u}\|_{L^2}$	# of iter.	CPUs
Gauss	2^4	6.3004×10^{-3}		0.69 s
	2^5	1.8866×10^{-3}		13.5 s
	2^6	4.3221×10^{-4}		19 m 28 s
	2^7			> 10 h
	2^8		out of memory	
		$C_\alpha = 1.4, \alpha = 1.93$		
CGS	2^4	6.3004×10^{-3}	15	0.48 s
	2^5	1.8866×10^{-3}	20	4.75 s
	2^6	4.3221×10^{-4}	24	2 m 53 s
	2^7	2.8420×10^{-5}		7 h 18 m 40 s
	2^8		out of memory	
		$C_\alpha = 10.3, \alpha = 2.55$		
FCGS	2^4	6.3004×10^{-3}	15	0.09 s
	2^5	1.8866×10^{-3}	20	0.25 s
	2^6	4.3221×10^{-4}	24	2.33 s
	2^7	2.8420×10^{-5}	27	10.6 s
	2^8	8.2241×10^{-6}	35	54.5 s
		$C_\alpha = 9.3, \alpha = 2.52$		

From Tables 2–4, The following results can be observed: (i) The L^2 errors of the numerical solutions of (1) become small while s decreases. (ii) The number of iterations in CGS and FCGS falls while s decreases. (iii) FCGS requires less memory than the traditional solvers like Gauss and CGS. Specifically, we can observe from Tables 2–4 that when N is equal to 2^8, the memory required in Gauss and CGS is not enough. However, FCGS does not have this problem because of less memory requirement. (iv) From Tables 2–4, we can observe that FCGS needs less CPU time than Gauss and CGS especially when N is chosen to be bigger. For example, when $s = -1/2$ and $N = 2^7$, the CPU time required by CGS exceeds 5 hours, but FCGS needs only about 19 seconds.

5 Conclusions

In this paper, a fast and faithful collocation method is developed for a two-dimensional variable-coefficient linear nonlocal diffusion model. Using Taylor expansion to the variable coefficient $\alpha(x,y)$, we see that the coefficient matrix resulting from the bilinear collocation method can be approximated by a sum of the products of diagonal matrix and BTTB matrix, which result in an approximate collocation method to the original physical problem. This fast method can reduce the computational operations in each Krylov subspace iteration from $O(N^2)$ to $O(N \log N)$ and the memory requirement of the coefficient matrix from $O(N^2)$ to $O(N)$. The numerical experiments show the credibility and utility of this fast collocation method.

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 partially supported by the National Natural Science Foundation of China under Grants 91630207.

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 (Shandong Academy of Sciences), Jinan, 250100, China. [2]School of Mathematics, Shandong University, Jinan, 250100, China.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 30 June 2020 Accepted: 26 August 2020 Published online: 09 September 2020

References

1. Stavroulakis, I.P., Tersian, S.A.: Partial Differential Equations, 2th edn. World Scientific, Singapore (2004)
2. 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)
3. 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)
4. Viglialoro, G., Woolley, T.E.: Boundedness in a parabolic-elliptic chemotaxis system with nonlinear diffusion and sensitivity and logistic source. Math. Methods Appl. Sci. **41**, 1809–1824 (2018)
5. Benson, D., Schumer, R., Meerschaert, M.M., Wheatcraft, S.W.: Fractional dispersion, Levy motions, and the made tracer tests. Transp. Porous Media **42**, 211–240 (2001)
6. Benson, D., Wheatcraft, S.W., Meerschaert, M.M.: The fractional-order governing equation of Levy motion. Water Resour. Res. **36**, 1413–1423 (2000)
7. Hilfer, R.: Applications of Fractional Calculus in Physics. World Scientific, Singapore (2000)
8. Silling, S.A.: Reformulation of elasticity theory for discontinuities and long-range forces. J. Mech. Phys. Solids **48**, 175–209 (2000)
9. Askari, E., Bobaru, F., Lehoucq, R.B., Parks, M.L., Silling, S.A., Weckner, O.: Peridynamics for multiscale materials modeling. J. Phys. Conf. Ser. **125**, 012078 (2008)
10. Du, Q., Lehoucq, R.B., Tartakovsky, A.M.: Integral approximations to classical diffusion and smoothed particle hydrodynamics. Comput. Methods Appl. Mech. Eng. **286**, 216–229 (2015)
11. Kao, C.Y., Lou, Y., Shen, W.: Random dispersal vs. non-local dispersal. Discrete Contin. Dyn. Syst. **26**, 551–596 (2010)
12. Li, T., Viglialoro, G.: Boundedness for a nonlocal reaction chemotaxis model even in the attraction-dominated regime (2020) https://arxiv.org/abs/2004.10991
13. Liu, Z., Li, X.: Two fast and efficient semi-implicit approaches with unconditional energy stability for nonlocal phase field crystal equation. Appl. Numer. Math. **150**, 491–506 (2020)
14. Madenci, E., Oterkus, E.: Peridynamic Theory and Its Applications. Springer, Berlin (2014)
15. Wang, K., Wang, H.: A fast characteristic finite difference method for fractional advection-diffusion equations. Adv. Water Resour. **34**, 810–816 (2011)
16. Wang, H., Du, N.: A fast finite difference method for three-dimensional time-dependent space-fractional diffusion equations and its efficient implementation. J. Comput. Phys. **253**, 50–63 (2013)
17. Wang, H., Du, N.: Fast alternating-direction finite difference methods for three-dimensional space-fractional diffusion equations. J. Comput. Phys. **258**, 305–318 (2013)
18. 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)
19. 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)
20. Zhang, X., Li, X., Cheng, A., Wang, H.: A preconditioned fast collocation method for a linear bond-based peridynamic model. Adv. Differ. Equ. **2020**, 244 (2020)
21. Liu, Z., Li, X., Zhang, X.: A fast high-order compact difference method for the fractal mobile/immobile transport equation. Int. J. Comput. Math. **97**, 1860–1883 (2020)
22. Mengesha, T., Du, Q.: Nonlocal constrained value problems for a linear peridynamic Navier equation. J. Elast. **116**, 27–51 (2014)
23. Wang, C., Wang, H.: A fast collocation method for a variable-coefficient nonlocal diffusion model. J. Comput. Phys. **330**, 114–126 (2017)
24. 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)
25. Jin, X.Q.: Preconditioning Techniques for Toeplitz Systems, 1th edn. Higher Education Press, China (2010)
26. Zhou, K., Du, Q.: Mathematical and numerical analysis of linear peridynamic models with nonlocal boundary condition. SIAM J. Numer. Anal. **48**, 1759–1780 (2010)