MULTILEVEL AND MULTISCALE SCHEMES FOR FRACTIONAL PARTIAL DIFFERENTIAL EQUATIONS

ZHJJANG ZHANG and WEIHUA DENG

Abstract. The wavelet numerical methods for the classic PDEs have been well developed, but they are still not discussed for the fractional PDEs. This paper focuses on investigating the applications of wavelet bases to numerically solving fractional PDEs and digging out the potential benefits of wavelet methods comparing with other numerical methods, especially in the aspects of realizing preconditioning, adaptivity, and keeping the Toeplitz structure. More specifically, the contributions of this paper are as follows: 1. the techniques of efficiently generating stiffness matrix with computational cost $O(2^J)$ are provided for first, second, and any order bases; 2. theoretically and numerically discuss the effective multilevel preconditioner for time-independent equation and multiresolution multigrid method for time-dependent equation, respectively; 3. the wavelet multiscale adaptivity is experimentally discussed and numerically applied to solve the time-dependent (independent) equations. In fact, having reliable, simple, and local regularity indicators is the striking benefit of the wavelet in adaptively solving fractional PDEs.

1991 Mathematics Subject Classification. 35R11, 65T60, 65F08, 65M55.

1. Introduction

The continuous time random walk (CTRW), a fundamental model in statistic physics, is a stochastic process with arbitrary distributions of jump lengths and waiting times. When the jump length and/or waiting time distribution(s) are power law and the second order moment of jump lengths and/or the first order moment of waiting times are divergent, the CTRW describes the anomalous diffusion, i.e., the super and sub diffusive cases; and its Fokker-Planck equation has space and/or time fractional derivative(s). It can be noted that the corresponding fractional PDEs are essentially dealing with the multiscale phenomena; and generally the fractional PDEs have weaker regularity at the area close to boundary and initial time. Besides anomalous diffusion, the fractional models are also used to characterize the memory and hereditary properties inherent in various materials and processes and, recently, much more scientific applications are found in a variety of fields; see, e.g., 19, 21, 27, 39 and the references therein.

The obtained analytical solutions of fractional PDEs are usually in the form of transcendental functions or infinite series; and in much more cases, the analytical solutions are not available. Then the approximation and numerical techniques for solving the fractional PDEs become essential and have been developed very fast.
When our attention on its efficiency by proposing and testing the adaptive algorithms that concentrate the degrees of freedom. They have the closed-form expression, which is of course attractive for the fractional operators. In this regard, the wavelet numerical methods will be presented in [10]; this will enable us to focus on the wavelet numerical methods themselves.

Recently, such as, the finite difference method [5][20][30][38][41], the finite element method [9][11][32][36], and the spectral method [10][17][40]. But the computational expenses and nonuniform regularity are still the big challenges that one faces in numerically solving the fractional PDEs, owing to the nonlocality and potential multiscale characteristics of the fractional derivatives; and basing on the preconditioning, adaptivity, and fast transform techniques to develop high efficient methods seems to be a new trend. The preconditioning techniques are discussed in [23][37], where the Krylov subspace projection is their common theme. Fast transform method and multigrid method are provided in [34] and [24], respectively.

So far, there seems to be very limited works [13][28][29][33] to solve the fractional PDEs or ODEs by wavelet, although the wavelet numerical methods for classical PDEs or ODEs have been well developed [5][31]. The goal of this paper is to dig out the potential advantages of wavelets in treating the fractional operators, including preconditioning, multigrid, adaptivity, and keeping the quasi-Toeplitz structure for arbitrary order wavelet bases. More concretely, the clearly obtained benefits of wavelets for fractional operator consist of the following: 1) stiffness matrix of fractional operator is Toeplitz for scaling bases because of their shift-invariant property (it is not always true for the familiar finite element bases, such as the quadratic or cubic element) and a simple diagonal scaling usually produces a good preconditioner; 2) multiscale coefficients indicate the local regularity, and they can be used as the indicator (local posteriori error estimate seems hard to be obtained for the adaptive finite element method because of the global property of the operator) in the adaptive mesh refinement for controlling the entire computational process and increasing the efficiency, i.e., one only needs to make the local refinement on the subdomain where the wavelet coefficients are larger compared with those of other places. For avoiding all non indispensable complications, we present the main ideas and techniques in their simplest form and restrict ourselves to the following homogeneously space fractional PDE [10][30][32]:

\[ q u_t + Au = f \quad \text{on } \Omega, \tag{1.1} \]

where \( \Omega = (0, 1) \), \( q = 0 \) or 1; and \( A \) is a \((2 - \beta)\)-th \((0 \leq \beta < 1)\) order differential operator

\[ Au := -\kappa_\beta D \left( p \, D_x^{-\beta} + (1 - p) \, D_1^{-\beta} \right) Du \tag{1.2} \]

with \( \kappa_\beta > 0 \) being the generalized diffusivity; \( 0 \leq p \leq 1 \); \( D \) represents a single spatial derivative; \( D_x^{-\beta} \) and \( D_1^{-\beta} \) are the left and right fractional integral operators [22], being, respectively, defined as

\[ D_x^{-\beta} u := \frac{1}{\Gamma(\beta)} \int_0^x (x - s)^{\beta-1} u(s) \, ds, \tag{1.3} \]

\[ D_1^{-\beta} u := \frac{1}{\Gamma(\beta)} \int_x^1 (s - x)^{\beta-1} u(s) \, ds. \tag{1.4} \]

When \( q = 1 \), one gets the fractional initial boundary value problem (IBVP) with an additional initial condition \( u(x, 0) = g(x) \); and when \( q = 0 \), it is the fractional boundary value problem (BVP), which can also be regarded as the steady state equation of the associated IBVP. Considering the homogeneous boundary condition and using integration by parts, one can easily get \( D_0 D_x^{-\beta} Du = D_0 D_x^{-\beta} u \) and \( D_0 D_1^{-\beta} Du = D_0 D_1^{-\beta} u \). Then one can reduce the model [14] to a more familiar form, and a basic theoretical framework for its variational solution has been presented in [10]; this will enable us to put focus on the wavelet numerical methods themselves.

This paper is organized as follows. In Section 2, we give a brief recall to the spline scaling and wavelet functions. They have the closed-form expression, which is of course attractive for the fractional operators. In Section 3, we study the computational formulation with respect to the uniform grids. We first discuss the effective way to construct the algebraic system, and then derive the multilevel wavelet preconditioning and the multiresolution multigrid schemes (MMG) for solving the BVP and IBVP, respectively. In Section 4, we give some heuristically adaptive algorithms and show how singularities can be easily detected by wavelet, and put our attention on its efficiency by proposing and testing the adaptive algorithms that concentrate the degrees...
of freedom in the neighborhood of near singularities. The numerical results are shown in Section 5 and we conclude the paper with some remarks in the last section.

2. Preliminaries

In this section, we collect/present some essential properties of the scaling functions and wavelets to make the paper self-contained and more readable. For the details, refer to [25,26] and [8,31]. First, we give the definitions of freedom in the neighborhood of near singularities. The numerical results are shown in Section 5 and we choose the Schoenberg sequence of knots to define the scaling function sets $\Phi_j$ of $L_2$ and $S_j$.

Let $[\phi_j] : \boldsymbol{supp} \phi_j \cap \boldsymbol{supp} \phi_{j+1} \leq 1$; it forms a stable Riesz basis of $S_j$, i.e.,

$$c_\Phi \parallel e_j \parallel_{L_2(\Delta_j)} \leq \sum_{k \in \Delta_j} c_{j,k} \phi_{j,k} \parallel_{L_2(I)} \leq C_\Phi \parallel e_j \parallel_{L_2(\Delta_j)}. $$

and $S_j$ satisfies the Jackson and Bernstein estimates, i.e.,

$$\inf_{v_j \in S_j} \parallel v - v_j \parallel_{L_2(I)} \leq 2^{-jd} \parallel v \parallel_{H^d(I)} \ \forall v \in H^d_0(I),$$

$$\parallel v_j \parallel_{H^s(I)} \leq 2^{js} \parallel v_j \parallel_{L_2(I)} \ \forall v_j \in S_j, \ 0 \leq s \leq \gamma,$
where $\gamma := \sup \{ \nu \in \mathbb{R} : v_j \in H^\nu(I) \forall v_j \in S_j \}$ and by $A \lesssim B$ we mean that $A$ can be bounded by a multiple of $B$, independent of the parameters they may depend on.

Since $\Phi_j$ is a Riesz basis of $S_j$, there exists a dual MRA sequence $\tilde{S}_j = \text{span}\{\tilde{\Phi}_j\}$, which also forms a MRA of $L_2(\Omega)$. And one can define the biorthogonal projector:

$$P_j : L_2(\Omega) \rightarrow S_j, \quad P_j v := \sum_{k \in \Delta_j} \left( v, \tilde{\phi}_{j,k} \right) \phi_{j,k},$$

(2.11)

where $(\cdot, \cdot)$ denotes the $L^2$ inner product. Then $P_{j+1}P_j = P_j$, and for $0 \leq s \leq 1$,

$$\|v - P_j v\|_{H^s(\Omega)} \lesssim 2^{j(s-\gamma)} \|v\|_{H^s(\Omega)}, \quad 0 \leq s < \gamma < d. \tag{2.12}$$

One can also construct the interval biorthogonal wavelet sets $\Psi_j = \{\psi_{j,k}, k \in \nabla_j\}$ and $\tilde{\Psi}_j = \{\tilde{\psi}_{j,k}, k \in \nabla_j\}$; it holds the norm equivalence, i.e., there exist $\tilde{\sigma}, \sigma > 0$ such that for the Sobolev space $H^s(\Omega)$,

$$\left\| \sum_{j \geq j_0-1} \sum_{k \in \nabla_j} d_{j,k}^2 \psi_{j,k} \right\|_{H^s(\Omega)} \approx \sum_{j \geq j_0-1} \sum_{k \in \nabla_j} 2^{2js} |d_{j,k}|^2 \quad \forall s \in (-\tilde{\sigma}, \sigma),$$

(2.13)

where $\psi_{j_0-1,k} := \phi_{j_0,k}, \nabla_{j_0-1} := \triangle_{j_0}, d_{j_0-1,k} := c_{j_0,k}; J_0$ denotes the lowest level. It also means that $\bigcup_{j=J_0-1}^{\infty} 2^{-js}\Psi_j$ is a Riesz basis of $H^0_0(\Omega)$.

Moreover, denote $W_j = \text{span}\{\psi_j\}$. Then the operator $Q_j := P_{j+1} - P_j$ is a projection onto the space $W_j$, having the representation

$$Q_j v = \sum_{k \in \nabla_j} \left( f, \tilde{\psi}_{j,k} \right) \psi_{j,k}. \tag{2.14}$$

And for $0 < \gamma < d$, there exists

$$\left| \left( f, \tilde{\psi}_{j,k} \right) \right| \lesssim \inf_{p \in P_{j+1}} \|f - p\|_{L_2(\text{supp} \tilde{\psi}_{j,k})} \left\| \tilde{\psi}_{j,k} \right\| \lesssim 2^{-j\gamma} \left\| f(\gamma) \right\|_{L_2(\text{supp} \tilde{\psi}_{j,k})}. \tag{2.15}$$

This shows that the wavelet coefficients are small provided that the function is locally smooth, which is the foundation to design wavelet adaptive algorithms.

Note that based on the scaling function sets $\Phi_j$, some other special wavelets can also be constructed. If one demands $W_j = W_j$ and $S_j = S_j$, the semiorthogonal wavelets $\Psi_j$ can be obtained [7]; if one chooses $d = 2$, $\psi(x) = \frac{1}{\sqrt{2}} \phi_{1,1}(x), \Psi_j = \{\psi_{j,k} = 2^j \psi(2^j x - k), k \in \nabla_j\}$, then the interpolation wavelet $\{\Psi_j\}_{j \geq -1}$ is obtained, and it also satisfies the norm equivalence for $s \in (1, \frac{3}{2})$ (Page 605 of [3]); and if $d = 4$, let $\Phi_j = \{\phi_{j,k}, k \in \triangle_j/\{1, 2^j + 1\}\}$, and define $\phi(x), \phi^b(x), \psi(x)$ and $\psi^b(x)$ by

$$\phi(x) = \frac{1}{6} \sum_{i=0}^{4} \left( \begin{array}{c} 4 \\ i \end{array} \right) (-1)^i (x - i)^3_+, \tag{2.16}$$

$$\phi^b(x) = \frac{3}{2} x^2_+ - \frac{11}{12} x^3_+ + \frac{3}{2} (x - 1)^3_+ - \frac{3}{4} (x - 2)^3_+ + \frac{(x - 3)^3_+}{6}, \tag{2.17}$$

$$\psi(x) = -\frac{1}{4} \phi(2x) + \phi(2x - 1) - \frac{1}{4} \phi(2x - 2), \tag{2.18}$$

$$\psi^b(x) = \phi^b(2x) - \frac{1}{4} \phi(2x). \tag{2.19}$$

One has the semi-interpolation spline wavelet $\Psi_j = \{\psi^b(2^j x), \psi_{j,k} \mid 0 \leq k \leq (2^j - 3), \psi^b(2^j (1 - x))\}$, which satisfies the so-called point value vanishing property, and the wavelet expansion coefficients also indicate the regularity.
of the approximation function $[2]$. In practice, the base functions $\phi_{j,1}(\cdot)$ and $\phi_{j,2^j+1}(\cdot)$ are usually added for removing the limitation that the first order derivative of the (to be approximated) function at the boundary needs to be zero.

Finally, we point out that there exists the refinement relations

$$\Phi_j^T = \Phi_{j+1}^T M_{j,0}, \quad \Psi_j^T = \Phi_{j+1}^T M_{j,1}. \quad (2.20)$$

And the space $S_j$ can be written as $S_j = S_{j_0} \oplus W_{j_0} \oplus \cdots \oplus W_{j-1}$. For any $u_j \in S_j$, it follows that

$$u_j = \sum_{k \in \Delta_j} c_{j,k} \phi_{j,k} = \sum_{k \in \Delta_{j_0}} c_{j_0,k} \phi_{j_0,k} + \sum_{j=J_0}^{J-1} \sum_{k \in \Delta_j} d_{j,k} \psi_{j,k}. \quad (2.21)$$

Denote $c_j = (c_{j,k})_{k \in \Delta_j}$, $d_j = (d_{j,k})_{k \in \Delta_j}$, and $d_j = (c_{j_0}, d_{j_0}, \ldots, d_{j-1})$. Then there exists a fast wavelet transform (FWT) between the single-scale and the multiscale representations, i.e.,

$$c_j = M d_j, \quad (2.22)$$

which can be performed with the cost $O(2^j)$; here

$$M = \begin{pmatrix} M_{J-1} & 0 & 0 \\ 0 & I_{J-1} & 0 \\ 0 & 0 & I_{J-2} \end{pmatrix} \cdots \begin{pmatrix} M_{J_0} & 0 \\ 0 & I_{J_0} \end{pmatrix} \quad (2.23)$$

and $M_j = (M_{j,0}, M_{j,1})$.

3. Uniform Schemes

The nonlocal property of fractional operator makes the matrix of its discretizations inevitably dense. We will show that the chosen bases being the dilation and translation of one single function render the matrix to have a special structure, which greatly reduces the cost of computing and storing the entries. In this sense, these kind of bases are superior to the other possible bases, such as the usually used finite element or spectral polynomial bases. Meanwhile, based on the benefits of these bases, a simple diagonal preconditioner and the fast transform are presented to enhance the effectiveness of the widely used nonlinear or linear iterative schemes.

We first consider the BVP of (1.1) with $q = 0$. It has the variational formulation: Find $u \in H^\alpha_0(\Omega)$ with $\alpha = 1 - \beta/2 (0 \leq \beta < 1)$, such that

$$a(u,v) = (f,v) \quad \forall v \in H^\alpha_0(\Omega). \quad (3.1)$$

More precisely, using integration by parts and the adjoint property of fractional integral operator $[9]$ leads to

$$a(u,v) = \left\langle -\kappa_\beta D(p \alpha D^-x + (1 - p) x D^-1 \beta)Du, v \right\rangle \quad (3.2)$$

$$= \kappa_\beta \left\langle p \alpha D^-x D^-\beta Du + (1 - p) x D^-1 \beta Du, Dv \right\rangle$$

$$= \kappa_\beta \left\langle p \alpha D^-x \beta/2 Du, x D^-1 \beta/2 Dv \right\rangle + \kappa_\beta \left( (1 - p) x D^-1 \beta/2 Du, p \alpha D^-x \beta/2 Dv \right).$$

The bilinear form $a(\cdot, \cdot) : H^\alpha_0(\Omega) \times H^\alpha_0(\Omega) \rightarrow \mathbb{R}$ is continuous and coercive $[11]$, i.e.,

$$|a(u,v)| \lesssim \|u\|_\alpha \|v\|_\alpha, \quad a(u,u) \gtrsim \|u\|_\alpha^2. \quad (3.3)$$

For $f \in L^2(\Omega)$, Eq. (3.1) admits a unique solution. Letting $S_J$ be a subspace of $H^\alpha_0(\Omega)$ with order $d$, the Galerkin approximation $u_J$ belonging to $S_J$ satisfies

$$a(u_J, v_J) = (f, v_J) \quad \forall v_J \in S_J. \quad (3.4)$$
If \( u \) is sufficiently smooth, following (2.12) and the Ceâ’s lemma, one gets
\[
\|u - u_J\|_\alpha \lesssim \inf_{v_J \in S_J} \|u - v_J\|_\alpha \lesssim 2^{\left\lfloor (a-d)\right\rfloor} \|u\|_{\mathcal{H}^\alpha(\Omega)}. \tag{3.5}
\]

For space discretization, one can either use the scaling basis \( \Phi_J \) or the multiscale basis \( \Psi^J = \{\Psi_J\}_{j=J_0-1}^J \), generating the following linear systems, respectively,
\[
\begin{align*}
A_J c_J &= F_J, \tag{3.6} \\
\hat{A}_J d_J &= \tilde{F}_J, \tag{3.7}
\end{align*}
\]
where \( A_J = a(\Phi_J, \Phi_J) \), \( F_J = (f, \Phi_J) \), \( \hat{A}_J = a(\Psi^J, \Psi^J) \), \( \tilde{F}_J = (f, \Psi^J) \). There are the following Lemmas.

**Lemma 3.1.** Let \( \phi(x) \in \mathcal{H}^\alpha_0(\Omega) \), \( \text{supp}(\phi) = [0, d] \) and \( \phi_{J,k}(x) := 2^{J/2}\phi(2^Jx - k), 0 \leq k \leq 2^J - d, k \in \mathbb{N} \). Then \( a(\phi_{J,k_1}, \phi_{J,k_2}) = a(\phi_{J,k_1}', \phi_{J,k_2}') \) if and only if \( k_2 - k_1 = k_2' - k_1' \).

**Proof.** For \( \phi(x) \in \mathcal{H}^\alpha_0(\Omega) \), there holds
\[
\begin{align*}
&\left( 0 D_x^{-\beta/2} D\phi_{J,k_2}, 0 D_x^{-\beta/2} D\phi_{J,k_1} \right) \\
&= \frac{1}{\Gamma(\beta)} \int_0^1 \int_0^x (x-s)^{\beta-1} \phi_{J,k_1}'(s) ds \phi_{J,k_2}'(x) dx \\
&= \frac{2^{2J}}{\Gamma(\beta)} \int_{2^{-J}k_2}^{2^{-J}(d+k_2)} \int_0^x (x-s)^{\beta-1} \phi'(2^J s - k_1) ds \phi'(2^J x - k_2) dx \\
&= \frac{2^{2J}}{\Gamma(\beta)} \int_0^d \int_{-k_2}^{2^{-J}(x+k_2)} (2^{-J}(k_2 + x) - s)^{\beta-1} \phi'(2^J s - k_1) ds \phi'(x) dx \\
&= \frac{2^{2J}}{\Gamma(\beta)} \int_0^d \int_{-k_1}^{x+k_2-k_1} (k_2 + x - s - k_1)^{\beta-1} \phi'(s) ds \phi'(x) dx \\
&= \frac{2^{2J}}{\Gamma(\beta)} \int_0^d \int_{-k_1}^{x+k_2-k_1} (x - s + k_2 - k_1)^{\beta-1} \phi'(s) ds \phi'(x) dx,
\end{align*}
\]
which just depends on the value of \( k_2 - k_1 \). The second part of (3.2) can be expressed by its first part, i.e.,
\[
\left( D_x^{-\beta/2} D\phi_{J,k_2}, D_x^{-\beta/2} D\phi_{J,k_1} \right) = \left( D_x^{-\beta/2} D\phi_{J,k_1}, D_x^{-\beta/2} D\phi_{J,k_2} \right). \tag{3.8}
\]

Then the desired result is obtained.

**Lemma 3.2.** Let \( \phi(x) \) and \( \phi_{J,k}(x) \) be given as above, and \( \phi(d/2-x) = \phi(d/2+x) \). Define \( \theta_{J,i}(x) := 2^{J/2}\theta_i(2^J x) \) and \( \tilde{\theta}_{J,i}(x) := 2^{J/2}\tilde{\theta}_i(2^J (1-x)) \) with \( \theta_i(x) \in \mathcal{H}^\alpha(\Omega) \) and \( \text{supp}(\theta_i) = [0, d_i] \), where \( 0 < d_i < d \) and \( i = 1, 2 \). Then
\[
\left( 0 D_x^{-\beta/2} D\theta_{J,i}, D_x^{-\beta/2} D\phi_{J,k} \right) = \left( 0 D_x^{-\beta/2} D\phi_{J,i}, D_x^{-\beta/2} D\tilde{\theta}_{J,i} \right). \tag{3.9}
\]

**Proof.** Similar to Lemma 3.1 it follows that
\[
\begin{align*}
&\left( 0 D_x^{-\beta/2} D\theta_{J,i}, D_x^{-\beta/2} D\phi_{J,k} \right) \\
&= \frac{1}{\Gamma(\beta)} \int_0^1 \int_0^x (x-s)^{\beta-1} \theta_{J,i}'(s) ds \phi_{J,k}'(x) dx \\
&= \frac{2^{2J}}{\Gamma(\beta)} \int_0^d \int_0^{x+k} (x + k - s)^{\beta-1} \tilde{\theta}_{J,i}'(s) ds \phi'(x) dx.
\end{align*}
\]
By the properties of symmetry and compact support, there exists

\[
\begin{aligned}
\left(0D_x^{-\beta/2}D\phi_{J,2^j-k}, xD_1^{-\beta/2}D\tilde{\theta}_{J,i}\right) \\
= \frac{1}{\Gamma(\beta)} \int_0^1 \int_0^x (x-s)^{\beta-1} \phi'_{J,2^j-k}(s) \, ds \, \tilde{\theta}_{J,i}(x) \, dx \\
= \frac{2^{2J}}{\Gamma(\beta)} \int_0^1 \int_0^{1-2^{-j}x} (1-s-2^{-j}x)^{\beta-1} \phi' (2^j s - 2^j + d + k) \, ds \, \theta'_{i}(x) \, dx \\
= \frac{2^{2J\alpha}}{\Gamma(\beta)} \int_0^d \int_{\min\{2^j-k,d\}}^{\max\{0,x-k\}} (s + k - x)^{\beta-1} \phi' (s) \, ds \, \theta'_{i}(x) \, dx \\
= \frac{2^{2J\alpha}}{\Gamma(\beta)} \int_0^d \int_0^{x+k} (x + k - s)^{\beta-1} \theta'_{i}(s) \, ds \, \phi'(x) \, dx,
\end{aligned}
\]

where the Fubini-Tonelli theorem and \( \min\{2^j - k, d\} = d \) are used.

It is also easy to check that for any \( i_1, i_2 \in \{1, 2\} \),

\[
\begin{aligned}
\left(0D_x^{-\beta/2}D\theta_{J,i_1}, xD_1^{-\beta/2}D\theta_{J,i_2}\right) &= \left(0D_x^{-\beta/2}D\tilde{\theta}_{J,i_1}, xD_1^{-\beta/2}D\tilde{\theta}_{J,i_2}\right), \\
\left(0D_x^{-\beta/2}D\phi_{J,k}, xD_1^{-\beta/2}D\theta_{J,i}\right) &= \left(0D_x^{-\beta/2}D\tilde{\theta}_{J,k}, xD_1^{-\beta/2}D\tilde{\theta}_{J,i}\right).
\end{aligned}
\]

Now, from the structure of \( \Phi_{J} \), and the above lemmas, one knows that the matrix \( A_1 := \left(0D_x^{-\beta/2}D\Phi_{J}, xD_1^{-\beta/2}D\Phi_{J}\right) \)
has a quasi-Toeplitz structure, that is, it is a Toeplitz matrix after removing very few rows and columns near the boundaries. More precisely, for \( d = 2 \), it is a full Toeplitz matrix, but for \( d = 3 \) and \( d = 4 \), they have the following structures, respectively,

\[
\begin{pmatrix}
    a_1 & r(a_2)^T & 0 \\
    a_1 & H(2^j-2) & a_2 \\
    a_2 & r(a_1)^T & a_1
\end{pmatrix}_{2^j \times 2^j},
\]

\[
\begin{pmatrix}
    a_1 & a_2 & r(a_2)^T & 0 & 0 \\
    a_3 & a_4 & r(a_2) & a_1 & 0 \\
    a_3 & a_4 & H(2^j-3) & a_2 & a_1 \\
    a_5 & a_6 & r(a_4)^T & a_4 & a_2 \\
    a_7 & a_5 & r(a_3) & a_3 & a_1
\end{pmatrix}_{(2^j+1) \times (2^j+1)}
\]

where \( a_i \) are real numbers; \( a_i \) are vectors, \( r(a_i) \) the reverse order of \( a_i \); and \( H_{N \times N} \) is Toeplitz matrix.

The fact that the bases are obtained by dilating and translating of a single function and the symmetry of the bases are essential for obtaining the above results, recalling that they do not hold for the general finite element (except linear element) and spectral methods. For the high order finite difference methods, the similar results can be got after modifying the approximation near the boundary for recovering the desired accuracy [3,42], but it seems that the general theoretical results (stability, convergence and so on) are hard to obtain. Further results for the generated matrix are

\[
A_r := \left(0D_1^{-\beta/2}D\Phi_{J}, 0D_x^{-\beta/2}D\Phi_{J}\right) = \left(0D_x^{-\beta/2}D\Phi_{J}, xD_1^{-\beta/2}D\Phi_{J}\right)^T = A_1^T,
\]

\[
\left(0D_x^{-\beta/2}D\phi_{J,k_1}, xD_1^{-\beta/2}D\phi_{J,k_2}\right) = 0 \quad \forall k_2 - k_1 \leq -d.
\]
Remark 3.1. The structure of $\Phi_J$ also allows one to compute its Riemann-Liouville fractional derivative easily, which can greatly reduce the computational complexity of generating the differential matrix in Galerkin and collocation methods (see Section 5). As an example, we present the techniques for $d = 4$ in Appendix, being similar for other values of $d$. Then combining with (3.13) and (3.17), the left differential matrix $A_l = (0D_{x}^{1-\beta}\Phi_J, D\Phi_J)$ can be calculated exactly or numerically with the cost $O(N)$, being superior to the traditional finite element and spectral approximation with the cost $O(N^2)$ [14][17].

3.1. Multilevel Preconditioning

For an algebraic system with dense matrix, a well convergent iterative method generally has the computational cost $O(N \log(N))$ or $O(N^2)$, which is much less than the cost $O(N^3)$ of the direct method. Moreover, a well conditional number and “bunching of eigenvalues” usually bring good numerical stability and fast convergence speed [13]. In general, for a linear system $Ax = b$, a satisfactory preconditioned system $Bx' = b'$ should have the property

$$\|B\| \leq C, \quad \|B^{-1}\| \leq C, \quad C \text{ is a moderate-sized constant independent of } N;$$

and the computational cost for the preconditioning step is cheap. Here, both the matrix $A_J$ and $\hat{A}_J$ are dense, and their condition numbers are of order $O(2^{2J_J})$; see Table 7 for Example 5.2. But with the aid of wavelet bases, by the norm equivalence, a simple diagonal scaling can lead to a good preconditioned system. In fact, define

$$\nabla^J = \triangle J_0 \cup \nabla J_0 \cup \cdots \cup \nabla J_{-1},$$

$$K = \text{diag}\left(\frac{2}{2-2^{J_0}}, \ldots, \frac{2}{2-2^{J_0}}, \ldots, \frac{2}{2-2^{J_0}}, \ldots, \frac{2}{2-(J-1)}, \ldots, \frac{2}{2-(J-1)}\right).$$

Combining the ellipticity [33], norm equivalence [2.13], and the Riesz representation theorem, one gets that for all $x \in l_2(\nabla^J),$

$$\|K\hat{A}_JKx\|_{l_2(\nabla^J)} = \sup_{y \in l_2(\nabla^J)} \left(\frac{K\hat{A}_JKx, y}_{l_2(\nabla^J)}\right),$$

$$= \sup_{y \in l_2(\nabla^J)} \frac{a(x^T K\Phi^J, y^T K\Phi^J)}{\|y\|_{l_2(\nabla^J)}} \lesssim \|x^T K\Phi^J\|_a \|y^T K\Phi^J\|_a \lesssim \|x\|_{l_2(\nabla^J)},$$

$$\|K\hat{A}_JKx\|_{l_2(\nabla^J)} \lesssim \frac{\|x^T K\Phi^J\|_a}{\|x\|_{l_2(\nabla^J)}} \lesssim \|x\|_{l_2(\nabla^J)}. \quad (3.18)$$

Therefore, there exist $C_1, C_2$ not depending on $J$ such that

$$C_1 \|x\|_{l_2(\nabla^J)} \leq \|K\hat{A}_JKx\|_{l_2(\nabla^J)} \leq C_2 \|x\|_{l_2(\nabla^J)}.$$

(3.19)

Now, one arrives at

$$\|K\hat{A}_JK\| \lesssim C_2, \quad \|(K\hat{A}_JK)^{-1}\| \lesssim (1/C_1), \quad \text{cond}_2(K\hat{A}_JK u_J) = \|K\hat{A}_JK\|/(K\hat{A}_JK)^{-1} \lesssim (C_2/C_1). \quad (3.20)$$

(3.21)

The norm equivalence implies $a(\psi_{j,k}, \psi_{j,k}) \sim 2^{2j_J}$, so one can also define matrix $K$ by the inverse square root of the diagonal of $\hat{A}_J$, and (3.19) and (3.21) still hold. Usually the current $K$ performs better since it uses the information directly from the stiffness matrix, and we will use it in Section 5. Moreover, the cost of generating $K$ is only $O(J)$; this is because that by using the translation property of the inner wavelet on the same level,
one just needs to calculate the entries $a(\psi_j, k, \psi_j, k)$ near the boundaries and one in the inner part without the necessity to assemble $A_J$.

Now, one can rewrite (3.7) as the two-sided preconditioned form

$$K \hat{A}_J K^{-1} d_J = K \hat{F}_J.$$  (3.22)

Further using (2.22), one gets that

$$KM^T A_J MK^{-1} M^{-1} c_J = KM^T F.$$  (3.23)

A straightforward product of $A_J$ or $\hat{A}_J$ to a given vector needs a computational cost $O(2^J)$. But if one uses the quasi-Toeplitz structure of the matrix, the computational cost can be reduced to $O(J2^J)$. In fact, one can rewrite $A_J$ as

$$A_J = \text{diag}(K_1) A_t + \text{diag}(K_2) A_r,$$  (3.24)

where $K_1$ and $K_2$ denote the coefficient vectors, formed by the coefficient of space fractional derivative taking values at the discretized intervals, and $A_t$ and $A_r$ are quasi-Toeplitz matrices. Using the FFT to the matrix-vector product makes the computational cost $O(J2^J)$ [31]. Finally, because the FWT (having the matrix representation $M$ or $M^T$, which denotes the primal reconstruction or the dual decomposition [34]) can be implemented with the cost $O(2^J)$, if the CG scheme (symmetric) is applied to (3.23) or to the corresponding normal equation (asymmetric), the well conditioned number of the matrix implies that the convergence rate is independent of the level $J$; then we can solve it with the total operations $O(J2^J)$. For the general iterative schemes, such as GMRES or Bi-CGSTAB, usually one can show that the system with clustered spectrum and well conditioned number after preconditioning has an accelerated convergence. What’s more, compared with the most existing preconditioners which require the solving of a linear system (see, e.g., the ILU [18] and the Strang [15]), the wavelet preconditioning operation reduces to the matrix-vector product, where FWT can be used.

3.2. Multiresolution Multigrid Method

The multiresolution method based on the finite difference discretization for solving the fractional IBVPs have been developed in [0,24], where the transition operators (restriction and prolongation operators) between the grids are chosen as the full weight and interpolation operators. In this Subsection, we investigate the MMG method for solving fractional IBVPs. We will show that the transition operators in the MRA background can be more straightforwardly defined. And using the techniques presented in the above content, the MMG scheme can also be fast implemented.

Denoting $S_j$ as the subspace, $A_J : S_j \rightarrow S_j$ with $(A_J \omega, v_j) := a(\omega, v) \forall v_j \in S_J$ and $Q_J : L^2 \rightarrow S_J$ with $(Q_J \rho, v_j) := (\rho, v) \forall v_j \in S_J$, one arrives at the semidiscrete form: Find $u_J(t) \in S_J, t \geq 0$ such that

$$\left\{ \begin{array}{l}
\frac{\partial u_J}{\partial t} + A_J u_J = f_J(t) \\
u_J(0) = u_J^0 \in S_J.
\end{array} \right.$$  (3.25)

Taking the time mesh as $0 \equiv t_0 < t_1 < \cdots < t_{N-1} < t_N \equiv T$ and the stepsizes $\Delta t_n = t_{n+1} - t_n$, $n = 0, \ldots, N-1$, one gets the backward Euler multiresolution Galerkin method (B-MGM)

$$(U_{J,n+1}^n, v_j) + \Delta t_n a(U_{J,n+1}^n, v_j) = (U_{J,n}^n + \Delta t_n f(t_{n+1}), v_j) \forall v_j \in S_J;$$  (3.26)

and the Crank-Nicolson multiresolution Galerkin method (CN-MGM)

$$\left( \frac{\partial U_{J,n+1}^n}{\partial t}, v_j \right) + a\left( \frac{U_{J,n+1}^n + U_{J,n}^n}{2}, v_j \right) = (f(t_{n+1/2}), v_j) \forall v_j \in S_J,$$  (3.27)
where \( \mathcal{B}U_j^{n+1} = (U_j^{n+1} - U_j^n)/\Delta t_n \). Introduce the bilinear form \( B_{n+1}(u,v) := (u,v) + \lambda \Delta t_n a(u,v) \), where \( \lambda = 1 \) for the B-MGM and \( \lambda = 1/2 \) for the CN-MGM. Define \( B_j^{n+1} : S_j \rightarrow S_j \) with \( (B_j^{n+1}\rho_j, v_j) = B_{n+1}(\rho_j, v_j) \) \( \forall v_j \in S_j \), and the operator \( P_{n+1} : \mathcal{H}_0^1(\Omega) \rightarrow S_j \) with \( B_{n+1}(P_{n+1}\rho, v_j) = B_{n+1}(\rho, v_j) \) \( \forall v_j \in S_j \). Then the MGM schemes can be rewritten uniformly as the form

\[
B_j^{n+1}U_j^{n+1} = g_j^{n+1},
\]

where \( g_j^{n+1} := U_j^n + \Delta t_nQ_jf(t_{n+1}) \) for the B-MGM and \( g_j^{n+1} := -\Delta t_nA_jU_j^n + U_j^n + \Delta t_nQ_jf(t_{n+1}/2) \) for the CN-MGM, respectively. Suppose that \( U_j = \sum_{k \in \Delta_j} c_j,k \phi_j,k \in S_j \), and define \( c_j,k, \tilde{g}_j \in \mathbb{R}^{p(\Delta_j)} \), \( (c_j)_k := c_j,k, (\tilde{g}_j)_k := (g_j, \phi_j,k) \), \( k \in \Delta_j \). Denoting \( B_j^{n+1} = (B_{n+1}(\phi_j,k, \phi_{j,i}))_{k,i \in \Delta_j} \), one also gets the algebraic representation of (3.28) given by \( B_j^{n+1}\phi_j,k = \tilde{g}_j^{n+1} \).

The basic iteration algorithm for the operator equation (3.28) is given by (3.29)

\[
U_{j+1}^{n+1,l} = U_{j}^{n+1,l} + R_j^{n+1}\left( g_j^{n+1} - B_j^{n+1}U_j^{n+1,l} \right), \quad l = 0,1,2,\ldots
\]

with the error propagation operator \( K_j^{n+1} := I - R_j^{n+1}B_j^{n+1} \) and the iterator \( R_j^{n+1} : S_j \rightarrow S_j \). For the damped Richardson and Jacobi methods, the iterators are, respectively, given by

\[
R_j^{n+1}g = \omega \sigma(B_j^{n+1})^{-1} \sum_{k \in \Delta_j} (g, \phi_j,k) \phi_j,k \quad \forall g \in S_j; \quad (3.30)
\]

\[
R_j^{n+1}g = \omega \sum_{k \in \Delta_j} B_{n+1}(\phi_j,k, \phi_j,k)^{-1} (g, \phi_j,k) \phi_j,k \quad \forall g \in S_j; \quad (3.31)
\]

they can also be regarded as the correction with subspaces decomposition \( S_j = \sum_{k \in \Delta_j} V^k_j \) with \( V^k_j = \text{span} \{ \phi_j,k \} \); and one can also rewrite the Jacobi iterator as

\[
R_j^{n+1} = \omega \sum_{k \in \Delta_j} P_j^{n+1,k}(B_j^{n+1})^{-1}, \quad (3.32)
\]

where \( P_j^{n+1,k} : S_j \rightarrow V^k_j \) with \( B_{n+1}(P_j^{n+1,k}v_j, \phi_j,k) = B_{n+1}(v_j, \phi_j,k) \) \( \forall v_j \in S_j \). Now, the MMG V-cycle algorithm for (3.28) reads

\[
U_{j+1}^{n+1,l+1} = U_{j}^{n+1,l} + \mathcal{M}_j^{n+1}\left( g_j^{n+1} - B_j^{n+1}U_j^{n+1,l} \right), \quad l = 0,1,2,\ldots
\]

and the multigrid iterator \( \mathcal{M}_j^{n+1} \) is defined in Algorithm 4 by induction, where \( R_j^{n+1} \) and \( K_j^{n+1} : S_j \rightarrow S_j \) are defined in the same way as \( R_j^{n+1} \) and \( K_j^{n+1} \). Obviously, the MMG error propagation operator satisfies

\[
I - \mathcal{M}_j^{n+1}B_j^{n+1} = (K_j^{n+1})^{m_2(j+1)}(I - \mathcal{M}_j^{n+1}Q_jB_j^{n+1}) (K_j^{n+1})^{m_1(j+1)}
\]

\[
= (K_j^{n+1})^{m_2(j+1)}(I - P_j^{n+1}) (K_j^{n+1})^{m_1(j+1)}
\]

\[
+ (K_j^{n+1})^{m_2(j+1)} (I - \mathcal{M}_j^{n+1}B_j^{n+1}) P_j^{n+1} (K_j^{n+1})^{m_1(j+1)}
\]

where the relation \( Q_jB_j^{n+1} = B_j^{n+1}P_j^{n+1} \) has been used; \( I \) just denotes the usual two-grid error propagation operator; and \( m_1(j+1) \) (or \( m_2(j+1) \)) means that the iterative times \( m_1 \) (or \( m_2 \)) may depend on the level \( j+1 \).
Algorithm 1 MMG V-CYCLE ITERATOR

1: Fix \( t = t_{n+1} \); for \( j = J_0 \), define \( \mathcal{M}_0^{n+1} = (\mathcal{B}_0^{n+1})^{-1} \). Assume that \( \mathcal{M}_j^{n+1} : S_{j-1} \rightarrow S_j \) is defined. For \( g \in S_j \), define the iterator \( \mathcal{M}_j^{n+1} : S_j \rightarrow S_j \) through the following steps:

2: (1) Pre-smoothing: For \( x_0^{n+1} = 0 \in S_j \) and \( l = 1, \ldots, m_1(j) \),
\[
x_l^{n+1} = x_{l-1}^{n+1} + B_j^{n+1} (g - B_j^{n+1} x_{l-1}^{n+1})
\]

3: (2) Coarse grid correction:
\[
x_m^{n+1}(j) + l = x_m^{n+1}(j) + \mathcal{M}_j^{n+1} Q_{j-1} \left( g - B_j^{n+1} x_m^{n+1}(j) \right)
\]

4: (3) Post-smoothing: For \( l = m_1(j) + 2, \ldots, m_1(j) + m_2(j) + 1 \),
\[
x_l^{n+1} = x_{l-1}^{n+1} + B_j^{n+1} (g - B_j^{n+1} x_{l-1}^{n+1})
\]

5: Define \( \mathcal{M}_j^{n+1} g = x_m^{n+1}(j) + m_2(j) + 1 \)

Recall that the refinement relation \([22,20]\), one can get the prolongation matrix \( M_{j,0} \) straightforward, and it holds
\[
\begin{align*}
\rho_j^{n+1} &= M_{j,0} \rho_j^{n+1} + U_j^{n+1} \in S_j \subset S_{j+1}; \\
Q_j^{n+1} &= M_{j,0} Q_j^{n+1} + U_j^{n+1} \in S_{j+1},
\end{align*}
\]
where the meaning of \( r_j^{n+1} \) is defined after Eq. \((3.28)\). This means that the transpose of \( M_{j,0} \) is just the restriction matrix. Noticing that \( B_j^{n+1} U_j^{n+1} = Q_j B_{j+1}^{n+1} U_j^{n+1} \forall U_j^{n+1} \in S_j \), there holds
\[
B_j^{n+1} \rho_j^{n+1} = B_j^{n+1} U_j^{n+1} = M_{j,0} B_{j+1}^{n+1} U_j^{n+1} = M_{j,0} B_{j+1}^{n+1} M_{j,0} \rho_j^{n+1},
\]
i.e.,
\[
B_j^{n+1} = M_{j,0} B_{j+1}^{n+1} M_{j,0},
\]
which actually is the Galerkin identity, facilitating the convergence analysis, but this is not true for the difference method. Note that the quasi-Topelitz structure of \( B_j^{n+1} \) makes it feasible to be generated directly with the cost \( O(2^j) \). Using the fast algorithms (FFT and FWT), the matrix-vector product can preformed with the cost \( O(j 2^j) \). So the total computational count per MMG step is \( O(j 2^j) \) and the storage cost is \( O(2^j) \).

In the following, we present the convergence analysis of the MMG when \( A \) is a Riesz derivative and \( m_1(j) = m_2(j) = m_0 \); see Algorithm 1. It is easy to check that \( B_j^{n+1} \) is symmetric, \( P_j^{n+1} \) and \( Q_j^{n+1} \) are \( A \)-orthogonal projectors, and \( K_j^{n+1} \) and \( I - M_j^{n+1} B_j^{n+1} \) are \( A \)-selfadjoint; all of them are considered with respect to \( B_{n+1}(\cdot, \cdot) \).

Lemma 3.3 (see \[3\]). Assume that \( B_j^{n+1} : S_j \rightarrow S_j \) is symmetric with respect to \( (\cdot, \cdot) \), positive semi-definite, and satisfies
\[
\begin{align*}
\left\{ B_{n+1} (K_j^{n+1}) v_j, v_j \right\} &\geq 0 \quad \forall v_j \in S_j, \\
\left\{ (B_j^{n+1})^{-1} v_j, v_j \right\} &\leq \epsilon B_{n+1} (v_j, v_j) \quad \forall v_j \in (I - P_j^{n+1}) S_j.
\end{align*}
\]
Then we have
\[
0 \leq B_{n+1} \left( (I - M_j^{n+1} B_j^{n+1}) v_j, v_j \right) \leq \delta B_{n+1} (v_j, v_j) \quad \forall v_j \in S_j,
\]
where \( \delta = \epsilon / (\epsilon + 2m_0) \).
Since \( I - M_j^{n+1}B_j^{n+1} \) is \( A \)-selfadjoint, \( (I - M_j^{n+1}B_j^{n+1}) \) actually means that its spectral radius

\[
\sigma(I - M_j^{n+1}B_j^{n+1}) = \|I - M_j^{n+1}B_j^{n+1}\|_A \quad \text{(3.40)}
\]

\[
= \sup_{0 \neq v \in S_j} \frac{B_{n+1}((I - M_j^{n+1}B_j^{n+1})v, v)}{B_{n+1}(v, v)} \leq \delta < 1.
\]

When \( \omega \in [c_0, 1] \), \( 0 < c_0 \leq 1 \), the Richardson method obviously satisfies the requirements of Lemma 3.3. Since the damped Jacobi iteration converges under the condition \( 0 < \omega < 2/\sigma(R_j^{n+1}B_j^{n+1}) \), for any \( v_j \in S_j \), there exists

\[
B_{n+1}(k_j^{n+1}v_j, k_j^{n+1}v_j) = B_{n+1}(v_j, v_j) - 2\omega B_{n+1}(R_j^{n+1}B_j^{n+1}v_j, v_j) + \omega^2 B_{n+1}(R_j^{n+1}B_j^{n+1}v_j, R_j^{n+1}B_j^{n+1}v_j)
\]

\[
= B_{n+1}(v_j, v_j) - 2\omega \left( \left( R_j^{n+1}B_j^{n+1}v_j \right) + \omega B_{n+1}(v_j, v_j) \right) + \omega^2 \left( \left( R_j^{n+1}B_j^{n+1}v_j \right) \left( R_j^{n+1}B_j^{n+1}v_j \right) \right)
\]

\[
\leq B_{n+1}(v_j, v_j) - \omega \left( 2 - \omega \sigma(R_j^{n+1}B_j^{n+1}) \right) \left( R_j^{n+1}B_j^{n+1}v_j, R_j^{n+1}B_j^{n+1}v_j \right).
\]

Then it is sufficient to take \( 0 < \omega < 1/\sigma(R_j^{n+1}B_j^{n+1}) \) for getting the first condition in (3.38).

\[
((R_j^{n+1})^{-1}v_j, v_j) = \sum_{k \in \Delta_j} ((R_j^{n+1})^{-1}v_j, c_{j,k}^n \phi_j,k) = \sum_{k \in \Delta_j} B_{n+1}(c_{j,k}^n \phi_j,k)
\]

\[
\leq \sqrt{\sum_{k \in \Delta_j} B_{n+1}(\mathcal{E}_j^{n+1}v_j, \mathcal{E}_j^{n+1}v_j)} \sqrt{\sum_{k \in \Delta_j} B_{n+1}(c_{j,k}^n \phi_j,k, c_{j,k}^n \phi_j,k)}
\]

\[
= \sqrt{\sum_{k \in \Delta_j} (\mathcal{E}_j^{n+1}v_j, (R_j^{n+1})^{-1}v_j)} \sqrt{\sum_{k \in \Delta_j} B_{n+1}(c_{j,k}^n \phi_j,k, c_{j,k}^n \phi_j,k)}
\]

\[
= \frac{1}{\omega} ((R_j^{n+1})^{-1}v_j, v_j) \sqrt{\sum_{k \in \Delta_j} B_{n+1}(c_{j,k}^n \phi_j,k, c_{j,k}^n \phi_j,k)},
\]

where \( v_j = \sum_{k \in \Delta_j} c_{j,k}^n \phi_j,k \in S_j \) and \( \mathcal{E}_j^{n+1} := P_j^{n+1,k}(B_j^{n+1})^{-1}(R_j^{n+1})^{-1} \).

By Bernstein estimate and uniform stability given in (2.10) and (2.8), one gets

\[
\sum_{k \in \Delta_j} B_{n+1}(c_{j,k}^n \phi_j,k, c_{j,k}^n \phi_j,k) \leq 2^{2j\alpha} \sum_{k \in \Delta_j} \|c_{j,k}^n \phi_j,k\|_{L^2(\Omega)}^2 \sim 2^{2j\alpha} \|v_j\|_{L^2(\Omega)}^2.
\]

By the Aubin-Nitscale trick \( \Omega_1 \theta_0 \Omega_2 \), there exists

\[
\|(I - P_j^{n+1})v_j\|_{L^2(\Omega)}^2 \leq \frac{2^{-2j\alpha}}{2} \|(I - P_j^{n+1})v_j\|_{H^\alpha(\Omega)}^2 \sim 2^{-2j\alpha} B_{n+1}((I - P_j^{n+1})v_j, (I - P_j^{n+1})v_j).
\]

Noting that \((I - P_j^{n+1})v_j \in S_j\), then the second requirement in (3.38) holds. The proof of the convergence for MMG with Jacobi iterator is completed.

4. Multiscale Adaptive Schemes

Although there are works to discuss the low regularity (especially the weaker regularity at the area close to boundary) of the solutions for fractional PDEs, it seems few of them are for designing the adaptive algorithms,
which have been well developed for classical PDEs. In the classical finite elements approximation, adaptivity is usually driven by so-called local a posteriori error estimate (an efficient and reliable error indicator consisting of local terms and being easy to compute). In the following we first show the challenge when using the traditional finite element method to adaptively solve the fractional BVPs, then numerically demonstrate the performance of the wavelet adaptation algorithm.

For the linear element and \( \beta \in (1/2, 1) \), with a given positive integer \( K \), we define the mesh

\[
0 \equiv x_0 < x_1 < \cdots x_{K-1} < x_K \equiv 1, \quad I_i = (x_{i-1}, x_i).
\]

Then the posteriori error for the BVP is bounded by

\[
a(u - u_h, u - u_h) \lesssim \sum_{i=1}^K h_i^{2\alpha} \| f + \kappa_\beta (p_0 D_2^{2\alpha} u_h + (1 - p) p_2 D_1^{2\alpha} u_h) \|_{L^2(I_i)}^2, \tag{4.1}
\]

where \( h_i = x_i - x_{i-1} \), and \( \beta \in (1/2, 1) \) ensures that \( p_0 D_2^{2\alpha} u_h \) and \( p D_1^{2\alpha} u_h \) belong to \( L_2 \).

Now we prove (4.1). Denote by \( \Pi_h \) the operator for the piecewise linear interpolation associated with \( \{ x_i \} \) and \( e_h = u - u_h \), For any \( v \in H_0^\alpha(\Omega) \), there exists

\[
a(e_h, v) = a(e_h, v - \Pi_h v) = (f, v - \Pi_h v) - a(u_h, v - \Pi_h v).
\]

Combining (3.2), \((v - \Pi_h v)(x_{i-1}) = (v - \Pi_h v)(x_i) = 0\), and the regularity of \( u_h \) leads to

\[
a(e_h, v) = \sum_{i=1}^K \int_{I_i} \left( f + \kappa_\beta (p_0 D_2^{2\alpha} u_h + (1 - p) p_2 D_1^{2\alpha} u_h) \right) (v - \Pi_h v) \, dx
\]

\[
\leq \sum_{i=1}^K \| f + \kappa_\beta (p_0 D_2^{2\alpha} u_h + (1 - p) p_2 D_1^{2\alpha} u_h) \|_{L^2(I_i)} \| v - \Pi_h v \|_{L^2(I_i)}
\]

\[
\lesssim \sum_{i=1}^K h_i^{\alpha} \| f + \kappa_\beta (p_0 D_2^{2\alpha} u_h + (1 - p) p_2 D_1^{2\alpha} u_h) \|_{L^2(I_i)} \| v \|_{H^\alpha(\Omega)}
\]

\[
\lesssim \sqrt{\sum_{i=1}^K h_i^{2\alpha} \| f + \kappa_\beta (p_0 D_2^{2\alpha} u_h + (1 - p) p_2 D_1^{2\alpha} u_h) \|_{L^2(I_i)}^2} \| v \|_{H^\alpha(\Omega)}.
\]

Then (4.1) follows by taking \( v = e_h \) in the above inequality and using \( a(v, v) \sim \| v \|_{H^\alpha(\Omega)} \). The term \( \| \cdot \|_{L^2(I_i)}^2 \) involves nonlocal calculations; so it can not be used directly as a local error indicator. The following in Algorithm 2 we will show that for the wavelet methods of fractional BVPs, the local regularity indicator can be the wavelet coefficient when the (to be determined) solution is represented by the multiscale bases: small coefficient implies good local regularity while big one indicates the opposite. One of the main features of Algorithm 2 is that the finest grid resolution can be automatically determined by the given tolerance \( \epsilon(j) \). In order to gain a more robust and faster algebra solver, \( \psi_{j,k} \) has been scaled by the inverse square root of \( a(\psi_{j,k}, \psi_{j,k}) \), and the multiscale approximation of the solution at the current scale has been as an initial guess for the iteration in the finer scale obtained after adding the waveslets. For constructing the refined index set, thanks to the tree structure of wavelet singularity detection, we first include a coarsening step by thresholding the latest available wavelet coefficients to get a significant index set, then add all their children. If \( j = l + 1 \) and \( k \in \{ 2\lambda, 2\lambda + 1 \} \), then the wavelet indexed by \( (j, k) \) is called a child of the wavelet indexed by \( (l, \lambda) \). One can further extend the index set by including the horizontal neighbors of the wavelet indices already included. Such an extended index set associated with the index \( (l, \lambda) \) is called an adjacent zone, which is denoted by \( N_{l,\lambda} \). In Algorithm 2 the index set is continuously updated to resolve the local structures that appear in the solution. One can dynamically
Algorithm 2 ADAPTIVE WAVELET SOLVER FOR THE BVP

1: Given \( \epsilon(j), I_{\text{max}}, J_0 \)
2: \( m = 0 \)
3: Solve the equation in space \( V_{J_0+1} \) to get the initial approximation coefficients \((c_{J_0}^0, d_{J_0}^0)\) and the index \( \Lambda^m = (J_0, \lambda) \in \nabla_{J_0} \)
4: repeat
5: Determine the significant index set \( \Lambda \) by \( \epsilon(j) \)
6: Check the adjacent zone index set \( N_{l, \lambda} \) of each \((l, \lambda) \in \Lambda; \) denote \( \Lambda^{m+1} = \Lambda \cup \bigcup_{(l, \lambda) \in \Lambda} N_{l, \lambda} \) and establish \( \Lambda^{m+1} = \Lambda \cup \Lambda_N \)
7: for \((l, \lambda) \in \Lambda^{m+1}\) do
8: \[ \begin{cases} d_{l, \lambda}^{m+1} = d_{l, \lambda}^m, & (l, \lambda) \in \Lambda^m \\ 0, & \text{otherwise} \end{cases} \]
9: end for
10: \[ \begin{cases} c_{J_0}^{m+1} = c_{J_0}^m \\ d_{J_0}^{m+1} = \sum_{(l, \lambda) \in \Lambda^{m+1}} d_{l, \lambda}^{m+1} \\. \\
\end{cases} \]
11: Solve the algebraic matrix equation, resulted from the discretization in the nonlinear approximation space \( \hat{V}_{J_0+m+1}(\Omega) \subset \hat{V}_{J_0+m+1}(\Omega) \), by appropriate iterative scheme with the initial guess \((c_{J_0}^{m+1}, d_{J_0}^{m+1}, \ldots, d_{J_0+m+1})\)
12: Determine \( \Lambda = \{(l, \lambda) : (l, \lambda) \in \Lambda^{m+1}, |d_{l, \lambda}^m| \geq \epsilon(l)\} \)
13: \( m = m + 1 \)
14: until \( m > I_{\text{max}} \) or \( \Lambda = \emptyset \) (empty set)

adjust the number and locations of the wavelets used in the wavelet expansion, reducing significantly the cost of the scheme while providing enough resolution in the regions where the solution varies significantly. The \( m \)-th approximation of the solution is given by

\[
\hat{u}_{J_0+m} = \sum_{(l, \lambda) \in \Delta_{J_0} \cup \Lambda^m} d_{l, \lambda} \hat{\psi}_{l, \lambda},
\]

where \( \Delta_{J_0} \cup \Lambda^m \) is the irregular index set, and \( \hat{\psi}_{l, \lambda} \) represents the normalization of \( \psi_{l, \lambda} \). Finally, after getting the sufficiently accurate approximation, the corresponding single scaling representation can be got by the FWT.

The following we develop adaptive algorithm for time-dependent problem. With the time partition \( 0 \equiv t_0 < t_1 < \cdots < t_N \equiv T \) and stepsizes \( \Delta t_n = t_{n+1} - t_n, \) \( n = 0, \ldots, N - 1, \) for some time \( \tau \in [t_{n-1}, t_n] \), one arrives at a system of the form

\[
\partial_t u |_{\tau} + Au(\cdot, \tau) = f(\cdot, \tau).
\]

The numerical solution \( \overline{U}_j^m \) can be uniquely represented (a unique decomposition) in one of the subspaces of \( S_J := S_{J_0} \cup W_{J_0} \cup \cdots W_{J-1} \):

\[
\overline{U}_j^m(x) = \sum_{k \in \Delta_{J_0}} \overline{c}_{j_0, \lambda} \phi_{j_0, \lambda} + \sum_{j=J_0}^{J-1} \sum_{\lambda \in \nabla_j \cap G} \overline{d}_{j, \lambda} \psi_{j, \lambda},
\]

which is equivalent to the unique coefficient vector

\[
\overline{\psi}^m := \left( \overline{c}_{j_0}^0, \overline{d}_{J_0}, \ldots, \overline{d}_{J-1} \right).
\]

For establishing the algebraic system of \( \overline{\psi}^m \), one can still use the Galerkin scheme; as an extension, the collocation method based on the semi-interpolation wavelet (see eq. 2.16–2.19) can also be considered, which replaces the
test function \( \{ \phi_{j_0, \lambda} \} \cup \{ \psi_{j, \lambda} \}, (j, \lambda) \in \mathcal{G}^n \) of the Galerkin scheme by the Dirac distribution \( \delta \) centered at \( x_i \), being the collocation point corresponding to the index set \( \mathcal{G}^n \), a subset of \( \{ k/2^j \}_{k=1}^{2^j-1} \cup \{ (2k + 1)/2^{j+1} \}_{k=0,j=J_0} \\
\) Generally, the collocation method is more convenient and efficient for problems with variable coefficients and/or nonlinear terms. Details of such a scheme are provided in Algorithm 3, the steps are very similar to Algorithm

\section*{Algorithm 3: ADAPTIVE WAVELET SOLVER FOR THE IBVP}

1: Given time partition \( \{ t_n \}, J_{\text{max}} \) and threshold \( \epsilon(j) \)
2: Construct the initial irregular index set \( \mathcal{G}^0 \) and the multiscale coefficients \( \varphi^0 \) by \( u_0 \)
3: for \( n = 1, \ldots, N \) do
4: Based on \( \varphi^{n-1} \) to solve \( \varphi^n \) on the index set
5: Threshold \( \varphi^n \) to obtain the significant index set
6: \( \mathcal{G}^n := \{ (j, \lambda) \in \mathcal{G}^{n-1} : |d_{j, \lambda}| \geq \epsilon(j) \} \)
7: if \( \mathcal{G}^{n-1} \) and \( \mathcal{G}^n \) are different then
8: For every index \( (j, \lambda) \in \mathcal{G}^n \) not in \( \mathcal{G}^{n-1} \), the corresponding wavelet coefficients are initialized with 0, and denote the result by \( \varphi^n \)
9: end if
10: end for

\section{5. Numerical results}

In order to illustrate the accuracy and efficiency of the proposed numerical schemes, we apply them to solve the BVP and/or IBVP (1.1). Example 5.1 is used to discuss the implementations of the MGM for the BVP and the collocation method for the IBVP, and in particular the convergence orders are carefully verified. We use Example 5.2 to show the powerfulness of the provided multilevel preconditioner and MMG. And Example 5.3 is used to illustrate the effectiveness of the presented wavelet adaptive schemes.

\textbf{Example 5.1.}

Consider the MGM for the BVP (1.1) with \( q = 0 \) and \( p = \kappa_\beta = 1 \), and the source term

\[ f(x) = \frac{2x^\beta}{\Gamma(\beta + 1)} - \frac{\Gamma(\nu + 1)}{\Gamma(\nu + \beta - 1)} x^{\nu + \beta - 2}. \]

The exact solution of the problem is \( u(x) = x^\nu - x^2 \). It is well known that if \( \nu > 0 \) and \( \nu \notin \mathbb{N} \), then \( u \in \mathcal{H}^{\nu+1/2-\epsilon}(\Omega) \). For \( \beta = 4/5 \), the numerical results are listed in Tables 1 and 2, which confirm that if the analytical solution is smooth enough, the convergence order is \( d \) and \( d - \alpha \) in the \( L_2 \) and \( \mathcal{H}^\alpha \)-norm, respectively. Otherwise the convergence order is limited by the regularity of the solution, but the approximation accuracy is improved when the high order bases are used. Moreover, If the modified Galerkin method, e.g., the one proposed in [14], is used, for this type problem one can have a convergence rate \( d - \beta \) for the sufficient smooth source term \( f \); when \( f = 1 \), the numerical results are listed in Table 3. We want to emphasize that including the boundary bases is very important to ensure the polynomial exactness (known as the Strang-fix condition), which is the foundation to have the desired convergence results. The numerical results in Table 4 are for the...
Table 1. Numerical results of the BVP (1.1), solved by MGM, with \( q = 0, p = \kappa_\beta = 1, \) and \( \beta = 4/5. \)

| \( d \) | \( J \) | \( \nu = 4 \) | \( \nu = 17/10 \) |
|-------|-----|--------|----------------|
| \( \nu \) | \( L_2\)-Err | \( L_2\)-Rate | \( H^\alpha\)-Rate | \( L_2\)-Err | \( L_2\)-Rate |
| 6 | 1.17589e-04 | — | — | 1.4535e-05 | — |
| 7 | 4.3968e-05 | 2.0001 | 3.1635e-04 | 1.4374 | 2.0009 |
| 8 | 1.0993e-05 | 1.9999 | 1.1829e-04 | 1.4374 | 2.0009 |

Table 2. Numerical results of the BVP (1.1), solved by MGM, with \( q = 0, p = \kappa_\beta = 1, \) and \( \beta = 4/5. \)

| \( J \) | \( \nu = 11/10 \) | \( \nu = 21/10 \) |
|-------|----------------|----------------|
| \( d \) | \( \nu = 4 \) | \( \nu = 17/10 \) | \( \nu = 19/10 \) |
| \( d \) | \( \nu = 4 \) | \( \nu = 17/10 \) | \( \nu = 19/10 \) |

Table 3. Numerical results of the BVP (1.1), solved by MGM, with \( q = 0, p = \kappa_\beta = 1, \) and \( \beta = 4/5. \)

Table 4. Numerical results of the BVP (1.1), solved by inner MGM, with \( q = 0, p = \kappa_\beta = 1, \) and \( \beta = 4/5. \)

We further consider the collocation method for the variable-coefficient version of the IBVP (1.1). The collocation points are chosen as \( \{1/2, k/2^j, 1 - 1/2^J\} \), and the approximation properties of the cubic cases that the boundary base functions are absent (one base is removed for \( d = 3 \) and two for \( d = 4 \)). At this moment the exact solution after zero extension is required to have sufficient regularity to recover the desired convergence order. The similar observations are also detected for the finite difference methods, and the ways of recovering the optimal convergence orders are presented in [5, 42].
spline collocation method are discussed in the space \( S_J := \text{span}\{\Phi_J\} \) with \( \Phi_J = \{\phi_{J,k}, k \in \triangle_J, d = 4\} \). The considered equation is

\[
    u_t - (k_1 x^{2-\beta} \partial_x^2 D_\beta^2 x u + k_2 (1-x)^{2-\beta} \partial_x^2 D_\beta^2 x u) = f \quad t \in (0, T)
\]

with the right-hand term

\[
    f(x, t) = -12 \exp(-t) \left\{ x^2(1-x)^2 + \frac{1}{6} [k_1 x^2 + k_2 (1-x)^2] \right\} - \frac{1}{\beta + 1} \left[ k_1 x^3 + k_2 (1-x)^3 \right] + \frac{2}{(\beta+1)(\beta+2)} \left[ k_1 x^4 + k_2 (1-x)^4 \right]
\]

and the initial condition \( u(x, 0) = x^2(1-x)^2 \). Then it can be checked that the analytical solution is \( u(x, t) = \exp(-t)x^2(1-x)^2 \).

The Crank-Nicolson scheme is used to get the full discretization approximation of (5.1) with the time stepsize \( 1/2^{2J} \) and \( T = 1/2 \). Table 5 shows the expected convergence order \( 2 + \beta \) of collocation method for fractional PDE, agreeing with the classical conclusion when \( \beta = 0 \). Though the superconvergence can be obtained for classical PDE by carefully averaging the derivative values gotten in collocation points, it seems that this result maynot be directly extended to fractional PDE. For convenience, let’s consider the frequently used Hermite spline collocation method. Take the collocation space

\[
    V_J := \text{span} \{ \pi_2(2^J x + 1)\}_{\Omega}, \pi_1(2^J x - k), \pi_2(2^J x - k), \pi_2(2^J x - 2^J + 1)\}_{\Omega},
\]

where \( \Omega \) denotes the restriction in \( \Omega \), \( k = 0, 1, \cdots, 2^J - 2 \), and \( \pi_1, \pi_2 \) are the cubic Hermite compactly supported functions given as

\[
    \pi_1(x) = \begin{cases} 
    -x^2(2x-3) & 0 \leq x < 1, \\
    (x-2)^2(2x-1) & 1 \leq x \leq 2,
    \end{cases}
\]

\[
    \pi_2(x) = \begin{cases} 
    x^2(x-1) & 0 \leq x < 1, \\
    (x-2)^2(x-1) & 1 \leq x \leq 2.
    \end{cases}
\]

To determine the unknown coefficients, one need total \( 2^{J+1} \) collocation points. As well known for classic PDE (\( \beta = 0 \)), when the general collocation points such as the third-quarter points of every interval \([i/2^J, (i+1)/2^J])\), \( i = 0, 1, \cdots, 2^J - 1 \) are used, the convergence order is 2. But if the Gauss nodes are used, one arrives at the superconvergence result of order 4. Unfortunately, in both case the convergence order are \( 2 + \beta \) for the fractional PDE, except the approximation accuracy maybe improved. The numerical results are presented in Table 5 where the abbr ‘Equi’ and ‘Gauss’ denote the two types of collocation points mentioned above.

**Remark 5.1.** Unlike the Galerkin method, the differential matrix of right derivative in collocation method is not the transpose of its left twin. Instead, combining the symmetry of the scaling spline bases and collocation points, it is easy to prove that

\[
    A_r = A_l(\text{end} : -1 : 1, \text{end} : -1 : 1).
\]
Table 6. Convergence performance of the cubic Hermite collocation method with $k_1 = 1$ and $k_2 = 0$.

| $J$ | $\beta = 5/10$, Equi | $\beta = 5/10$, Gauss | $\beta = 0$, Equi | $\beta = 0$, Gauss |
|-----|---------------------|---------------------|---------------------|---------------------|
|     | $L_\infty$-Err     | $L_\infty$-Rate     | $L_\infty$-Err     | $L_\infty$-Rate     |
| 5   | 8.2952e-06         | —                   | 3.5693e-05         | —                   |
| 6   | 1.4663e-06         | 2.5001              | 2.2506e-09         | 3.9946              |
| 7   | 2.5912e-07         | 2.5005              | 2.2316e-10         | 3.9968              |

Example 5.2.

Now, we focus on the wavelet multilevel schemes for solving the fractional PDEs. The presented numerical results with $d = 2$, and in this case the coefficient matrix has a full Toeplitz structure. The matrix-vector product is performed by FFT. For the other bases, the computational procedure is almost the same after a slight modification, e.g., when $d = 3$, $A_1c_j := \left( aD_x^{-\beta/2}D\Phi_{J,x}D_1^{-\beta/2}D\Phi_{J} \right)c_j$ can be decomposed into several blocks with $H$ being the Toeplitz matrix:

$$(A_1c_j)(1) = \left[ a_1, r(a_2)^T, 0 \right]c_j,$$

$$(A_1c_j)(2: \text{end} - 1) = c_j(1)a_1 + Hc_j(2: \text{end} - 1) + c_j(\text{end})a_2,$$

$$(A_1c_j)(\text{end}) = \left[ a_2, r(a_1)^T, a_1 \right]c_j.$$

For the BVP, we first reveal that the multilevel preconditioning brings a uniform matrix condition number and an improved spectral distribution. Considering the BVP (1.1) with $\kappa_\beta = 1$, $p = 1$ and $\kappa_\beta = 1, p = 1/2$, the condition numbers for different $\beta$ are presented in Table 7; one can see that without preconditioning, the condition number of the stiffness matrix behaves like $O(2^{(2-\beta)})$, which means the conditional number increases fast with the refinement especially when $\beta$ is small. After preconditioning, the uniformly bounded condition numbers with different wavelet preconditioners are obtained; see Table 8, where ‘inte-’, ‘Semi-’, and ‘Bior- ($d$)’ denote the interpolation wavelet, semiorthogonal wavelet, and biorthogonal wavelet $^2\psi$, respectively, having been introduced in Section 2. Note that when performing the decomposition by semiorthogonal and biorthogonal wavelets, the interpolation wavelet has been used for $S_1$ and $S_2$. We also display the matrix eigenvalue distribution for $\beta = 1/5$ in Figures 1 and 2; they show the preconditioning benefits of a more concentrated eigenvalue distribution.

To explore the effectiveness of this preconditioned system, we numerically solve BVP (1.1) with

$$f = \frac{1}{\Gamma(1 + \beta)} (-2x^\beta + \beta x^{\beta-1}),$$

and

$$f = \frac{1}{2\Gamma(1 + \beta)} (-2x^\beta + \beta x^{\beta-1} - 2(1 - x)^\beta + \beta(1 - x)^{\beta-1}),$$
Table 8. Preconditioned condition numbers of the BVP (1.1) with $q = 0$, $\kappa_\beta = 1$, $d = 2$, and $J_0 = 0$.

| $p$ | $J$ | $\beta = 1/2$ | $\beta = 1/5$ |
|-----|-----|--------------|--------------|
|     |     | Inte.- Semi- Bior.- ($d = 2$) | Inte.- Semi- Bior.- ($d = 2$) |
| 8   | 3.0970 | 5.8363 | 13.3957 | 1.5953 | 10.2897 | 12.2315 |
| 9   | 3.2286 | 6.1158 | 14.4784 | 1.6269 | 10.5561 | 12.9767 |
| 10  | 3.3457 | 6.3622 | 15.4103 | 1.6540 | 10.7779 | 13.5788 |
| 8   | 3.2614 | 8.0344 | 12.7702 | 1.5935 | 11.1094 | 12.2830 |
| 9   | 3.4745 | 8.1648 | 13.6624 | 1.6296 | 11.4094 | 13.0063 |
| 10  | 3.6686 | 8.2634 | 14.4026 | 1.6608 | 11.6511 | 13.5854 |

Figure 1. Eigenvalue distribution of the BVP matrix with $p = 1$ (first two) and $p = 1/2$ (last two).

Figure 2. Eigenvalue distribution of the preconditioned BVP systems with the interpolation wavelet (first line) and the semiorthogonal wavelet (second line), respectively (the first two columns are for $p = 1$ and the last two columns for $p = 1/2$).

for $p = 1$ and $p = 1/2$, respectively. We use GMRES and Bi-CGSTAB to solve the algebraic system before and after preconditioning, and the numerical results are given in Tables 9 and 10, respectively. The comparisons for the two methods are made almost with the same $L_2$ approximation error, not listed in the tables. The stopping criterion for solving the linear systems is

$$\frac{\| r(k) \|_{L_2}}{\| r(0) \|_{L_2}} \leq 1e^{-8},$$

with $r(k)$ being the residual vector of linear systems after $k$ iterations. It should be noted that the GMRES method for $p = 1$ without preconditioning stops before reaching this criterion. In fact, by the two-dimension
FWT and the properties of tensor product, these proposed preconditioner can also easily apply to two dimension and it also works well for the algebraic systems generated by the finite difference methods, e.g., [20][30].

Table 9. Numerical results of the BVP (1.1), solved by GMRES and Bi-CGSTAB, with \( q = 0, \kappa = 1, \beta = 1/5, \) and \( d = 2. \)

| \( J \) | \( p = 1, \text{GMRES} \) | \( p = 1/2, \text{GMRES} \) | \( p = 1, \text{Bi-CGSTAB} \) | \( p = 1/2, \text{Bi-CGSTAB} \) |
|---|---|---|---|---|
| | \( \text{Iter} \) | \( \text{CPU(s)} \) | \( \text{Iter} \) | \( \text{CPU(s)} \) | \( \text{Iter} \) | \( \text{CPU(s)} \) | \( \text{Iter} \) | \( \text{CPU(s)} \) |
| 8 | 2.550e+02 | 0.0243 | 1.180e+02 | 0.0915 | 2.635e+02 | 0.0672 | 1.170e+02 | 0.0303 |
| 9 | 5.110e+02 | 1.5217 | 2.200e+02 | 0.3230 | 5.355e+02 | 0.2408 | 2.095e+02 | 0.1012 |
| 10 | 1.0230e+03 | 7.4783 | 4.120e+02 | 1.3219 | 1.1665e+03 | 0.6731 | 3.9150e+02 | 0.2280 |

Table 10. Numerical results of the BVP (1.1), solved by the preconditioned GMRES and Bi-CGSTAB, with \( q = 0, \kappa = 1, \beta = 1/5, \) and \( J_0 = 0. \)

| \( p \) | \( J \) | \( \text{GMRES, Inte.} \) | \( \text{GMRES, Semi.} \) | \( \text{Bi-CGSTAB, Inte.} \) | \( \text{Bi-CGSTAB, Semi.} \) |
|---|---|---|---|---|---|
| | | \( \text{Iter} \) | \( \text{CPU(s)} \) | \( \text{Iter} \) | \( \text{CPU(s)} \) | \( \text{Iter} \) | \( \text{CPU(s)} \) | \( \text{Iter} \) | \( \text{CPU(s)} \) |
| \( p = 1 \) | 8 | 13.0 | 0.0094 | 27.0 | 0.0203 | 2.3 | 0.0147 | 5.5 | 0.0261 |
| | 9 | 13.0 | 0.0115 | 28.0 | 0.0258 | 9.5 | 0.0133 | 20.0 | 0.0272 |
| | 10 | 13.0 | 0.0209 | 28.0 | 0.0452 | 9.5 | 0.0149 | 22.0 | 0.0376 |
| \( p = 1/2 \) | 8 | 9.0 | 0.0075 | 25.0 | 0.0227 | 6.5 | 0.0064 | 18.0 | 0.0201 |
| | 9 | 9.0 | 0.0084 | 26.0 | 0.0248 | 7.5 | 0.0087 | 20.0 | 0.0267 |
| | 10 | 9.0 | 0.0163 | 26.0 | 0.0370 | 8.0 | 0.0126 | 21.0 | 0.0363 |

Secondly, we use the MMG to solve the fractional IBVPs (1.1) with the exact solution \( u(x, t) = \exp(-t)(x^n - x^2), q = \kappa = 1, \) and the suitable source term and initial condition. It can be noted that because of the constant diagonal elements of the stiffness matrix, the Richardson and the Jacobi iterations used in MMG are actually equivalent. For \( m_1(j) = m_2(j) = 1, J_0 = 3, \) the numerical results of CN-MMG are given in Tables 11 and 12, where ‘Iter’ denotes the average iteration times and ‘CPU(s)’ the computation time also including the time of the calculation of coefficient matrix \( B^{n+1}_j, J = J_0, \cdots, J \) and the right term. The initial iteration vector at \( t_{n+1} \) is chosen as the approximation at \( t_n, \) and the stopping criterion is

\[
\left\| c_{J}^{n+1,1} - c_{J}^{n+1,1-1} \right\|_{\infty} \leq 2^{-J/2} \times (1e - 9),
\]

where \( c_{J}^{n+1,1} \) is the approximation vector after the \( l \) iteration. Of course, the FFT and the FWT are used to accelerate the process. ‘Gauss(s)’ denotes the computation time of the Gaussian elimination method; for fair comparison, the FFT is also used to get the matrix-vector product appeared in the right-hand term at the time \( t_{n+1}. \)

For \( p = 1/2, \) the coefficient matrix is symmetric. And if we choose \( \omega < 1/\lambda_{\text{max}}, \lambda_{\text{max}} = \sigma \left( \text{diag}(B_{J}^{n+1}B_{J}^{n+1}) \right) \), then the MMG is convergent and the average iteration number is slightly affected by the choice of \( \omega. \) It also seems that the restriction to \( \omega \) can be relaxed to some extent in real computation. When \( p \neq 1/2, \) even though there are no strict theoretical prediction, the numerical results show when \( \omega \geq \lambda_{\text{max}}, \) the iteration may be divergent; see Table 12. Here, we get the value of \( \lambda_{\text{max}} \) by the Matlab function \( \text{eigs}; \) it can also be estimated by the Gershgorin Theorem or the Power method.

Example 5.3.

In this example, we focus on the previously proposed ad-hoc wavelet adaptive algorithms for the fractional PDEs. The BVP is solved by the biorthogonal wavelet bases produced by \( 3,3\psi (d = 3, \tilde{d} = 3), \) and the IBVP
by the semi-interpolation wavelet bases. We first consider the BVP (1.1), the regularity of its exact solution, \( u(x) = (1 - x)^{11/10} - (1 - x) \), is weak at the area close to the right boundary; and the parameters \( \kappa_\beta = 1, p = 0 \), and the source term

\[
f(x) = -\frac{\Gamma(21/10)(1 - x)^{\beta - 9/10}}{\Gamma(\beta + 1/10)} + \frac{(1 - x)^{\beta - 1}}{\Gamma(\beta)}.
\]

In the algorithm, we take \( J_0 = 3, \epsilon(j) = 1e - 5 \). For every iteration step, the finally extended irregular indexes are obtained by firstly adding the children of all the significant indexes and then including two neighbors, i.e., the right and left neighbors, of each index of the extended irregular indexes. When \( \beta = 1/2 \), the sets of wavelet indices that corresponding to the adaptively chosen wavelets and the corresponding error \( u - \tilde{u}_{J_0 + m} \) are presented in Figures 3, where the blue bar denotes that we have used all the scaling bases in the coarsest level \( J_0 \). One can see that the algorithm in fact automatically recognizes the whereabouts of the boundary layer of the solution \( u \), and adds wavelets locally to there. It also reveals that the newly added computational costs are spented in the most needed place, and the large peaks of the errors are successively reduced. Moreover, for different \( \beta \), from the decreasing of the \( L_2 \) approximation error of the adaptive and uniform Galerkin schemes with the increasing of the freedom \( N \) (the loglog coordinate) in Figure 4, one can see that the adaptive MGM is remarkably superior to the uniform MGM.

Secondly, we consider the IBVP (1.1) with \( \kappa_\beta = p = 1, \beta = 5/10 \), the initial condition \( u(x, 0) = x^4(1 - x) \), and the source term

\[
f(x, t) = \exp(3t) x^{20t + 4}(1 - x)(3 + 20 \ln x) + \exp(3t) \frac{\Gamma(20t + 5)}{\Gamma(20t + 3 + \beta)} x^{20t + 2 + \beta} \left( \frac{20t + 5}{20t + 3 + \beta} x - 1 \right).
\]

Its exact solution is \( u = \exp(3t)x^{20t + 4}(1 - x) \), which has a strong gradient at somewhere as shown in Figure 4 (left). In the computation, based on Algorithm 3 the semi-interpolation adaptive wavelet collocation method is used; and the time step \( \Delta t = 1/2J_{\max} \), \( J_0 = 3, J_{\max} = 10, \epsilon(j) = 1e - 5 \), and \( T = 1 \). In this adaptive algorithm, for every time step, the index extension techniques being used are the same as the ones for the BVP,
and the chosen collocation points are just the ones corresponding to the reserved wavelet bases. For $t = 1$, the adaptive solution and the distribution of semi-interpolation wavelets are displayed in Figure 5. Further seeing the global picture, Figure 6 (right), one can easily notice that the high level wavelets and collocation points mainly concentrate on the area with steep gradient, being exactly as what we have desired.

6. Conclusion and Discussion

This paper focuses on digging out the potential benefits, providing the techniques, and performing the theoretical analysis and extensive numerical experiments in solving the fractional PDEs by wavelet numerical
methods. The multiscale (wavelet) bases show their strong advantages in treating the fractional operators which essentially arise from the multiscale problem. Even the scaling bases also display their powerfulness in saving computational cost when generating stiffness matrix, i.e., by using the scaling bases, the stiffness matrix has the Toeplitz structure. The way of generating effective preconditioner is presented for time-independent problem and multigrid scheme for time dependent problem is detailedly discussed. We numerically show that the heuristic wavelet adaptive scheme works very well for fractional PDEs; in particular, it is still easy to get the local regularity indicator even for the fractional (nonlocal) problem; and the algorithm descriptions are provided.
After finishing this work, one of the directions of our further research appears, i.e., applying the wavelet compression property to fractional operator. A key difference between the fractional and classical operators is that the former is non-local, and then both the matrixes generated by the scaling and the multiscale bases are no longer sparse. Fortunately, the wavelet compression not only allows one to obtain a sparse representation of functions, but it seems also effective for the fractional operators. Considering the discretization of the operator:

$$A u = -D \left( \frac{2}{3} D_x^{-\beta} + \frac{1}{3} D_i^{-\beta} \right) Du$$

in the approximation space $S_J$ with $J = 10$, we first compute the matrix $A_J$ or $\hat{A}_J$ (here the multiscale wavelet bases also have been normalized with $D$, proposed in Section 3). Then we get the compressed matrix by setting all entries of $A_J$ or $\hat{A}_J$ with modulus less than $\epsilon = 10^{-4} \times 2^{-J}$ to zeros. The comparison results are displayed in Table 13 and Figure 7, where $(\cdot \%)$ denotes the percentage of the non-zero entries of the compressed matrix.

It can be seen that many entries in $\hat{A}_J$ are so small that they can be omitted to retrieve the famous finger structure, whereas essentially all entries in $A_J$ are significant. In the future, we will investigate the effective ways of using wavelet compression to get the paralleled sparse approximate inverse (SPAI) preconditioner and to perform the low-cost multiscale matrix-vector product.

![Figure 7. Sparsity patterns of the compression matrix $\hat{A}_J$ by using the semiorthogonal with $\beta = 5/10$ and $d = 2$ (left) and the biorthogonal wavelet $2^4\psi$ with $d = 2$ and $\tilde{d} = 4$ bases (right).](image)

| $\psi$ | $\beta = 8/10$ | $\beta = 5/10$ | $\beta = 2/10$ | Note |
|-------|---------------|---------------|---------------|------|
| $A_J$ | $A_J$         | $A_J$         | $A_J$         |      |
| $d = 2, \text{Inte}$ | 99.80% | 99.97% | 99.80% | 80.38 | 99.80% | 48.04% |
| $d = 2, \text{Semi}$ | 99.80% | 6.66% | 99.80% | 6.14% | 99.80% | 5.31% |
| $d = 2, \tilde{d} = 4$ | 99.80% | 8.15% | 99.80% | 7.89% | 99.80% | 7.15% |
| $d = 3, \tilde{d} = 3$ | 99.80% | 10.18% | 100% | 9.52% | 100% | 8.30% |

**Table 13. Compression capacity of the different bases for operator $A$.**

**Appendix**

Here we present the techniques for computing the left fractional derivative of the base function $\Phi_J = \{\phi_{J,k}, k \in \Delta_J, d = 4\}$. Besides $\phi(x)$ and $\phi_b(x)$ given by (2.16) and (2.17), define

$$\phi_a(x) = 3x - \frac{9}{2} x_+^2 + \frac{7}{4} x_+^3 - 2(x - 1)_+^3 + \frac{1}{4} (x - 2)_+^3.$$
Then $\Phi J = 2^{J/2} \{ \phi_a (2^J x), \phi_b (2^J x), \phi (2^J x - k)^{J/2 - 4} \}_{k=0}^J, \phi_b (2^J (1 - x)), \phi_a (2^J (1 - x)) \}$ is a Riesz bases of $S_J$. For $x_0 \geq 0$, it is easy to check that

$$aD_{x}^{1-\beta} (H(x-x_0)v(x)) = H(x-x_0)x_0 D_{x}^{1-\beta} v(x),$$

where $H(x)$ denotes the Heaviside function. By the well-known formulae

$$aD_{x}^{1-\beta} (x-a)^{\nu} = \frac{\Gamma(\nu + 1)(x-a)^{\nu+\beta-1}}{\Gamma(\nu + \beta)}, \quad \nu \in \mathbb{N},$$

$$(b - ax)_+^k = (b - ax)_+^k + (-1)^{k-1} (ax - b)_+^k, \quad k \in \mathbb{N}^+,$$

for $b/a \geq 0$, $k \in \mathbb{N}^+$, there exist

$$aD_{x}^{1-\beta} (ax - b)_+^k = a^{2-\beta} \frac{\Gamma(k+1)}{\Gamma(k+\beta-1)} (ax - b)_+^{k+\beta-1},$$

$$aD_{x}^{1-\beta} (b - ax)_+^k = (-1)^{k-1} aD_{x}^{1-\beta} (ax - b)_+^k + \sum_{m=0}^{k} \binom{k}{m} a^{k-m} b^m (m!)^(-m) \frac{m!}{\Gamma(m+\beta)} b_+^{m+\beta-1},$$

Define

$$M_1(x) := aD_{x}^{1-\beta} \phi_a (x) = \frac{3}{\Gamma(\beta + 2)} \left( (\beta + 1)x_+^\beta - 3x_+^{\beta+1} + \frac{7}{2(\beta + 2)} x_+^{\beta+2} \right)$$

$$+ \frac{3}{2\Gamma(\beta + 3)} \left( -8(x-1)_+^{\beta+2} + (x-2)_+^{\beta+2} \right),$$

$$M_2(x) := aD_{x}^{1-\beta} \phi_b (x) = \frac{1}{\Gamma(\beta + 2)} \left( 3x_+^{\beta+1} - \frac{11}{2(\beta + 2)} x_+^{\beta+2} \right)$$

$$+ \frac{1}{2\Gamma(\beta + 3)} \left( 18(x-1)_+^{\beta+2} - 9(x-2)_+^{\beta+2} + 2(x-3)_+^{\beta+2} \right),$$

$$M_3(x) := aD_{x}^{1-\beta} \phi (x) = \frac{1}{\Gamma(\beta + 3)} \sum_{i=0}^{4} \binom{4}{i} (-1)^{i} (x-i)_+^{\beta+2},$$

$$M_4(x,l) := aD_{x}^{1-\beta} \phi (l - x) = \frac{-1}{\Gamma(\beta + 2)} \left( 3(x-l)_+^{\beta+1} + \frac{11}{2(\beta + 2)} (x-l)_+^{\beta+2} \right)$$

$$+ \frac{1}{2\Gamma(\beta + 3)} \left( 18(x-l+1)_+^{\beta+2} - 9(x-l+2)_+^{\beta+2} + 2(x-l+3)_+^{\beta+2} \right).$$

Then we have

$$aD_{x}^{1-\beta} \left( 2^{J/2} \phi_a (2^J x) \right) = 2^{J(3/2-\beta)} M_i (2^J x), \quad (i, a_i) = (1, a) \text{ or } (2, b),$$

$$aD_{x}^{1-\beta} \left( 2^{J/2} \phi (2^J x - k) \right) = 2^{J(3/2-\beta)} M_3 (2^J x - k),$$

$$aD_{x}^{1-\beta} \left( 2^{J/2} \phi_b (2^J (1 - x)) \right) = 2^{J(3/2-\beta)} M_4 (2^J x, 2^J).$$

The similar formulae can also be derived for $2^{J/2} \phi_a (2^J (x - 1))$.

**References**

[1] M. Benzi, Preconditioning techniques for larger linear systems: a survey. *J. Comput. Phys.* 182 (2002) 418-477.
Z.J. ZHANG AND W.H. DENG

[2] W. Cai and J. Wang, An adaptive multiresolution collocation methods for initial boundary value problems of nonlinear PDEs. *SIAM J. Numer. Anal.* 33 (1996) 92-126.

[3] F. F. Campos and J. S. Rollett, Analysis of preconditioners for conjugate gradients through distribution of eigenvalues. *Int. J. Comput. Math.* 58 (1995) 135-158.

[4] Z. Chen and H. Wu, *Selected Topics in Finite Element Methods*. Science Press, Beijing (2010).

[5] M. H. Chen and W. H. Deng, High order algorithms for the fractional substantial diffusion equation with truncated Lévy flights. *SIAM J. Sci. Comput.* 37 (2015) A890-A917.

[6] M. H. Chen, Y. T. Wang, X. Cheng and W. H. Deng, Second-order LOD multigrid method for multidimensional Riesz fractional diffusion equation. *BIT* 54 (2014) 623-647.

[7] C. K. Chui and E. Quak, Wavelets on a bounded interval, in *Numerical methods in approximation theory*, D. Braess and L. Schumaker, eds., Birkäuser, Basel (1992) 53-75.

[8] A. Cohen, Wavelet methods in numerical analysis, in Handbook of *Numerical Analysis*, P. Ciarlet and J. Lions, eds., Elsevier North-Holland (2000) 417-711.

[9] W. H. Deng, Finite element method for the space and time fractional Fokker-Planck equation. *SIAM J. Numer. Anal.* 47 (2009) 204-226.

[10] V. J. Ervin and J. P. Roop, Variational formulation for the stationary fractional advection dispersion equation. *Numer. Methods Partial Differential Equations* 22 (2005) 558-576.

[11] V. J. Ervin, N. Heuer, and J. P. Roop, Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation. *SIAM J. Numer. Anal.* 45 (2007) 572-591.

[12] J. Huang, N. Nie, and Y. Tang, A second order finite difference-spectral method for space fractional diffusion equation. *Sci. China Math.* 57 (2014) 1303-1317.

[13] H. Jafari, S. A. Yousefi, M. A. Firoozjaee, S. Momani, and C. M. Khalique, Application of Legendre wavelets for solving fractional differential equations. *Comput. Math. Appl.* 62 (2011) 1038-1045.

[14] B. T. Jin, R. Lazarov, X. L. Lu, and Z. Zhou, A simple finite element method for boundary value problems with a Riemann-Liouville derivative. *J. Comput. Appl. Math.* 293 (2016) 94-111.

[15] S. Lei and H. Sun, A circulant preconditioner for fractional diffusion equations. *J. Comput. Phys.* 242 (2013) 715-725.

[16] X. Li and C. Xu, A space-time spectral method for the time fractional diffusion equation. *SIAM J. Numer. Anal.* 47 (2009) 2108-2131.

[17] X. Li and C. Xu, Existence and uniqueness of the weak solution of the space-time fractional diffusion equation and a spectral method approximation. *Commun. Comput. Phys.* 8 (2010) 1016-1051.

[18] F. Lin, S. Yang, and X. Jin, Preconditioned iterative methods for fractional diffusion equation. *J. Comput. Phys.* 256 (2014) 109-117.

[19] F. Mainardi, *Fractional Calculus and Waves in Linear Viscoelasticity*. World Scientific (2010).

[20] M. M. Meerschaert and C. Tadjeran, Finite difference approximations for fractional advection-dispersion flow equations. *J. Comput. Appl. Math.* 172 (2004) 65-77.

[21] M. M. Meerschaert and A. Sikorskii, *Stochastic Models for Fractional Calculus*. Walter de Gruyter (2011).

[22] R. Metzler and J. Klafter, The random walk’s guide to anomalous diffusion: A fractional dynamics approach. *Phys. Rep.* 339 (2000) 1-77.

[23] T. Moroney and Q. Yang, A banded preconditioner for the two-sided, nonlinear space-fractional diffusion equation. *Comput. Math. Appl.* 66 (2013) 659-667.

[24] H. Pang and H. Sun, Multigrid method for fractional diffusion equations. *J. Comput. Phys.* 231 (2012) 693-703.

[25] S. Daleke, M. Fornasier, M. Pröbsch, T. Rauch, M. Werner, Nonlinear and adaptive frame approximation schemes for elliptic PDEs: Theory and numerical experiments. *Numer. Methods Partial Differential Equations* 25 (2009) 1261-1510.

[26] M. Pröbsch, New stable biorthogonal spline-wavelets on the interval. *Results Math.* 57 (2010) 121-162.

[27] M. Raberto, E. Scalas, and F. Mainardi, Waiting-times and returns in high-frequency financial data: An empirical study. *Phys. A* 314 (2002) 749-755.

[28] A. Saadatmandi, M. Dehghan, and Mo. R. Azizic, The Sinc-Legendre collocation method for a class of fractional convection-diffusion equations with variable coefficients. *Commun. Nonlinear Sci. Numer. Simul.* 17 (2012) 4125-4136.

[29] U. Saeed and M. Rehman, Haar wavelet-quasilinearization technique for fractional nonlinear differential equations. *Appl. Math. Comput.* 220 (2013) 630-648.

[30] W. Y. Tian, H. Zhou, and W. H. Deng, A class of second order difference approximation for solving space fractional diffusion equations. *Math. Comp.* 84 (2015) 1703-1727.

[31] K. Urban, *Wavelet Methods for Elliptic Partial Differential Equations*. Oxford University Press, Oxford, New York (2009).

[32] H. Wang, D. Yang, and S.F. Zhu, Inhomogeneous dirichlet boundary-value problems of space-fractional diffusion equations and their finite element approximations. *SIAM J. Numer. Anal.* 52 (2014) 1292-1310.

[33] L. F. Wang, Y. P. Ma, and Z. J. Meng, Harr wavelet method for solving fractional partial differential equations numerically. *Appl. Math. Comput.* 227 (2014) 66-76.

[34] H. Wang, K. Wang, and T. Sircar, A direct $O(N \log^2 N)$ finite difference method for fractional diffusion equations. *J. Comput. Phys.* 239 (2010) 8095-8104.
[35] J. Xu, An introduction to multilevel methods, in: M. Ainsworth, J. Levesley, W. A. Light and M. Marletta (Eds.), Wavelets, Multilevel Methods and Elliptic PDEs, Leicester, 1996, Oxford University Press, New York (1997) 213-302.

[36] Q. W. Xu and J. S. Hesthaven, Discontinuous Galerkin method for fractional convection-diffusion equations. SIAM J. Numer. Anal. 52 (2014) 405-423.

[37] Q. Yang, I. Turner, F. Liu, and M. lič, Novel numerical methods for solving the time-space fractional diffusion equation in 2D, SIAM J. Sci. Comput. 33 (2011) 1159-1180.

[38] Q. Yang, I. Turner, and F. Liu, Numerical methods for fractional partial differential equations with Riesz space fractional derivatives. Appl. Math. Model. 34 (2010) 200-218.

[39] M. Zaslavsky, Chaos, fractional kinetics, and anomalous transport. Phys. Rep. 371 (2002) 461-580.

[40] M. Zayernouri and G. E. Karniadakis, Fractional Sturm-Liouville eigen-problems: theory and numerical approximation. J. Comput. Phys. 252 (2013) 495-517.

[41] Y. N. Zhang, Z. Z. Sun, and H. L. Liao, Finite difference methods for the time fractional diffusion equation on non-uniform meshes. J. Comput. Phys. 265 (2014) 195-210.

[42] L. J. Zhao and W. H. Deng, A series of high order quasi-compact schemes for space fractional diffusion equations based on the superconvergent approximations for fractional derivatives. Numer. Methods Partial Differential Equations 31 (2015) 1345-1381.
t = 0.099999s, Error = 2.1824e−05
t = 0.4s, Error = 1.0396e-05
$t = 0.7s, \text{ Error} = 3.202e{-05}$
t=0.099999s, N=64
$t=0.4s, N=73$
t=0.7s, N=81

Level

Location
2nd iteration, N=44
4th iteration, \( N=64 \)
5th iteration,  N=72
7th iteration, \( N=80 \)
8th iteration, $N=88$
$x \times 10^{-4}$ Difference to exact solution
The graph shows the difference to the exact solution for $x \times 10^{-5}$.
$x \times 10^{-6}$ Difference to exact solution
Difference to exact solution

$2 \times 10^{-6}$
