SPECTRAL, TENSOR AND DOMAIN DECOMPOSITION METHODS FOR FRACTIONAL PDES

(DEDICATED TO OUR FRIEND FRANCISCO-JAVIER SAYAS)

TIANYI SHI, HARBIR ANTIL, AND DREW P. KOURI

Abstract. Fractional PDEs have recently found several geophysics and imaging science applications due to their nonlocal nature and their flexibility in capturing sharp transitions across interfaces. However, this nonlocality makes it challenging to design efficient solvers for such problems. In this paper, we introduce a spectral method based on an ultraspherical polynomial discretization of the Caffarelli-Silvestere extension to solve such PDEs on rectangular and disk domains. We solve the discretized problem using tensor equation solvers and thus can solve higher-dimensional PDEs. In addition, we introduce both serial and parallel domain decomposition solvers. We demonstrate the numerical performance of our methods on a 3D fractional elliptic PDE on a cube as well as an application to optimization problems with fractional PDE constraints.

1. Introduction

Fractional partial differential equations (PDEs) have recently received a tremendous amount of attention, which can be attributed to the flexibility of fractional operators in capturing long-range effects, due to their nonlocal nature. In addition, they have fewer regularity requirements than their classical counterparts. In particular, the fractional Laplacian has been successfully used as a regularizer in imaging science in place of the total variation regularization [3, 4]. Moreover, the fractional Helmholtz equation was derived in [32], using first principle arguments combined with a constitutive relation, to model geophysical electromagnetism. Other applications include: Quasi-geostrophic flow [13], phase field models [2, 3], porous media [14], and quantum mechanics [20]. Motivated by these applications, we introduce a new approach to solve the fractional PDE

$$\begin{aligned} (-\Delta)^s u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1.1}$$

where $\Omega \subset \mathbb{R}^n$ is a bounded open domain with boundary $\partial\Omega$, and $s \in (0,1)$ is the fractional exponent. The operator $(-\Delta)^s$ denotes the spectral fractional Laplacian, whose rigorous definition will be provided in Section 2. The nonlocality of the fractional Laplacian makes it challenging to realize in practice [11, 29]. However, several approaches exist. For example, the authors in [28] use a spectral discretization in space and discuss computing the spectrum of the Laplacian to realize the fractional Laplacian. Unfornately, computing the spectrum of an operator in general domains can

2010 *Mathematics Subject Classification.* 49J20, 49K20, 35S15, 65R20, 65N30 .

Key words and phrases. Ultraspherical polynomial; fractional PDEs; tensor methods; domain decomposition; 3D problems; optimal control .

This work is partially supported by the Air Force Office of Scientific Research under award numbers: FA9550-19-1-0036 and F4FGA09135G001 and NSF grants DMS-2110263 and DMS-1913004.

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under grant DE-NA-0003525. This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

be expensive. Whereas, the authors in [10] discuss an alternative approach based on the so-called Kato formula. Here, spatial discretization is carried out using the finite element method. For the standard problem such as (1.1), the Kato formula approach is desirable, but is delicate, especially when lower-order terms are present [32].

Another popular approach is based on the so-called Caffarelli–Silvestre extension. This extension was originally introduced by Caffarelli and Silvestre in [11] for the case of unbounded Ω and was extended to bounded domains by Stinga and Torrea in [29]. The main idea is to rewrite the nonlocal problem (1.1) as a local problem in one additional spatial dimension (i.e., the domain of the resulting local problem is $\Omega \times (0, \infty)$ with dimension $n + 1$). This idea was exploited in [24], where the authors introduced a finite element method (FEM) to solve the extended problem on a bounded domain $\Omega \times (0, R)$ with $R < \infty$. The choice of R in [24] is motivated by the fact that the solution in the extended direction decays exponentially. Recently, the authors in [7, 23] introduced *hp*-FEM that reduces the complexity of the $(n + 1)$-dimensional problem to *log-linear* with respect to the number of degrees of freedom in Ω. We also mention that an interesting hybrid FEM-spectral method based on approximation of the Laplacian eigenvalues was considered in [1].

An additional issue with the Caffarelli–Silvestre extension is that the solution in the extended dimension suffers from low regularity when $s \neq 1/2$. As a result, spectral methods do not provide the typical exponential convergence. This has been thoroughly investigated in [12], where the authors use generalized Laguerre functions as a basis in the extended dimension to solve a weak formulation of (1.1). To overcome this regularity issue and improve the convergence rate, the authors apply an enrichment technique for spectral methods. The convergence rate improves, as the enrichment terms are included, but the matrices for the discretized problem are generally dense. In addition, the system becomes ill-conditioned, and if there are too many singular terms included, the convergence rate can deteriorate significantly due to ill-conditioning. Finally, we mention extensive work on numerical methods for fractional PDEs with different fractional derivatives such as Riemann Liouville. See, for example [34], and the review article [21].

The goal of this paper is to introduce a new efficient and robust spectral method to solve fractional PDEs based on the Caffarelli–Silvestre extension. For simplicity, we present our method for $n = 2$ on two types of domains: disks and rectangles. By utilizing a tensor equation solver, our method is easily generalized to even higher dimensions. For example, as shown in our numerical examples in Section 5, our method directly applies to $n = 3$ on a cube. In addition, with spherical polar coordinates, our method can solve the problem on a ball. Our spectral method utilizes ultraspherical, Chebyshev, and Fourier polynomial discretizations. The resulting discrete systems are tensor equations, which we solve with a direct solver based on the real Schur decomposition. For the disk domain, we rewrite the extended problem in polar coordinates and use the Double Fourier Sphere (DFS) method [33] to overcome the singular behavior when the extended dimension $z = 0$. Using DFS, $z = 0$ is no longer treated as a boundary. Afterward, we consider two cases: (i) $z^{1/s}$ can be approximated well by a polynomial; (ii) $z^{1/s}$ cannot be approximated well by a polynomial. The former case occurs, for example, when $1/s$ is an integer. In the latter case, we employ piecewise polynomials over subdomains. In this manuscript, we focus on fewer subdomains with higher polynomial degrees for simpler systems to solve. One can alternatively choose to use more subdomains with lower polynomial degrees [6]. For our numerical results, we use `chebfun` [16] to automatically generate the subdomains. We then solve the extended PDE directly using ultraspherical, Fourier, and Chebyshev spectral discretizations for the radius, angle, and extended direction. For rectangular domains, we spectrally discretize the extended PDE with ultraspherical polynomials in the space domain and Chebyshev polynomials for the extended direction. In addition, when $z^{1/s}$ is approximated by a piecewise polynomial, we design a domain decomposition solver that handles

each piece independently. As a result, this solver is easily parallelized. The convergence analysis for ultraspherical spectral methods can be found in [26]. In practice, we use the polynomial coefficients of the solution to determine if the solver has converged. Specifically, when a coefficient falls below a given threshold (e.g., we choose 10^{-10} in this paper), we terminate the spectral method and use the result as the discrete approximation of the solution to the fractional PDE.

In all numerical examples and for all values of s, we observe exponential convergence with just a few degrees of freedom. In this sense, the potential benefits of our method are clear. Additionally, the proposed approach has several other advantages over the existing spectral methods. We can combine our spectral methods on rectangles with the ultraspherical spectral element method (UltraSEM) [17] to develop solvers on polygonal domains with unstructured quadrilateral or triangular meshes. To be specific, we can partition the general domain Ω into quadrilateral subdomains, convert them into rectangles with a change of variables, and apply our solver. As a result, the total number of degrees of freedom is the sum of those used on each subdomain. Finally, when $n = 2$, we emphasize that our method can be generalized to solve fractional PDEs of the form:

$$\begin{aligned} \mathcal{L}^s u &= f \quad \text{in } \Omega \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \tag{1.2}$$

where the operator $\mathcal{L}$ is the general elliptic operator $\mathcal{L}u = -\nabla \cdot (A\nabla u) + cu$. Here, $A \in \mathbb{R}^{2\times 2}$ is matrix function that is symmetric and positive definite and $0 \leq c \in \mathbb{R}$ is a function. This is feasible due to [17]. However, [17] only considers the case with $n = 2$, motivating our restriction to $n = 2$ in (1.2).

The remainder of the paper is organized as follows. In Section 2, we review preliminary results on fractional PDEs as well as the polynomial and tensor notation that we use in the subsequent sections. In Section 3, we introduce the direct solver for the disk domain. In Section 4, we solve the extended PDE on rectangles both directly and with a parallel domain-decomposition solver. Finally, in Section 5, we demonstrate our method on two applications: solving the fractional elliptic PDE on the cube, and solving a fractional PDE-constrained optimal control problem.

2. Notation and Preliminary Results

Let $\Omega \subset \mathbb{R}^n$ be a bounded open set with Lipschitz boundary $\partial\Omega$. Let $-\Delta$ be the $L^2(\Omega)$ realization of the standard Laplace operator with zero Dirichlet boundary conditions. It then follows that $-\Delta$ has compact resolvent and its eigenvalues can be arranged as $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_k \leq \cdots$ with $\lim_{k\to\infty} \lambda_k = \infty$. Let us denote by $\varphi_k \in H_0^1(\Omega)$ the eigenfunctions corresponding to λ_k. These eigenfunctions form an orthonormal basis of $L^2(\Omega)$.

For $s \geq 0$, we let

$$\mathbb{H}^s(\Omega) := \left\{ u = \sum_{k=1}^{\infty} u_k \varphi_k \;\middle|\; \|u\|^2_{\mathbb{H}^s(\Omega)} := \sum_{k=1}^{\infty} \lambda_k^s u_k^2 < \infty, \;\text{ where }\; u_k = \int_\Omega u\varphi_k dx \right\}.$$

For a relation between $\mathbb{H}^s(\Omega)$ and the classical fractional-order Sobolev space $H^s(\Omega)$, we refer to [5, 15], and the references therein. We shall denote the dual of $\mathbb{H}^s(\Omega)$ by $\mathbb{H}^{-s}(\Omega)$. Specifically in this manuscript, we focus on $0 < s < 1$. Now, to define the fractional Laplacian: $(-\Delta)^s$ is the mapping

$$(-\Delta)^s : \mathbb{H}^s(\Omega) \to \mathbb{H}^{-s}(\Omega)$$

defined for all $u \in \mathbb{H}^s(\Omega)$ by

$$(-\Delta)^s u = \sum_{k=1}^{\infty} \lambda_k^s u_k \varphi_k.$$

To solve (1.1), we use the Caffarelli–Silvestre extension [11, 29]. This requires introducing an extension variable ζ, and the change of variable $z = \left(\frac{\zeta}{2s}\right)^{2s}$. Then, the resulting problem aims to find $U : \Omega \times [0, \infty) \to \mathbb{R}$ that satisfies the following equation

$$\begin{aligned} \Delta_x U + z^\alpha U_{zz} &= 0 && \text{in } \Omega \times (0, \infty), \\ U(x, z) &= 0 && \text{on } \partial\Omega \times (0, \infty), \\ \partial_\nu U(x, 0) &= d_s f(x) && \text{on } \Omega \times \{0\}, \end{aligned} \tag{2.1}$$

where $\alpha = 2 - \frac{1}{s}$, and $d_s = s^{2s-1}\frac{\Gamma(1-s)}{\Gamma(s)}$. Here, Δ_x denotes the Laplacian with respect to the original domain Ω and U_{zz} denotes the second derivative with respect to the extended dimension z. After solving for U, we can recover the solution to (1.1) as $u(x) = U(x, 0)$.

In general, approximating functions by polynomials on an unbounded domain is a challenging problem. Motivated by the fact that the solution U in the z-direction decays exponentially [24] (also confirmed by our numerical experiments), we consider the following truncated problem:

$$\begin{aligned} z^{(1/s)}\Delta_x U + z^2 U_{zz} &= 0 && \text{in } \Omega \times (0, R), \\ U(x, z) &= 0 && \text{on } \partial\Omega \times (0, R), \\ U(x, R) &= 0 && \text{on } \Omega \times \{R\}, \\ \partial_\nu U(x, 0) &= d_s f(x) && \text{on } \Omega \times \{0\}, \end{aligned} \tag{2.2}$$

where $R > 0$ is the truncation parameter. In our numerical experiments, the choice of R is motivated by [24]. In particular, we set $R = \mathcal{O}(\log(\mathrm{DoF}_\Omega))$ for the rectangular domains, where DoF_Ω is the total degrees of freedom used for Ω. Experimentally, we notice that for the disc domain it is more appropriate to choose $R = \mathcal{O}(\mathrm{DoF}_\Omega^{1/3})$. We emphasize that the additional variable z introduced by the extension requires that we solve a problem of one dimension higher. In particular, although Ω is chosen to be rectangles or disks in this paper, we must solve (2.2) in hexahedron or cylinders. In the subsequent sections, we review some basic polynomial bases for discretization and tensor operations for the discrete operators.

2.1. Ultraspherical Polynomial Basis and Spectral Methods.

Ultraspherical (or Gegenbauer) polynomials are a special family of polynomials that are usually denoted by $C_n^{(\lambda)}(x)$. Here, $\lambda > 0$ is a coefficient and n is the polynomial degree of x [25, Table 18.3.1]. They are orthogonal on the interval $(-1, 1)$ with respect to the weight function $w(x) = (1 - x^2)^{\lambda - 1/2}$ and satisfy the three-term recurrence [25, Table 18.9.1]

$$\begin{aligned} C_0^{(\lambda)}(x) &= 1, \\ C_1^{(\lambda)}(x) &= 2\lambda x, \\ C_{n+1}^{(\lambda)}(x) &= \frac{2(n+\lambda)}{n+1} x C_n^{(\lambda)}(x) - \frac{n + 2\lambda - 1}{n+1} C_{n-1}^{(\lambda)}(x). \end{aligned} \tag{2.3}$$

For notational convenience, we use $\tilde{C}_n^{(\lambda)}(x)$ to denote the $L^2(-1, 1)$-normalized ultraspherical polynomials with respect to the weight $w(x)$. Two well-known classes of polynomials, Chebyshev polynomials of the second kind and Legendre polynomials, are both special cases of ultraspherical polynomials, with coefficient $\lambda = 1$ and $\lambda = 1/2$, respectively.

2.2. **Tensor Notations.** We follow the notation for tensors found in [19], which we briefly review now for the reader's convenience. A tensor is a multidimensional array. We focus on three-dimensional tensors, denoted by calligraphic upper case letters, such as $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$. Comparatively, matrices are represented by upper case letters, such as $X \in \mathbb{C}^{n_1 \times n_2}$. We emphasize that the following discussion directly extends to four-dimensional tensors, see Section 5.

To represent submatrices, and fibers and slices of tensors, we employ MATLAB notation. In particular, "$m : n$" means that we take all integers between m and n, including m and n, and a simple ":" means that we take all available indices. For example, $A_{1:4,:}$ is a submatrix of A containing the first four rows and all the columns. We also use the keyword "end" to indicate the last index. For example, $A_{\text{end},:}$ represents the last row of A. Fibers are higher-order analogues of matrix rows and columns, formed by fixing all but one index. A three-dimensional tensor $\mathcal{X}$ has column fibers $\mathcal{X}_{:,j,k}$, row fibers $\mathcal{X}_{i,:,k}$, and tube fibers $\mathcal{X}_{i,j,:}$. Slices are two-dimensional sections of tensors made by fixing all but two indices. A three-dimensional tensor $\mathcal{X}$ has horizontal slices $\mathcal{X}_{i,:,:}$, lateral slices $\mathcal{X}_{:,j,:}$, and frontal slices $\mathcal{X}_{:,:,k}$. The k-fold (or k-mode) product of a tensor $\mathcal{X} \in \mathbb{C}^{n_1 \times n_2 \times n_3}$ with a matrix $A \in \mathbb{C}^{n_k \times n_k}$ is denoted by $\mathcal{X} \times_k A$, and defined elementwise as

$$\begin{aligned}
(\mathcal{X} \times_1 A)_{i,j,k} &= \sum_{i_1=1}^{n_1} \mathcal{X}_{i_1,j,k} A_{i,i_1}, \\
(\mathcal{X} \times_2 A)_{i,j,k} &= \sum_{i_2=1}^{n_2} \mathcal{X}_{i,i_2,k} A_{j,i_2}, \\
(\mathcal{X} \times_3 A)_{i,j,k} &= \sum_{i_3=1}^{n_3} \mathcal{X}_{i,j,i_3} A_{k,i_3}.
\end{aligned} \tag{2.4}$$

The k-fold product corresponds to each mode-k fiber of X being multiplied by the matrix A.

3. Spectral Discretization for Fractional PDEs on a Disk

In this section, we solve (2.2) on a disk domain. Without loss of generality, we let Ω be the unit circle. Otherwise, we can easily convert to this problem by scaling with the radius. We use polar coordinates to rewrite (2.2) as

$$\begin{aligned}
z^{1/s}\left(U_{\rho\rho} + \frac{1}{\rho}U_\rho + \frac{1}{\rho^2}U_{\theta\theta}\right) + z^2 U_{zz} &= 0 && \text{in } [0,1) \times [-\pi,\pi] \times (0,R), \\
U(1,\theta,z) &= 0 && \text{on } [-\pi,\pi] \times (0,R), \\
U(\rho,\theta,R) &= 0 && \text{on } [0,1) \times [-\pi,\pi], \\
\partial_\nu U(\rho,\theta,0) &= d_s f(\rho,\theta) && \text{on } [0,1) \times [-\pi,\pi].
\end{aligned} \tag{3.1}$$

To avoid the singularity at $\rho = 0$, we use the DFS method [33] to extend to $\rho \in (-1,1)$ by setting

$$\tilde{U}(\rho,\theta,z) = \begin{cases} U(\rho,\theta,z) & (\rho,\theta,z) \in [0,1) \times [-\pi,\pi] \times (0,R) \\ U(-\rho,\theta+\pi,z) & (\rho,\theta,z) \in (-1,0] \times [-\pi,\pi] \times (0,R) \end{cases},$$

$$\tilde{f}(\rho,\theta) = \begin{cases} f(\rho,\theta) & (\rho,\theta) \in [0,1) \times [-\pi,\pi] \\ f(-\rho,\theta+\pi) & (\rho,\theta) \in (-1,0] \times [-\pi,\pi] \end{cases}.$$

Notice that both $\tilde{U}$ and $\tilde{f}$ are now continuous at $\rho = 0$, which leads to simpler spectral discretization with smaller polynomial degrees. From this point, we work directly with these "doubled" functions

so that the singularity at $\rho = 0$ does not require additional consideration. The disk domain allows us to assume that both the solution $\tilde{U}$ and the function $\tilde{f}$ have Fourier expansions:

$$\tilde{U}(\rho,\theta,z) \approx \sum_{k=-m/2}^{m/2-1} \tilde{U}_k(\rho,z)e^{ik\theta} \qquad \text{and} \qquad \tilde{f}(\rho,\theta) \approx \sum_{k=-m/2}^{m/2-1} \tilde{f}_k(\rho)e^{ik\theta}.$$

In this way, we can decouple (3.1) into differential equations for each Fourier mode:

$$\begin{aligned} z^{1/s}\left((\tilde{U}_k)_{\rho\rho} + \frac{1}{\rho}(\tilde{U}_k)_\rho - \frac{k^2}{\rho^2}\tilde{U}_k\right) + z^2(\tilde{U}_k)_{zz} &= 0 && \text{in } (-1,1)\times(0,R), \\ \tilde{U}_k(\pm 1,z) &= 0 && \text{on } (0,R), \\ \tilde{U}_k(\rho,R) &= 0 && \text{on } (-1,1), \\ \partial_\nu \tilde{U}_k(\rho,0) &= d_s\tilde{f}_k(\rho) && \text{on } (-1,1). \end{aligned} \tag{3.2}$$

Following [18, § 4.1.2], we assume that the ansatz for $\tilde{U}_k$ is given by

$$\tilde{U}_k(\rho,z) = (1-\rho^2)\rho^{\min(|k|,2)}\tilde{V}_k(\rho,z), \tag{3.3}$$

where the term $(1-\rho^2)$ incorporates the boundary condition $\tilde{U}_k(\pm 1,z) = 0$. Then, we solve for $\tilde{V}_k$. The choice of the above ansatz only imposes partial regularity on $\tilde{U}_k$, and we refer to [18, § 4.1.2] for a detailed discussion on why it is challenging to impose full regularity.

Since the fractional exponent s can be any number between 0 and 1, the function $z^{1/s}$ makes the development of solvers for (3.2) a challenging task for our spectral method. To overcome this difficulty, we develop different solvers for varied values of s.

3.1. **Polynomial Approximation of $z^{1/s}$.** We first consider the case in which $s \in (0,1)$ is such that the function $z \mapsto z^{(1/s)}$ can be approximated accurately by a polynomial of low degree. For the discretized problem, the multiplication by $z^{(1/s)}$ is transformed into a matrix-matrix product. Consequently, the polynomial degree is directly related to the discretization size, and the definition of "low degree" in the above statement is related to the size of the discretized system one is capable of solving. For example, one may interpret "low degree" to mean "degree less than 50". In this case, any value of s such that $1/s$ is an integer between 1 and 50 falls into this class.

To make the spectral discretization easier, we make the change-of-variables $w = \frac{2}{R}z - 1 \in (-1,1)$. In this way, the PDE in (3.2) becomes:

$$\begin{aligned} \left(\frac{R(w+1)}{2}\right)^{1/s}\left((\tilde{U}_k)_{\rho\rho} + \frac{1}{\rho}(\tilde{U}_k)_\rho - \frac{k^2}{\rho^2}\tilde{U}_k\right) + (w+1)^2(\tilde{U}_k)_{ww} &= 0 && \text{in } (-1,1)^2, \\ \tilde{U}_k(\pm 1,w) &= 0 && \text{on } (-1,1), \\ \tilde{U}_k(\rho,1) &= 0 && \text{on } (-1,1), \\ \partial_\nu \tilde{U}_k(\rho,-1) &= Rd_s\tilde{f}_k(\rho)/2 && \text{on } (-1,1), \end{aligned} \tag{3.4}$$

and we solve for

$$\tilde{U}_k(\rho,w) = (1-\rho^2)\rho^{\min(|k|,2)}\tilde{V}_k(\rho,w). \tag{3.5}$$

Conventionally, one spectrally discretizes $\tilde{V}_k$ with Chebyshev polynomials for both ρ and w, but the matrices used to represent differentiation and function multiplication in the discretized equations

are dense and hard to manipulate. Instead, we set

$$\tilde{V}_k(\rho, w) = \sum_{i=0}^{n_1-1} \sum_{j=0}^{n_2-1} X_{ij}^{(k)} \tilde{C}_i^{(3/2)}(\rho) T_j(w), \quad \tilde{f}_k(\rho) = \sum_{i=0}^{n_1-1} F_i^{(k)} \tilde{C}_i^{(3/2)}(\rho), \tag{3.6}$$

where $T_j(w)$ is the Chebyshev polynomial of the first kind of degree j, $X^{(k)}$ is the matrix of coefficients of $\tilde{V}_k$ in the $\tilde{C}^{(3/2)}$ basis, and $F^{(k)}$ is the vector of coefficients of $\tilde{f}_k$ in the Chebyshev basis. Although $\tilde{C}^{(3/2)}$ is an uncommon polynomial basis, it is easy and efficient to transform coefficients in the $\tilde{C}^{(3/2)}$ basis to Chebyshev coefficients [18]. Therefore, users of the solver do not need to know about the special ultrashperical polynomial basis.

In this way, (3.5) indicates that we have three cases:

- If $|k| \geq 2$, then $\tilde{U}_k = \rho^2(1-\rho^2)\tilde{V}_k$, and (3.2) becomes:

$$\begin{aligned} \left(\frac{R}{2}\right)^{1/s} (w+1)^{1/s}[\rho^2(1-\rho^2)(\tilde{V}_k)_{\rho\rho} + \rho(5-9\rho^2)(\tilde{V}_k)_\rho + (4-k^2+(k^2-16)\rho^2)\tilde{V}_k], & \\ +\rho^2(1-\rho^2)(w+1)^2(\tilde{V}_k)_{ww} = 0 & \quad \text{in } (-1,1)^2, \\ \tilde{V}_k(\rho, 1) = 0 & \quad \text{on } (-1,1), \\ \rho^2(1-\rho^2)\partial_\nu \tilde{V}_k(\rho,-1) = d_s R \tilde{f}_k(\rho)/2 & \quad \text{on } (-1,1). \end{aligned} \tag{3.7}$$

  Following [18] on operations related to $\tilde{C}^{(3/2)}$ and [26] on operations related to Chebyshev polynomials, (3.7) can be discretized to the following matrix equations:

$$\begin{aligned} (M_{\rho^2}D + 5M_\rho M D_u + (14-k^2)M - 10I)X^{(k)}(S_{2,0}B_1)^T + (M_{\rho^2}M)X^{(k)}(B_2D_2)^T = 0 \\ (M_{\rho^2}M)X^{(k)}L^T = H^{(k)}, \end{aligned} \tag{3.8}$$

  where D is a diagonal matrix representing the second derivatives of $(1-\rho^2)\tilde{C}^{(3/2)}(\rho)$, M is a symmetric penta-diagonal matrix with 0 super and sub diagonals representing the operation of multiplying $\tilde{C}^{(3/2)}(\rho)$ by $(1-\rho^2)$, $M_{\rho^2} = I - M$ represents the operation of multiplying $\tilde{C}^{(3/2)}(\rho)$ by ρ^2, M_ρ is a tridiagonal matrix derived from the three-term recurrence (2.3) to represent the operation of multiplying $\tilde{C}^{(3/2)}(\rho)$ by ρ, D_u is a dense matrix representing the first derivatives of $\tilde{C}^{(3/2)}(\rho)$, but MD_u is tridiagonal [25, (18.9.8) & (18.9.19)], B_1 represents the operation of multiplying the Chebyshev polynomials $T(w)$ by $(R/2)^{1/s}(1+w)^{1/s}$, B_2 represents the operation of multiplying $C^{(2)}(x)$—ultraspherical polynomials with coefficient 2—by $(1+w)^2$, D_2 represents the second derivative of the Chebyshev basis, $S_{2,0}$ converts the Chebyshev basis to the $C^{(2)}$ basis,

$$L = \begin{bmatrix} T_0'(-1) & T_1'(-1) & \dots \\ T_0(1) & T_1(1) & \dots \end{bmatrix},$$

  and $H^{(k)}$ is a matrix with two columns:

$$H^{(k)} = \begin{bmatrix} -d_s R F^{(k)}/2 & 0 \end{bmatrix}.$$

  One can find a visualization of these operational matrices in Appendix A.

- If $|k| = 1$, then $\tilde{U}_k = \rho(1-\rho^2)\tilde{V}_k$. Using the same matrix notation, the discretized matrix equation is:

$$\begin{aligned} (M_\rho D + 3MD_u - 6M_\rho)X^{(k)}(S_{2,0}B_1)^T + (M_\rho M)X^{(k)}(B_2D_2)^T = 0 \\ (M_\rho M)X^{(k)}L^T = H^{(k)}. \end{aligned} \tag{3.9}$$

- If $k = 0$, then $\tilde{U}_k = (1-\rho^2)\tilde{V}_k$, and the discretized version of (3.2) is:

$$\begin{aligned}(M_{\rho^2}D + M_\rho M D_u - 2M_{\rho^2})X^{(0)}(S_{2,0}B_1)^T + (M_{\rho^2}M)X^{(0)}(B_2D_2)^T = 0\\ MX^{(0)}L^T = H^{(0)}.\end{aligned} \tag{3.10}$$

We can merge the linear equation with the linear constraint into one matrix equation in all three cases, see for instance [31] for a similar approach. It is desirable to keep the structure and the sparsity of the matrices related to ultraspherical discretization while solving the equation. However, due to the variable s, we do not know the spectrum of the matrices associated with the Chebyshev basis, which is essential if we want to use fast iterative solvers such as an alternating direction implicit (ADI) method [9]. Therefore, we treat all matrices as general dense matrices and solve all matrix equations with the Bartels–Stewart algorithm [8]. Nevertheless, QZ decompositions on sparse penta-diagonal matrices are cheaper and more stable than general dense matrices. Thus, we gain from using ultraspherical polynomials. The solutions $X^{(k)}$ can be transformed to the Chebyshev coefficient matrix for both variables, and then they form the frontal slices of the solution $\mathcal{Z}$ to the discretized DFS version of (3.1). Finally, the coefficient matrix that represents the solution of (1.1) can be calculated via

$$\mathcal{Z} \times_2 \begin{bmatrix} T_0(-1) & T_1(-1) & \cdots \end{bmatrix}.$$

We summarize this solver in Algorithm 1.

Algorithm 1 Fractional PDE solver on the unit disk — Polynomial approximation of $z^{1/s}$

1: **Input**: The coefficient matrix F of f after DFS extension in Chebyshev and Fourier bases.
2: **Output**: The coefficient matrix W of the solution u in Chebyshev and Fourier bases.
3: Convert the columns of F into coefficients in $\tilde{C}^{(3/2)}$ basis.
4: **while** $\max_i X^{(k)}_{i,n_2} > 10^{-10}$ for any k in (3.6) **do**
5: Increase n_2
6: Solve (3.8), (3.9), and (3.10) for $X^{(k)}$.
7: Stack all $X^{(k)}$ in the tube direction to form $\mathcal{Z}$.
8: **end while**.
9: Calculate $W = \mathcal{Z} \times_2 \begin{bmatrix} T_0(-1) & T_1(-1) & \ldots \end{bmatrix}$.
10: Convert the columns of W into coefficients in Chebyshev basis.

Remark 3.1 (Numerical convergence)**.** *As a test, consider $s = 1/2$ so that the extended PDE is a Laplace equation on a cylinder. When $f = J_0(s_{01}\rho)$, where J_0 is the first Bessel function of the first kind, and s_{01} is the first nonzero root of J_0, the solution is*

$$U = \frac{1}{s_{01}\cosh(s_{01}R)} J_0(s_{01}\rho)\sinh(s_{01}R(1-z)/2).$$

Figure 1 shows the reduction of approximation error obtained by our adaptive solver.

3.2. **Piecewise Polynomial Approximation of $z^{1/s}$.** The more challenging case is when $z^{1/s}$ is not well-approximated by a polynomial. For these values of s, we approximate $z^{1/s}$ with piecewise polynomials, i.e.,

$$z^{1/s} \approx p_i(z), \quad z_i \le z \le z_{i+1}, \tag{3.11}$$

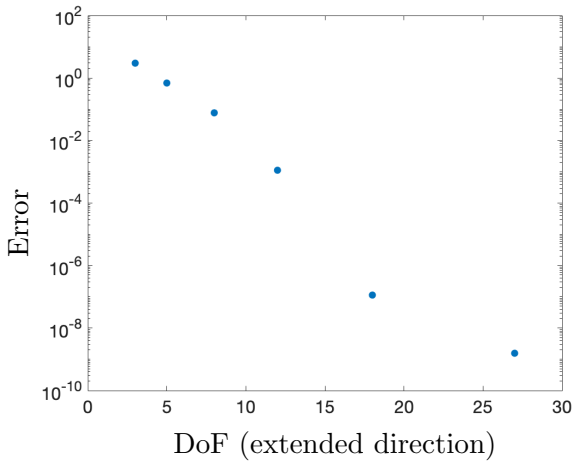

FIGURE 1. Solving fractional PDE for $f = J_0(s_{01}\rho)$ and $s = 1/2$ on the unit disk with our spectral solver. By increasing the degrees of freedom in the extended direction, the error between the analytic and numerical solutions decreases.

where p_i is a polynomial for $0 \le i \le \ell - 1$, $z_0 = 0$, and $z_\ell = R$. Then, on the interval (z_i, z_{i+1}), we have that

$$p_i(z)\left((\tilde{U}_k^{(i)})_{\rho\rho} + \frac{1}{\rho}(\tilde{U}_k^{(i)})_\rho - \frac{k^2}{\rho^2}\tilde{U}_k^{(i)}\right) + z^2(\tilde{U}_k^{(i)})_{zz} = 0 \qquad \text{in } (-1,1)\times[-\pi,\pi]\times(z_i, z_{i+1}),$$
$$\tilde{U}_k^{(i)}(\pm 1, z) = 0 \qquad \text{on } (z_i, z_{i+1}),$$

with bottom and top boundary conditions

$$\begin{cases} \tilde{U}_k^{(0)}(\rho, z_1) = \tilde{\phi}_1(\rho), \\ \partial_\nu \tilde{U}_k^{(0)}(\rho, z_0) = d_s \tilde{f}_k(\rho), \\ \partial_\nu \tilde{U}_k^{(0)}(\rho, z_1) = \tilde{\psi}_1(\rho), \end{cases} \qquad \begin{cases} \tilde{U}_k^{(\ell-1)}(\rho, z_{\ell-1}) = \tilde{\phi}_{\ell-1}(\rho), \\ \tilde{U}_k^{(\ell-1)}(\rho, R) = 0, \\ \partial_\nu \tilde{U}_k^{(\ell-1)}(\rho, z_{\ell-1}) = -\tilde{\psi}_{\ell-1}(\rho), \end{cases}$$

and

$$\begin{cases} \tilde{U}_k^{(i)}(\rho, z_i) = \tilde{\phi}_i(\rho), \\ \tilde{U}_k^{(i)}(\rho, z_{i+1}) = \tilde{\phi}_{i+1}(\rho), \\ \partial_\nu \tilde{U}_k^{(i)}(\rho, z_i) = -\tilde{\psi}_i(\rho), \\ \partial_\nu \tilde{U}_k^{(i)}(\rho, z_{i+1}) = \tilde{\psi}_{i+1}(\rho), \end{cases} \qquad 1 \le i \le l-2, \tag{3.12}$$

where $\tilde{\phi}_i$ and $\tilde{\psi}_i$ are implicitly defined. We use these implicit boundary conditions to ensure that $\tilde{U}_i$ can form an overall continuous solution. On intersection surfaces, solutions on the two sides have identical Dirichlet conditions and opposite Neumann conditions.

On each interval, we perform a change of variables to produce similar PDEs as (3.4). These PDEs can be discretized into $m\ell$ matrix equations. We solve on all intervals simultaneously by combining the matrix equations corresponding to the same Fourier mode into one equation. We use the matrix notation from Section 3.1 and get the following three matrix equations:

- If $|k| \geq 2$, then we have:

$$(M_{\rho^2}D + 5M_\rho M D_u + (14-k^2)M - 10I)Y^{(k)}E_1^T + (M_{\rho^2}M)Y^{(k)}E_2^T = 0$$
$$(M_{\rho^2}M)Y^{(k)}B_z^T = H^{(k)}. \tag{3.13}$$

- If $|k| = 1$, then we have:

$$(M_\rho D + 3MD_u - 6M_\rho)Y^{(k)}E_1^T + (M_\rho M)Y^{(k)}E_2^T = 0$$
$$(M_\rho M)Y^{(k)}B_z^T = H^{(k)}. \tag{3.14}$$

- If $k = 0$, then we have:

$$(M_{\rho^2}D + M_\rho M D_u - 2M_{\rho^2})Y^{(0)}E_1^T + (M_{\rho^2}M)Y^{(0)}E_2^T = 0$$
$$MY^{(0)}B_z^T = H^{(0)}. \tag{3.15}$$

In the matrix equations above, we set

$$Y^{(k)} = \begin{bmatrix} X_0^{(k)} & \cdots & X_{\ell-1}^{(k)} \end{bmatrix},$$

where $X_j^{(k)}$ is the kth Fourier mode solution on (z_j, z_{j+1}). The first column of $H^{(k)}$ is $-d_s R\tilde{f}_k(\rho)/2$ and all other columns are zero. E_1 and E_2 are block diagonal matrices with diagonal blocks $S_{2,0}B_i$ and K_iD_2, respectively, where B_i represents the multiplication of p_i and K_i represents the multiplication of the Chebyshev basis by $(w + \frac{z_{i+1}+z_i}{z_{i+1}-z_i})^2$. In addition, we set

$$B_z = \begin{bmatrix} P_{-1} & & & & & \\ \frac{2}{z_1}P_1 & -\frac{2}{z_2-z_1}P_{-1} & & & & \\ Q_1 & -Q_{-1} & & & & \\ & \frac{2}{z_2-z_1}P_1 & -\frac{2}{z_3-z_2}P_{-1} & & & \\ & Q_1 & -Q_{-1} & & & \\ & & & & \ddots & \\ & & & & & Q_1 \end{bmatrix},$$

where $P_{-1} = [T_0'(-1)\ T_1'(-1)\cdots]$, $P_1 = [T_0'(1)\ T_1'(1)\cdots]$, $Q_{-1} = [T_0(-1)\ T_1(-1)\cdots]$, and $Q_1 = [T_0(1)\ T_1(1)\cdots]$. Then, we can form X_0 from $Y^{(k)}$ and obtain the coefficient matrix of the solution u of (1.1).

By construction, the first two columns of D_2 are 0. As a result, E_2 has 2ℓ scattered zero columns. This property is undesirable when using the solver in [31]. Instead, we permute the columns of E_2 so that the first 2ℓ columns are 0, and permute E_1, $Y^{(k)}$, B_z and $H^{(k)}$ accordingly. After solving the permuted matrix equations, we can easily obtain the original solution. We note that the two solvers are, in fact, equivalent for different values of s. The solver in this subsection can be thought of as a generalized version of the solver in Section 3.1, where the solution only has one piece and no implicit boundary conditions are needed. This generalized solver is described in Algorithm 2.

Remark 3.2. *In our numerics, we use* `Chebfun` [16] *to automatically partition the extended direction and to approximate the map $z \mapsto z^{1/s}$ with Chebyshev polynomials on each domain with desired accuracy level, i.e., to determine p_i and z_i in (3.11).* `Chebfun` *only divides the domain when a high degree polynomial cannot achieve the resolution, so the piecewise polynomial approximation tends to have as few pieces as possible. From the above discretized systems, one notices that more pieces*

Algorithm 2 Fractional PDE solver on the unit disk — Piecewise approximation of $z^{1/s}$

Input: The coefficient matrix F of f after DFS extension in Chebyshev and Fourier basis.
Output: The coefficient matrix W of the solution u in Chebyshev and Fourier basis.
Convert the columns of F into coefficients in $\tilde{C}^{(3/2)}$ basis.
while $\max_i \left(X_j^{(k)}\right)_{i,n_2^{(j)}} > 10^{-10}$ for any k or j **do**
    Increase $n_2^{(j)}$.
    Construct the matrices in (3.13), (3.14), and (3.15), and permute E_1, E_2, $Y^{(k)}$, B_z and $H^{(k)}$ such that the first 2ℓ columns of E_2 are zero columns while the equations still hold.
    Solve for $Y^{(k)}$.
    Stack $X_0^{(k)}$ to form $\mathcal{X}_0$.
end while.
Calculate $W = \mathcal{X}_0 \times_2 \begin{bmatrix} T_0(-1) & T_1(-1) & \dots \end{bmatrix}$.
Convert the columns of W into coefficients in Chebyshev basis.

lead to more complicated matrix equations to solve. Consequently, it is beneficial to use `Chebfun` *for approximations. Figure 2 shows the number of polynomial segments needed to approximate the map* $z \mapsto z^{1/s}$ *on* $(0, 10)$ *for varying* $s \in (0, 1)$ *with an accuracy of* 10^{-12} *using* `Chebfun`*. If we strive for machine precision, the numbers need to be larger. However, we found in practice that* 10^{-12} *gives sufficiently accurate PDE solutions. Comparatively, it is also possible to partition the* z*-direction into more segments so that* $z^{1/s}$ *can be approximated by a low degree polynomial on each interval. For this way of partitioning, we point the readers to [6] for more details.*

4. Spectral Discretization for Fractional PDEs on a Rectangle

In this section, we solve (1.1) in the unit square $(-1, 1)^2$ by spectrally discretizing the truncated, extended PDE (2.2). General rectangles are straightforward to scale to unit squares by a change-of-variables and are thus easy to solve with the same method. It is also worth noting that the solver can be easily generalized to higher-dimensional domains. To this end, we demonstrate the solver application on a cube in Section 5.

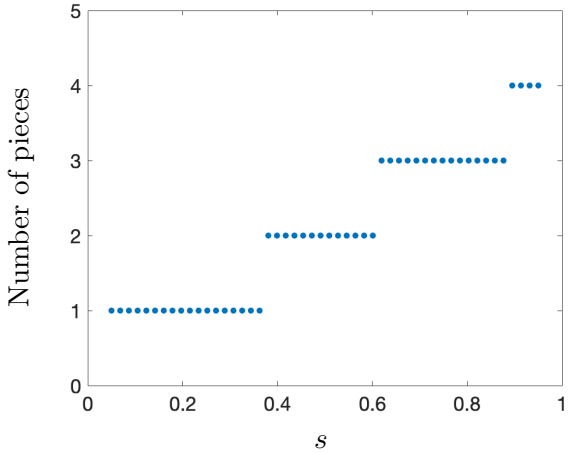

Figure 2. The number of segments for a piecewise polynomial approximation of $z^{1/s}$ on $(0, 10)$ with accuracy 10^{-12}.

We first introduce a direct solver that is similar to the solvers in Section 3. The direct solver, can encounter efficiency issues as the equations become large when $z^{1/s}$ needs to be approximated by a piecewise polynomial with many pieces. In those scenarios, we design a parallelizable solver using domain decomposition in Section 4.2.

4.1. **Direct Solver.** We first approximate $z^{1/s}$ as an ℓ-piece piecewise polynomial (3.11), with the simplest case being $\ell = 1$. On each interval (z_i, z_{i+1}), we have the PDE:

$$\begin{aligned} p_i(z)\left(U^{(i)}_{xx} + U^{(i)}_{yy}\right) + z^2 U^{(i)}_{zz} &= 0 && \text{in } (-1,1)\times(-1,1)\times(z_i, z_{i+1}),\\ U^{(i)}(\pm 1, y, z) &= 0 && \text{on } (-1,1)\times(z_i, z_{i+1}),\\ U^{(i)}(x, \pm 1, z) &= 0 && \text{on } (-1,1)\times(z_i, z_{i+1}), \end{aligned}$$

with bottom and top boundary conditions

$$\begin{cases} U^{(0)}(x,y,z_1) &= \phi_1(x,y),\\ \partial_\nu U^{(0)}(x,y,z_0) &= d_s f_k(x,y),\\ \partial_\nu U^{(0)}(x,y,z_1) &= \psi_1(x,y), \end{cases} \quad \begin{cases} U^{(\ell-1)}(x,y,z_{\ell-1}) &= \phi_{\ell-1}(x,y),\\ U^{(\ell-1)}(x,y,R) &= 0,\\ \partial_\nu U^{(\ell-1)}(x,y,z_{\ell-1}) &= -\psi_{\ell-1}(x,y), \end{cases}$$

and

$$\begin{cases} U^{(i)}(x,y,z_i) &= \phi_i(x,y),\\ U^{(i)}(x,y,z_{i+1}) &= \phi_{i+1}(x,y),\\ \partial_\nu U^{(i)}(x,y,z_i) &= -\psi_i(x,y),\\ \partial_\nu U^{(i)}(x,y,z_{i+1}) &= \psi_{i+1}(x,y), \end{cases} \quad 1 \le i \le l-2, \tag{4.1}$$

where ϕ_i and ψ_i are implicitly defined to ensure continuity of the entire solution. We employ the change the variables $w = \frac{2}{z_{i+1}-z_i}(z - z_i) - 1$ on (z_i, z_{i+1}), and assume the ansatz:

$$\begin{aligned} U^{(i)} &= (1-x^2)(1-y^2)\sum_{p=0}^{n_1}\sum_{q=0}^{n_2}\sum_{r=0}^{n_3^{(i)}} \mathcal{X}^{(i)}_{pqr}\tilde{C}_p^{(3/2)}(x)\tilde{C}_q^{(3/2)}(y)T_r(w),\\ f &= \sum_{p=0}^{n_1}\sum_{q=0}^{n_2} F_{pq}\tilde{C}_p^{(3/2)}(x)\tilde{C}_q^{(3/2)}(y), \end{aligned} \tag{4.2}$$

where $\tilde{C}_p^{(3/2)}(x)$ is the pth normalized ultraspherical polynomial with coefficient $3/2$, $T_r(w)$ is the rth Chebyshev polynomial, $\mathcal{X}^{(i)}$ is a 3D tensor, and F is a matrix. As before, $\mathcal{X}^{(i)}$ and F contain the coefficients of $U^{(i)}$ and f in $\tilde{C}^{(3/2)}$ and Chebyshev basis, respectively. We can then write the discretized problem as a tensor equation by stacking $\mathcal{X}^{(i)}$ in the tube direction to form $\mathcal{Y}$:

$$\begin{aligned} \mathcal{Y}\times_1 A\times_3 E_1 + \mathcal{Y}\times_2 A\times_3 E_1 + \mathcal{Y}\times_1 A\times_2 A\times_3 E_2 &= 0,\\ \mathcal{Y}\times_1 M\times_2 M\times_3 B_z &= \mathcal{G}, \end{aligned} \tag{4.3}$$

where $A = D^{-1}M$, the first frontal slice of $\mathcal{G}$ is $-d_s RF/2$ and the remaining slices are zero, and all other matrices are defined in Section 3. In order to use a tensor analogue of the solver in [31], we first perform a column permutation on E_1, E_2 and B_z, and a frontal slice permutation on $\mathcal{Y}$ and $\mathcal{G}$ so that the first 2ℓ columns of E_2 are zero and (4.3) still holds.

The linear constraint $\mathcal{Y}\times_1 M\times_2 M\times_3 B_z = \mathcal{G}$ in (4.3) can be rewritten as $\mathcal{Y}\times_3 L = \mathcal{H}$, where $L = SB_z = \begin{bmatrix} I & \tilde{L}\end{bmatrix}$ such that the leftmost $2\ell\times 2\ell$ submatrix of L is the identity matrix, and

$\mathcal{H} = \mathcal{G} \times_1 M^{-1} \times_2 M^{-1} \times_3 S$. Then, (4.3) can be combined into a single equation:

$$\begin{aligned}\mathcal{Y} \times_1 A \times_3 (E_1 - (E_1)_{1:2\ell} L) + \mathcal{Y} \times_2 A \times_3 (E_1 - (E_1)_{1:2\ell} L) + \mathcal{Y} \times_1 A \times_2 A \times_3 (E_2 - (E_2)_{1:2\ell} L) \\ = -\mathcal{H} \times_1 A \times_3 (E_1)_{1:2\ell} - \mathcal{H} \times_2 A \times_3 (E_1)_{1:2\ell} - \mathcal{H} \times_1 A \times_2 A \times_3 (E_2)_{1:2\ell}, \quad (4.4)\end{aligned}$$

where $(E_1)_{1:2\ell}$ represents the first 2ℓ columns of E_1. Since the first 2ℓ columns of E_2 are 0 and the leftmost $2\ell \times 2\ell$ submatrix of L is the identity, we know that the first 2ℓ frontal slices of $\mathcal{Y}$ do not influence the solution. Therefore, we can solve for $\mathcal{Y}_2$, which contains the rest of the frontal slices, by solving a smaller Sylvester equation:

$$\begin{aligned}\mathcal{Y}_2 \times_1 A \times_3 R_1 + \mathcal{Y}_2 \times_2 A \times_3 R_1 + \mathcal{Y}_2 \times_1 A \times_2 A \times_3 R_2 \\ = -\mathcal{H} \times_1 A \times_3 (E_1)_{1:2\ell} - \mathcal{H} \times_2 A \times_3 (E_1)_{1:2\ell}, \quad (4.5)\end{aligned}$$

where R_1 is the first $\left(\sum_i \left(n_3^{(i)}\right) - 2\ell\right)$ rows of $(E_1)_{2\ell+1:\sum_i \left(n_3^{(i)}\right)} - (E_1)_{1:2\ell}\tilde{L}$, and R_2 is the first $\left(\sum_i \left(n_3^{(i)}\right) - 2\ell\right)$ rows of $(E_1)_{2\ell+1:\sum_i \left(n_3^{(i)}\right)}$. After computing $\mathcal{Y}_2$, it is then straightforward to use the linear constraint to calculate the first 2ℓ frontal slices by $\mathcal{Y}_1 = \mathcal{H} - \mathcal{Y}_2 \times_3 \tilde{L}$.

Since we do not know the behavior of the spectrum of R_1 and R_2, we use a tensor analogue of the Bartels–Stewart algorithm [8] to solve for $\mathcal{Y}_2$. To be specific, we take real Schur decompositions of the pairs of matrices using the QZ decomposition [8], and then solve for each column fiber of $\mathcal{Y}_2$. Finally, we recover $\mathcal{X}^{(0)}$ as the first $n_3^{(0)}$ frontal slices of $\mathcal{Y}$, convert it to the Chebyshev coefficient tensor $\mathcal{Z}$ and get the coefficient matrix of the solution of (1.1) by $\mathcal{Z} \times_3 \begin{bmatrix} T_0(-1) & T_1(-1) & \dots \end{bmatrix}$. This direct solver on a unit square is summarized in Algorithm 3.

Algorithm 3 Direct fractional PDE solver on the unit square

1: **Input**: The coefficient matrix F of f in Chebyshev basis.
2: **Output**: The coefficient matrix W of the solution u in Chebyshev basis.
3: Convert both columns and rows of F into coefficients in $\tilde{C}^{(3/2)}$ basis.
4: **while** $\max_{p,q} \left|\mathcal{X}^{(i)}_{p,q,n_3^{(i)}}\right| > 10^{-10}$ for any i in (4.2) **do**
5: Increase $n_3^{(i)}$.
6: Solve (4.5) for $\mathcal{Y}_2$ and compute $\mathcal{Y}_1 = \mathcal{H} - \mathcal{Y}_2 \times_3 \tilde{L}$.
7: Stack $\mathcal{Y}_1$ and $\mathcal{Y}_2$ in the tube direction, and form $\mathcal{X}^{(0)}$ to be the first $n_3^{(1)}$ frontal slices of $\mathcal{Y}$.
8: **end while**.
9: Calculate $W = \mathcal{X}^{(0)} \times_2 \begin{bmatrix} T_0(-1) & T_1(-1) & \cdots \end{bmatrix}$.
10: Convert both columns and rows of W into coefficients in Chebyshev basis.

Remark 4.1 (Numerical convergence)**.** *As a numerical example, we consider the case that* $u = \sin(\pi x)\sin(\pi y) + \sin(2\pi x)\sin(2\pi y)$*, then* $f = (2\pi^2)^s \sin(\pi x)\sin(\pi y) + (8\pi^2)^s \sin(2\pi x)\sin(2\pi y)$ *by the spectral definition. Figure 3 (Left) shows the coefficient decay along the extended direction of the discretized tensor solution for different values of* s*. This plot demonstrates that when the algorithm terminates, the coefficient of the polynomial terms of the discretized solution is small enough. Figure 3 (Right) shows the accuracy improvements of our adaptive algorithm. With the increase of polynomial degree to approximate the extended domain, we obtain a reduction of error between the numerical and analytic solution.*

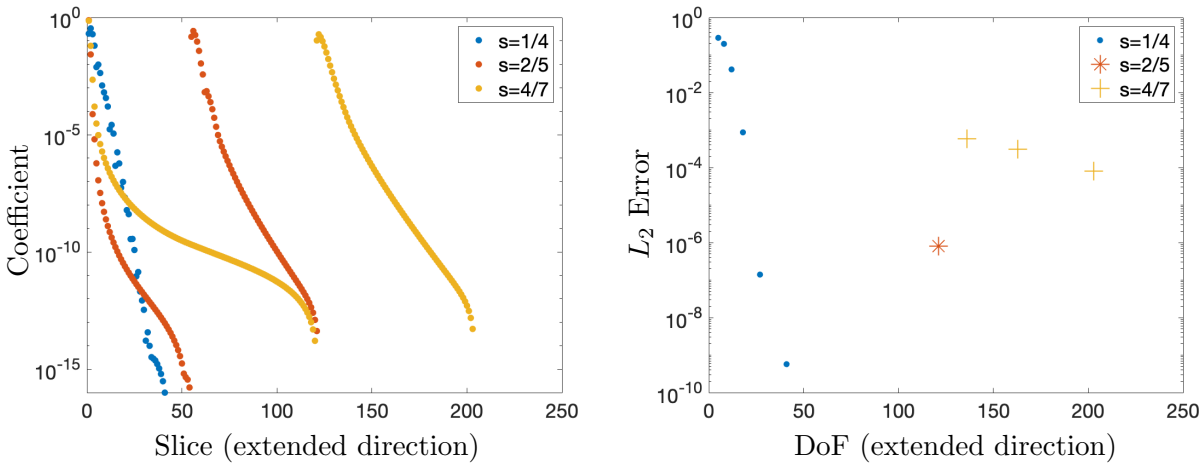

FIGURE 3. Solving the fractional PDE for $f = (2\pi^2)^s \sin(\pi x)\sin(\pi y) + (8\pi^2)^s \sin(2\pi x)\sin(2\pi y)$ and different values of s on $\Omega = (-1,1)^2$ with our spectral solver. **Left:** the largest coefficient in magnitude on each slice along the extended direction in the discretized solution, i.e., the largest $\mathcal{X}_{pqr}$ when r is fixed in (4.2). When $s = 2/5$ and $s = 4/7$, $z^{(1/s)}$ is approximated by piecewise polynomials with two pieces. The coefficient patterns show the decay of both pieces. **Right:** the accuracy achieved with different degrees of freedom in the extended direction. For $s = 1/4$, we achieve sufficient coefficient decay within five iterations. For $s = 4/7$, we require two iterations and for $s = 2/5$, the coefficients of a $28 \times 28 \times 121$ discretization decay below 10^{-13} in the first iteration, resulting in a single point on the plot.

Remark 4.2 (Numerical convergence for non-compatible datum)**.** *As another example, we consider a non-compatible case with $f = 2$ and $s < 1/2$, leading to low regularity of the solution to the fractional PDE* [24, Sec. 6.3]*. Figure 4 shows the coefficient decay along the extended direction of the discretized tensor solution for different values of s. Again, when the algorithm terminates, the coefficient of the polynomial terms of the discretized solution is small enough, and the number of coefficients to achieve this decay is smaller than that in Remark 4.1.*

4.2. **Domain Decomposition Solver.** In the previous section, we solved (1.1) on the unit square by jointly solving (4.1) for all segments. However, the solution of (1.1) only corresponds to $\mathcal{X}^{(0)}$, the solution on the first domain. Therefore, we design a domain decomposition solver, inspired by the hierarchical Poincaré–Steklov method [22], to solve only for $\mathcal{X}^{(0)}$.

Suppose we know the Robin boundary conditions for $\mathcal{X}^{(0)}$ and Dirichlet boundary conditions for all other $\mathcal{X}^{(i)}$. It is then straightforward to write a tensor equation for each $\mathcal{X}^{(i)}$ in the form of (4.4):

$$(\mathcal{X}^{(i)} \times_1 A + \mathcal{X}^{(i)} \times_2 A) \times_3 (C_i - (C_i)_{1:2} L_i) + \mathcal{X}^{(i)} \times_1 A \times_2 A \times_3 (M_2 - (M_2)_{1:2} L_i) = -\mathcal{H}^{(i)} \times_1 A \times_3 (C_i)_{1:2} - \mathcal{H}^{(i)} \times_2 A \times_3 (C_i)_{1:2}, \quad (4.6)$$

where $C_i = S_{2,0} B_i$, $M_2 = K_i D_2$, $L_0 = N_0 \begin{bmatrix} P_{-1} \\ Q_1 \end{bmatrix} = [I \ \tilde{L}_0]$, $L_i = N_i \begin{bmatrix} Q_{-1} \\ Q_1 \end{bmatrix} = [I \ \tilde{L}_i]$ for $i > 1$, $\mathcal{H}^{(i)} = \mathcal{G}^{(i)} \times_1 M^{-1} \times_2 M^{-1} \times_3 N_i$, and the two frontal slices of $\mathcal{G}^{(i)}$ correspond to the boundary

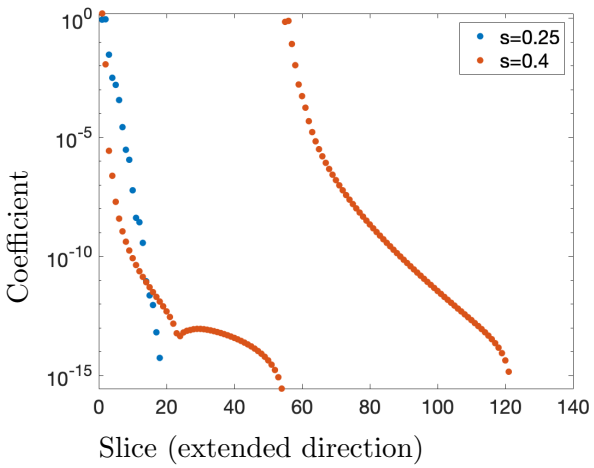

FIGURE 4. Solving the fractional PDE for $f = 2$ and different values of $s < 1/2$ on $\Omega = (-1, 1)^2$ with our spectral solver. The plot shows the largest coefficient in magnitude on each slice along the extended direction in the discretized solution, i.e., the largest $\mathcal{X}_{pqr}$ when r is fixed in (4.2). When $s = 2/5$, $z^{(1/s)}$ is approximated by a piecewise polynomial with two pieces. The coefficient patterns show the decay of both pieces.

conditions of $\mathcal{X}^{(i)}$ that are assumed to be known. This means that, once we have $\mathcal{G}^{(i)}$, we are able to solve for $\mathcal{X}^{(i)}$.

Equation (4.6) allows us to construct a solution map $S^{(i)} \in \mathbb{R}^{n_1 n_2 n_3^{(i)} \times 2n_1 n_2}$:

$$S^{(i)} = -[(C_i - (C_i)_{1:2}L_i) \otimes I \otimes A + (C_i - (C_i)_{1:2}L_i) \otimes A \otimes I + (M_2 - (M_2)_{1:2}L_i) \otimes A \otimes A]^{-1} \\ (((C_i)_{1:2}N_i) \otimes (I \otimes A + A \otimes I)), \tag{4.7}$$

such that $\text{vec}\left(\mathcal{X}^{(i)}\right) = S^{(i)} \text{ vec}\left(\mathcal{G}^{(i)}\right)$, where $\text{vec}\left(\mathcal{X}^{(i)}\right)$ reshapes all elements in $\mathcal{X}^{(i)}$ to a vector. In addition, we define the Dirichlet-to-Neumann (DtN) map $K^{(i)} \in \mathbb{R}^{2n_1 n_2 \times 2n_1 n_2}$ by

$$K^{(i)} = \frac{2}{z_i - z_{i-1}} \left(\begin{bmatrix} P_{-1} \\ P_1 \end{bmatrix} \otimes I \otimes I \right) S^{(i)}.$$

The map $K^{(i)}$ converts the Dirichlet boundary conditions $\mathcal{G}^{(i)}$ into the Neumann values $\mathcal{B}^{(i)}$ on the boundaries.

We solve for each column of $S^{(i)}$ by solving a tensor Sylvester equation:

$$\mathcal{S}_j^{(i)} \times_1 A \times (C_i - (C_i)_{1:2}L_i) + \mathcal{S}_j^{(i)} \times_2 A \times_3 (C_i - (C_i)_{1:2}L_i) + \mathcal{S}_j^{(i)} \times_1 A \times_2 A \times_3 (M_2 - (M_2)_{1:2}L_i) = \mathcal{W}_j^{(i)},$$

where $\mathcal{S}_j^{(i)}$ the jth column of $S^{(i)}$ reshaped to an $n_1 \times n_2 \times n_3^{(i)}$ tensor, and $\mathcal{W}_j^{(i)}$ is the jth column of $((C_i)_{1:2}N_i) \otimes (I \otimes A + A \otimes I)$ reshaped to an $n_1 \times n_2 \times n_3^{(i)}$ tensor. This suggests that the computation of $\mathcal{S}_j^{(i)}$ is a parallelizable process. We can calculate the operator $K^{(i)}$ by

$$\mathcal{K}_j^{(i)} = \frac{2}{z_i - z_{i-1}} \mathcal{S}_j^{(i)} \times_3 \begin{bmatrix} P_{-1} \\ P_1 \end{bmatrix},$$

where $\mathcal{K}_j^{(i)}$ is the jth vector of $K^{(i)}$ reshaped to a $2 \times n_1 \times n_2$ tensor.

Our goal is to use the solution operator $S^{(0)}$ to solve for $\mathcal{X}^{(0)}$. Thus, we must obtain the implicit Dirichlet boundary condition. Next, we show that we can carefully merge the solution maps $S^{(i)}$ and DtN maps $K^{(i)}$. As a result, we obtain solution maps and DtN maps that work on several domains simultaneously, and these merged maps can help us find the desired boundary condition.

The boundaries for each domain consist of one upper surface and one lower surface so that we can separate the boundary conditions into two parts and the DtN operator into four parts, i.e.,

$$\operatorname{vec}\left(\mathcal{G}^{(i)}\right) = \begin{bmatrix} \mathcal{G}_v^{(i)} \\ \mathcal{G}_u^{(i)} \end{bmatrix}, \quad \operatorname{vec}\left(\mathcal{B}^{(i)}\right) = \begin{bmatrix} \mathcal{B}_v^{(i)} \\ \mathcal{B}_u^{(i)} \end{bmatrix}, \quad K^{(i)} = \begin{bmatrix} K_{v,v}^{(i)} & K_{v,u}^{(i)} \\ K_{u,v}^{(i)} & K_{u,u}^{(i)} \end{bmatrix}.$$

Then we can follow [22] to merge the solution and DtN operators:

$$S^{(i,i+1)} = \left(K_{u,u}^{(i)} - K_{v,v}^{(i+1)}\right)^{-1} \begin{bmatrix} -K_{u,v}^{(i)} & K_{v,u}^{(i+1)} \end{bmatrix},$$

$$K^{(i,i+1)} = \begin{bmatrix} K_{v,v}^{(i)} & 0 \\ 0 & K_{u,u}^{(i+1)} \end{bmatrix} + \begin{bmatrix} K_{v,u}^{(i)} \\ K_{u,v}^{(i+1)} \end{bmatrix} S^{(i,i+1)},$$

so that

$$\mathcal{G}_u^{(i)} = \mathcal{G}_v^{(i+1)} = S^{(i,i+1)} \begin{bmatrix} \mathcal{G}_v^{(i)} \\ \mathcal{G}_u^{(i+1)} \end{bmatrix}, \quad \begin{bmatrix} \mathcal{B}_v^{(i)} \\ \mathcal{B}_u^{(i+1)} \end{bmatrix} = K^{(i,i+1)} \begin{bmatrix} \mathcal{G}_v^{(i)} \\ \mathcal{G}_u^{(i+1)} \end{bmatrix}.$$

In other words, given the Dirichlet boundary conditions on the lower surface of the ith domain and the upper surface of the $(i+1)$st domain, $K^{(i,i+1)}$ enables one to compute the Neumann conditions on those surfaces, and $S^{(i,i+1)}$ allows one to calculate the Dirichlet condition of the overlapping surface.

We need $\mathcal{G}_u^{(0)}$ or $\mathcal{G}_v^{(1)}$ to compute $\mathcal{X}^{(0)}$. To achieve this, we can merge the operators for $\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(\ell-1)}$ to get $S^{(1,\ell-1)}$ and $K^{(1,\ell-1)}$, and then merge them with $S^{(0)}$ and $K^{(0)}$ in the final step. In particular, there are two ways of merging the operators:

(a) Starting from the top piece, we form $S^{(\ell-1)}$ and $K^{(\ell-1)}$. We then iterate downwards from $i = \ell - 1$ to $i = 0$, form new operators for each piece, and merge them with the operators from the previous iteration. This is merging in a sequential way.
(b) Since the maps for each piece are independent, we can form the operators on all domains in parallel. Then, we merge in a hierarchical manner, merging two of them simultaneously, and these merging operations are parallelizable.

In summary, this domain decomposition solver is Algorithm 4.

Remark 4.3. *We can also use domain decomposition to construct a parallel solver for the disk domain. The solution operator $S^{(i)} \in \mathbb{C}^{n_1 n_2^{(i)} m \times 2n_1 m}$ on the ith domain takes in the Dirichlet boundary coefficients on the top and the bottom surfaces, and returns the coefficients of $\tilde{U}^{(i)}$. Although we cannot construct $S^{(i)}$ in one setting due to partial regularity, we discover that row $\left((k-1)n_1 n_2^{(i)} + 1\right)$ to row $\left(kn_1 n_2^{(i)}\right)$ of $S^{(i)}$ corresponds to the $(k - 1 - m/2)$th Fourier mode of the solution. In this way, we can construct these rows with the linear system converted from the generalized Sylvester equation. It is then straightforward to construct the DtN map $K^{(i)}$ from $S^{(i)}$, and we can use the hierarchical Poincaré–Steklov method described for the rectangle domain.*

Algorithm 4 Domain decomposition fractional PDE solver on the unit square

Input: The coefficient matrix F of f in Chebyshev basis.
Output: The coefficient matrix W of the solution u in Chebyshev basis.
Convert both columns and rows of F into coefficients in $\tilde{C}^{(3/2)}$ basis.
while $\max_{p,q} \left| \mathcal{X}^{(i)}_{p,q,n_3^{(i)}} \right| > 10^{-10}$ for any i in (4.6) **do**
    Increase $n_3^{(i)}$.
    Form solution and DtN maps on each domain.
    Merge the maps to get $S^{(1,\ell-1)}$ and $K^{(1,\ell-1)}$.
    Merge with $S^{(0)}$ and $K^{(0)}$ to get solution tensor $\mathcal{X}^{(0)}$ on the first domain.
end while.
Calculate $W = \mathcal{X}^{(0)} \times_2 \begin{bmatrix} T_0(-1) & T_1(-1) & \cdots \end{bmatrix}$.
Convert both columns and rows of W into coefficients in Chebyshev basis.

5. Numerical Example and Application to Optimal Control Problems

In this section, we present two examples. In the first example, we extend our 2D solver to 3D to solve the fractional elliptic PDE on $\Omega = (0,1)^3$. The second example is an optimal control problem with a fractional PDE constraint.

5.1. Fractional PDE on the Cube.

We can extend our solver from Section 4 to solve fractional PDEs on the unit cube. Using ultraspherical polynomials for the cube dimensions and Chebyshev polynomials for the extended direction, the discretized problem is the following:

$$\begin{aligned} \mathcal{Y} \times_1 A \times_2 A \times_4 E_1 + \mathcal{Y} \times_2 A \times_3 A \times_4 E_1 + \mathcal{Y} \times_1 A \times_3 A \times_4 E_1 + \mathcal{Y} \times_1 A \times_2 A \times_3 A \times_4 E_2 = 0, \\ \mathcal{Y} \times_1 M \times_2 M \times_3 M \times_4 B_z = -d_s R \mathcal{F}/2, \end{aligned} \tag{5.1}$$

where $\mathcal{Y}$ is formed by stacking 4D tensor solutions of the discretized problem on each interval along the fourth dimension, $\mathcal{F}$ is a 3D tensor representing the polynomial coefficients of the initial condition, and all other matrices have been defined in Section 4.

We solve (5.1) by merging the linear constraint into the tensor equation and using a 4D Bartels–Stewart algorithm analogue to obtain the solution directly. For a numerical example, we consider the simple case that $f = (3\pi^2)^s \sin(\pi x)\sin(\pi y)\sin(\pi z) + (12\pi^2)^s \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$ so that the analytic solution is $u = \sin(\pi x)\sin(\pi y)\sin(\pi z) + \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$. Figure 5 (Left) shows the coefficient decay along the extended direction of the discretized tensor solution for different values of s. This plot shows that the coefficient of the discretized solution is small enough when the algorithm finishes. Figure 5 (Right) shows the accuracy improvements in our adaptive algorithm when we increase the degrees of freedom by allowing higher polynomial degrees to approximate the solution in the extended direction. Similar to the problem on the square, polynomial coefficients of the discretized solution decay exponentially along the extended direction, which allows us to find an accurate numerical solution with only a few degrees of freedom.

5.2. Optimal Control Problem.

We consider the optimization problem given by

$$\begin{aligned} &\min_{(u,q)} \left\{ J(u,q) := \frac{1}{2}\int_\Omega |u - u_d|^2 + \frac{\alpha}{2}\int_\Omega |q|^2 \right\}, \\ &\text{subject to} \qquad (-\Delta)^s u = q \quad \text{in } \Omega, \end{aligned} \tag{5.2}$$

where u_d is a given function and α is the control penalty parameter. We solve this problem via a direct solver. Specifically, we express the optimality condition as two fractional PDEs, discretize

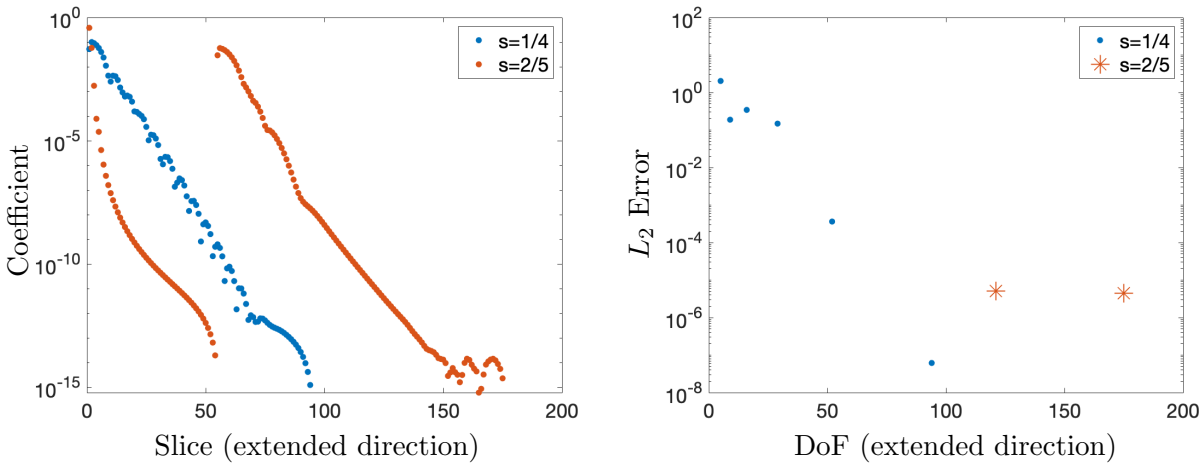

FIGURE 5. Solving the fractional PDE for $f = (3\pi^2)^s \sin(\pi x)\sin(\pi y)\sin(\pi z) + (12\pi^2)^s \sin(2\pi x)\sin(2\pi y)\sin(2\pi z)$ and different values of s on $\Omega = (-1,1)^3$ with our spectral solver. **Left:** the largest coefficient in magnitude on each slice along the extended direction in the discretized solution, i.e. the largest coefficient when the fourth index of the discretized tensor solution is fixed. When $s = 2/5$, $z^{(1/s)}$ is approximated by a piecewise polynomial with two pieces. In this case, the coefficient patterns show the decay of both pieces. **Right:** the accuracy achieved with different degrees of freedom in the extended direction. When $s = 1/4$, the algorithm performs five iterations, increasing the degrees of freedom along the extended direction to achieve sufficient decay of the coefficients. On the other hand, when $s = 2/5$, the solution admits sufficient coefficient decay in the first iteration because the adaptive solver adds additional degrees of freedom.

both, combine them into one tensor equation and obtain the best u and z directly. In particular, we solve

$$\begin{aligned} (-\Delta)^s u &= -\frac{1}{\alpha} p, \\ (-\Delta)^s p &= u - u_d, \end{aligned} \tag{5.3}$$

where we have eliminated the so-called gradient equation. For simplicity, we consider solving (5.3) directly on the unit square. Let $\mathcal{U}$ and $\mathcal{P}$ be the coefficient tensors for the extensions of u and p, respectively. Then, $\mathcal{U}$ and $\mathcal{P}$ satisfy the following tensor equations:

$$\begin{aligned} \mathcal{U} \times_1 A \times_3 E_1^{(u)} + \mathcal{U} \times_2 A \times_3 E_1^{(u)} + \mathcal{U} \times_1 A \times_2 A \times_3 E_2^{(u)} &= 0, \\ \mathcal{U} \times_3 B_y^{(u)} &= \mathcal{G}^{(u)}, \\ \mathcal{P} \times_1 A \times_3 E_1^{(p)} + \mathcal{P} \times_2 A \times_3 E_1^{(p)} + \mathcal{U} \times_1 A \times_2 A \times_3 E_2^{(p)} &= 0, \\ \mathcal{P} \times_3 B_y^{(p)} &= \mathcal{G}^{(p)}, \end{aligned} \tag{5.4}$$

where the first frontal slice of $\mathcal{G}^{(u)}$ is $\frac{d_s z_1}{2\alpha} \mathcal{P} \times_3 \begin{bmatrix} T_0(-1) & \dots & T_{n_3^{(p)}}(-1) & 0 & \dots & 0 \end{bmatrix}$, the first frontal slice of $\mathcal{G}^{(p)}$ is $-\frac{d_s z_1}{2} \left(\mathcal{U} \times_3 \begin{bmatrix} T_0(-1) & \dots & T_{n_3^{(u)}}(-1) & 0 & \dots & 0 \end{bmatrix} - U_d \right)$, and U_d is the coefficient matrix of u_d. To distinguish between matrices used in the equations for u and p, we use superscripts (u) and (p). The matrices E_1, E_2 and B_y are defined in Section 4. We can then rearrange (5.4) so

that we only have one tensor Sylvester equation and a linear constraint:

$$\mathcal{Y} \times_1 A \times_3 \begin{bmatrix} E_1^{(u)} & \\ & E_1^{(p)} \end{bmatrix} + \mathcal{Y} \times_2 A \times_3 \begin{bmatrix} E_1^{(u)} & \\ & E_1^{(p)} \end{bmatrix} + \mathcal{Y} \times_1 A \times_2 A \times_3 \begin{bmatrix} E_2^{(u)} & \\ & E_2^{(p)} \end{bmatrix} = 0, \quad \mathcal{Y} \times_3 B_y = \mathcal{G}, \tag{5.5}$$

where $\mathcal{Y}$ is formed by stacking U and P along the tube direction, $\mathcal{G}$ is a zero tensor except for the last frontal slice of $\frac{1}{2} d_s z_1 U_d$,

$$B_y = \begin{bmatrix} B_y^{(u)}(1) & & -\frac{d_s z_1}{2\alpha} \begin{bmatrix} T_0(-1) & \dots & T_{n_3^{(p)}}(-1) & 0 & \dots & 0 \end{bmatrix} \\ B_y^{(u)}(2:\text{end}) & & \\ & B_y^{(p)}(2:\text{end}) & \\ \frac{d_s z_1}{2} \begin{bmatrix} T_0(-1) & \dots & T_{n_3^{(u)}}(-1) & 0 & \dots & 0 \end{bmatrix} & B_y^{(p)}(1) & \end{bmatrix},$$

and $B_y^{(u)}(i)$ is the ith row of $B_y^{(u)}$. We can then solve for $\mathcal{Y}$, get $\mathcal{U}$ and $\mathcal{P}$, and calculate the coefficient tensor $\mathcal{Q}$ of q by $\mathcal{Q} = -\frac{1}{\alpha}\mathcal{P}$.

For numerical demonstration, we take $\Omega = (-1,1)^2$, $\alpha = 10^{-2}$, and $u_d = (1+\alpha(2\pi^2)^{2s})\sin(\pi x)\sin(\pi y)$. The analytic solution of (5.2) with this data is $u = \sin(\pi x)\sin(\pi y)$ and $q = (2\pi^2)^s \sin(\pi x)\sin(\pi y)$. Figure 6 shows the performance of our adaptive direct solver. As demonstrated, our method generates accurate numerical solutions for both u and q with only a few degrees of freedom in the extended direction.

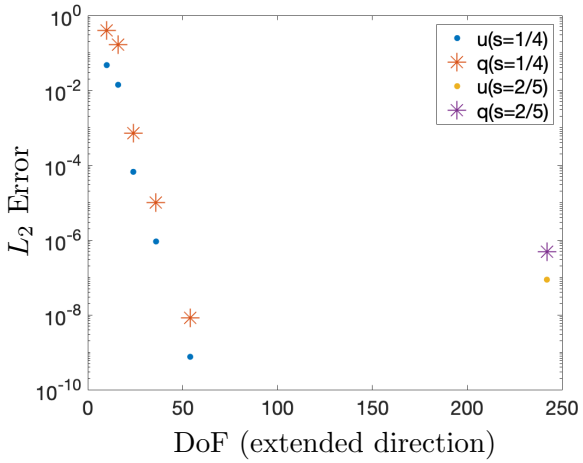

FIGURE 6. Solving the optimal control problem (5.2) with different values of s for $\alpha = 10^{-2}$, $u_d = (1+\alpha(2\pi^2)^{2s})\sin(\pi x)\sin(\pi y)$ and $\Omega = (-1,1)^2$. As the degrees of freedom in the extended direction increase, the approximation of both u and q improves. When $s = 1/4$, we perform four iterations, which add more degrees of freedom along the extended direction, before our algorithm terminates. When $s = 2/5$, our first trial guarantees enough decay in the coefficients for both u and q, resulting in a single point for each on the plot.

6. Conclusions

In this paper, we present a spectral method that uses ultraspherical and Fourier polynomials to solve fractional Laplacian equations on square and disk domains via the Caffarelli–Silvestre extension. Based on the value of the fractional exponent s, we decompose the PDE along the extended domain. We show a direct method that finds solutions on all sub-domains through one tensor equation, and a parallelizable domain decomposition solver generated from the hierarchical Poincaré–Steklov method. Numerical tests suggest that coefficients of the solutions decay exponentially along the extended direction, and we can recover accurate discretized solutions with a few degrees of freedom. Our method is easily generalized to problems of higher dimensions, such as solving fractional PDEs on cubes, and it can be used to accurately compute solutions of optimal control problems. For future work, we will develop spectral solvers for fractional operators with variable coefficients (1.2) through the extension scheme, yielding an approach that can be used to solve more general PDEs and optimal control problems as in [27].

Appendix A. Visualization of Matrices in Ultraspherical Discretization

In this appendix, we provide the readers with some details of the matrices we use during discretization in Sections 3 and 4.

- D is a diagonal matrix representing second derivative of $(1-\rho^2)\tilde{C}^{(3/2)}(\rho)$, with diagonal elements $D_{j,j} = -(j(j+3)+2)$.
- M is a symmetric penta-diagonal matrix with 0 super and sub diagonals representing multiplication of $1-\rho^2$ in $\tilde{C}^{(3/2)}$ basis, with elements

$$M_{j,j} = \frac{2(j+1)(j+2)}{(2j+1)(2j+5)},\ M_{j,j+1} = 0,\ M_{j,j+2} = \frac{-1}{(2j+3)(2j+5)}\sqrt{\frac{(j+4)!(2j+3)}{j!(2j+7)}}.$$

- B_1 and B_2 are two banded matrices representing multiplications in different ultraspherical polynomial basis, where the bandwidth is determined by the degree of the polynomial approximation in the respective basis. In addition, B_1 can be shown to be a *Toeplitz-plus-Hankel-plus-rank-1* operator [26], and both B_1 and B_2 satisfy three-term recurrence [30, Chpt.6].
- D_2 represents second derivative of Chebyshev basis, with the form

$$D_2 = \begin{bmatrix} 0 & 0 & 2 & & & \\ & & & 3 & & \\ & & & & 4 & \\ & & & & & \ddots \end{bmatrix}.$$

Here, elements of D_2 on the diagonal and first upper top diagonal are all 0.

- $S_{2,0}$ converts Chebyshev basis to $C^{(2)}$ basis, and can be calculated by $S_{2,0} = S_1 S_0$, where

$$S_0 = \begin{bmatrix} 1 & 0 & -1/2 & & \\ & 1/2 & 0 & -1/2 & \\ & & 1/2 & 0 & \ddots \\ & & & 1/2 & \ddots \\ & & & & \ddots \end{bmatrix}, \qquad S_1 = \begin{bmatrix} 1 & 0 & -1/3 & & \\ & 1/2 & 0 & -1/4 & \\ & & 1/3 & 0 & \ddots \\ & & & 1/4 & \ddots \\ & & & & \ddots \end{bmatrix}.$$

References

[1] M. Ainsworth and C. Glusa. Hybrid finite element–spectral method for the fractional Laplacian: Approximation theory and efficient solver. *SIAM J. Sci. Comput.*, 40(4):A2383–A2405, 2018.

[2] M. Ainsworth and Z. Mao. Analysis and approximation of a fractional Cahn–Hilliard equation. *SIAM J. Numer. Anal.*, 55(4):1689–1718, 2017.

[3] H. Antil and S. Bartels. Spectral Approximation of Fractional PDEs in Image Processing and Phase Field Modeling. *Comput. Methods Appl. Math.*, 17(4):661–678, 2017.

[4] H. Antil, Z. W. Di, and R. Khatri. Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography. *Inverse Problems*, 36(6):064001, may 2020.

[5] H. Antil, J. Pfefferer, and S. Rogovs. Fractional operators with inhomogeneous boundary conditions: Analysis, control, and discretization. *Commun. Math. Sci.*, 16(5):1395–1426, 2018.

[6] I. Babuška and B. Guo. The h, p and hp version of the finite element method; basis theory and applications. *Adv. Eng. Software*, 15(3-4):159–174, 1992.

[7] L. Banjai, J. M. Melenk, R. H. Nochetto, E. Otárola, A. J. Salgado, and C. Schwab. Tensor FEM for spectral fractional diffusion. *Found. Comput. Math.*, 19(4):901–962, 2019.

[8] R. H. Bartels and G. W. Stewart. Solution of the matrix equation AX+ XB= C. *Communications of the ACM*, 15(9):820–826, 1972.

[9] P. Benner, R.-C. Li, and N. Truhar. On the ADI method for Sylvester equations. *J. Comput. Appl. Math.*, 233(4):1035–1045, 2009.

[10] A. Bonito and J. Pasciak. Numerical approximation of fractional powers of elliptic operators. *Math. Comp.*, 84(295):2083–2110, 2015.

[11] L. Caffarelli and L. Silvestre. An extension problem related to the fractional Laplacian. *Comm. Part. Diff. Eqs.*, 32(7-9):1245–1260, 2007.

[12] S. Chen and J. Shen. An efficient and accurate numerical method for the spectral fractional Laplacian equation. *J. Sci. Comput.*, 82(1):Paper No. 17, 25, 2020.

[13] P. Constantin and J. Wu. Behavior of solutions of 2D quasi-geostrophic equations. *SIAM J. Math. Anal.*, 30(5):937–948, 1999.

[14] A. de Pablo, F. Quirós, A. Rodríguez, and J. L. Vázquez. A fractional porous medium equation. *Adv. Math.*, 226(2):1378–1409, 2011.

[15] E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. *Bull. Sci. Math.*, 136(5):521–573, 2012.

[16] T. A. Driscoll, N. Hale, and L. N. Trefethen. Chebfun guide, 2014.

[17] D. Fortunato, N. Hale, and A. Townsend. The ultraspherical spectral element method. *J. Comput. Phys*, 2020.

[18] D. Fortunato and A. Townsend. Fast Poisson solvers for spectral methods. *IMA J. Numer. Anal.*, 40(3):1994–2018, 11 2019.

[19] T. G. Kolda and B. W. Bader. Tensor decompositions and applications. *SIAM Rev.*, 51(3):455–500, 2009.

[20] N. Laskin. Fractional quantum mechanics and Lévy path integrals. *Phys. Let. A*, 268(4-6):298–305, 2000.

[21] A. Lischke, G. Pang, M. Gulian, and et al. What is the fractional Laplacian? A comparative review with new results. *J. Comput. Phys.*, 404:109009, 62, 2020.

[22] P. Martinsson. The hierarchical Poincaré–Steklov (HPS) solver for elliptic PDEs: A tutorial. *arXiv preprint arXiv:1506.01308*, 2015.

[23] D. Meidner, J. Pfefferer, K. Schürholz, and B. Vexler. *hp*-finite elements for fractional diffusion. *SIAM J. Numer. Anal.*, 56(4):2345–2374, 2018.

[24] R. Nochetto, E. Otárola, and A. Salgado. A PDE approach to fractional diffusion in general domains: A priori error analysis. *Found. Comput. Math.*, 15(3):733–791, 2015.
[25] F. W. J. Olver, D. W. Lozier, R. F. Boisvert, and C. W. Clark, editors. *NIST handbook of mathematical functions*. U.S. Department of Commerce, National Institute of Standards and Technology, Washington, DC; Cambridge University Press, Cambridge, 2010. With 1 CD-ROM (Windows, Macintosh and UNIX).
[26] S. Olver and A. Townsend. A fast and well-conditioned spectral method. *SIAM Rev.*, 55(3):462–489, 2013.
[27] B. Schmitt, B. N. Khoromskij, V. Khoromskaia, and V. Schulz. Tensor method for optimal control problems constrained by fractional three-dimensional elliptic operator with variable coefficients. *Numer. Linear Algebra Appl.*, page e2404, 2021.
[28] F. Song, C. Xu, and G. E. Karniadakis. Computing fractional Laplacians on complex-geometry domains: algorithms and simulations. *SIAM J. Sci. Comput.*, 39(4):A1320–A1344, 2017.
[29] P. Stinga and J. Torrea. Extension problem and Harnack's inequality for some fractional operators. *Comm. Part. Diff. Eqs.*, 35(11):2092–2122, 2010.
[30] A. Townsend. *Computing with functions in two dimensions*. PhD thesis, Citeseer, 2014.
[31] A. Townsend and S. Olver. The automatic solution of partial differential equations using a global spectral method. *J. Comput. Phys.*, 299:106–123, 2015.
[32] C. J. Weiss, B. G. van Bloemen Waanders, and H. Antil. Fractional operators applied to geophysical electromagnetics. *Geophys. J. Intern.*, 220(2):1242–1259, 2020.
[33] H. Wilber, A. Townsend, and G. B. Wright. Computing with functions in spherical and polar geometries II. The disk. *SIAM J. Sci. Comput.*, 39(3):C238–C262, 2017.
[34] M. Zayernouri, M. Ainsworth, and G. E. Karniadakis. A unified Petrov-Galerkin spectral method for fractional PDEs. *Comput. Methods Appl. Mech. Engrg.*, 283:1545–1569, 2015.

T. Shi. Center for Applied Mathematics, 657 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853, USA.
Email address: ts777@cornell.edu

H. Antil. The Center for Mathematics and Artificial Intelligence (CMAI) and Department of Mathematical Sciences, George Mason University, Fairfax, VA 22030, USA.
Email address: hantil@gmu.edu

D.P. Kouri. Optimization and Uncertainty Quantification, Sandia National Laboratories, Albuquerque, NM 87185-1320, USA.
Email address: dpkouri@sandia.gov