A SIMPLE SOLVER FOR THE FRACTIONAL LAPLACIAN IN MULTIPLE DIMENSIONS[*]

VICTOR MINDEN[†] AND LEXING YING[‡]

Abstract. We present a simple discretization scheme for the hypersingular integral representation of the fractional Laplace operator and solver for the corresponding fractional Laplacian problem. Through singularity subtraction, we obtain a regularized integrand that is amenable to the trapezoidal rule with equispaced nodes, assuming a high degree of regularity in the underlying function (i.e., $u \in C^6(\mathbf{R}^d)$). The resulting quadrature scheme gives a discrete operator on a regular grid that is translation-invariant and thus can be applied quickly with the fast Fourier transform. For discretizations of problems related to space-fractional diffusion on bounded domains, we observe that the underlying linear system can be efficiently solved via preconditioned Krylov methods with a preconditioner based on the finite-difference (nonfractional) Laplacian. We show numerical results illustrating the error of our simple scheme as well the efficiency of our preconditioning approach, both for the elliptic (steady-state) fractional diffusion problem and the time-dependent problem.

Key words. hypersingular integrals, quadrature, fractional diffusion

AMS subject classifications. 65R20, 35R11, 65N06

DOI. 10.1137/18M1170406

1. Introduction. Fractional powers of the Laplacian operator arise naturally in the study of anomalous diffusion, where the fractional operator plays an analogous role to that of the standard Laplacian for ordinary diffusion (see, e.g., the review articles by Metzler and Klafter [38, 39] and Vázquez [50]). By replacing Brownian motion of particles with Lévy flights [32], whose increments are drawn from the α-stable Lévy distribution for $\alpha \in (0, 2)$, we obtain a fractional diffusion equation (or fractional kinetic equation) in terms of the fractional Laplacian operator of order α [46], defined for sufficiently nice functions $u: \mathbf{R}^d \to \mathbf{R}$ via the Cauchy principal value (p.v.) integral

$$(1) \qquad (-\Delta)^{\alpha/2} u(\mathbf{x}) \equiv \text{p.v.} \int_{\mathbf{R}^d} C_{\alpha,d} \left[\frac{u(\mathbf{x}) - u(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^{d+\alpha}} \right] d\mathbf{y}, \quad \mathbf{x} \in \mathbf{R}^d,$$

with known normalizing constant $C_{\alpha,d}$ [29].

For a bounded domain $\Omega \subset \mathbf{R}^d$ with complement $\Omega^{\mathsf{c}} \equiv \mathbf{R}^d \setminus \Omega$, we consider fractional diffusion with homogeneous extended Dirichlet conditions given in terms of (1) by

$$(2) \qquad \begin{cases} \partial_t u(\mathbf{x}, t) = -(-\Delta)^{\alpha/2} u(\mathbf{x}, t) + f(\mathbf{x}, t), & \mathbf{x} \in \Omega, \quad t > 0, \\ u(\mathbf{x}, t) = 0, & \mathbf{x} \in \Omega^{\mathsf{c}}, \quad t > 0, \\ u(\mathbf{x}, 0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega. \end{cases}$$

[†]Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305. Current address: Center for Computational Biology, Flatiron Institute, Simons Foundation, New York, NY 10017 (victorminden@gmail.com).

[‡]Department of Mathematics and Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305 (lexing@stanford.edu).

Also of interest is the related elliptic problem

$$
(3) \qquad \begin{cases} (-\Delta)^{\alpha/2} u(\mathbf{x}) = f(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \phantom{(-\Delta)^{\alpha/2}} u(\mathbf{x}) = 0, & \mathbf{x} \in \Omega^{\mathrm{c}}. \end{cases}
$$

Somewhat unintuitively, the nonlocality of (1) implies that the solutions of (2) and (3) depend on data prescribed everywhere outside Ω [12, 14, 43], though other definitions of the fractional Laplacian on a bounded domain are also in common use [50]. Further, a more general formulation of fractional diffusion involves augmenting (2) by incorporating fractional time derivatives of Caputo or Riemann–Liouville type. We focus in this work on the case of space-fractional diffusion and do not discuss the discretization of time-fractional differential operators, though the latter is of independent interest [30, 35, 56, 57, 59].

1.1. Contribution. The contribution of this paper is a simple discretization scheme for (2) and (3) on Cartesian grids and an efficient algorithm for solving the resulting linear systems. The discretization generalizes easily to domains that can be represented as occluded Cartesian grids, i.e., domains given by taking a regular grid and removing a subset of grid points and corresponding subdomains to obtain, e.g., an "L"-shaped domain.

Our approach is based on using a Taylor expansion around each point $\mathbf{x}$ to replace the singular integrand in (1) with a sufficiently smooth function of $\mathbf{y}$ on all of $\mathbf{R}^d$ via singularity subtraction. The resulting integral can be easily discretized using the trapezoidal rule on a regular grid of N points, leading to a translation-invariant linear operator that can be applied at a cost of $O(N \log N)$ using the fast Fourier transform (FFT). The resulting discrete linear system approximating (3) can then be efficiently solved using standard Krylov methods. As $\alpha \to 2$, the resulting linear systems can exhibit the ill-conditioning characteristic of discretizations of the Laplacian operator on a regular grid. To circumvent this, we develop an efficient preconditioning strategy based on the fact that our discrete fractional Laplacian operator may be written as the sum of a standard finite-difference Laplacian and another matrix with mostly small entries.

When the solution u to (3) is sufficiently smooth, standard results on convergence of the trapezoidal rule and finite-difference operators imply that the error of our approach for computing the fractional Laplacian at a point goes to zero as $O(h^2)$, where h is the linear spacing between grid points, which we show in section 2. In general, however, the solution to the fractional Laplace problem on bounded domains is only $\lfloor \alpha/2 \rfloor$ times continuously differentiable [44], leading to a natural deterioration of the rate of convergence of our simple approach.

1.2. Related work. A discretization scheme similar to that presented here appears in Pozrikidis [42], though without discussion of accuracy or the importance of windowing for singularity subtraction. Huang and Oberman [26, 27] derive a scheme for the one-dimensional case based on singularity subtraction and finite-difference approximation but do not tackle the multidimensional case (see also Tian and Du [47], Gao et al. [20], and Duo, Van Wyk, and Zhang [13]). Chen et al. [10] consider a multidimensional discretization and fast preconditioners based on multigrid, but their scheme uses the so-called coordinate fractional Laplacian that takes a tensor product of one-dimensional operators and is not equivalent to (1) (see also related finite-difference approaches with different operators [36, 37, 51, 58]).

Other similar work on efficient solution of fractional Laplacian systems using fast preconditioned iterative methods includes Pang and Sun [41] and Wang and

collaborators [17, 18, 19, 52]. While limited to one spatial dimension, this work also exploits the structure of the discrete operator for fast matrix-vector products and preconditioned solves, and the latter line of work includes treatment of time-fractional operators.

Another family of approaches on discretizing the fractional Laplacian operator is based on finite elements [2, 4, 6, 7, 8, 48, 49]. Compared to our scheme, such approaches are typically more amenable to general geometries (as is typical for finite elements) but are also more involved. Other notable schemes for discretizing the fractional Laplacian based on different ideas include work based on the Caffarelli–Silvestre extension [9, 24, 40], spectral approaches [3, 33, 53, 55], and hybrid schemes [5]. General references for fractional Laplacians on bounded domains include, e.g., Ros-Oton [43], D'Elia and Gunzburger [12], Felsinger, Kassmann, and Voigt [14], and Lischke et al. [31].

2. Spatial discretization of the fractional Laplacian. To begin, we outline our scheme for discretization of (1) in the one-dimensional case where the function u vanishes outside of some interval. Following that, we give more details in our discussion of the multidimensional case.

2.1. Singularity subtraction in one dimension. Concretely, consider the task of approximating the principal value integral

$$(4) \qquad (-\Delta)^{\alpha/2} u(x) = \text{p.v.} \int_{-\infty}^{\infty} C_{\alpha,1} \left[\frac{u(x) - u(y)}{|x - y|^{1+\alpha}} \right] dy,$$

where $u(y) = 0$ for $|y| > 1$. For $\alpha > 1$, this integral is hypersingular due to the high-order pole at $x = y$, which generally leads to large inaccuracies when simple quadrature schemes are applied directly to (4). Therefore, we proceed by regularizing the integrand to remove the singularity and obtain an integral for which simple quadratures are accurate.

Assuming that the function u is sufficiently smooth, we may write a Taylor series expansion about the point x to obtain

$$(5) \qquad u(y) = u(x) + \frac{1}{2} u''(x)(y - x)^2 + u_{\text{odd}}(y) + R_4(y),$$

where the smooth remainder $R_4(y) = O(|y - x|^4)$ as $y \to x$. For brevity, we have grouped terms that are odd about x into u_{odd}, as they will not play an explicit role in what follows.

Our regularization strategy is singularity subtraction based on adding and subtracting a calculable integral that matches terms in the Taylor series. Suppose w is a sufficiently smooth windowing function with compact support such that $w(0) = 1$ and $w(y) = w(-y)$. Then we may write

$$(-\Delta)^{\alpha/2} u(x) = C_{\alpha,1} \int_{-\infty}^{\infty} \frac{u(x) - u(y) + w(x - y) \left[\frac{1}{2} u''(x)(x - y)^2 - u_{\text{odd}}(y) \right]}{|x - y|^{1+\alpha}} dy$$

$$- C_{\alpha,1} \int_{-\infty}^{\infty} \frac{w(x - y) \left[\frac{1}{2} u''(x)(x - y)^2 - u_{\text{odd}}(y) \right]}{|x - y|^{1+\alpha}} dy$$

$$(6) \qquad \equiv (\text{I}) + (\text{II}),$$

where we define (I) to be the first integral and (II) to be the second. By construction, (I) is no longer hypersingular, as we see from (5) that the integrand can be equivalently written

$$\frac{[w(x-y)-1]\left[\frac{1}{2}u''(x)(x-y)^2 - u_{\mathrm{odd}}(y)\right] + R_4(y)}{|x-y|^{1+\alpha}}.$$

By our smoothness assumptions on u and w, as $y \to x$ this integrand decays and is continuously differentiable with a second derivative that is integrable. This implies that the standard trapezoidal rule would exhibit second-order convergence when applied to (I); see Cruz-Uribe and Neugebauer [11]. Of course, this requires knowledge of $u''(x)$ and u_{odd} in general, which we do not assume. In the context of discretization of the integral using a uniform grid, however, the situation simplifies.

2.2. The first integral in one dimension. Consider discretizing (I) using the trapezoidal rule on a one-dimensional lattice $\{y_j\}_{j\in\mathbf{Z}} = \{jh\}_{j\in\mathbf{Z}}$, and take $x = y_i$ to be one of the lattice points. Without loss of generality, we may shift the domain such that $x = y_0 = 0$. This discretization yields the second-order accurate approximation

$$\text{(I)} \approx C_{\alpha,1}h \sum_{j\neq 0}\left[\frac{u(0) - u(y_j) + w(y_j)\left[\frac{1}{2}u''(0)(y_j)^2 - u_{\mathrm{odd}}(y_j)\right]}{|y_j|^{1+\alpha}}\right]$$

$$= C_{\alpha,1}h\left[\sum_{j\neq 0}\frac{u(0)}{|y_j|^{1+\alpha}} - \sum_{j\neq 0}\frac{u(y_j)}{|y_j|^{1+\alpha}} + \frac{u''(0)}{2}\sum_{j\neq 0}\frac{w(y_j)}{|y_j|^{\alpha-1}} - \sum_{j\neq 0}\frac{w(y_j)u_{\mathrm{odd}}(y_j)}{|y_j|^{1+\alpha}}\right]$$

$$= C_{\alpha,1}h\left[A_1 u(0) - \sum_{j\neq 0}\frac{u(y_j)}{|y_j|^{1+\alpha}} + A_2 u''(0)\right],$$

where $A_1 = \sum_{j\neq 0}|y_j|^{-(1+\alpha)}$ and $A_2 = \frac{1}{2}\sum_{j\neq 0}w(y_j)|y_j|^{1-\alpha}$ are constants independent of x and j and the last sum in the second line is identically zero due to oddness considerations. We note that the sum remaining on the final line is over a finite range, as u is compactly supported. Since u is assumed to be smooth enough, we replace $u''(0)$ with the finite-difference approximation

$$u''(0) \approx L_{\mathrm{FD}}u(0) \equiv \frac{u(h) - 2u(0) + u(-h)}{h^2},$$

which gives our final approximation for (I),

$$\text{(7)} \qquad \text{(I)} \approx C_{\alpha,1}h\left[A_1 u(0) - \sum_{j\neq 0}\frac{u(y_j)}{|y_j|^{1+\alpha}} + A_2 L_{\mathrm{FD}}u(0)\right].$$

2.3. The second integral in one dimension and final quadrature. Having established a method for approximating the integral (I) in (6), we turn to (II). Again using oddness considerations, we see that the contribution from u_{odd} vanishes such that

$$\text{(II)} = -\frac{C_{\alpha,1}u''(0)}{2}\int_{-\infty}^{\infty}\frac{w(y)}{|y|^{\alpha-1}}\,dy = C_{\alpha,1}hA_3 u''(0),$$

where the constant A_3 given by

$$A_3 = -\frac{1}{2h}\int_{-\infty}^{\infty}\frac{w(y)}{|y|^{\alpha-1}}\,dy$$

is well-defined (since w is compactly supported) and we again take $x = 0$ for convenience. We once again replace the second derivative $u''(0)$ with its finite-difference approximation to obtain (II) $\approx C_{\alpha,1} h A_3 L_{\mathrm{FD}} u(0)$. Combining this with our quadrature for (I) gives our approximation for $(-\Delta)^{\alpha/2} u(0)$,

$$
(-\Delta)^{\alpha/2} u(0) \approx C_{\alpha,1} h \left[A_1 u(0) - \sum_{j \neq 0} \frac{u(y_j)}{|y_j|^{1+\alpha}} + (A_2 + A_3) L_{\mathrm{FD}} u(0) \right],
$$

which applies equally well not only to $x = 0$ but in general to $x = y_i$ for any grid point y_i, i.e.,

$$
(8) \qquad (-\Delta)^{\alpha/2} u(y_i) \approx C_{\alpha,1} h \left[A_1 u(y_i) - \sum_{j \neq i} \frac{u(y_j)}{|y_i - y_j|^{1+\alpha}} + (A_2 + A_3) L_{\mathrm{FD}} u(y_i) \right].
$$

This is our final quadrature for the fractional Laplacian in one dimension.

2.4. Singularity subtraction in higher dimensions. We turn now to the multidimensional integral, i.e., (1) with $d = 2$ or $d = 3$. Once again we will assume that the function u is compactly supported and sufficiently smooth, as we will make explicit. Our basic strategy is the same as in one dimension.

LEMMA 1. *Suppose that $u \in C^k(\mathbf{R}^d)$ and let $w \in C^p(\mathbf{R})$ be a windowing function symmetric about $z = 0$ such that $1 - w(z) = O(|z|^r)$ as $z \to 0$. Let the third-order Taylor approximation of u about the point $\mathbf{x} \in \mathbf{R}^d$ be given in multi-index notation by*

$$
(9) \qquad u(\mathbf{y}) = \sum_{|\beta| \leq 3} \frac{D^\beta u(\mathbf{x})}{\beta!} (\mathbf{y} - \mathbf{x})^\beta + \sum_{|\tilde{\beta}|=4} R_{\tilde{\beta}}(\mathbf{y})(\mathbf{y} - \mathbf{x})^{\tilde{\beta}},
$$

where the remainder is given in explicit form as

$$
R_{\tilde{\beta}}(\mathbf{y}) \equiv \frac{|\tilde{\beta}|}{\tilde{\beta}!} \int_0^1 (1-t)^{|\tilde{\beta}|-1} D^{\tilde{\beta}} u(\mathbf{x} + t(\mathbf{y} - \mathbf{x})) \, dt.
$$

Then, defining the function

$$
(10) \qquad \tilde{u}(\mathbf{y}) \equiv u(\mathbf{y}) - u(\mathbf{x}) - w(|\mathbf{x} - \mathbf{y}|) \sum_{1 \leq |\beta| \leq 3} \frac{D^\beta u(\mathbf{x})}{\beta!} (\mathbf{y} - \mathbf{x})^\beta,
$$

we have that $\tilde{u} \in C^s(\mathbf{R}^d)$ and $D^\beta \tilde{u}(\mathbf{y}) = O(|\mathbf{y}-\mathbf{x}|^{t-|\beta|})$ as $\mathbf{y} \to \mathbf{x}$ for $s = \min(k-4, p)$, $t = \min(1+r, 4)$, and $0 \leq |\beta| \leq \min(s, t)$.

Proof. It is clear that

$$
\tilde{u}(\mathbf{y}) = (1 - w(|\mathbf{x} - \mathbf{y}|)) \sum_{1 \leq |\beta| \leq 3} \frac{D^\beta u(\mathbf{x})}{\beta!} (\mathbf{y} - \mathbf{x})^\beta + \sum_{|\tilde{\beta}|=4} R_{\tilde{\beta}}(\mathbf{y})(\mathbf{y} - \mathbf{x})^{\tilde{\beta}}.
$$

By inspection, the order of differentiability of $\tilde{u}(\mathbf{y})$ is limited by that of $w(|\mathbf{x} - \mathbf{y}|)$ and of $R_{\tilde{\beta}}(\mathbf{y})$. Given the explicit form of $R_{\tilde{\beta}}(\mathbf{y})$, it is at least in $C^{k-4}(\mathbf{R}^d)$ as a function of $\mathbf{y}$, whereas $w \in C^p(\mathbf{R})$ by assumption. Further, $\tilde{u}(\mathbf{y}) = O(|\mathbf{y} - \mathbf{x}|^t)$ for $t = \min(1+r, 4)$, since the first summand is $O(|\mathbf{y} - \mathbf{x}|^{1+r})$ and the second summand is at least $O(|\mathbf{y} - \mathbf{x}|^4)$. Explicit term-by-term differentiation of $\tilde{u}(\mathbf{y})$ with the product rule concludes the proof. $\square$

By subtracting off the windowed multivariate Taylor series we obtain an integral that is no longer hypersingular. In particular, we write

$$(-\Delta)^{\alpha/2} u(\mathbf{x}) = C_{\alpha,d} \int_{\mathbf{R}^d} \frac{u(\mathbf{x}) - u(\mathbf{y}) + w(|\mathbf{x} - \mathbf{y}|) \sum_{1 \le |\beta| \le 3} \frac{D^\beta u(\mathbf{x})}{\beta!} (\mathbf{y} - \mathbf{x})^\beta}{|\mathbf{x} - \mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

$$- C_{\alpha,d} \int_{\mathbf{R}^d} \frac{w(|\mathbf{x} - \mathbf{y}|) \sum_{1 \le |\beta| \le 3} \frac{D^\beta u(\mathbf{x})}{\beta!} (\mathbf{y} - \mathbf{x})^\beta}{|\mathbf{x} - \mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

$$(11) \qquad \equiv (\mathrm{I}d) + (\mathrm{II}d),$$

where we define (Id) to be the first integral and (IId) to be the second.

2.5. The first integral in higher dimensions. To numerically approximate (Id) we use a quadrature rule on a uniform lattice $\{\mathbf{y_j}\}_{\mathbf{j} \in \mathbf{Z}^d} = \{\mathbf{j}h\}_{\mathbf{j} \in \mathbf{Z}^d}$. We assume the lattice is constructed such that the point $\mathbf{x}$ coincides with with some lattice point $\mathbf{y_i}$, which we take to be $\mathbf{x} = \mathbf{y_0} = \mathbf{0}$ without loss of generality.

Replacing the integral with a weighted sum over the lattice, we obtain

$$(\mathrm{I}d) \approx C_{\alpha,d} h^d \sum_{\mathbf{j} \ne \mathbf{0}} \frac{u(\mathbf{y_i}) - u(\mathbf{y_j}) + w(|\mathbf{y_i} - \mathbf{y_j}|) \sum_{1 \le |\beta| \le 3} \frac{D^\beta u(\mathbf{y_i})}{\beta!} (\mathbf{y_j} - \mathbf{y_i})^\beta}{|\mathbf{y_i} - \mathbf{y_j}|^{d+\alpha}}$$

$$= C_{\alpha,d} h^d \sum_{\mathbf{j} \ne \mathbf{0}} \frac{u(\mathbf{0}) - u(\mathbf{y_j}) + w(|\mathbf{y_j}|) \sum_{1 \le |\beta| \le 3} \frac{D^\beta u(\mathbf{0})}{\beta!} (\mathbf{y_j})^\beta}{|\mathbf{y_j}|^{d+\alpha}},$$

which we note does not include a term for $\mathbf{j} = \mathbf{0}$. This corresponds to the standard trapezoidal rule for $d = 2$ and the punctured trapezoidal rule for $d = 3$, though more involved quadrature corrections may be used (see, e.g., Marin, Runborg, and Tornberg [34]). Assuming w is symmetric about the origin, we see that for many values of the multi-index β the corresponding summand vanishes due to oddness considerations. Taking these symmetries into account, we let $\mathbf{e}_1^T \mathbf{y_j}$ denote the first coordinate of $\mathbf{y_j}$ and observe that

$$\sum_{\mathbf{j} \ne \mathbf{0}} \sum_{1 \le |\beta| \le 3} \frac{w(|\mathbf{y_j}|) \frac{D^\beta u(\mathbf{0})}{\beta!} (\mathbf{y_j})^\beta}{|\mathbf{y_j}|^{d+\alpha}} = \frac{\Delta u(\mathbf{0})}{2} \sum_{\mathbf{j} \ne \mathbf{0}} \frac{w(|\mathbf{y_j}|)(\mathbf{e}_1^T \mathbf{y_j})^2}{|\mathbf{y_j}|^{d+\alpha}},$$

which we plug back into our quadrature scheme to obtain

$$(\mathrm{I}d) \approx C_{\alpha,d} h^d \sum_{\mathbf{j} \ne \mathbf{0}} \frac{u(\mathbf{0}) - u(\mathbf{y_j}) + \frac{\Delta u(\mathbf{0})}{2} w(|\mathbf{y_j}|)(\mathbf{e}_1^T \mathbf{y_j})^2}{|\mathbf{y_j}|^{d+\alpha}}$$

$$= C_{\alpha,d} h^d \left[\left(\sum_{\mathbf{j} \ne \mathbf{0}} \frac{1}{|\mathbf{y_j}|^{d+\alpha}} \right) u(\mathbf{0}) - \sum_{\mathbf{j} \ne \mathbf{0}} \frac{u(\mathbf{y_j})}{|\mathbf{y_j}|^{d+\alpha}} + \left(\frac{1}{2} \sum_{\mathbf{j} \ne \mathbf{0}} \frac{w(|\mathbf{y_j}|)(\mathbf{e}_1^T \mathbf{y_j})^2}{|\mathbf{y_j}|^{d+\alpha}} \right) \Delta u(\mathbf{0}) \right]$$

$$\equiv C_{\alpha,d} h^d \left[A_{1,d} u(\mathbf{0}) - \sum_{\mathbf{j} \ne \mathbf{0}} \frac{u(\mathbf{y_j})}{|\mathbf{y_j}|^{d+\alpha}} + A_{2,d} \Delta u(\mathbf{0}) \right]$$

with correspondingly defined constants

$$(12) \qquad A_{1,d} \equiv \left(\sum_{\mathbf{j} \neq \mathbf{0}} \frac{1}{|\mathbf{y_j}|^{d+\alpha}} \right), \quad A_{2,d} \equiv \left(\frac{1}{2} \sum_{\mathbf{j} \neq \mathbf{0}} \frac{w(|\mathbf{y_j}|)(\mathbf{e}_1^T \mathbf{y_j})^2}{|\mathbf{y_j}|^{d+\alpha}} \right).$$

THEOREM 2. *Suppose the same setup as Lemma 1 with $k = 6$, $p = 3$, and $r = 3$ such that $t = 4$ and $s = 2$. Assume further u and w are compactly supported with $0 \leq w(z) \leq 1$ for all z. Then the above approximation for (Id) is second-order accurate. That is,*

$$C_{\alpha,d} \int_{\mathbf{R}^d} \frac{u(\mathbf{0}) - u(\mathbf{y}) + w(|\mathbf{y}|) \sum_{1 \leq |\beta| \leq 3} \frac{D^\beta u(\mathbf{0})}{\beta!}(\mathbf{y})^\beta}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

$$= C_{\alpha,d} h^d \left[A_{1,d} u(\mathbf{0}) - \sum_{\mathbf{j} \neq \mathbf{0}} \frac{u(\mathbf{y_j})}{|\mathbf{y_j}|^{d+\alpha}} + A_{2,d} \Delta u(\mathbf{0}) \right] + O(h^2)$$

with $A_{1,d}$ and $A_{2,d}$ as in (12).

Proof. The described approximation is numerically equivalent to the (punctured) trapezoidal rule, so this amounts to bounding the error of the trapezoidal rule applied in d dimensions with integrand $\tilde{u}(\mathbf{y})/|\mathbf{y}|^{d+\alpha}$, where $\tilde{u}(\mathbf{y})$ is as in Lemma 1 with $\mathbf{x} = \mathbf{0}$. Letting $R > h$ be such that both $u(\mathbf{y}) = 0$ and $w(|\mathbf{y}|) = 0$ for $|\mathbf{y}| > R$, we proceed by breaking the integral into three contributions: one for the subdomain $B_h \equiv [-h, h]^d$ "near" the singularity, one for the "midrange" subdomain $B_R \backslash B_h \equiv [-R, R]^d \backslash [-h, h]^d$, and one for the "far" subdomain $\mathbf{R}^d \backslash B_R$. We write

$$\int_{\mathbf{R}^d} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} = \int_{B_h} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} + \int_{B_R \backslash B_h} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} + \int_{\mathbf{R}^d \backslash B_R} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y},$$

each piece of which we analyze separately.

Near the singularity, we see due to symmetry considerations that

$$\int_{B_h} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} = \sum_{1 \leq |\beta| \leq 3} \frac{D^\beta u(\mathbf{0})}{\beta!} \int_{B_h} \frac{(1 - w(|\mathbf{y}|))(\mathbf{y})^\beta}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} + \sum_{|\tilde{\beta}|=4} \int_{B_h} \frac{R_{\tilde{\beta}}(\mathbf{y})(\mathbf{y})^{\tilde{\beta}}}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

$$= \frac{\Delta u(\mathbf{0})}{2} \int_{B_h} \frac{(1 - w(|\mathbf{y}|)) \left(\mathbf{e}_1^T \mathbf{y} \right)^2}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} + \sum_{|\tilde{\beta}|=4} \int_{B_h} \frac{R_{\tilde{\beta}}(\mathbf{y})(\mathbf{y})^{\tilde{\beta}}}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

$$= O\left(h^{4-\alpha} \right),$$

where under our assumptions the integrands are both $O(h^{4-d-\alpha})$ and Δu is bounded. Since $\tilde{u}(\mathbf{y}) = O(|\mathbf{y}|^t)$, we see $\frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} = O(|\mathbf{y}|^{t-d-\alpha})$, which implies that the corresponding (punctured) trapezoidal rule approximation to the integral is $O(h^{t-\alpha})$, since we gain a factor of h^d due to the quadrature weights. Therefore, the contribution to the error from the integral over the near subdomain is $O(h^{4-\alpha}) = O(h^2)$, since $\alpha \in (0, 2)$.

In the midrange subdomain, we explicitly use the composite nature of the trapezoidal rule to write

$$\int_{B_R \backslash B_h} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} = \sum_\ell \int_{\Omega_\ell} \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

and then consider the error of the trapezoidal rule in approximating the integral over each Ω_ℓ separately, where the square/cubic subdomains $\{\Omega_\ell\}$ in the trapezoidal rule

are pairwise disjoint and are such that $\bigcup_\ell \Omega_\ell = B_R \setminus B_h$. Since we are away from the origin, on each subdomain Ω_ℓ the integrand $\phi(\mathbf{y}) \equiv \frac{\tilde{u}(\mathbf{y})}{|\mathbf{y}|^{d+\alpha}}$ is in $C^2(\Omega_\ell)$ which means the standard error bound for the trapezoidal rule on Ω_ℓ gives an error contribution of no more than $Ch^{d+2} \sum_{|\beta|=2} \|D^\beta \phi\|_{L_\infty(\Omega_\ell)}$ for some constant C independent of h. However, the term $\|D^\beta \phi\|_{L_\infty(\Omega_\ell)}$ does depend on h. Since $D^\beta \tilde{u}(\mathbf{y}) = O(|\mathbf{y}|^{t-|\beta|})$ from Lemma 1, the product rule gives $D^\beta \phi(\mathbf{y}) = O(1 + |\mathbf{y}|^{t-|\beta|-d-\alpha})$. With this we can bound the total error on $\mathbf{R}^d \setminus B_h$ as

$$\sum_\ell Ch^{d+2} \sum_{|\beta|=2} \|D^\beta \phi\|_{L_\infty(\Omega_\ell)} \leq C' h^{d+2} \sum_\ell \|1 + |\mathbf{y}|^{t-2-d-\alpha}\|_{L_\infty(\Omega_\ell)}$$

$$\leq C'' h^2 \left(1 + \int_0^R r^{t-3-\alpha}\, dr\right) = C''' h^2,$$

where we have bounded

$$h^d \sum_\ell \|1 + |\mathbf{y}|^{t-2-d-\alpha}\|_{L_\infty(\Omega_\ell)} \leq c \int_{B_R} \left(1 + |\mathbf{y}|^{t-2-d-\alpha}\right)\, d\mathbf{y} + c'$$

(up to some geometry-dependent factors that are independent of h) due to concavity of the summand. Therefore, the error contribution from the midrange subdomain is $O(h^2)$.

Finally, for the far subdomain, we observe that the integrand is in $C^2(\mathbf{R}^d \setminus B_R)$ and its smoothness is independent of h in this region, so the standard composite trapezoidal error bound of $O(h^2)$ applies. Therefore, the overall error is $O(h^2)$. $\quad\square$

Remark. Being based on singularity subtraction via Taylor series expansion, the theoretical results in Lemma 1 and Theorem 2 apply directly only for relatively smooth functions u. As discussed, however, it is known that in the general case solutions to (3) exhibit only mild Hölder regularity on the whole space but typically better regularity on Ω (i.e., $u \in C^{0,\alpha/2}(\mathbf{R}^d)$, but u is more regular than f on Ω) [44]. This lack of regularity across the boundary of Ω substantially complicates error analysis of any translation-invariant numerical approach such as is presented here.

While smoothness is not generally a property of solutions to (3), examples can be concocted. For example, inside the unit ball $B \equiv \{\mathbf{x} \mid |\mathbf{x}|^2 \leq 1\} \subset \mathbf{R}^d$ one family of smooth solutions is given by observing that for $q > 0$ and $s \in (0,1)$ we have

$$(-\Delta)^{-s}\left[(1 - |\mathbf{x}|^2)_+^q\right] = K \times {}_2F_1\left(\frac{d}{2} - s, -q - s; \frac{d}{2}; |\mathbf{x}|^2\right),\ |\mathbf{x}| \leq 1,$$

for known constant K [25, equation 9], where ${}_2F_1$ is the Gauss hypergeometric function [1]. Applying the negative Laplacian to either side and letting $s = 1 - \alpha/2$ we see

$$(-\Delta)^{\alpha/2}\left[(1 - |\mathbf{x}|^2)_+^q\right] = -\Delta\left[K \times {}_2F_1\left(\frac{d+\alpha}{2} - 1, \frac{\alpha}{2} - q - 1; \frac{d}{2}; |\mathbf{x}|^2\right)\right],\ |\mathbf{x}| \leq 1.$$

This gives a family of smooth solutions to (3) on B, and related formulas can be used to obtain $(-\Delta)^{\alpha/2}\left[(1 - |\mathbf{x}|^2)_+^q\right]$ for $|\mathbf{x}| > 1$ (and thus to extend the problem domain beyond B). Beyond such examples, the theoretical accuracy of Theorem 2 is chiefly useful when studying the fractional Laplacian *forward operator* applied to smooth functions. That said, in section 4 we empirically observe linear convergence of the solution to (3) for $\alpha > 1$.

2.6. The second integral in higher dimensions and final quadrature. We now consider the second integral (IId) in (11). Assuming without loss of generality that $\mathbf{x} = 0$ and using symmetry and oddness considerations as before, we see that

$$(\text{IId}) = -C_{\alpha,d} \int_{\mathbf{R}^d} \sum_{1 \leq |\beta| \leq 3} \frac{w(|\mathbf{y}|) \frac{D^\beta u(\mathbf{0})}{\beta!}(\mathbf{y})^\beta}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y} = -\frac{C_{\alpha,d} \Delta u(0)}{2} \int_{\mathbf{R}^d} \frac{w(|\mathbf{y}|) \left(\mathbf{e}_1^T \mathbf{y}\right)^2}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y}.$$

Defining the constant

$$(13) \qquad\qquad A_{3,d} \equiv -\frac{h^{-d}}{2} \int_{\mathbf{R}^d} \frac{w(|\mathbf{y}|) \left(\mathbf{e}_1^T \mathbf{y}\right)^2}{|\mathbf{y}|^{d+\alpha}} \, d\mathbf{y}$$

and combining this with our quadrature for (Id) gives

$$(-\Delta)^{\alpha/2} u(\mathbf{0}) \approx C_{\alpha,d} h^d \left[A_{1,d} u(\mathbf{0}) - \sum_{\mathbf{j} \neq \mathbf{0}} \frac{u(\mathbf{y_j})}{|\mathbf{y_j}|^{d+\alpha}} + (A_{2,d} + A_{3,d}) \Delta u(\mathbf{0}) \right]$$

or, more generally,

$$(-\Delta)^{\alpha/2} u(\mathbf{y_i}) \approx C_{\alpha,d} h^d \left[A_{1,d} u(\mathbf{y_i}) - \sum_{\mathbf{j} \neq \mathbf{i}} \frac{u(\mathbf{y_j})}{|\mathbf{y_i} - \mathbf{y_j}|^{d+\alpha}} + (A_{2,d} + A_{3,d}) \Delta u(\mathbf{y_i}) \right].$$

Of course, as written this approximation requires second derivative information in the form of $\Delta u(\mathbf{y_i})$. For smooth u, however, we may replace this with a finite-difference stencil involving the neighbors of $\mathbf{y}_i$ in the lattice,

$$\Delta u(\mathbf{y_i}) \approx L_{\text{FD}} u(\mathbf{y_i}) \equiv \frac{1}{h^2} \left(\sum_{\|\mathbf{i}-\mathbf{j}\|_1 = 1} u(\mathbf{y_j}) - 2^d u(\mathbf{y_i}) \right),$$

just as in the one-dimensional case.

2.7. Summary of quadrature for fractional Laplacian. We briefly summarize our complete approach for discretizing the fractional Laplacian applied to a function u. First, we regularize the integrand of (1) by adding to the numerator a windowed Taylor series approximation of u about $\mathbf{x}$ with window function w to obtain (Id) in (11). This gives an integral that is nice enough to admit discretization with the trapezoidal rule or related schemes. Then, by exploiting symmetries of the problem, we rewrite the discretization in terms of the constants $A_{1,d}$ and $A_{2,d}$ in (12), which do not depend on u. Finally, we derive an expression for the correction term (IId) in terms of another constant $A_{3,d}$ given in (13), which when combined with (Id) and a finite-difference stencil approximation gives a nice expression for $(-\Delta)^{\alpha/2} u(\mathbf{y_i})$ as a linear function of u evaluated on a regular grid.

A few details of the procedure remain to be discussed. First, there are a number of possibilities for the windowing function w. In this paper, we use the piecewise-polynomial window

$$(14) \qquad w(r) = W_\delta(r) \equiv \begin{cases} 1 - 35 \left(\frac{r}{\delta}\right)^4 + 84 \left(\frac{r}{\delta}\right)^5 - 70 \left(\frac{r}{\delta}\right)^6 + 20 \left(\frac{r}{\delta}\right)^7, & r < \delta, \\ 0 & \text{else.} \end{cases}$$

Of course, this is by no means the only sufficiently smooth choice. Further, we note that the requirement that w be compactly supported can be relaxed so long as W decays sufficiently quickly as $r \to \infty$ such that the necessary integrals and sums may be computed.

On that note, we also must still compute the constants $A_{1,d}$, $A_{2,d}$, and $A_{3,d}$. For our choice of polynomial window, the integral defining $A_{3,d}$ can be computed explicitly; for other choices the integral may be numerically computed to high precision offline using, e.g., adaptive quadrature in MATLAB. For compactly supported w, the sum defining $A_{2,d}$ has a finite number of nonzero terms and is easily computable. Finally, the infinite lattice sum $A_{1,d}$ is given in terms of the Riemann zeta function for $d = 1$ and may otherwise be well-approximated using far-field compression techniques related to the fast multipole method (FMM) [22, 54]. We use Chebyshev polynomials for far-field compression in the vein of Fong and Darve [16], though we do not require the full FMM machinery as we are interested only in the lattice sum and not a full approximate operator.

We remark that the analysis of this section gives a bound for the "apply error" when the approximate operator is applied to an appropriately smooth function. While we use standard regularity assumptions to prove convergence of the finite-difference quadrature approximation, such regularity does not hold in general for solutions to (3), particularly near the boundary $\partial\Omega$ [45]. Thus, these results do not apply directly to the "solve error" (error in approximating u), and in practice we expect lower rates of convergence for the solve error, as we explore numerically in section 4.

3. Solving the fractional differential equations on a bounded domain. Having developed our trapezoidal rule scheme for evaluating (1) given u, we turn now to the fractional differential equations (2) and (3) concerning fractional diffusion on a bounded domain Ω with homogeneous extended Dirichlet conditions. We focus on the case $\Omega = [0, 1]^d$ for ease of exposition.

3.1. The elliptic case: Steady-state fractional diffusion. To solve the elliptic problem (3), we discretize Ω using a regular grid of $N = (n - 1)^d$ points $\{\mathbf{y_j}\}$ with linear spacing $h = \frac{1}{n+1}$, where $\mathbf{j} = (j_1, \ldots, j_d)$ and $\mathbf{y_j} = h\mathbf{j}$. For notational convenience, we define the index set $\mathcal{J} \equiv [n]^d \subset \mathbf{Z}^d$. Then, replacing the fractional Laplacian with our quadrature-based approximation gives

$$(15) \quad C_{\alpha,d} h^d \left[A_{1,d} u_\mathbf{i} - \sum_{\substack{\mathbf{j} \in \mathcal{J} \\ \mathbf{j} \neq \{\mathbf{i}\}}} \frac{u_\mathbf{j}}{|\mathbf{y_i} - \mathbf{y_j}|^{d+\alpha}} + (A_{2,d} + A_{3,d}) L_{\mathrm{FD}} u_\mathbf{i} \right] = f(\mathbf{y_i}) \quad \forall \mathbf{i} \in \mathcal{J},$$

which is a linear system to be solved for the variables $\{u_\mathbf{j}\} \approx \{u(\mathbf{y_j})\}$. We remark that the "boundary conditions" affect the system in two ways. First, the center sum has been reduced from an infinite number of terms (in general) to a more manageable finite sum. Second, evaluating the finite-difference stencil L_{FD} for $\mathbf{i}$ near the boundary of the domain will require the prescribed value of $u(\mathbf{y})$ on the boundary, as in the standard (nonfractional) case.

We write (15) in matrix form as

$$(16) \qquad\qquad\qquad\qquad \mathsf{M}\mathbf{u} = \mathbf{f},$$

where now $\mathbf{u} \in \mathbf{R}^N$ and $\mathbf{f} \in \mathbf{R}^N$ are vectors with corresponding entries $\{u_\mathbf{j}\}$ and $\{f(\mathbf{y_j})\}$ and $\mathsf{M} \in \mathbf{R}^{N \times N}$ contains the coefficients implied by (15).

Forward operator and application with FFT. By construction, the approximate fractional Laplacian operator involved in (15) is translation-invariant, which means that the matrix M is block Toeplitz with Toeplitz blocks (BTTB) under any natural ordering of the unknowns. As is well known, this in turn implies that M may be applied efficiently using the FFT at a cost of $O(N \log N)$ FLOPs per application and stored with storage cost $O(N)$.

Further, investigation of the constants $A_{1,d}$, $A_{2,d}$, and $A_{3,d}$ reveal that M is symmetric positive definite. When coupled with the previous observatiton, this leads naturally to the use of the conjugate gradient method (CG) [23] or related iterative methods for solving (16). However, while the FFT ensures low complexity per iteration, the number of iterations required to achieve a specified iteration can be large unless an effective preconditioner is used. This is of particular concern as $\alpha \to 2$, whereupon we recover the standard (ill-conditioned) Laplacian.

Preconditioning: Laplacian pattern and fast Poisson solver. To construct an efficient preconditioner for (16), we observe that M may be decomposed as the sum of two matrices $\mathsf{M} = C_{\alpha,d} h^d (\mathsf{K} + \mathsf{L})$, where

$$
\mathsf{K_{ij}} = \left\{ \begin{array}{ll} -\frac{1}{|\mathbf{y_i} - \mathbf{y_j}|^{d+\alpha}}, & \mathbf{i} \neq \mathbf{j}, \\ A_{1,d}, & \mathbf{i} = \mathbf{j}, \end{array} \right. \quad \text{and} \quad \mathsf{L_{ij}} = \left\{ \begin{array}{ll} \frac{(A_{2,d} + A_{3,d})}{h^2}, & \|\mathbf{i} - \mathbf{j}\|_1 = 1, \\ -\frac{2^d (A_{2,d} + A_{3,d})}{h^2}, & \mathbf{i} = \mathbf{j}, \\ 0 & \text{else}, \end{array} \right.
$$

and we note that $A_{2,d} + A_{3,d} < 0$. The sparse matrix L is (up to a proportionality constant) the typical finite-difference approximation of the negative Laplacian, whereas the matrix K has entries that quickly decay away from $\mathbf{i} = \mathbf{j}$, particularly for larger α. This motivates using L itself as a preconditioner when using CG to solve (16). Because L is effectively a finite-difference discretization of Poisson's equation on a regular grid with homogeneous Dirichlet boundary conditions, application of L^{-1} may be accomplished with the FFT at a cost of $O(N \log N)$ using typical fast Poisson solver techniques [28, Chapter 12]. For nonrectangular domains, the FFT-based approach is no longer feasible, but the same preconditioner can be used with, e.g., nested dissection [21] or related methods.

We remark that other choices of preconditioner are possible. For example, rather than using L^{-1} as our preconditioner we could instead use $\widetilde{\mathsf{M}}^{-1}$, where $\widetilde{\mathsf{M}}_{\mathbf{ij}} = \mathsf{M}_{\mathbf{ij}}$ if $\mathsf{L}_{\mathbf{ij}} \neq 0$ and zero otherwise. Preliminary experiments with this approach (not shown) did not seem to show measurable benefit.

3.2. The time-dependent case: Time-dependent fractional diffusion. We turn now to the full time-dependent problem (2). For spatial discretization we use the approximate fractional Laplacian just as in subsection 3.1, which we combine with a Crank–Nicolson scheme for the discretization of temporal derivatives. This leads to the implicit time-stepping method

$$
(17) \qquad \left(\mathsf{I} + \frac{\Delta t}{2} \mathsf{M} \right) \mathbf{u}^{(k+1)} = \left(\mathsf{I} - \frac{\Delta t}{2} \mathsf{M} \right) \mathbf{u}^{(k)} + \frac{\Delta t}{2} \left(\mathbf{f}^{(k+1)} + \mathbf{f}^{(k)} \right)
$$

to be solved for $\mathbf{u}^{(k+1)} \in \mathbf{R}^N$, where M is as in subsection 3.1 and now $\mathbf{u}^{(k)} \in \mathbf{R}^N$ and $\mathbf{f}^{(k)} \in \mathbf{R}^N$ have entries $\{u_{\mathbf{j}}^{(k)}\} \approx \{u(\mathbf{y_j}, t_k)\}$ and $\{f_{\mathbf{j}}^{(k)}\} = \{f(\mathbf{y_j}, t_k)\}$ for $t_k = k\Delta t$.

Just as in subsection 3.1, we exploit BTTB structure to apply M such that (17) may be solved efficiently with CG at each time step. Compared to the steady-state problem, the system matrix $\left(\mathsf{I} + \frac{\Delta t}{2} \mathsf{M} \right)$ here is much better conditioned due to the

addition of the identity. However, we still find that the number of iterations is reduced substantially via preconditioning, where we use the matrix $I + \frac{\Delta t}{2} C_{\alpha,d} h^d L$ as preconditioner.

4. Numerical results. To demonstrate and profile our approach to discretizing and solving fractional diffusion problems on bounded Cartesian domains, we implemented a number of examples. All computations were performed in MATLAB R2017a on a 64-bit Ubuntu laptop with a dual-core Intel Core i7-7500U processor at 2.70 GHz and 16GB of RAM. All reported timings are in seconds.

4.1. Elliptic examples in one dimension.

A relatively smooth solution. We begin with a one-dimensional elliptic example on the interval $\Omega = [-1, 1]$, discretizing and solving (3) with right-hand side

$$(18) \qquad f(x) = {}_2F_1\left(\frac{1+\alpha}{2}, -2; \frac{1}{2}; x^2\right).$$

In this case, the analytic solution on Ω is known and is given up to a known constant of proportionality K_α by $u(x) = K_\alpha^{-1}(1-x^2)^{2+\frac{\alpha}{2}}$ [25]. We observe that this solution is relatively smooth when extended to $\mathbf{R}$ due to vanishing second derivatives as $x \to \pm 1$.

We discretize the interval Ω with regularly spaced points as in (15) with $d = 1$, choosing δ in (14) as a function of the number of discretization points N, such that w is supported inside a ball with a radius of 20 discretization points. The time t_{con} to construct the discrete operator M is less than 3ms in all cases for the one-dimensional case.

Using the known solution $u(x)$ for right-hand side (RHS) (18), we measure the apply error of our discretization as $e_{\mathrm{app}} \equiv \|M\mathbf{u}_{\mathrm{true}} - \mathbf{f}\|/\|\mathbf{f}\|$, where $\mathbf{u}_{\mathrm{true}}$ is the analytic solution sampled on the discrete grid points and $\mathbf{f}$ is the discretized RHS. To demonstrate the solution error of our discretization scheme we take the same RHS as before and use CG to solve the resulting linear system (16). This gives a discrete solution $\mathbf{u}$ that we can compare to $\mathbf{u}_{\mathrm{true}}$ by computing the relative solution error $e_{\mathrm{sol}} \equiv \|\mathbf{u} - \mathbf{u}_{\mathrm{true}}\|/\|\mathbf{u}_{\mathrm{true}}\|$. These metrics are all shown in Table 1 for four different choices of α, with correponding plots in Figure 1. For convenience, at the bottom of Table 1 we give an estimate of the asymptotic decay rate of the error as N is increased, given by a least-squares fit of the log-error to $\log N$.

We show in Table 2 the runtime t_{CG} and iterations n_{CG} required by CG to solve the linear system (16) for two different choices of relative ℓ_2-norm residual tolerance ϵ_{res}. We give results and approximate rates of runtime growth for both the preconditioned system (where the preconditioner L is a finite-difference Laplacian as described

TABLE 1
Relative apply and solve errors for $\alpha \in \{0.75, 1.25, 1.50, 1.75\}$ for the one-dimensional elliptic example with RHS (18). The last row gives an estimate of the rate of growth as N is increased, i.e., γ in $O(N^\gamma)$.

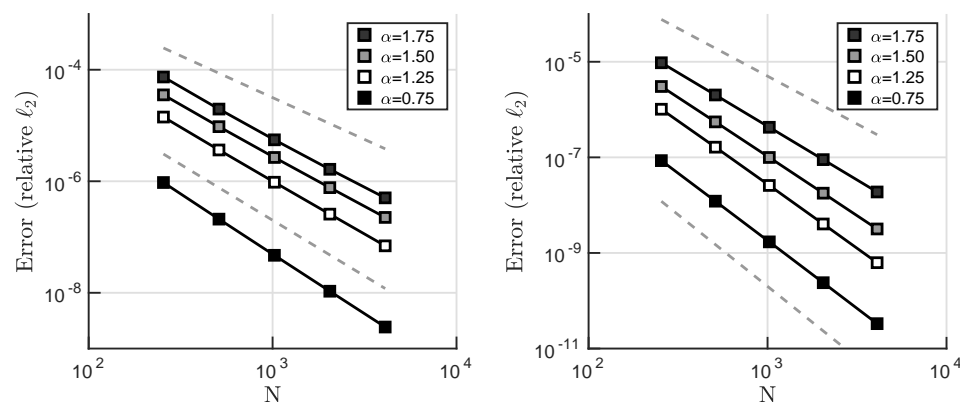| N	$e_{\mathrm{app},0.75}$	$e_{\mathrm{app},1.25}$	$e_{\mathrm{app},1.5}$	$e_{\mathrm{app},1.75}$	$e_{\mathrm{sol},0.75}$	$e_{\mathrm{sol},1.25}$	$e_{\mathrm{sol},1.5}$	$e_{\mathrm{sol},1.75}$
511	2.1e−07	3.6e−06	9.5e−06	2.0e−05	1.2e−08	1.6e−07	5.5e−07	2.0e−06
1023	4.7e−08	9.6e−07	2.7e−06	5.5e−06	1.7e−09	2.6e−08	1.0e−07	4.3e−07
2047	1.1e−08	2.6e−07	7.7e−07	1.6e−06	2.4e−10	4.0e−09	1.8e−08	9.0e−08
4095	2.4e−09	7.0e−08	2.2e−07	5.0e−07	3.3e−11	6.3e−10	3.2e−09	1.9e−08
Rate:	-2.1	-1.9	-1.8	-1.8	-2.8	-2.7	-2.5	-2.2

FIG. 1. *For the one-dimensional example, we plot the relative ℓ_2 apply error e_{app} (left) and solve error e_{sol} (right) as tabulated in Table 1. In each case we see steady convergence, though with differing rates (note the difference in y-axis scale between figures). On the left, the top trend line is $O(N^{-1.5})$ and the bottom is $O(N^{-2})$. On the right, the top trend line is $O(N^{-2})$ and the bottom is $O(N^{-3})$.*

TABLE 2
Runtime t_{CG} and number of iterations n_{CG} required to solve the one-dimensional elliptic example using CG with/without preconditioning based on the finite-difference Laplacian. The parenthesized quantities indicate the corresponding test did not converge within 1000 iterations. We omit results for $\alpha = 0.75$, as for $\alpha < 1$ our preconditioning scheme is unnecessary.

| α | N | $\epsilon_{\text{res}} = 10^{-6}$ | | $\epsilon_{\text{res}} = 10^{-9}$ | |
		t_{CG}	n_{CG}	t_{CG}	n_{CG}
	511	4.8e−3 / 1.8e−2	22 / 104	7.2e−3 / 2.0e−2	35 / 123
	1023	1.4e−2 / 5.7e−2	27 / 162	1.5e−2 / 5.0e−2	44 / 190
1.25	2047	2.6e−2 / 1.5e−1	35 / 251	3.9e−2 / 1.8e−1	57 / 295
	4095	6.3e−2 / 4.3e−1	43 / 389	9.5e−2 / 5.1e−1	72 / 457
	Rate:	1.2 / 1.5	*	1.3 / 1.6	*
	511	3.6e−3 / 2.5e−2	15 / 156	6.4e−3 / 2.9e−2	23 / 174
	1023	8.1e−3 / 6.9e−2	20 / 263	1.2e−2 / 8.8e−2	28 / 294
1.50	2047	1.8e−2 / 2.7e−1	22 / 445	2.9e−2 / 3.1e−1	34 / 497
	4095	3.7e−2 / 8.5e−1	26 / 752	5.7e−2 / 9.7e−1	40 / 839
	Rate:	1.1 / 1.7	*	1.1 / 1.7	*
	511	2.7e−3 / 3.3e−2	11 / 216	3.7e−3 / 3.8e−2	15 / 229
	1023	4.5e−3 / 9.9e−2	12 / 397	7.9e−3 / 1.1e−1	17 / 422
1.75	2047	1.1e−2 / 4.3e−1	13 / 731	1.7e−2 / 4.8e−1	18 / 776
	4095	2.7e−2 / (1e+0)	15 / (1000)	4.2e−2 / (1e+0)	21 / (1000)
	Rate:	1.1 / 1.8	*	1.2 / 1.8	*

in subsection 3.1) and the unpreconditioned system. Because this is a one-dimensional problem, use of a fast Poisson solver to apply L^{-1} is not stricly necessary for efficiency. Instead, we use a sparse Cholesky factorization with negligible overhead. The corresponding timing results are plotted in Figure 2 (left), where we see that our simple preconditioning scheme is effective for reducing the time to solution, especially for larger α.
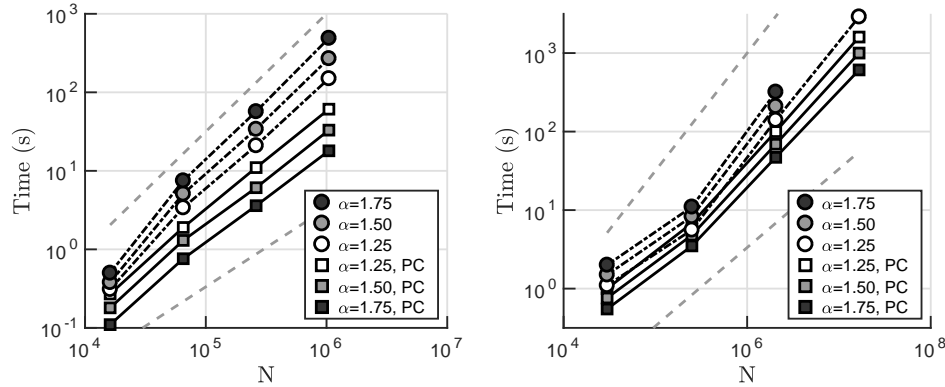
FIG. 2. *For the two-dimensional (left) and three-dimensional (right) examples, we plot the runtime t_{CG} required for CG to attain an accuracy of $\epsilon_{res} = 10^{-9}$ as tabulated in Tables 6 and 10, both with (square markers; PC) and without (circular markers) preconditioning. Note that some points in the right plot are absent due to excessive runtime. In both plots the top trend line is $O(N^{1.5})$ and the bottom is $O(N)$.*

TABLE 3
Relative solve errors for $\alpha \in \{0.25, 0.50, 0.75, 1.00, 1.25, 1.50, 1.75\}$ for the one-dimensional elliptic example with RHS $\mathbf{f} = 1$ and discrete solution $\mathbf{u}$ approximating (19).

N	$e_{\mathrm{sol},0.25}$	$e_{\mathrm{sol},0.50}$	$e_{\mathrm{sol},0.75}$	$e_{\mathrm{sol},1.00}$	$e_{\mathrm{sol},1.25}$	$e_{\mathrm{sol},1.50}$	$e_{\mathrm{sol},1.75}$
511	3.2e−03	3.5e−03	3.0e−03	2.5e−03	2.0e−03	1.4e−03	8.1e−04
1023	2.1e−03	2.1e−03	1.7e−03	1.3e−03	1.0e−03	7.2e−04	4.1e−04
2047	1.4e−03	1.2e−03	9.2e−04	6.8e−04	5.0e−04	3.6e−04	2.0e−04
4095	8.8e−04	7.4e−04	5.0e−04	3.5e−04	2.6e−04	1.8e−04	1.0e−04
Rate:	-0.63	-0.75	-0.86	-0.93	-0.98	-0.99	-1.00

A less smooth solution. As a second one-dimensional example, we follow Huang and Oberman [27, section 7] and take an RHS corresponding to $f(x) = 1$. This leads to an analytic solution on Ω given by (up to known constant K'_α)

$$(19) \qquad u(x) = K'_\alpha (1 - x^2)^{\alpha/2},$$

which when extended to $\mathbf{R}$ is only continuous as $x \to \pm 1$, in contrast to the previous example.

As in Huang and Oberman, applying the discrete forward operator M to (19) is inaccurate near the boundary due to the lack of differentiability (not shown). However, taking $\mathbf{f} = 1$ as the RHS in the discretization of (3), we still observe steady convergence of the relative solution error e_{sol} as N increases in Table 3, though due to reduced regularity of the solution the observed rate of convergence deteriorates to $O(N^\gamma)$ with $\gamma \approx \min(1, 1/2 + \alpha/2)$.

4.2. Elliptic example in two dimensions: Square domain. For a two-dimensional example, we use a square domain $\Omega = [0,1]^2$ discretized with a regular grid of N DOFs, showing the time to construct (t_{con}) and apply (t_{app}) the discrete operator M in Table 4. For these and the remainder of our examples, we focus on the case $\alpha > 1$ for brevity, as for $\alpha < 1$ the linear system (15) may be solved efficiently without any preconditioning.

TABLE 4

Runtimes t_{con} for the construction of the operator M and t_{app} for application via FFT for the two-dimensional elliptic example.

N	t_{con}	t_{app}
127^2	8.4e−2	4.5e−3
255^2	9.8e−2	3.0e−2
511^2	1.9e−1	1.1e−1
1023^2	6.1e−1	5.0e−1
Rate:	0.5	1.1

TABLE 5

Grid error estimates R_2 and R_∞ for both $\mathbf{u}$ and $\mathbf{f}$ in (16) for the two-dimensional elliptic example.

	Grid rate, $\mathbf{u}$		Grid rate, $\mathbf{f}$	
α	R_2	R_∞	R_2	R_∞
1.25	0.98	0.87	2.65	2.63
1.50	0.99	0.82	2.38	2.35
1.75	0.99	0.87	2.16	2.15

Unlike the one-dimensional case, in two dimensions there is no RHS f for which (3) has a simple known solution. Instead, we use standard grid error estimates based on Richardson extrapolation to estimate the rate of convergence. Concretely, for our application error grid estimate we use the function

$$g_1(\mathbf{x}) = \prod_{i=1}^{2} \frac{1}{4}(1 + \cos(2\pi x_i - \pi))^2, \tag{20}$$

which is nice when truncated to $\Omega = [0,1]^2$. Using a coarse grid of size 255^2, a medium-scale grid of size 511^2, and a fine-scale grid of size 1023^2, we obtained three corresponding estimates of the fractional Laplacian of g_1 evaluated on the common coarse grid: $\{\mathbf{f}_{\text{c}}, \mathbf{f}_{\text{m}}, \mathbf{f}_{\text{f}}\} \subset \mathbf{R}^{255^2}$. The Richardson error rate estimate is then given by

$$R_p \equiv \frac{\log \|\mathbf{f}_{\text{f}} - \mathbf{f}_{\text{m}}\|_p - \log \|\mathbf{f}_{\text{m}} - \mathbf{f}_{\text{c}}\|_p}{\log 1/2}, \tag{21}$$

where $\|\cdot\|_p$ is the ℓ_p norm. For solution error, we obtain analogous error rate estimates for the solution u to the extended Dirichlet problem using RHS $g_2(\mathbf{x}) = 1$. These rate estimates can be seen in Table 5, where we observe that the solution error rates are empirically limited to first-order due again to the general lack of smoothness of u near the boundary $\partial\Omega$ [44].

In Table 6 and Figure 2 (left) we give CG convergence results for the square example, analogous to the one-dimensional results in Table 2. Note that unlike the one-dimensional case, here it is clearly advantageous to use a fast Poisson solver to apply the preconditioner. While the reduced number of iterations is roughly offset by the cost of applying the preconditioner at each iteration for smaller α and N, the utility of our preconditioning approach becomes clear for the larger, more ill-conditioned problems.

As discussed, the Richardson convergence results for non-smooth $\mathbf{u}$ in Table 5 are limited to roughly first-order accuracy. To validate the accuracy of our approach

TABLE 6

Runtime t_{CG} and number of iterations n_{CG} required to solve the two-dimensional elliptic example using CG with/without preconditioning.

α	N	$\epsilon_{\mathrm{res}} = 10^{-6}$		$\epsilon_{\mathrm{res}} = 10^{-9}$	
		t_{CG}	n_{CG}	t_{CG}	n_{CG}
1.25	127^2	1.7e−1 / 2.2e−1	21 / 61	2.7e−1 / 3.1e−1	33 / 77
	255^2	1.1e+0 / 2.3e+0	28 / 95	1.9e+0 / 3.4e+0	43 / 121
	511^2	6.5e+0 / 1.6e+1	36 / 148	1.1e+1 / 2.1e+1	57 / 188
	1023^2	3.7e+1 / 1.2e+2	47 / 231	6.1e+1 / 1.5e+2	74 / 292
	Rate:	1.3 / 1.5	*	1.3 / 1.5	*
1.50	127^2	1.3e−1 / 3.3e−1	15 / 86	1.8e−1 / 3.8e−1	23 / 108
	255^2	8.6e−1 / 4.0e+0	18 / 147	1.3e+0 / 5.1e+0	28 / 184
	511^2	4.0e+0 / 2.7e+1	21 / 250	6.1e+0 / 3.4e+1	33 / 312
	1023^2	2.2e+1 / 2.2e+2	26 / 425	3.3e+1 / 2.7e+2	40 / 528
	Rate:	1.2 / 1.5	*	1.2 / 1.6	*
1.75	127^2	8.6e−2 / 4.1e−1	11 / 120	1.1e−1 / 5.0e−1	15 / 149
	255^2	5.7e−1 / 6.1e+0	12 / 223	7.6e−1 / 7.5e+0	17 / 278
	511^2	2.5e+0 / 4.6e+1	13 / 413	3.6e+0 / 5.7e+1	19 / 516
	1023^2	1.2e+1 / 4.0e+2	14 / 766	1.8e+1 / 4.9e+2	21 / 962
	Rate:	1.2 / 1.6	*	1.2 / 1.6	*

TABLE 7

Relative ℓ_2 solve error for $\alpha \in \{0.75, 1.25, 1.50, 1.75\}$ for the two-dimensional elliptic example with RHS generated numerically by sampling as described in text. The last row gives an estimate of the rate of growth as $n = \sqrt{N} \sim h^{-1}$ is increased, i.e., γ in $O(n^\gamma)$.

N	$e_{0.75}$	$e_{1.25}$	$e_{1.5}$	$e_{1.75}$
127^2	2.1e−06	9.4e−06	2.7e−05	8.2e−05
255^2	1.6e−07	1.8e−06	6.0e−06	2.1e−05
511^2	1.5e−08	2.9e−07	1.2e−06	4.8e−06
1023^2	1.6e−09	4.3e−08	2.1e−07	9.9e−07
Rate:	-3.4	-2.6	-2.3	-2.1

on smooth solutions, we generate a smooth synthetic example in two dimensions as follows. Taking $u \equiv g_1$ in (20), we sample u on a regular grid of 4095^2 DOFs to obtain $\mathbf{u}_{\mathrm{true}}^{(4095)}$. With $\mathsf{M}^{(4095)}$ as our discrete operator of the corresponding size, we compute $\mathbf{f}^{(4095)} = \mathsf{M}^{(4095)}\mathbf{u}_{\mathrm{true}}^{(4095)}$. For a given problem size $N = n^2$ we obtain the RHS vector $\mathbf{f}^{(n)}$ and "true solution" $\mathbf{u}_{\mathrm{true}}^{(n)}$ by appropriately subsampling $\mathbf{f}^{(4095)}$ and $\mathbf{u}_{\mathrm{true}}^{(4095)}$, respectively, which then permits computing the relative ℓ_2 error norm $e = \|\mathbf{u}_{\mathrm{true}}^{(n)} - \mathbf{u}^{(n)}\|/\|\mathbf{u}_{\mathrm{true}}^{(n)}\|$, where the discrete solution satisfies the problem on the smaller grid $\mathsf{M}^{(n)}\mathbf{u}^{(n)} = \mathbf{f}^{(n)}$. Results can be seen in Table 7, where we note that rates given are in terms of $n = \sqrt{N}$ as appropriate for error rates on a regular two-dimensional grid. For this problem with smooth solution, we see better error rates compared to Table 5, aligned with our theory.

4.3. Elliptic example in three dimensions. In three dimensions for the hypercube case $\Omega = [0,1]^3$ we repeat experiments analogous to those in subsection 4.2 (timing results shown in Table 8).

TABLE 8

Runtimes t_{con} for the construction of the operator M and t_{app} for application via FFT for the three-dimensional elliptic example.

N	t_{con}	t_{app}
31^3	2.3e−1	3.0e−2
63^3	3.1e−1	8.7e−2
127^3	1.3e+0	1.3e+0
255^3	1.8e+1	2.0e+1
Rate:	0.7	1.1

TABLE 9

Grid error estimates R_2 and R_∞ for both $\mathbf{u}$ and $\mathbf{f}$ in (16) for the three-dimensional elliptic example.

α	Grid rate, $\mathbf{u}$		Grid rate, $\mathbf{f}$	
	R_2	R_∞	R_2	R_∞
1.25	1.02	0.90	4.35	4.55
1.50	1.10	0.87	4.22	4.62
1.75	1.30	1.13	3.90	4.99

To compute our three-dimensional grid error estimates (21), we use the three-dimensional analogue of (20),

$$g_1(\mathbf{x}) = \prod_{i=1}^{3} \frac{1}{4}(1 + \cos(2\pi x_i - \pi))^2,$$

for the apply error and again $g_2(\mathbf{x}) \equiv 1$ for the solution error. We use coarse, medium-scale, and fine grids with sizes $63^3, 127^3$, and 255^3, respectively, and give the results in Table 9. We remark that the error rates reported for $\mathbf{f}$ appear artifically inflated, likely due to the fact that the grid error estimate is an asymptotic approximation that holds in the limit of large N, and $N = 63^3$ is not large.

In Table 10 and Figure 2 (right) we give CG convergence results for the three-dimensional example, just like those for the two-dimensional example. Just as in two dimensions, the utility of our simple preconditioner is clear for larger problems and for larger α, where iterative approaches to solving the linear system start to become prohibitively expensive without preconditioning.

4.4. Elliptic examples in two dimensions: An "L"-shaped domain and octagonal domain. Before moving to time-dependent examples, we include a final demonstration showing our method applied to problems where the domain is an occluded Cartesian grid (i.e., a regular discretization that is not a hypercube). The first such example is a problem on an "L"-shaped domain obtained by taking a regular grid of $(n-1)^2$ points as before and then removing $(n/2)^2$ contiguous points corresponding to a single corner of the domain; see Figure 3 (left and center).

Because the "L"-shaped domain is discretized as a subset of a regular grid, the operator M can still be applied quickly via FFT as before. However, use of a fast Poisson solver to apply the preconditioner is no longer possible. Instead, we use a sparse Cholesky factorization as in our one-dimensional examples. We use the default MATLAB permutation options for sparse Cholesky (which corresponds to an approximate minimum degree ordering of the unknowns) though other methods are possible.

TABLE 10

Runtime t_{CG} and number of iterations n_{CG} required to solve the three-dimensional elliptic example using CG with/without preconditioning. The parenthesized quantities indicate the corresponding test did not converge within 250 iterations. For the rate computations we omit $N = 31^3$ due to the clear nonasymptotic behavior in Figure 2.

| α	N	$\epsilon_{\text{res}} = 10^{-6}$		$\epsilon_{\text{res}} = 10^{-9}$	
		t_{CG}	n_{CG}	t_{CG}	n_{CG}
	31^3	6.4e−1 / 7.7e−1	13 / 27	1.0e+0 / 1.1e+0	20 / 37
	63^3	4.3e+0 / 4.6e+0	16 / 47	6.8e+0 / 5.6e+0	24 / 63
1.25	127^3	6.9e+1 / 1.0e+2	21 / 75	1.0e+2 / 1.4e+2	32 / 100
	255^3	1.0e+3 / 2.2e+3	27 / 117	1.6e+3 / 2.9e+3	42 / 156
	Rate:	1.3 / 1.5	*	1.3 / 1.5	*
	31^3	5.3e−1 / 1.1e+0	10 / 36	7.6e−1 / 1.5e+0	14 / 50
	63^3	3.5e+0 / 5.9e+0	12 / 66	4.8e+0 / 8.3e+0	17 / 91
1.50	127^3	4.7e+1 / 1.5e+2	14 / 113	6.9e+1 / 2.1e+2	21 / 154
	255^3	7.2e+2 / 3.6e+3	18 / 195	1.0e+3 / (4.6e+3)	26 / (250)
	Rate:	1.3 / 1.5	*	1.3 / 1.5	*
	31^3	4.1e−1 / 1.6e+0	7 / 49	5.5e−1 / 2.0e+0	10 / 65
	63^3	2.4e+0 / 8.3e+0	8 / 93	3.5e+0 / 1.1e+1	12 / 123
1.75	127^3	3.4e+1 / 2.4e+2	10 / 175	4.7e+1 / 3.2e+2	14 / 229
	255^3	4.6e+2 / (4.6e+3)	11 / (250)	6.1e+2 / (4.6e+3)	15 / (250)
	Rate:	1.3 / 1.6	*	1.2 / 1.6	*
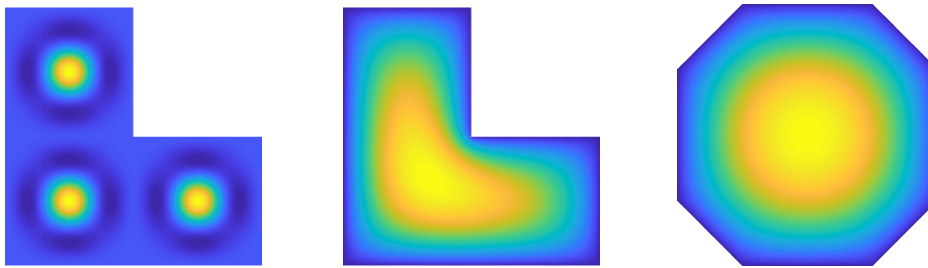
FIG. 3. *For a qualitative demonstration on an "L"-shaped domain, we plot the RHS (left) and solution with $\alpha = 1.75$ (center). As in our other examples, the solution is forced to zero outside of the domain. For results on an irregular octagonal domain (right), we build an octagon with horizontal and vertical sides of length 2 and all other sides of length $\sqrt{2}$.*

In Table 11 we show results for the "L"-shaped domain for choices of N ranging from $N = 12033$ (i.e., $127^2 - 64^2$) to $N = 784385$ (i.e., $1023^2 - 512^2$). We focus on the time t_{PC} to construct the factored preconditioner using sparse Cholesky as well as the runtime and number of iterations for CG both with and without our preconditioning scheme, as before. For brevity we give only results for $\alpha = 1.75$, as results for smaller α follow the same trends as in the case of a square domain. We remark that, while the time to factor the preconditioner is nonzero, it is still small compared to the time to solve the systems with CG, and the runtimes t_{CG} in Table 11 are comparable to those in Table 6 (albeit one must adjust for the slighly different system sizes).

For another example discretized on a subset of a regular grid, we construct an irregular octagon whose vertices coincide with points on a regular grid; see Figure 3 (right). As with the previous example, we use a sparse Cholesky factorization to build the preconditioner and give analogous results for $\alpha = 1.75$ and varying N and ϵ_{res} in Table 12.

TABLE 11

Runtime t_{PC} for construction of the preconditioner for the two-dimensional elliptic example on the "L"-shaped domain, as well as time t_{CG} and number of iterations n_{CG} required for CG to converge to tolerances 10^{-6} and 10^{-9} with/without preconditioning for the case $\alpha = 1.75$. The parenthesized quantities indicate the corresponding test did not converge within 1000 iterations.

| α | N | t_{PC} | $\epsilon_{res} = 10^{-6}$ | | $\epsilon_{res} = 10^{-9}$ | |
			t_{CG}	n_{CG}	t_{CG}	n_{CG}
	12033	1.9e−2	6.2e−2 / 4.9e−1	10 / 141	8.0e−2 / 6.4e−1	14 / 191
	48641	8.4e−2	4.3e−1 / 7.2e+0	11 / 268	5.8e−1 / 9.2e+0	16 / 354
1.75	195585	3.9e−1	2.1e+0 / 5.4e+1	13 / 497	2.7e+0 / 7.0e+1	18 / 655
	784385	1.8e+0	1.0e+1 / 4.5e+2	14 / 925	1.4e+1 / (4.9e+2)	20 / (1000)
	Rate:	1.1	1.2 / 1.6	*	1.2 / 1.7	*

TABLE 12

Runtime t_{PC} for construction of the preconditioner for the two-dimensional elliptic example on the octagonal domain, as well as time t_{CG} and number of iterations n_{CG} required for CG to converge to tolerances 10^{-6} and 10^{-9} with/without preconditioning for the case $\alpha = 1.75$.

| α | N | t_{PC} | $\epsilon_{res} = 10^{-6}$ | | $\epsilon_{res} = 10^{-9}$ | |
			t_{CG}	n_{CG}	t_{CG}	n_{CG}
	14145	5.3e−2	9.3e−2 / 3.8e−1	10 / 119	1.2e−1 / 4.9e−1	14 / 145
	56961	1.4e−1	4.5e−1 / 5.4e+0	11 / 220	6.4e−1 / 6.9e+0	16 / 267
1.75	228609	5.5e−1	2.0e+0 / 4.0e+1	13 / 407	2.8e+0 / 4.9e+1	18 / 493
	915969	2.7e+0	1.1e+1 / 3.7e+2	15 / 752	1.4e+1 / 4.5e+2	20 / 910
	Rate:	0.9	1.1 / 1.6	*	1.1 / 1.6	*

4.5. Time-dependent example in two and three dimensions. Finally, we turn to the time-dependent case. As described in subsection 3.2, our approach to the time-dependent fractional diffusion problem (2) involves first computing the discrete fractional Laplacian operator as before and then using a Crank–Nicolson method to time-step the solution. Here we demonstrate the efficiency of our preconditioning scheme for the time-dependent problem and give grid error estimates for $\Omega = [0, 1]^d$.

For smooth solutions, the Crank-Nicolson scheme is locally second-order in time and our spatial discretization is locally second-order in space. Thus, we choose our temporal step size in d dimensions as $\Delta t = (N^{1/d} + 1)^{-1}$ such that $\Delta t \approx h$ but the number of time steps required to reach a final time of $T = 0.25$ is integral. However, we remark that, just as in the elliptic case, we cannot expect better than first-order convergence in general [15].

For our grid error estimates we take $f \equiv 0$ in (2) and initial condition

$$u_0(\mathbf{x}) = \prod_{i=1}^{d} \frac{1}{4}(1 + \cos(2\pi\nu_i x_i - \pi))^2,$$

with $\nu_1 = 3$, $\nu_2 = 11$, and $\nu_3 = 2$. Using (17) to time-step the solution to final time $T = 0.25$, we then compute grid error estimates for simultaneous refinement in space and time, which are given in Table 13. For the two-dimensional case we use spatial grids with 255^2, 511^2, and 1023^2 points for the coarse, medium-scale, and fine grids, respectively. For the three-dimensional case we analogously use 31^3, 63^3, and 127^3 points in space, as we are limited by the runtime requirements of solving the largest

TABLE 13

Grid error estimates R_2 and R_∞ for the solution at time $T = 0.25$ to the parabolic problem (2) in both two (2D) and three (3D) dimensions as described in subsection 4.5.

	2D		3D	
α	R_2	R_∞	R_2	R_∞
1.25	0.97	0.87	2.22	3.11
1.50	0.97	0.82	2.30	3.24
1.75	0.97	0.85	3.02	3.39

TABLE 14

Runtime t_{CG} and number of iterations n_{CG} required to perform a single time step for the time-dependent example using CG with/without preconditioning in both two (2D) and three (3D) dimensions. In all cases we use $\epsilon_{res} = 10^{-9}$.

		2D			3D		
α	N	t_{CG}	n_{CG}	N	t_{CG}	n_{CG}
1.25	255^2	5.4e−1 / 6.3e−1	12 / 25	63^3	2.9e+0 / 2.1e+0	11 / 25
	511^2	2.1e+0 / 2.7e+0	12 / 27	127^3	3.5e+1 / 4.5e+1	12 / 28
	1023^2	9.4e+0 / 1.5e+1	12 / 29	255^3	4.7e+2 / 5.8e+2	12 / 30
	Rate:	1.0 / 1.1	*	Rate:	1.2 / 1.3	*
1.50	255^2	5.4e−1 / 1.4e+0	12 / 56	63^3	2.7e+0 / 3.9e+0	10 / 47
	511^2	2.3e+0 / 6.5e+0	13 / 67	127^3	3.8e+1 / 7.7e+1	12 / 60
	1023^2	1.1e+1 / 4.3e+1	14 / 79	255^3	5.0e+2 / 1.4e+3	13 / 73
	Rate:	1.1 / 1.2	*	Rate:	1.2 / 1.4	*
1.75	255^2	4.4e−1 / 2.8e+0	10 / 121	63^3	2.2e+0 / 6.5e+0	8 / 81
	511^2	1.9e+0 / 1.5e+1	11 / 162	127^3	2.9e+1 / 1.6e+2	9 / 122
	1023^2	9.3e+0 / 1.2e+2	12 / 212	255^3	4.1e+2 / 3.2e+3	10 / 170
	Rate:	1.1 / 1.4	*	Rate:	1.2 / 1.5	*

problems. As in the elliptic setting, we observe an artificial inflation of the Richardson rate in three dimensions.

In Table 14 we give the CG results for a single time step with random RHS, i.e., the results for a single linear system. In both two dimensions and three dimensions, we use the preconditioner described in section 2 applied with a modified fast Poisson solver. Compared to the elliptic setting, we see that the time-dependent system matrix is better conditioned, and thus preconditioning for $\alpha = 1.25$ is not necessary in two dimensions and not helpful in three dimensions. However, for larger α there is a clear benefit.

We remark that in practical settings with multiple time steps the number of iterations is reduced slightly from the current setting because the old solution $\mathbf{u}^{(k)}$ can be used as an initial guess for the solution $\mathbf{u}^{(k+1)}$, but the difference is not substantial in general.

5. Conclusions. We introduced a simple discretization scheme for the fractional Laplacian operator in one, two, and three dimensions based on singularity subtraction combined with the regularly spaced trapezoidal rule. When applied to sufficiently smooth functions u, the resulting discretization is provably second-order accurate in the grid spacing h, whereas for rougher u we observe first-order accuracy in the ℓ_2 solution error.

When the order α of the fractional Laplacian is close to two, the discrete operator is ill conditioned, reflecting the underlying ill-conditioning of the continuous (integer-order) Laplacian. To efficiently solve linear systems with the discrete fractional Laplacian, we demonstrated the utility of a simple preconditioning scheme based on fast Poisson solvers.

For higher-order schemes, it is necessary to forsake the simplicity of our approach to more precisely handle solutions u that exhibit only fractional-order smoothness near the boundary of Ω for both (2) and (3). While we intend to pursue this in future work, we have shown that the scheme presented here provides a fast, simple alternative for situations amenable to lower-order accuracy.

REFERENCES

[1] M. ABRAMOWITZ AND I. STEGUN, *Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables*, Dover Publications, New York, 1964.

[2] G. ACOSTA AND J. P. BORTHAGARAY, *A fractional Laplace equation: Regularity of solutions and finite element approximations*, SIAM J. Numer. Anal., 55 (2017), pp. 472–495, https://doi.org/10.1137/15M1033952.

[3] G. ACOSTA, J. P. BORTHAGARAY, O. BRUNO, AND M. MAAS, *Regularity theory and high order numerical methods for the (1D)-fractional Laplacian*, Math. Comp., 87 (2018), pp. 1821–1857.

[4] M. AINSWORTH AND C. GLUSA, *Aspects of an adaptive finite element method for the fractional Laplacian: A priori and a posteriori error estimates, efficient implementation and multigrid solver*, Comput. Methods Appl. Mech. Engrg., 327 (2017), pp. 4–35, https://doi.org/10.1016/j.cma.2017.08.019.

[5] M. AINSWORTH AND C. GLUSA, *Hybrid finite element–spectral method for the fractional Laplacian: Approximation theory and efficient solver*, SIAM J. Sci. Comput., 40 (2018), pp. A2383–A2405, https://doi.org/10.1137/17M1144696.

[6] M. AINSWORTH AND C. GLUSA, *Towards an efficient finite element method for the integral fractional Laplacian on polygonal domains*, in Contemporary Computational Mathematics, J. Dick, F. Y. Kuo, and H. Wozniakowski, eds., Springer International Publishing, Cham, 2018, pp. 17–57.

[7] A. BONITO, J. P. BORTHAGARAY, R. H. NOCHETTO, E. OTAROLA, AND A. J. SALGADO, *Numerical methods for fractional diffusion*, Comput. Vis. Sci., 19 (2018), pp. 19–46.

[8] A. BONITO, W. LEI, AND J. E. PASCIAK, *Numerical approximation of the integral fractional Laplacian*, Numer. Math., 142 (2019), pp. 235–278.

[9] L. CAFFARELLI AND L. SILVESTRE, *An extension problem related to the fractional Laplacian*, Comm. Partial Differential Equations, 32 (2007), pp. 1245–1260, https://doi.org/10.1080/03605300600987306.

[10] M. CHEN, Y. WANG, X. CHENG, AND W. DENG, *Second-order LOD multigrid method for multidimensional Riesz fractional diffusion equation*, BIT, 54 (2014), pp. 623–647, https://doi.org/10.1007/s10543-014-0477-1.

[11] D. CRUZ-URIBE AND C. NEUGEBAUER, *Sharp error bounds for the trapezoidal rule and Simpson's rule*, J. Inequal. Pure Appl. Math., 3 (2002), 49.

[12] M. D'ELIA AND M. GUNZBURGER, *The fractional Laplacian operator on bounded domains as a special case of the nonlocal diffusion operator*, Comput. Math. Appl., 66 (2013), pp. 1245–1260, https://doi.org/10.1016/j.camwa.2013.07.022.

[13] S. DUO, H. W. VAN WYK, AND Y. ZHANG, *A novel and accurate finite difference method for the fractional Laplacian and the fractional Poisson problem*, J. Comput. Phys., 355 (2018), pp. 233–252, https://doi.org/10.1016/j.jcp.2017.11.011.

[14] M. FELSINGER, M. KASSMANN, AND P. VOIGT, *The Dirichlet problem for nonlocal operators*, Math. Z., 279 (2015), pp. 779–809, https://doi.org/10.1007/s00209-014-1394-3.

[15] X. FERNÁNDEZ-REAL AND X. ROS-OTON, *Regularity theory for general stable operators: Parabolic equations*, J. Funct. Anal., 272 (2017), pp. 4165–4221, https://doi.org/10.1016/j.jfa.2017.02.015.

[16] W. FONG AND E. DARVE, *The black-box fast multipole method*, J. Comput. Phys., 228 (2009), pp. 8712–8725, https://doi.org/10.1016/j.jcp.2009.08.031.

[17] H. Fu, M. K. Ng, AND H. Wang, *A divide-and-conquer fast finite difference method for space–time fractional partial differential equation*, Comput. Math. Appl., 73 (2017), pp. 1233–1242, https://doi.org/10.1016/j.camwa.2016.11.023.

[18] H. Fu AND H. Wang, *A preconditioned fast finite difference method for space-time fractional partial differential equations*, Fract. Calc. Appl. Anal., 20 (2017), pp. 88–116.

[19] H. Fu AND H. Wang, *A preconditioned fast parareal finite difference method for space-time fractional partial differential equation*, J. Sci. Comput., 78 (2019), pp. 1724–1743, https://doi.org/10.1007/s10915-018-0835-2.

[20] T. Gao, J. Duan, X. Li, AND R. Song, *Mean exit time and escape probability for dynamical systems driven by Lévy noises*, SIAM J. Sci. Comput., 36 (2014), pp. A887–A906, https://doi.org/10.1137/120897262.

[21] A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363, https://doi.org/10.1137/0710032.

[22] L. Greengard AND V. Rokhlin, *A fast algorithm for particle simulations*, J. Comput. Phys., 73 (1987), pp. 325–348.

[23] M. Hestenes AND E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Natl. Bureau Standards, 49 (1952), pp. 409–436.

[24] Y. Hu, C. Li, AND H. Li, *The finite difference method for Caputo-type parabolic equation with fractional Laplacian: One-dimension case*, Chaos Solitons Fractals, 102 (2017), pp. 319–326, https://doi.org/10.1016/j.chaos.2017.03.038.

[25] Y. Huang, *Explicit Barenblatt profiles for fractional porous medium equations*, Bull. Lond. Math. Soc., 46 (2014), pp. 857–869, https://doi.org/10.1112/blms/bdu045.

[26] Y. Huang AND A. Oberman, *Numerical methods for the fractional Laplacian: A finite difference-quadrature approach*, SIAM J. Numer. Anal., 52 (2014), pp. 3056–3084, https://doi.org/10.1137/140954040.

[27] Y. Huang AND A. Oberman, *Finite Difference Methods for Fractional Laplacians*, arXiv e-prints, 2016, https://arxiv.org/abs/1611.00164.

[28] A. Iserles, *A First Course in the Numerical Analysis of Differential Equations*, Cambridge University Press, Cambridge, UK, 1996.

[29] M. Kwaśnicki, *Ten equivalent definitions of the fractional Laplace operator*, Fract. Calc. Appl. Anal., 20 (2017), pp. 7–51, https://doi.org/10.1515/fca-2017-0002.

[30] Y. Lin AND C. Xu, *Finite difference/spectral approximations for the time-fractional diffusion equation*, J. Comput. Phys., 225 (2007), pp. 1533–1552, https://doi.org/10.1016/j.jcp.2007.02.001.

[31] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. M. Meerschaert, M. Ainsworth, AND G. E. Karniadakis, *What is the fractional Laplacian? A comparative review with new results*, J. Comput. Phys., 404 (2020), 109009.

[32] B. Mandelbrot, *The Fractal Geometry of Nature*, Henry Holt and Company, New York, 1982.

[33] Z. Mao AND G. E. Karniadakis, *A spectral method (of exponential convergence) for singular solutions of the diffusion equation with general two-sided fractional derivative*, SIAM J. Numer. Anal., 56 (2018), pp. 24–49, https://doi.org/10.1137/16M1103622.

[34] O. Marin, O. Runborg, AND A.-K. Tornberg, *Corrected trapezoidal rules for a class of singular functions*, IMA J. Numer. Anal., 34 (2014), pp. 1509–1540, https://doi.org/10.1093/imanum/drt046.

[35] W. McLean AND K. Mustapha, *A second-order accurate numerical method for a fractional wave equation*, Numer. Math., 105 (2007), pp. 481–510, https://doi.org/10.1007/s00211-006-0045-y.

[36] M. M. Meerschaert, H.-P. Scheffler, AND C. Tadjeran, *Finite difference methods for two-dimensional fractional dispersion equation*, J. Comput. Phys., 211 (2006), pp. 249–261, https://doi.org/10.1016/j.jcp.2005.05.017.

[37] M. M. Meerschaert AND C. Tadjeran, *Finite difference approximations for two-sided space-fractional partial differential equations*, Appl. Numer. Math., 56 (2006), pp. 80–90, https://doi.org/10.1016/j.apnum.2005.02.008.

[38] R. Metzler AND J. Klafter, *The random walk's guide to anomalous diffusion: A fractional dynamics approach*, Phys. Rep., 339 (2000), pp. 1–77, https://doi.org/10.1016/S0370-1573(00)00070-3.

[39] R. Metzler AND J. Klafter, *The restaurant at the end of the random walk: Recent developments in the description of anomalous transport by fractional dynamics*, J. Phys. A, 37 (2004), R161.

[40] R. H. Nochetto, E. Otárola, AND A. J. Salgado, *A PDE approach to fractional diffusion in general domains: A priori error analysis*, Found. Comput. Math., 15 (2015), pp. 733–791, https://doi.org/10.1007/s10208-014-9208-x.

[41] H.-K. Pang and H.-W. Sun, *Multigrid method for fractional diffusion equations*, J. Comput. Math., 231 (2012), pp. 693–703, https://doi.org/10.1016/j.jcp.2011.10.005.

[42] C. Pozrikidis, *The Fractional Laplacian*, CRC Press, Boca Raton, FL, 2016.

[43] X. Ros-Oton, *Nonlocal equations in bounded domains: A survey*, Publ. Mat., 60 (2016), pp. 3–26.

[44] X. Ros-Oton and J. Serra, *The Dirichlet problem for the fractional Laplacian: Regularity up to the boundary*, J. Math. Pures Appl. (9), 101 (2014), pp. 275–302, https://doi.org/10.1016/j.matpur.2013.06.003.

[45] X. Ros-Oton and J. Serra, *Regularity theory for general stable operators*, J. Differential Equations, 260 (2016), pp. 8675–8715, https://doi.org/10.1016/j.jde.2016.02.033.

[46] A. I. Saichev and G. M. Zaslavsky, *Fractional kinetic equations: Solutions and applications*, Chaos, 7 (1997), pp. 753–764, https://doi.org/10.1063/1.166272.

[47] X. Tian and Q. Du, *Analysis and comparison of different approximations to nonlocal diffusion and linear peridynamic equations*, SIAM J. Numer. Anal., 51 (2013), pp. 3458–3482, https://doi.org/10.1137/13091631X.

[48] X. Tian, Q. Du, and M. Gunzburger, *Asymptotically compatible schemes for the approximation of fractional Laplacian and related nonlocal diffusion problems on bounded domains*, Adv. Comput. Math., 42 (2016), pp. 1363–1380, https://doi.org/10.1007/s10444-016-9466-z.

[49] P. N. Vabishchevich, *Numerically solving an equation for fractional powers of elliptic operators*, J. Comput. Phys., 282 (2015), pp. 289–302, https://doi.org/10.1016/j.jcp.2014.11.022.

[50] J. L. Vázquez, *The mathematical Theories of Diffusion. Nonlinear and Fractional Diffusion*, arXiv e-prints, 2017, https://arxiv.org/abs/1706.08241.

[51] H. Wang and T. S. Basu, *A fast finite difference method for two-dimensional space-fractional diffusion equations*, SIAM J. Sci. Comput., 34 (2012), pp. A2444–A2458, https://doi.org/10.1137/12086491X.

[52] H. Wang and N. Du, *A superfast-preconditioned iterative method for steady-state space-fractional diffusion equations*, J. Comput. Phys., 240 (2013), pp. 49–57, https://doi.org/10.1016/j.jcp.2012.07.045.

[53] K. Xu and E. Darve, *Spectral Method for the Fractional Laplacian in 2D and 3D*, arXiv e-prints, 2018, arXiv:1812.08325.

[54] L. Ying, G. Biros, and D. Zorin, *A kernel-independent adaptive fast multipole algorithm in two and three dimensions*, J. Comput. Phys., 196 (2004), pp. 591–626, https://doi.org/10.1016/j.jcp.2003.11.021.

[55] M. Zayernouri and G. E. Karniadakis, *Fractional spectral collocation method*, SIAM J. Sci. Comput., 36 (2014), pp. A40–A62, https://doi.org/10.1137/130933216.

[56] F. Zeng, C. Li, F. Liu, and I. Turner, *The use of finite difference/element approaches for solving the time-fractional subdiffusion equation*, SIAM J. Sci. Comput., 35 (2013), pp. A2976–A3000, https://doi.org/10.1137/130910865.

[57] F. Zeng, Z. Zhang, and G. E. Karniadakis, *Fast difference schemes for solving high-dimensional time-fractional subdiffusion equations*, J. Comput. Phys., 307 (2016), pp. 15–33, https://doi.org/10.1016/j.jcp.2015.11.058.

[58] M. Zhao, H. Wang, and A. Cheng, *A fast finite difference method for three-dimensional time-dependent space-fractional diffusion equations with fractional derivative boundary conditions*, J. Sci. Comput., 74 (2018), pp. 1009–1033, https://doi.org/10.1007/s10915-017-0478-8.

[59] X. Zhao, Z.-z. Sun, and G. E. Karniadakis, *Second-order approximations for variable order fractional derivatives*, J. Comput. Phys., 293 (2015), pp. 184–200, https://doi.org/10.1016/j.jcp.2014.08.015.