East Asian Journal on Applied Mathematics
doi: 10.4208/eajam.2024-183.020325

Vol. **16**, No. 1, pp. 129-151
February 2026

# A Fast Discontinuous Galerkin Finite Element Method for a Bond-Based Linear Peridynamic Model

Jincheng Dong[1], Ning Du[1,*] and Zhiwei Yang[2]

[1]*School of Mathematics, Shandong University, Jinan 250100, China.*
[2]*School of Qilu Transportation, Shandong University, Jinan 250002, China.*

*Received 6 November 2024; Accepted (in revised version) 2 March 2025.*

**Abstract.** A fast discontinuous Galerkin finite element algorithm on a non-uniform mesh for solution of a one-dimensional bond-based linear peridynamic model with fractional noise is developed. It is based on the approximation of the stiffness matrix corresponding to the discontinuous Galerkin finite element method by its hierarchical representation. The fast algorithm reduces the storage requirement for the stiffness matrix from $\mathcal{O}(N^2)$ to $\mathcal{O}(kN)$, where $k$ is a parameter controlling the accuracy of hierarchical matrices. The computational complexities of assembling the stiffness matrix and the Krylov subspace method for solving linear systems are also reduced from $\mathcal{O}(N^2)$ to $\mathcal{O}(kN)$. Numerical results show the utility of the numerical method.

**AMS subject classifications**: 65C30, 65B99

**Key words**: Peridynamic model, discontinuous Galerkin finite element, hierarchical matrix, fast algorithm, fractional noise.

## 1. Introduction

The peridynamic (PD) theory aimed to model problems with evolving discontinuities has been successfully used in various applications, including polycrystal fracture [11], quasi-static crack propagation [14], brittle fracture [15], damage in concrete [10], geomaterial fragmentation by impulse loads [19], failure and damage in composite laminates [23], failure in a stiffened composite curved panel [23]. To solve PD models, a number of numerical methods have been proposed. In particular, Chen and Gunzburger [4] developed a Galerkin finite element method for the one-dimensional steady-state PD model

$$\begin{aligned} &\frac{1}{\delta^2}\int_{x-\delta}^{x+\delta}\frac{u(x)-u(x')}{|x-x'|}dx' = b(x), && x\in\Omega,\\ &u(x)=g(x), && x\in\Gamma. \end{aligned} \tag{1.1}$$

[*] Corresponding author. *Email address:* duning@sdu.edu.cn (N. Du)

Discontinuous Galerkin finite element method has been applied to various models and extended to various novel discretization schemes [22,26,29]. In PD models (1.1), continuous Galerkin finite element is not suitable if the solutions contains jump discontinuities. Therefore, in this paper a discontinuous Galerkin (DG) finite element method is used. However, numerical methods for peridynamic models usually yield dense stiffness matrices due to their non-local property, thus the direct solvers require $\mathscr{O}(N^2)$ memory and $\mathscr{O}(N^3)$ computational complexity, where $N$ is the number of spatial unknowns. Meanwhile Krylov subspace iterative methods require $\mathscr{O}(N^2)$ computational complexity per iteration, which can be also very expensive. Extensive efforts have been made to improve the computational efficiency [8,21,25]. Wang and Tian [25] developed a fast method for a linear steady-state bond-based peridynamic model by utilizing the positive-definite tridiagonal-plus-Toeplitz structure of the stiffness matrix. Du *et al.* [8] developed a fast collocation method for a state-based linear PD model equation in one space dimension. Both approaches reduce the computational cost from $\mathscr{O}(N^3)$ to $\mathscr{O}(N\log^2 N)$ and the memory requirements from $\mathscr{O}(N^2)$ to $\mathscr{O}(N)$. So far, main progress has been made on reducing the computational cost and memory requirement based on a uniform mesh. However, the existing fast methods are not applicable to PD models if a non-uniform mesh is used.

In this work, we present a fast discontinuous Galerkin (FDG) finite element method for a steady-state one-dimensional PD model based on a non-uniform mesh using Hierarchical matrices ($\mathscr{H}$-matrices) [2]. $\mathscr{H}$-matrices are data-sparse approximations of dense matrices. They are successfully used for solving integral equations [1,2], fractional differential equations [20,28] and elliptic partial differential equations [18]. Matrices stored in $\mathscr{H}$-matrix format provide the reduction of storage requirement from $\mathscr{O}(N^2)$ to $\mathscr{O}(kN)$ and the computational complexity from $\mathscr{O}(N^2)$ to $\mathscr{O}(kN)$ per matrix-vector multiplication, where $k$ is a parameter controlling the accuracy. The key aim of the $\mathscr{H}$-matrix method is to approximate the stiffness matrix $\mathbb{A}$ by $\tilde{\mathbb{A}}$ which can be stored in a data-sparse (not necessary sparse) format. To this end, truncated Taylor expansion is adopted to approximate the kernel of the PD model. Let $k$ be the number of terms in the truncated Taylor expansion, we will prove theoretically that the approximation error of the $\mathscr{H}$-matrices $\tilde{\mathbb{A}}$ decays as $\mathscr{O}(3^{-k})$.

Meanwhile, most of the problems are affected by ubiquitous noise perturbations. This motivates the study of mathematical models driven by stochastic noise and numerical methods for solving them [6,27]. For finding correlated random fluctuations occurred in more general systems see [5,7,12]. An approximate model for correlated noise is the fractional Brownian motion (fBm) $W_B(x)$ [3,17,24], which satisfies

$$Cov(x,y) := E\big(W_B(x)W_B(y)\big) = \frac{1}{2}\left(|x|^{2B} + |y|^{2B} - |x-y|^{2B}\right),$$

where $x,y \in [0,\infty)$ and $B \in (0,1)$ is the Hurst index. Here, fBm denotes the Brownian motion if $B = 1/2$. Otherwise the increments of fBm are dependent.

The rest of the paper is organized as follows. In Section 2, we introduce a bond-based PD model with fractional noise and its discontinuous Galerkin finite element discretization based on a non-uniform mesh. In Section 3, we present the $\mathscr{H}$-matrix representation of stiffness matrices corresponding to the discretization scheme and present the error analysis. In Section 4, numerical experiments are presented to demonstrate the accuracy and computational efficiency of the fast algorithm. Conclusions and discussion of future works are summarized in Section 5. In Appendix A, we present an explanation of the construction of the $\mathscr{H}$-matrix structure.

## 2. PD Model and Discontinuous Galerkin Finite Element Discretization

Let

$$\Omega=(\alpha,\beta),\quad \Omega'=(\alpha-\delta,\beta+\delta),\quad \Gamma=[\alpha-\delta,\alpha]\cup[\beta,\beta+\delta],$$

where we assume the horizon width $\delta$ is a fixed constant independent of the forthcoming mesh. We consider a discontinuous Galerkin finite element method for the steady-state one-dimensional PD model with fractional noise

$$\begin{aligned} &\frac{1}{\delta^2}\int_{x-\delta}^{x+\delta}\frac{u(x)-u(x')}{|x-x'|}dx' = b(x)+\dot{W}_B(x), && x\in\Omega, \\ &u(x)=g(x), && x\in\Gamma, \end{aligned} \tag{2.1}$$

where $b(x)$ is the load term, $g(x)$ the boundary condition, and $W_B(x)$ a fractional Brownian motion [9] defined on a complete probability space with the fractional noise

$$\begin{aligned} &\dot{W}_B(x)=c_B\left(\sum_{n=1}^{\infty}\frac{\cos(\alpha_n x)}{\alpha_n^B J_{1-B}(\alpha_n)}\xi_n+\sum_{n=1}^{\infty}\frac{\sin(\beta_n x)}{\beta_n^B J_{-B}(\beta_n)}\zeta_n\right), \\ &c_B=\sqrt{2/\pi}\Gamma^{1/2}(1+2B)\sin^{1/2}(\pi B). \end{aligned} \tag{2.2}$$

Note that $\alpha_n$ and $\beta_n$ are respectively the positive zeros of the Bessel function $J_{-B}$ and $J_{1-B}$, while $\xi_n$ and $\zeta_n$ are mutually independent standard Gaussian random variables, and $B$ is the Hurst index.

We approximate the fractional noise $W_B$ by employing the truncation in (2.2) and substitute $\dot{W}_B(x)$ by $\dot{W}_B^{(Q)}$ in the numerical scheme [3],

$$\dot{W}_B^{(Q)}(x)=c_B\left(\sum_{n=1}^{Q}\frac{\cos(\alpha_n x)}{\alpha_n^B J_{1-B}(\alpha_n)}\xi_n+\sum_{n=1}^{Q}\frac{\sin(\beta_n x)}{\beta_n^B J_{-B}(\beta_n)}\zeta_n\right),$$

thus obtaining the following semi-discrete scheme:

$$\frac{1}{\delta^2}\int_{x-\delta}^{x+\delta}\frac{u_Q(x)-u_Q(x')}{|x-x'|}dx' = b(x)+\dot{W}_B^{(Q)}(x).$$

After that, we consider the full-discrete scheme. A Galerkin finite element method for the peridynamics model was proposed in [4]. Define $L^2(\Omega')$ as the space of all mean square integral functions with the standard Euclidean norm $\|f\|=\sqrt{\mathbb{E}[|f|^2]}$. Let $S(\Omega')\subset L^2(\Omega')$

be a Banach space and $S^h \subset S(\Omega')$ a family of finite-dimensional subspaces parameterized by $h$. Then the finite element approximation consists in finding $u_Q^h(x) \in S_g^h$ such that

$$\begin{aligned}
&\int_\alpha^\beta v^h(x) \int_{x-\delta}^{x+\delta} \frac{u_Q^h(x) - u_Q^h(x')}{|x-x'|} dx'dx \\
&= \delta^2 \int_\alpha^\beta \left(b(x) + \dot{W}_B^{(Q)}(x)\right) v^h(x)dx, \quad \forall v^h(x) \in S_0^h(\Omega'), \\
&u_Q^h(x) = g^h(x), \quad g^h(x) \in S^h|_\Gamma,
\end{aligned}$$

where $g^h(x)$ is an approximation of $g(x)$ — e.g. one can use the $L^2$ projection of $g(x) \in L^2(\Gamma)$ defined by

$$\left(g^h - g, v_h\right) = 0 \quad \text{for all} \quad v_h \in S^h|_\Gamma.$$

We define

$$\begin{aligned}
S_g^h &:= \left\{u^h(x) \in S^h \mid u^h(x) = g^h(x) \text{ a.e. on } \Gamma\right\}, \\
S_0^h &:= \left\{v^h(x) \in S^h \mid v^h(x) = 0 \text{ a.e. on } \Gamma\right\}.
\end{aligned}$$

## 2.1. Piecewise constant finite element approximations

Here we define a mesh (generally non-uniform) on $\Omega' = [\alpha - \delta, \alpha + \delta]$

$$\begin{aligned}
\alpha - \delta = x_{-K_1} < \cdots < x_{-1} < \alpha = x_0 < x_1 < \cdots < x_N = \beta \\
< x_{N+1} < \cdots < x_{N+K_2} = \beta + \delta
\end{aligned} \tag{2.3}$$

with $h_i := x_{i+1} - x_i$, $h := \max\{h_i\}$ and the ratio $r := \max\{h_i\}/\min\{h_j\}$, where $i, j = -K_1, \ldots, N + K_2 - 1$.

Let $\{\varphi_i(x)\}_{i=-K_1+1}^{N+K_2}$ be the set of discontinuous basis functions for piecewise constant finite element space $S^h$,

$$\varphi_j(x) = \begin{cases} 1, & x \in (x_{j-1}, x_j), \\ 0, & \text{otherwise}, \end{cases} \qquad j = -K_1 + 1, \ldots, N + K_2.$$

We have

$$g^h(x) = \sum_{j=-K_1+1}^{0} g_j \varphi_j(x) + \sum_{j=N+1}^{N+K_2} g_j \varphi_j(x).$$

Let

$$u_Q^h(x) = \sum_{j=1}^{N} u_j \varphi_j(x) + \sum_{j=-K_1+1}^{0} g_j \varphi_j(x) + \sum_{j=N+1}^{N+K_2} g_j \varphi_j(x), \quad v^h(x) = \varphi_i(x).$$

Then we obtain

$$
\begin{aligned}
&\sum_{j=1}^{N} u_j \int_{\alpha}^{\beta} \varphi_i(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx \\
=&\ \delta^2 \int_{\alpha}^{\beta} \left( b(x) + \dot{W}_B^{(Q)}(x) \right) \varphi_i(x) dx \\
&- \left( \sum_{j=-K_1+1}^{0} + \sum_{j=N+1}^{N+K_2} \right) g_j \int_{\alpha}^{\beta} \varphi_i(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx.
\end{aligned} \tag{2.4}
$$

Setting $\mathbf{U} = (u_1, u_2, \cdots, u_N)^T$, we write the linear system

$$\mathbb{A}\mathbf{U} = \mathbf{b}, \tag{2.5}$$

where the entries of the matrix $\mathbb{A} = [\mathbb{A}_{ij}]_{i,j=1}^{N}$ and vector $\mathbf{b} = [b_i]_{i=1}^{N}$ have the form

$$
\begin{aligned}
\mathbb{A}_{ij} =& \int_{\alpha}^{\beta} \varphi_i(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx, \qquad (2.6)\\
b_i =&\ \delta^2 \int_{\alpha}^{\beta} \left( b(x) + \dot{W}_B^{(Q)}(x) \right) \varphi_i(x) dx \\
&- \left( \sum_{j=-K_1+1}^{0} + \sum_{j=N+1}^{N+K_2} \right) g_j \int_{\alpha}^{\beta} \varphi_i(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx.
\end{aligned}
$$

Note that the matrix $\mathbb{A}$ is generally not sparse since $\delta$ is fixed and independent of the mesh.

## 2.2. Piecewise linear finite element approximations

Let $S^h$ be the discontinuous piecewise linear finite element space [4]. Now we will use the basis functions

$$
\varphi_{2j}(x) = \begin{cases} \dfrac{x - x_{j-1}}{h_j}, & x \in (x_{j-1}, x_j), \quad j = -K_1+1, \ldots, N+K_2, \\ 0, & \text{otherwise}, \end{cases}
$$

$$
\varphi_{2j-1}(x) = \begin{cases} \dfrac{x_j - x}{h_j}, & x \in (x_{j-1}, x_j), \quad j = -K_1+1, \ldots, N+K_2, \\ 0, & \text{otherwise}, \end{cases}
$$

defined on the mesh (2.3).

Let

$$
\begin{aligned}
g^h(x) &= \sum_{j=-2K_1+1}^{0} g_j \varphi_j(x) + \sum_{j=2N+1}^{2N+2K_2} g_j \varphi_j(x), \\
u_Q^h &= \sum_{j=1}^{2N} u_j \varphi_j(x) + \sum_{j=-2K_1+1}^{0} g_j \varphi_j(x) + \sum_{j=2N+1}^{2N+2K_2} g_j \varphi_j(x).
\end{aligned}
$$

Similar to (2.4), we have

$$\begin{aligned}
&\sum_{j=1}^{2N} u_j \int_\alpha^\beta \varphi_i(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx \\
&= \delta^2 \int_\alpha^\beta \left(b(x)+\dot{W}_B^{(Q)}\right)\varphi_i(x)dx \\
&\quad -\left(\sum_{j=-2K_1+1}^{0} + \sum_{j=2N+1}^{2N+2K_2}\right) g_j \int_\alpha^\beta \varphi_i(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx.
\end{aligned}$$

The linear system can be written in matrix form

$$\mathbb{C}\mathbf{U} = \mathbf{b}, \tag{2.7}$$

where $\mathbb{C}$ is the $(2N)\times(2N)$ matrix

$$\mathbb{C} := \begin{pmatrix} \mathbb{A}^1 & \mathbb{A}^2 \\ \mathbb{A}^3 & \mathbb{A}^4 \end{pmatrix}$$

with entries

$$\begin{aligned}
\mathbb{A}^1_{ij} &= \int_{x_{i-1}}^{x_i} \varphi_{2i}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_{2j}(x)-\varphi_{2j}(x')}{|x-x'|} dx' dx, \\
\mathbb{A}^2_{ij} &= \int_{x_{i-1}}^{x_i} \varphi_{2i}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_{2j+1}(x)-\varphi_{2j+1}(x')}{|x-x'|} dx' dx, \\
\mathbb{A}^3_{ij} &= \int_{x_{i-1}}^{x_i} \varphi_{2i-1}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_{2j}(x)-\varphi_{2j}(x')}{|x-x'|} dx' dx, \\
\mathbb{A}^4_{ij} &= \int_{x_{i-1}}^{x_i} \varphi_{2i-1}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_{2j+1}(x)-\varphi_{2j+1}(x')}{|x-x'|} dx' dx, \quad i,j=1,2,\ldots,N.
\end{aligned}$$

Besides, the matrices $\mathbf{U}$ and $\mathbf{b}$ have the form $\mathbf{U}=[\mathbf{u_1},\ \mathbf{u_2}]^T$, $\mathbf{b}=[\mathbf{b_1},\ \mathbf{b_2}]^T$, where

$$\begin{aligned}
&\mathbf{u_1} = [u_2, u_4, \cdots, u_{2N}], \quad \mathbf{u_2} = [u_1, u_3, u_5, \cdots, u_{2N-1}], \\
&\mathbf{b_1} = [b_2, b_4, \cdots, b_{2N}], \quad \mathbf{b_2} = [b_1, b_3, \cdots, b_{2N-1}]
\end{aligned}$$

and

$$\begin{aligned}
b_{2i} = &\int_{x_{i-1}}^{x_i} \left(b(x)+\dot{W}_B^{(Q)}\right)\varphi_{2i}(x)dx \\
&- \sum_{j=-2K_1+1}^{0} g_j \int_{x_{i-1}}^{x_i} \varphi_{2i}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx' dx
\end{aligned}$$

$$
\begin{aligned}
&-\sum_{j=2N+1}^{2N+2K_2} g_j \int_{x_{i-1}}^{x_i} \varphi_{2i}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx'dx, \\
b_{2i+1} = &\int_{x_{i-1}}^{x_i} \left(b(x)+\dot{W}_B^{(Q)}\right) \varphi_{2i+1}(x)dx \\
&-\sum_{j=-2K_1+1}^{0} g_j \int_{x_{i-1}}^{x_i} \varphi_{2i+1}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx'dx \\
&-\sum_{j=2N+1}^{2N+2K_2} g_j \int_{x_{i-1}}^{x_i} \varphi_{2i+1}(x) \int_{x-\delta}^{x+\delta} \frac{\varphi_j(x)-\varphi_j(x')}{|x-x'|} dx'dx, \quad i=1,\dots,N.
\end{aligned}
$$

## 3. $\mathscr{H}$-Matrix Representation

In this section, we approximate the matrix $\mathbb{A}$ by an $\mathscr{H}$-matrix $\tilde{\mathbb{A}}$. The latter one can be stored in a data-sparse format.

### 3.1. Taylor expansion of the kernel

Let the index set $\mathscr{I} := \{1,2,\dots,N\}$ contains the indices of basis functions $\varphi_i$ in the finite element discretisation. Let $t$ and $s$ be two subsets of the set $\mathscr{I}$ such that

$$
\tau := \bigcup_{i\in t} (\text{supp } \varphi_i), \quad \sigma := \bigcup_{i\in s} (\text{supp } \varphi_i)
$$

are both connected intervals. Let $x \in \tau := [a,b]$ and $x' \in \sigma := [c,d]$. We say that $\tau \times \sigma$ is admissible if $\tau \cap \sigma = \emptyset$ and the following condition is satisfied:

$$
\text{diam}(\tau) \le \text{dist}(\tau,\sigma), \tag{3.1}
$$

where $\text{diam}(\tau) = b-a$ and

$$
\text{dist}(\tau,\sigma) = \begin{cases} c-b, & \text{if} \quad c \ge b, \\ a-d, & \text{if} \quad a > d. \end{cases}
$$

The blocks $t\times s \subset \mathscr{I}\times\mathscr{I}$ of indices are called admissible if corresponding $\tau\times\sigma$ is admissible, conversely, the blocks are inadmissible.

**Remark 3.1.** In admissible condition (3.1), we define the ratio

$$
\lambda := \frac{\text{dist}(\tau,\sigma)}{\text{diam}(\tau)} \ge 1.
$$

In fact, $\lambda$ can be more flexible. As was shown in [20,28], the bigger the ratio, the smaller the error lead by the $\mathscr{H}$-matrix format.

Let $G(x,x') := 1/|x-x'|$. In this case, the singularity occurs only if $x = x'$. Using the truncated Taylor expansion with $x_0 = (a+b)/2$, for any $k \in \mathbb{N}$, we approximate the function $G(x)$ as

$$\tilde{G}(x,x') = \sum_{\nu=0}^{k-1} \frac{1}{\nu!} \partial_x^{\nu} G(x_0,x')(x-x_0)^{\nu} =: \sum_{\nu=0}^{k-1} p_\nu(x) h_\nu(x'), \tag{3.2}$$

where $p_\nu(x) := (x-x_0)^\nu$ and $h_\nu(x') := (1/\nu!)\partial_x^{\nu} G(x_0,x')$. Note that

$$h_\nu(x') = \begin{cases} (-1)^{\nu-1}(x_0-x')^{-(\nu+1)}, & x' > x_0, \\ (-1)^{\nu}(x_0-x')^{-(\nu+1)}, & x' < x_0. \end{cases}$$

**Lemma 3.1.** *If $x \in \tau = [a,b], x' \in \sigma = [c,d]$ $(b < c)$, and $x_0 = (a+b)/2$, then for any $k \in \mathbb{N}^+$, the truncated Taylor series*

$$\tilde{G}(x,x') = \sum_{\nu=0}^{k-1} \frac{1}{\nu!} \partial_x^{\nu} G(x_0,x')(x-x_0)^{\nu}$$

*approximates the kernel $G(x,x')$ with the error*

$$\left|G(x,x') - \tilde{G}(x,x')\right| \le \frac{3^{-k+1}}{2|c-b|}.$$

*Proof.*

$$\begin{aligned} \left|G(x,x') - \tilde{G}(x,x')\right| &= \sum_{\nu=k}^{\infty} \frac{1}{\nu!} \partial_x^{\nu} G(x_0,x')(x-x_0)^{\nu} \\ \le \left| \sum_{\nu=k}^{\infty} (x'-x_0)^{-(\nu+1)}(x-x_0)^{\nu} \right| &\le \left| \sum_{\nu=k}^{\infty} (x'-x_0)^{-1} \cdot \left(\frac{x-x_0}{x'-x_0}\right)^{\nu} \right| \\ \le (x'-x_0)^{-1} \cdot \sum_{\nu=k}^{\infty} \left|\frac{x-x_0}{x_0-x'}\right|^{\nu} &\le \frac{1}{|c-b|} \cdot \sum_{\nu=k}^{\infty} \left(\frac{|x_0-a|}{|x_0-a|+|c-b|}\right) \\ = \left(\frac{1}{|c-b|} + \frac{|x_0-a|}{|c-b|^2}\right) \cdot \left(1 + \frac{|c-b|}{|x_0-a|}\right)^{-k}. \end{aligned}$$

Since the admissible condition (3.1) is satisfied — i.e. $|a-b| \le |b-c|$, we obtain the estimate

$$\left|G(x,x') - \tilde{G}(x,x')\right| \le \frac{3^{-k+1}}{2|c-b|},$$

which finishes the proof. □

### 3.2. Low rank approximation of matrix blocks

In view of (2.6), the entries in the stiffness matrix can be computed with different concrete integral forms. The case which accounts for a large proportion is that the domain of integration is a rectangle, i.e. the entries in $\mathbb{A}$ can be written by

$$\mathbb{A}_{ij} = -\int_{x_{i-1}}^{x_i} \varphi_i(x) \int_{x_{j-1}}^{x_j} \frac{\varphi_j(x')}{|x-x'|} dx'dx. \tag{3.3}$$

If $\mathbb{A}_{ij}$ has the form (3.3), then we approximate the kernel function by (3.2) and

$$\begin{aligned}\mathbb{A}_{ij} &\approx -\int_{x_{i-1}}^{x_i} \varphi_i(x) \int_{x_{j-1}}^{x_j} \varphi_j(x') \sum_{\nu=0}^{k-1} p_\nu(x) h_\nu(x') dx'dx \\ &= -\sum_{\nu=0}^{k-1} \int_{x_{i-1}}^{x_i} \varphi_i(x) p_\nu(x) dx \int_{x_{j-1}}^{x_j} \varphi_j(x') h_\nu(x') dx' =: \tilde{\mathbb{A}}_{ij}. \end{aligned} \tag{3.4}$$

If all entries of the submatrix $\mathbb{A}_{t\times s}$ can be written as (3.4), then the block $t \times s$ and the corresponding $\tau \times \sigma$ are called feasible.

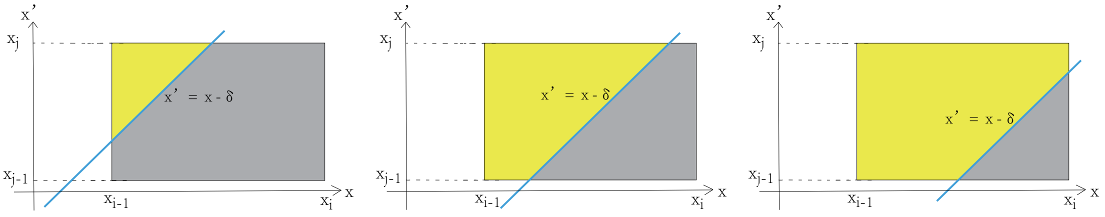
Let $D_{ij} = [x_{i-1}, x_i] \times [x_{j-1}, x_j]$, now we consider the entries $\mathbb{A}_{ij}$ whose corresponding rectangular areas $D_{ij}$ intersect the lines $x' = x + \delta$ and $x' = x - \delta$. For clarity, in Figs. 1(a)-1(c), we only present three cases when the areas intersect the line $x' = x - \delta$. The entries can be written as

$$(a)\ \ \mathbb{A}_{ij} = -\int_{x_{i-1}}^{x_j+\delta} \varphi_i(x) \int_{x-\delta}^{x_j} \frac{\varphi_j(x')}{|x-x'|} dx'dx, \tag{3.5}$$

$$(b)\ \ \mathbb{A}_{ij} = -\int_{x_{j-1}}^{x_j} \varphi_j(x') \int_{x_{i-1}}^{x'+\delta} \frac{\varphi_i(x)}{|x-x'|} dxdx', \tag{3.6}$$

$$\begin{aligned}(c)\ \ \mathbb{A}_{ij} = &-\bigg( \int_{x_{i-1}}^{x_{j-1}+\delta} \varphi_i(x) \int_{x_{j-1}}^{x_j} \frac{\varphi_j(x')}{|x-x'|} dx'dx \\ &+ \int_{x_{j-1}+\delta}^{x_i} \varphi_i(x) \int_{x-\delta}^{x_j} \frac{\varphi_j(x')}{|x-x'|} dx'dx \bigg),\end{aligned} \tag{3.7}$$

respectively, the other cases are similar due to symmetry.



(a) (b) (c)

Figure 1: Different cases of submatrix blocks. The double integrals over general regions are highlighted by yellow color.

Obviously, we cannot apply (3.4) to such cases as (3.5)-(3.7) because of dependence of the outer integral on the inner one in the double integral. Meanwhile block $t \times s$ and the corresponding $\tau \times \sigma$ are called unfeasible.

Note that the submatrix $\tilde{\mathbb{A}}_{t\times s}$ with feasible and admissible $t \times s$ can be represented in the factorised form

$$\begin{aligned} &\tilde{\mathbb{A}}_{t\times s} = -\sum_{\nu=0}^{k-1} \int_{x_{i-1}}^{x_i} \varphi_i(x) p_\nu(x) dx \int_{x_{j-1}}^{x_j} \varphi_j(x') h_\nu(x') dx' =: CR^T, \\ &C \in \mathbb{R}^{t\times\{0,\dots,k-1\}}, \quad R \in \mathbb{R}^{s\times\{0,\dots,k-1\}}, \end{aligned} \tag{3.8}$$

where

$$C_{i\nu} = -\int_{x_{i-1}}^{x_i} \varphi_i(x) p_\nu(x) dx, \quad R_{j\nu} = \int_{x_{j-1}}^{x_j} \varphi_j(x') h_\nu(x') dx'.$$

A submatrix $\tilde{\mathbb{A}}_{t\times s}$ obtained by utilizing the discretisation (3.8) is called a low rank matrix, so we only need to store $C$ and $R$ matrices in the form (3.8). Besides, a submatrix $\mathbb{A}_{t\times s}$ obtained by using the form (2.6) is called the full matrix, which does not lead to errors of the fast method. All block submatrices can be stored in low rank or full matrix format and the low rank matrices account for the most overwhelming proportions in assembling the stiffness matrix. Thus, the computational complexity with respect to the full matrices can be ignored for large enough $N$.

**Theorem 3.1.** *The global approximation error for stiffness matrix $\tilde{\mathbb{A}}$ stored in $\mathscr{H}$-matrix format is bounded by*

$$\|\mathbb{A} - \tilde{\mathbb{A}}\|_F \le \frac{3^{-k+1} r^2 (\beta - \alpha)}{2} \tag{3.9}$$

*in the Frobenius norm $\|M\|_F^2 := \sum M_{ij}^2$.*

*Proof.* Applying Lemma 3.1 gives

$$\begin{aligned} |\mathbb{A}_{ij} - \tilde{\mathbb{A}}_{ij}| &= \int_{x_{i-1}}^{x_i} \int_{x_{j-1}}^{x_j} |G(x,x') - \tilde{G}(x,x')| dx' dx \\ &\le \int_{x_{i-1}}^{x_i} \int_{x_{j-1}}^{x_j} \frac{3^{-k+1}}{2|c_{ij} - b_{ij}|} dx' dx \le \frac{3^{-k+1}}{2|c_{ij} - b_{ij}|} h^2, \end{aligned}$$

where $b_{ij}, c_{ij}$ denote $b, c$ corresponding to $\mathbb{A}_{ij}$. $b_{ij}$ and $c_{ij}$ may be different with different $\mathbb{A}_{ij}$ due to varied block submatrices.

The elements which are not in the admissible or feasible submatrices are computed without approximation and

$$Nh = N(rh_{\min}) \le r(\beta - \alpha).$$

If $b_{ij} - c_{ij} \ge h_{\min}$, we get the following approximation error $\|\mathbb{A} - \tilde{\mathbb{A}}\|_F$ in the Frobenius norm:

$$\|\mathbb{A} - \tilde{\mathbb{A}}\|_F \le \frac{3^{-k+1}}{2h_{\min}} Nh^2 \le \frac{3^{-k+1}(\beta - \alpha)}{2h_{\min}} hr \le \frac{3^{-k+1} r^2 (\beta - \alpha)}{2}.$$

The proof is complete. □

**Remark 3.2.** From Theorem 3.1, we can easily obtain that $\|\mathbb{A}-\tilde{\mathbb{A}}\|_F \leq \mathscr{O}(3^{-k})$. Then we replace $\mathbb{A}$ with $\tilde{\mathbb{A}}$ to solve the perturbed linear system

$$\tilde{\mathbb{A}}\widetilde{\mathbf{U}}=\mathbf{b}, \tag{3.10}$$

where $\widetilde{\mathbf{U}}=(\tilde{u}_1,\tilde{u}_2,\cdots,\tilde{u}_N)^T$ is the approximation of $\mathbf{U}$.

Let $\mathbb{A}_\epsilon=\tilde{\mathbb{A}}-\mathbb{A}$, $\mathbf{U}_\epsilon=\widetilde{\mathbf{U}}-\mathbf{U}$, and $\text{cond}(\mathbb{A})=\|\mathbb{A}^{-1}\|\cdot\|\mathbb{A}\|$ denote the condition number of $\mathbb{A}$. If $\|\mathbb{A}_\epsilon\|<1/\|\mathbb{A}^{-1}\|$, the standard perturbation theory [13] gives

$$\|\mathbf{U}_\epsilon\| \leq \frac{\text{cond}(\mathbb{A})}{1-\|\mathbb{A}^{-1}\|\cdot\|\mathbb{A}_\epsilon\|}\cdot\frac{\|\mathbb{A}_\epsilon\|\|\mathbf{U}\|}{\|\mathbb{A}\|},$$

so that $\|\mathbf{U}_\epsilon\| \leq \mathscr{O}(3^{-k})$. Consequently, the error caused by $\mathscr{H}$-matrix method is negligible if $k$ is large enough. Nevertheless, the computational complexity of the fast method partially depends on $k$. Therefore, in practice, $k$ cannot be arbitrary large and we usually take $k=\mathscr{O}(\log N)$.

In view of that the piecewise linear and constant discontinuous finite element method converges in $L^2$ norm with a order $\mathscr{O}(h^2)$ and $\mathscr{O}(h)$, respectively, the parameter $k$ is more large when the former method is adopted. For example, we take $k=2\log N$ and $k=\log N$, respectively.

## 3.3. Block cluster tree

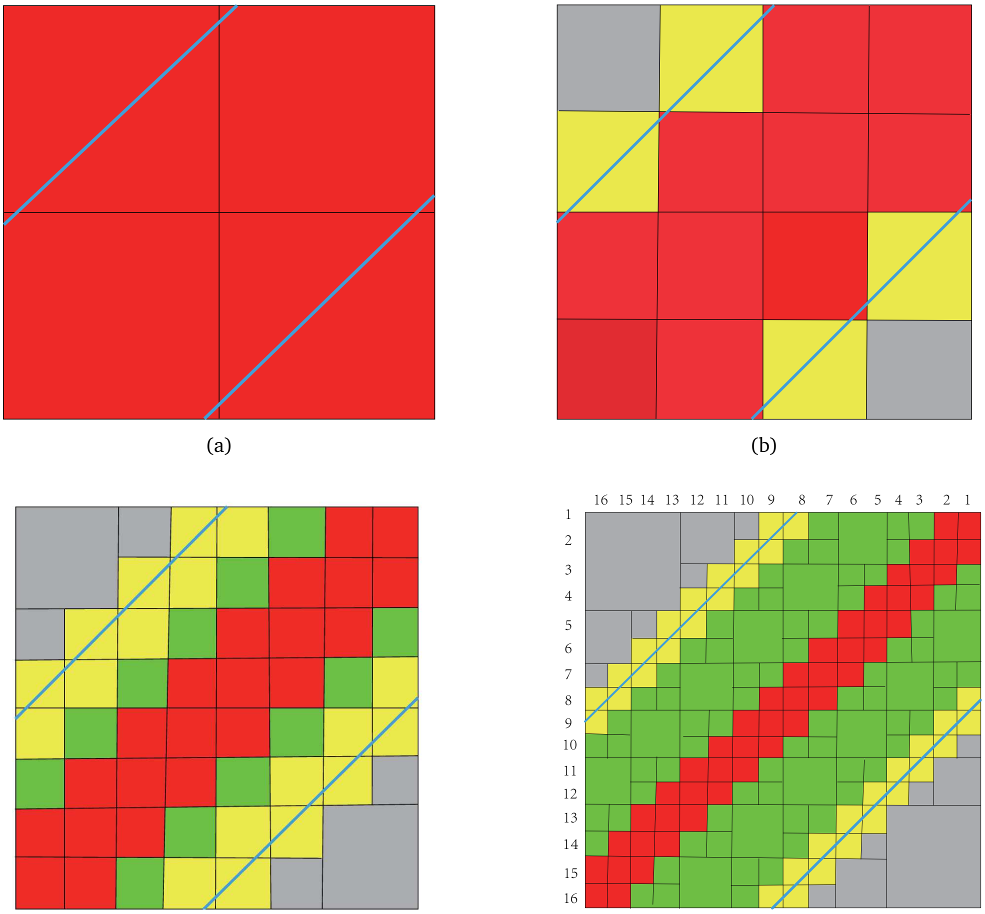
To efficiently realize the $\mathscr{H}$-matrix approximation of the stiffness matrix, we need a procedure called the block cluster tree. The method is to test blocks level by level starting with the root of $T_{\mathscr{I}\times\mathscr{I}}$ [16], which is called the quadtree. In Section A, we consider the example $N=2^4$ and assume the mesh is uniform to explain the construction of the quadtree. Here we only present the schematic diagram describing the construction of the $\mathscr{H}$-matrices and give an intuitive explanation. In Fig. 2(a), the block $\mathscr{I}\times\mathscr{I}$ is divided into four inadmissible subblocks which are colored red, then we further divide each of them into four parts as shown in Fig. 2(b). The lower right and upper left blocks represent zero matrices are colored gray. Meanwhile, there are four admissible blocks intersecting the blue lines $x'=x\pm\delta$ which are unfeasible and colored yellow. The rest blocks are inadmissible and colored red, thus we further divide the red and yellow blocks again to obtain Fig. 2(c). In this level, we first obtain some blocks which are admissible and feasible and colored green. Fig. 2(d) shows the final $\mathscr{H}$-matrix structure. The red and yellow blocks are stored in full matrices format while the green blocks are stored in low rank matrices form.

## 3.4. Matrix-vector multiplication

We consider Krylov subspace methods such as generalized minimal residual (GMRES) [13], where matrix-vector multiplication is the essential and most time consuming procedure in solving linear systems (3.9). Algorithm 3.1 presents a fast matrix-vector multiplication in $\mathscr{H}$ representation. According to Algorithm 3.1, for matrices $H$ stored in a full matrix form, the matrix-vector multiplication of $Hx$ can be determined by $\mathscr{O}(N^2)$ operations. If $H$ is stored in a low rank form, then $Hx$ can be determined by $\mathscr{O}(kN)$ operations only.

**Algorithm 3.1** Matrix-Vector Multiplication in $\mathscr{H}$-Matrix Representation

```
Procedure $y$=HMatVec($H,x$).
if $H$ is a full matrix — i.e. it is stored in a full matrix form then
    $y = Hx$,
else if $H$ is a low rank matrix — i.e. it is stored in low rank form $H = CR^T$ then
    $y = C(R^T x)$.
end if
```



(a) (b)

(c) (d)

Figure 2: The construction of $\mathscr{H}$-matrix representations. Red: Inadmissible. Green: Admissible. Gray: Zeros. Yellow: Blocks intersecting $y = x + \delta$ and $y = x - \delta$.

## 4. Numerical Experiments

In this section, we present several numerical experiments to investigate the accuracy and efficiency of the proposed methods. All the numerical experiments were implemented by using MATLAB R2020b.

In the following experiments, we use the discontinuous piecewise constant (2.5) and linear finite element scheme (2.7) on a non-uniform mesh with $r \leq 9$ and the number of sample paths $M = 200$. Let $N$ be the number of mesh elements of $\Omega$, $\omega$ belongs to a sample space, $u_h^N(\omega_j)$ and $\tilde{u}_h^N(\omega_j)$ be the $j$-th independent sample solutions of the linear systems (2.5) and (3.10), respectively. We define the error $e_N$ as

$$e_N := \left[ \frac{1}{M} \sum_{j=1}^{M} \left\| u(\omega_j) - u_h^N(\omega_j) \right\|_{L^2(\Omega)} \right]^{1/2}.$$

Similarly, we replace $u_h^N$ with $\tilde{u}_h^N$ to obtain $\tilde{e}_N$,

$$\tilde{e}_N := \left[ \frac{1}{M} \sum_{j=1}^{M} \left\| u(\omega_j) - \tilde{u}_h^N(\omega_j) \right\|_{L^2(\Omega)} \right]^{1/2}.$$

**Example 4.1** (Deterministic Continuous Solution Case)**.** We first consider model (1.1) without fractional noise. Let the load term $b(x) = 1$, the horizon width $\delta = 0.25, 0.5$, $[\alpha, \beta] = [0, 1]$. The corresponding exact solution is $u(x) = x(1 - x)$.

Table 1 shows the $L^2$ errors and convergence rates of the numerical solutions. Note that the convergence orders are around 1 which is optimal, as expected. Moreover, we observe that the $L^2$ errors of $u_h$ and $\tilde{u}_h$ are almost the same. Let CPU$\mathbb{A}$ and CPU$\tilde{\mathbb{A}}$ denote the CPU time of assembling the stiffness matrices $\mathbb{A}$ and the corresponding $\mathscr{H}$-matrices $\tilde{\mathbb{A}}$ respectively. Let CPU and CPUH be the CPU time of solving (2.5) and (3.10) by GMRES iteration respectively. In Table 2, we notice that CPU$\tilde{\mathbb{A}}$ is much less than CPU$\mathbb{A}$. Meanwhile,

Table 1: Example 4.1. $L^2$ errors and convergence rates for discontinuous piecewise constant approximation and the corresponding fast method, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^{10}$ | 1.63E-04 | - | 1.63E-04 | - |
| $2^{11}$ | 8.14E-05 | 1.00 | 8.14E-05 | 1.00 |
| $2^{12}$ | 4.07E-05 | 1.00 | 4.07E-05 | 1.00 |
| $2^{13}$ | 2.03E-05 | 1.00 | 2.03E-05 | 1.00 |
| $2^{14}$ | 1.02E-05 | 1.00 | 1.02E-05 | 1.00 |

Table 2: Example 4.1. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise constant approximations and the corresponding fast method, $\delta = 0.25$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^{10}$ | 0.30 | 0.05 | 0.21 | 0.05 |
| $2^{11}$ | 1.00 | 0.10 | 0.36 | 0.08 |
| $2^{12}$ | 3.87 | 0.70 | 0.85 | 0.23 |
| $2^{13}$ | 16.17 | 5.58 | 1.98 | 0.55 |
| $2^{14}$ | 65.78 | 28.69 | 4.78 | 1.30 |

Table 3: Example 4.1. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise constant approximations and the corresponding fast method, $\delta = 0.5$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^{10}$ | 0.28 | 0.03 | 0.15 | 0.03 |
| $2^{11}$ | 1.08 | 0.19 | 0.33 | 0.07 |
| $2^{12}$ | 4.63 | 1.71 | 0.80 | 0.20 |
| $2^{13}$ | 19.27 | 9.71 | 1.89 | 0.47 |
| $2^{14}$ | 78.88 | 47.91 | 4.47 | 1.11 |

Table 4: Example 4.1. $L^2$ errors and convergence rates for discontinuous piecewise linear approximation and the corresponding fast method, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^8$ | 2.16E-06 | - | 2.16E-06 | - |
| $2^9$ | 5.57E-07 | 1.98 | 5.57E-07 | 1.98 |
| $2^{10}$ | 1.32E-07 | 2.08 | 1.32E-07 | 2.08 |
| $2^{11}$ | 3.31E-08 | 1.99 | 3.31E-08 | 1.99 |
| $2^{12}$ | 8.75E-09 | 1.92 | 8.67E-09 | 1.93 |

Table 5: Example 4.1. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise linear approximations and the corresponding fast method, $\delta = 0.25$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^8$ | 0.22 | 0.12 | 0.24 | 0.06 |
| $2^9$ | 0.73 | 0.38 | 0.62 | 0.21 |
| $2^{10}$ | 2.56 | 1.96 | 1.50 | 0.52 |
| $2^{11}$ | 9.72 | 14.30 | 2.42 | 1.30 |
| $2^{12}$ | 35.77 | 68.97 | 5.90 | 4.12 |

the FDG method (3.10) using GMRES iteration consumes much less CPU time. For example, the time consumption for assembling the stiffness matrix decrease significantly from 67.71s to 5.47s, and that for GMRES iteration reduces drastically from 78.52s to 1.27s for a relatively finer mesh $N = 2^{14}$. Table 5 shows the CPU time for assembling the stiffness matrix and the GMRES iterations using the discontinuous piecewise linear finite element method, it can also be observed that the FDG has a reduced CPU time over the DG method while keeping the same convergence order of 2 as shown in Table 4.

Table 3 shows the numerical results for $\delta = 0.5$ compared with those for $\delta = 0.25$ in Table 2, we notice that the efficiency of the fast algorithm becomes more pronounced in assembling stiffness matrices as $\delta$ increases. For instance, when $N = 2^{14}$, the fast algorithm achieves savings of 94.33% and 97.68% for $\delta = 0.5$ in assembling the stiffness matrix and GMRES iteration, respectively, more than savings of 92.73% and 95.47% for $\delta = 0.25$. Theoretically, a larger $\delta$ results in a wider matrix bandwidth for the same $h$, leading to increased CPU time for both assembling the stiffness matrix and performing matrix-vector multiplications.

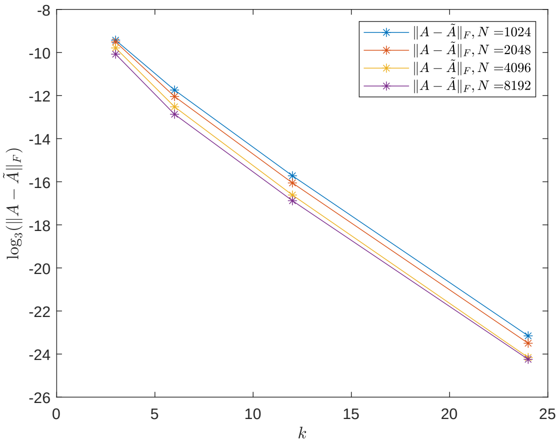


Figure 3: Example 4.1. Approximation error of the stiffness matrices $\|\mathbb{A}-\tilde{\mathbb{A}}\|_F$, $\delta=0.25$.

In this example, we also present the approximation errors of the stiffness matrices for different mesh sizes $N = 2^{10}, 2^{11}, 2^{12}, 2^{13}$ in Fig. 3. We can see that the Frobenius norm errors $\|\mathbb{A}-\tilde{\mathbb{A}}\|_F$ decrease when the number of terms in the truncated Taylor expansion $k$ grows, which is in accordance with the conclusion of Theorem 3.1. Meanwhile, the errors reduce with increasing $N$ and we can see that the error $\|A-\tilde{A}\|_F \approx 3^{-16}$ with $N = 4096$, which makes little influence on the convergence rate of the Galerkin finite element method.

**Example 4.2** (Deterministic Discontinuous Solution Case)**.** To clarify the advantage of $\mathscr{H}$-matrix method, we make up an example with a discontinuous point so that a non-uniform mesh with finer mesh on the discontinuous area is advantageous for the whole accuracy, and then the $\mathscr{H}$-matrix method can be applied for improving the computational efficiency. Let the exact solution be

$$u(x)=\begin{cases} x, & x<0.7,\\ x^2, & x\geq 0.7,\end{cases} \tag{4.1}$$

so that the load term $b(x)$ is

$$b(x)=\begin{cases} 0, & 0\leq x\leq 0.7-\delta,\\ \dfrac{1}{\delta^2}\left(\delta-\dfrac{\delta^2}{2}+\left(\dfrac{17}{10}-2\delta\right)x-\left(\log(\delta)-\dfrac{3}{2}\right)x^2\right. & \\ \left.\quad -x\left(\log\left(\dfrac{7}{10}-x\right)-\log(\delta)\right)+x^2\log\left(\dfrac{7}{10}-x\right)-\dfrac{91}{200}\right), & 0.7-\delta\leq x<0.7,\\ \dfrac{1}{\delta^2}\left(\delta-\dfrac{\delta^2}{2}-\left(\dfrac{17}{10}+2\delta\right)x+\dfrac{3}{2}x^2-x(x-1)\log\left(x-\dfrac{7}{10}\right)\right. & \\ \left.\quad +x(x-1)\log(\delta)+\dfrac{91}{200}\right), & 0.7\leq x<0.7+\delta,\\ -1, & 0.7+\delta\leq x\leq 1.\end{cases}$$

Take the horizon width $\delta=0.25$, $[\alpha,\beta]=[0,1]$.

The plot of solution (4.1) is presented in Fig. 4. We can see that the discontinuous point is $x = 0.7$. In this example, we compare the $L^2$ error on uniform and non-uniform meshes. For clarity, we define a piecewise uniform mesh

$$\begin{aligned}\alpha - \delta =: x_{-K} < \cdots < x_{-1} < \alpha =: x_0 < x_1 < \cdots < x_{l-1} < x_l < \cdots < x_r \\ < x_{r+1} < \cdots < x_{N-1} < x_N := \beta < x_{N+1} < \cdots < x_{N+K} := \beta + \delta,\end{aligned}$$

where the elements in $[x_l, x_r]$ are almost one-ninth the size of the other elements. In this example, the jump occurs at the cell boundary and inside the cell with $[x_l, x_r] = [0.6, 0.8]$ and $[x_l, x_r] = [0.63, 0.83]$, respectively.

In Tables 6 and 7, we present the numerical results for Example 4.2 with a uniform mesh and non-uniform mesh respectively. It is clear by comparing them that the non-uniform mesh yield more accurate approximations than the uniform mesh with the same number of partition element, together with the optimal convergence rate $\mathscr{O}(N^{-1})$ when the mesh point is at the discontinuous point. Meanwhile, as shown in Table 8, we verify again the fact that the $\mathscr{H}$-matrix method significantly reduce the computational cost.

In Tables 9 and 10, we present the $L^2$ error and the convergence rates when using discontinuous piecewise linear finite element method with a uniform and non-uniform meshes. Note that the $L^2$ error for non-uniform mesh is much smaller than for uniform mesh, while the $\mathscr{H}$-matrix method still has little effect on the numerical error. Furthermore, we note that the convergence order is about 0.5 for a uniform mesh and almost 2 for a non-uniform mesh. This is because that the mesh point is exactly at the location of the point where the jump discontinuity occurs. Table 11 shows the CPU time for constructing the stiffness matrix and solving the system of linear equations when using mesh where the jump occurs at the cell boundary, which once again verifies that the $\mathscr{H}$-matrix method can significantly reduce the computational cost. Tables 12 and 13 show the results for a mesh where the jump occurs inside the cell, the discontinuous piecewise linear finite element and the corresponding fast method are used to solve the Example 4.2, it can be observed that the errors

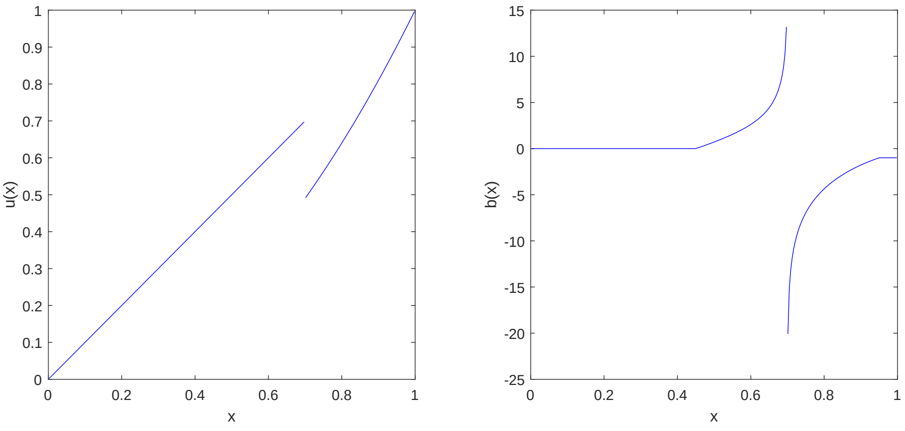


Figure 4: Example 4.2. Solution curves for exact solution $u(x)$ and load term $b(x)$.

Table 6: Example 4.2. $L^2$ errors and convergence rates for discontinuous piecewise constant approximation and the corresponding fast method, uniform mesh, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^{10}$ | 3.70E-03 | - | 3.70E-03 | - |
| $2^{11}$ | 2.58E-03 | 0.52 | 2.58E-03 | 0.52 |
| $2^{12}$ | 2.03E-03 | 0.35 | 2.03E-03 | 0.35 |
| $2^{13}$ | 1.28E-03 | 0.67 | 1.28E-03 | 0.67 |
| $2^{14}$ | 8.54E-04 | 0.58 | 8.54E-04 | 0.58 |

Table 7: Example 4.2. $L^2$ errors and convergence rates for discontinuous piecewise constant approximation and the corresponding fast method, non-uniform mesh, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^{10}$ | 7.31E-04 | - | 7.33E-04 | - |
| $2^{11}$ | 3.65E-04 | 1.00 | 3.67E-04 | 1.00 |
| $2^{12}$ | 1.83E-04 | 1.00 | 1.84E-04 | 1.00 |
| $2^{13}$ | 9.13E-05 | 1.00 | 9.22E-05 | 1.00 |
| $2^{14}$ | 4.57E-05 | 1.00 | 4.63E-05 | 0.99 |

Table 8: Example 4.2. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise constant approximation and the corresponding fast method, non-uniform mesh, $\delta = 0.25$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^{10}$ | 0.59 | 0.26 | 0.51 | 0.21 |
| $2^{11}$ | 3.02 | 1.86 | 1.26 | 0.86 |
| $2^{12}$ | 12.66 | 14.43 | 4.77 | 1.96 |
| $2^{13}$ | 46.49 | 71.49 | 9.47 | 4.18 |
| $2^{14}$ | 132.77 | 213.70 | 15.05 | 7.96 |

Table 9: Example 4.2. $L^2$ errors and convergence rates for discontinuous piecewise linear approximation and the corresponding fast method, uniform mesh, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^{8}$ | 3.59E-03 | - | 3.59E-03 | - |
| $2^{9}$ | 2.61E-03 | 0.46 | 2.61E-03 | 0.46 |
| $2^{10}$ | 1.80E-03 | 0.54 | 1.80E-03 | 0.54 |
| $2^{11}$ | 1.30E-03 | 0.46 | 1.30E-03 | 0.46 |
| $2^{12}$ | 9.01E-04 | 0.53 | 9.01E-04 | 0.53 |

Table 10: Example 4.2. $L^2$ errors and convergence rates for discontinuous piecewise linear approximation and the corresponding fast method, non-uniform mesh with jumps at cell boundaries, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^{8}$ | 1.71E-06 | - | 1.71E-06 | - |
| $2^{9}$ | 4.24E-07 | 2.00 | 4.24E-07 | 2.01 |
| $2^{10}$ | 1.06E-07 | 2.00 | 1.08E-07 | 1.98 |
| $2^{11}$ | 2.66E-08 | 2.00 | 2.74E-08 | 1.98 |
| $2^{12}$ | 7.79E-09 | 1.77 | 7.89E-09 | 1.79 |

Table 11: Example 4.2. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise linear approximation and the corresponding fast method, non-uniform mesh with jumps at cell boundaries, $\delta = 0.25$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^8$ | 0.24 | 0.05 | 0.18 | 0.06 |
| $2^9$ | 0.63 | 0.20 | 0.32 | 0.37 |
| $2^{10}$ | 2.50 | 1.67 | 0.86 | 0.99 |
| $2^{11}$ | 10.31 | 12.66 | 2.85 | 4.17 |
| $2^{12}$ | 45.39 | 71.85 | 6.75 | 11.42 |

Table 12: Example 4.2. $L^2$ errors and convergence rates for discontinuous piecewise linear approximation and the corresponding fast method, non-uniform mesh with interior jumps, $\delta = 0.25$.

| $N$ | $\|\|u-u_h\|\|_{L^2}$ | $\kappa$ | $\|\|u-\tilde{u}_h\|\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^6$ | 1.49E-03 | - | 1.49E-03 | - |
| $2^8$ | 1.16E-04 | 1.85 | 1.16E-04 | 1.85 |
| $2^{10}$ | 4.93E-05 | 0.61 | 4.93E-05 | 0.61 |
| $2^{12}$ | 2.15E-05 | 0.60 | 2.15E-05 | 0.60 |

Table 13: Example 4.2. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise linear approximation and the corresponding fast method, non-uniform mesh with interior jumps, $\delta = 0.25$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^6$ | 0.01 | 0.00 | 0.02 | 0.01 |
| $2^8$ | 0.16 | 0.04 | 0.11 | 0.07 |
| $2^{10}$ | 2.45 | 1.70 | 0.80 | 1.02 |
| $2^{12}$ | 47.20 | 80.73 | 6.53 | 11.79 |

are also very similar, and the error of the numerical method can be greatly reduced by using a refined partitioning near the discontinuity point. When $N = 2^6$, the error obtained is $1.49 \times 10^{-3}$, which is less than the error for $N = 2^8$ in Table 9. When $N = 2^{12}$, the error in Table 12 is $2.15 \times 10^{-5}$, nearly 1/40 of error $9.01 \times 10^{-4}$ in Table 9.

**Example 4.3** (Stochastic Case)**.** Let the load term $b(x) = 1$ in model (2.1), the horizon width $\delta = 0.25$, $[\alpha,\beta] = [0,1]$. Since it is hard to obtain the exact solution, we compute the reference solution by the same solver with very fine resolution, $N_{ref} = 2^{13}$. In this example, we use truncated fBm with $Q = 200$.

According to Table 14, the DG and FDG methods using $\mathscr{H}$-matrices have almost the same numerical accuracy. Table 15 demonstrates similar effect to that of Example 4.1. This verifies the utility of our proposed method for PD models with fBm noise. Note that the CPU time of iteration is much longer than in Example 4.1 because we have to simulate 200 sample paths in Example 4.3. This underscores the importance of $\mathscr{H}$-matrix method for computational efficiency in PD problems with stochastic noises.

Table 14: Example 4.3. $L^2$ errors and convergence rates for discontinuous piecewise constant approximation and the corresponding fast method, non-uniform mesh, $\delta = 0.25$.

| $N$ | $\|u-u_h\|_{L^2}$ | $\kappa$ | $\|u-\tilde{u}_h\|_{L^2}$ | $\kappa$ |
|---|---|---|---|---|
| $2^8$ | 4.47E-02 | - | 4.47E-04 | - |
| $2^9$ | 2.45E-02 | 0.87 | 2.45E-02 | 0.87 |
| $2^{10}$ | 1.27E-02 | 0.95 | 1.27E-02 | 0.95 |
| $2^{11}$ | 6.47E-03 | 0.97 | 6.47E-03 | 0.97 |

Table 15: Example 4.3. CPU time for matrix assembly and GMRES iterations for discontinuous piecewise constant approximation and the corresponding fast method, non-uniform mesh, $\delta = 0.25$.

| $N$ | CPU$\mathbb{A}$(s) | CPU(s) | CPU$\tilde{\mathbb{A}}$(s) | CPU$H$(s) |
|---|---|---|---|---|
| $2^8$ | 0.13 | 328.32 | 0.23 | 172.00 |
| $2^9$ | 0.12 | 1198.25 | 0.11 | 638.54 |
| $2^{10}$ | 0.36 | 4892.76 | 0.20 | 1844.31 |
| $2^{11}$ | 1.35 | 32924.12 | 0.44 | 4800.11 |

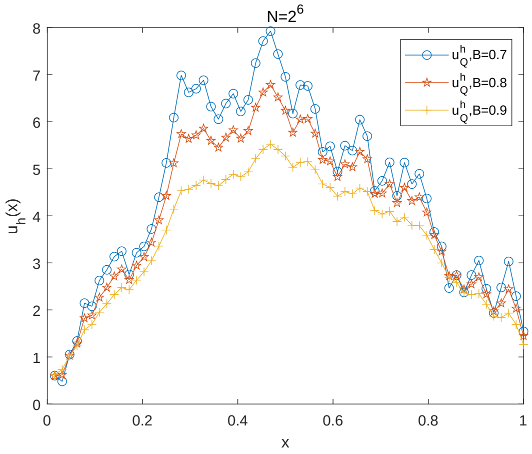


Figure 5: Example 4.3. Plots of the solution of (2.5) with $B = 0.7, 0.8, 0.9$.

Fig. 5 shows the numerical sample path of model (2.1) with same $\xi_n$ and $\zeta_n$. We note the difference between paths for different values of $B$ in model (2.1), viz. the smaller the value of $B$ is, the more pronounced is the fluctuation.

## 5. Conclusions

In this paper, we develop a fast discontinuous Galerkin finite element method for a bond-based linear peridynamic model in one spatial dimension with fractional noise. The fast algorithm is constructed on a generally non-uniform mesh by employing the $\mathscr{H}$-matrix approximation. This removes the strict limitation of some existing fast algorithms — i.e. the ones based on uniform meshes and the Toeplitz structure of the stiffness matrices. It reduces the storage requirement for assembling the stiffness matrices from $\mathcal{O}(N^2)$ to $\mathcal{O}(kN)$ and the computational complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(kN)$ at each Krylov subspace iteration. In future work, we want to use the fast algorithm to develop more efficient adaptive finite element methods for generalized PD models in multi-dimensional case.

## Appendix A

In this section, we consider the situation where $N = 2^4, \Omega = [0,1]$ and assume the mesh be uniform in order to demonstrate the construction of quadtree $T_{\mathcal{I}\times\mathcal{I}}$.

Fig. 2 shows the construction of an $\mathcal{H}$-matrix representation. The index set for the root in the quadtree $T_{\mathcal{I}\times\mathcal{I}}$ is $\{1,\ldots,16\} \times \{1,\ldots,16\}$. It is not admissible because the corresponding $\tau = [0,1], \sigma = [0,1]$ and

$$\operatorname{diam}(\tau) = 1 \nleq 0 = \operatorname{dist}(\tau,\sigma).$$

We divide $\{1,\ldots,16\} \times \{1,\ldots,16\}$ into 4 successive parts as shown in Fig. 2(a) and Fig. 6. Note that neither of the corresponding domains is admissible and divide them into 16 parts.

In Fig. 2(b), the red blocks are inadmissible and the yellow blocks are unfeasible which intersect $y = x \pm \delta$, so we need to divide these blocks again. It is reasonable to ignore the gray blocks because all the entries of them are zero (in fact, they are also admissible and feasible). Thus, we have the successors of the red or yellow blocks shown in Fig. 2(c). Now we have some admissible and feasible blocks — cf. e.g. $\{1,2\}\times\{5,6\}$. They are highlighted by green because they do not intersect $y = x \pm \delta$ and the corresponding $\tau = [0,1/8], \sigma = [1/4,3/8]$ satisfy

$$\operatorname{diam}(\tau) = \frac{1}{8} = \operatorname{dist}(\tau,\sigma).$$

Repeating these steps again, we obtain the final $\mathcal{H}$-matrix structure shown in the Fig. 2(d), where red and yellow blocks are stored in full matrix form and the green blocks in low rank matrix form. We display the quadtree in Fig. 6.

Following the above explanation, we consider successors $S(t,s)$ of $t \times s$ and present the following recursion Algorithm A.1.

**Algorithm A.1** Construct the Block Cluster Tree $T_{\mathcal{I}\times\mathcal{I}}$ Using Preorder Traversal

```
Procedure BuildBlockClusterTree($t,s$).
if $\tau \times \sigma$ is admissible then
    if $\tau \times \sigma$ is feasible then
        let $S(t,s)$ be an empty set,
    else if $\tau \times \sigma$ is unfeasible then
        let $S(t,s)$ be an empty set,
    end if
else if $\tau \times \sigma$ is inadmissible then
    let $S(t,s)$ be an empty set.
end if
```

```
Split $t \times s$ to four successive parts $S(t,s) = \{(t_1,s_1),(t_1,s_2),(t_2,s_1),(t_2,s_2)\}$.
for $(t',s') \in S(t,s)$ do
    BuildBlockClusterTree($t',s'$).
end for
```

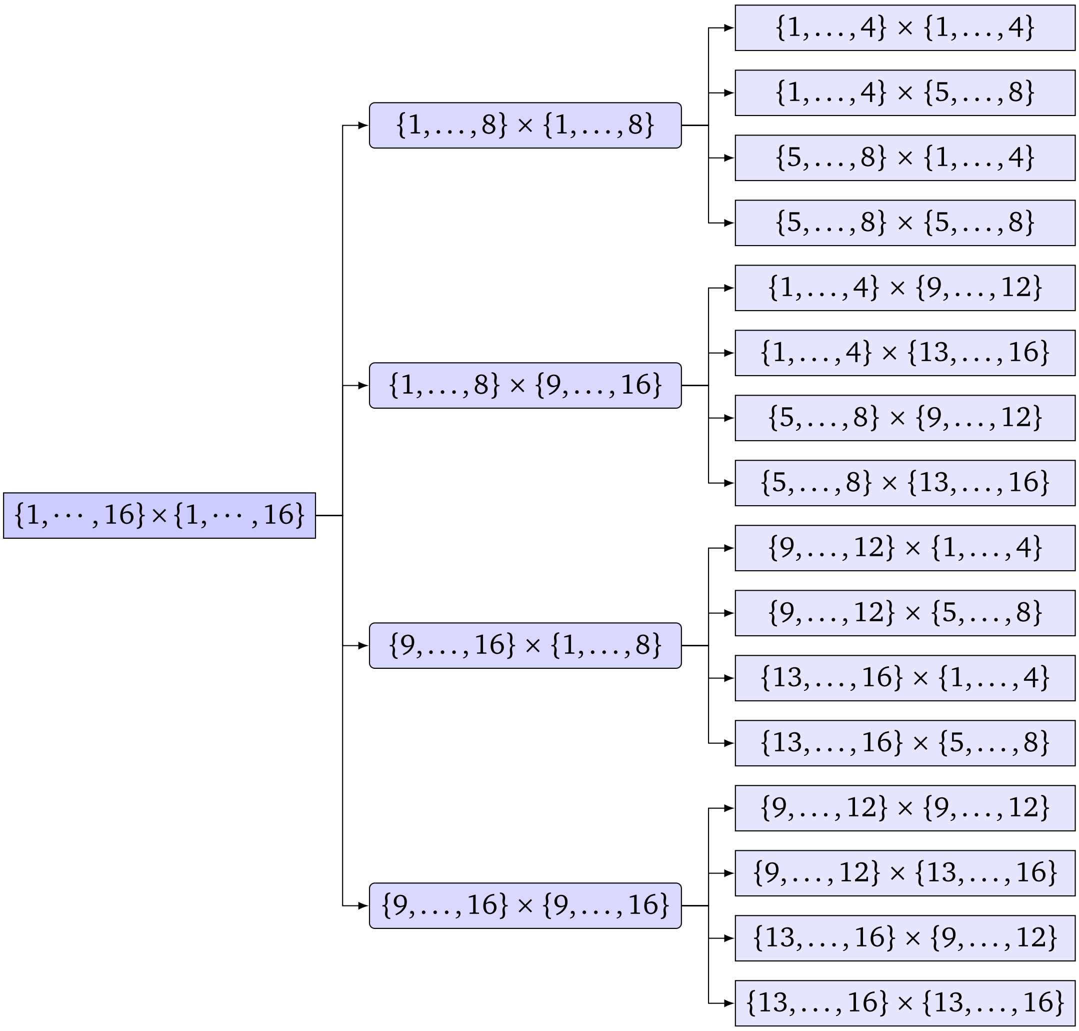


Figure 6: The diagram of a quadtree $T_{\mathcal{I}\times\mathcal{I}}$.

## Acknowledgments

This work was partially supported by the National Key R&D Program of China (Grant No. 2023YFA1009200), by the National Natural Science Foundation of China (Grant Nos. 12131014, 12401555) and by the Taishan Scholars Program of Shandong Province (Grant No. tsqn202507039).

## References

[1] S. Börm and L. Grasedyck, *Low-rank approximation of integral operators by interpolation*, Computing **72**, 325–332 (2004).

[2] S. Börm and L. Grasedyck, *Hybrid cross approximation of integral operators*, Numer. Math. **101**, 221–249 (2005).

[3] W. Cao, Z. Hao and Z. Zhang, *Optimal strong convergence of finite element methods for one-dimensional stochastic elliptic equations with fractional noise*, J. Sci. Comput. **91**, Paper No. 1 (2022).

[4] X. Chen and M. Gunzburger, *Continuous and discontinuous finite element methods for a peridynamics model of mechanics*, Comput. Methods Appl. Mech. Eng. **200**, 1237–1250 (2011).

[5] L. Chevillard, S.G. Roux, E. Lévêque, N. Mordant, J-F. Pinton and A. Arnéodo, *Intermittency of velocity time increments in turbulence*, Phys. Rev. Lett. **95**, Paper No. 064501 (2005).

[6] K. Dareiotis, C. Kumar and S. Sabanis, *On tamed Euler approximations of SDEs driven by Lévy noise with applications to delay equations*, SIAM J. Numer. Anal. **54**, 1840–1872 (2016).

[7] J. Davidson and N. Hashimzade, *Alternative frequency and time domain versions of fractional Brownian motion*, Econ. Theory **24**, 256–293 (2008).

[8] N. Du, X. Guo and H. Wang, *A fast state-based peridynamic numerical model*, Commun. Comput. Phys. **27**, 274–291 (2020).

[9] K. Dzhaparidze and J. Zanten, *A series expansion of fractional Brownian motion*, Probab. Theory Relat. Fields **130**, 39–55 (2004).

[10] W. Gerstle, N. Sau and S. Silling, *Peridynamic modeling of concrete structures*, Appl. Mech. Mater. **237**, 1250–1258 (2007).

[11] M. Ghajari, L. Iannucci and P. Curtis, *A peridynamic material model for the analysis of dynamic crack propagation in orthotropic media*, Comput. Methods Appl. Mech. Eng. **276**, 431–452 (2014).

[12] L.A. Gil-Alana, *A fractional multivariate long memory model for the US and the canadian real output*, Econ. Lett. **81**, 355–359 (2003).

[13] G.H. Golub and C.F. Van Loan, *Matrix Computations*, Johns Hopkins University Press (2013).

[14] J. Gwanghyun and Y.D. Ha, *Two-grid based sequential peridynamic analysis method for quasi-static crack propagation*, Eng. Fract. Mech. **269**, Paper No. 108549 (2022).

[15] Y.D. Ha and F. Bobaru, *Characteristics of dynamic brittle fracture captured with peridynamics*, Eng. Fract. Mech. **78**, 1156–1168 (2011).

[16] W. Hackbusch, *A sparse matrix arithmetic based on h-matrices. Part I: Introduction to h-matrices*, Computing **62**, 89–108 (1999).

[17] Z. Hao and Z. Zhang, *Numerical approximation of optimal convergence for fractional elliptic equations with additive fractional Gaussian noise*, SIAM/ASA J. Uncertain. Quantif. **9**, 1013–1033 (2021).

[18] K.L. Ho and L. Ying, *Hierarchical interpolative factorization for elliptic operators: Differential equations*, Commun. Pure Appl. Math. **69**, 1415–1451 (2016).

[19] X. Lai, B. Ren, H. Fan, S. Li, C.T. Wu, R.A. Regueiro and L. Liu, *Peridynamics simulations of geomaterial fragmentation by impulse loads: Peridynamics simulations of geomaterial fragmentation by impulse loads*, Appl. Mech. Mater. **39**, 1304–1330 (2015).

[20] X. Li, Z. Mao, N. Wang, F. Song, H. Wang and G.E. Karniadakis, *A fast solver for spectral elements applied to fractional differential equations using hierarchical matrix approximation*, Comput. Methods Appl. Mech. Eng. **366**, Paper No. 113053 (2020).

[21] H. Liu, A. Cheng and H. Wang, *A fast discontinuous Galerkin method for a bond-based linear peridynamic model discretized on a locally refined composite mesh*, J.Sci. Comput. **76**, 913–942

(2018).

[22] Y. Liu, C. Shu and M. Zhang, *Sub-optimal convergence of discontinuous Galerkin methods with central fluxes for linear hyperbolic equations with even degree polynomial approximations*, J. Comput. Math. **39**, 629–648 (2021).

[23] E. Oterkus, E. Madenci, O. Weckner, S. Silling, P. Bogert and A. Tessler, *Combined finite element and peridynamic analyses for predicting failure in a stiffened composite curved panel with a central slot*, Compos. Struct. **94**, 839–850 (2012).

[24] F. Russo and P. Vallois, *Fractional Brownian motion and related processes*, in: *Stochastic Calculus via Regularizations*, pp. 95–111, Springer (2022).

[25] H. Wang and H. Tian, *A fast Galerkin method with efficient matrix assembly and storage for a peridynamic model*, J. Comput. Phys. **231**, 7730–7738 (2012).

[26] N. Yi, Y. Huang and H. Liu, *A conservative discontinuous Galerkin method for nonlinear electromagnetic Schrödinger equations*, SIAM J. Sci. Comput. **41**, B1389–B1411 (2019).

[27] W. Zhan and Z. Li, *Sobolev-type fractional stochastic differential equations driven by fractional Brownian motion with non-Lipschitz coefficients*, J. Partial Differ. Equ. **32**, 144–155 (2019).

[28] X. Zhao, X. Hu, W. Cai and G.E. Karniadakis, *Adaptive finite element method for fractional differential equations using hierarchical matrices*, Comput. Methods Appl. Mech. Eng. **325**, 56–76 (2017).

[29] J. Zhu, C. Shu and J. Qiu, *High-order Runge-Kutta discontinuous Galerkin methods with a new type of multi-resolution WENO limiters on triangular meshes*, Appl. Numer. Math. **153**, 519–539 (2020).