MATHEMATICAL CONTROL AND
RELATED FIELDS

doi:10.3934/mcrf.2021004

MODEL REDUCTION FOR FRACTIONAL ELLIPTIC PROBLEMS USING KATO'S FORMULA

Huy Dinh
New York University, Warren Weaver Hall
251 Mercer Street
New York, NY 10012, USA

Harbir Antil
George Mason University
4400 University Drive, MS: 3F2
Fairfax, VA 22030, USA

Yanlai Chen
University of Massachusetts Dartmouth
285 Old Westport Road
North Dartmouth, MA 02747, USA

Elena Cherkaev and Akil Narayan*
University of Utah
155 South 1400 East
Salt Lake City, UT 84112-0090, USA

(Communicated by Enrique Zuazua)

Abstract. We propose a novel numerical algorithm utilizing model reduction for computing solutions to stationary partial differential equations involving the spectral fractional Laplacian. Our approach utilizes a known characterization of the solution in terms of an integral of solutions to local (classical) elliptic problems. We reformulate this integral into an expression whose continuous and discrete formulations are stable; the discrete formulations are stable independent of all discretization parameters. We subsequently apply the reduced basis method to accomplish model order reduction for the integrand. Our choice of quadrature in discretization of the integral is a global Gaussian quadrature rule that we observe is more efficient than previously proposed quadrature rules. Finally, the model reduction approach enables one to compute solutions to multi-query fractional Laplace problems with orders of magnitude less cost than a traditional solver.

1. **Introduction.** Differential equations involving fractional derivative powers have gained in popularity in recent years. These non-classical differential equations have shown potential to model nonlocal and time-delay effects, making them good candidates for modeling hysteretic and globally-coupled phenomena. For example, fractional differential equations have recently been used to model fluid mechanics, arterial blood flow, cardiac ischemia, and geophysics; as components of optimal control problems; and as ingredients in image denoising and image segmentation

2020 *Mathematics Subject Classification.* Primary: 35R11, 65M12.
Key words and phrases. Fractional Laplacian, model order reduction, numerical methods.
* Corresponding author: Akil Narayan.

[38, 30, 22, 2, 5]. In particular, optimal control problems frequently require optimization of an objective function that involves the solution to a partial differential equation (PDE). Algorithms for solving the optimization problem would require many evaluations of the objective, hence of the PDE. Thus, for control problems it is necessary to ensure that the computational solution of the PDE can be obtained as quickly as possible. In this paper we focus on the acceleration of computational algorithms for such PDEs, in particular those involving fractional elliptic operators, which are defined via spectral expansions; a prototypical example of an operator that we use throughout this paper is the fractional Laplacian.

With Δ the "standard" (classical) Laplacian operator, we are interested in computing the solution u to the partial differential equation

$$(-\Delta)^s u = f,$$

valid on some physical domain Ω (which is open, bounded, and Lipschitz) with appropriate boundary conditions on $\partial\Omega$, where $s \in (0,1)$ is the fractional order. The precise definition of the fractional operator $(-\Delta)^s$ involves the spectral expansion of the local operator $-\Delta$ (with appropriate associated boundary conditions), which we more precisely describe in Section 2. There are already several numerical algorithms for computing the solution to such an equation:

- Perhaps the most conceptually straightforward idea is to use the spectral expansion definition of $(-\Delta)^s$ to devise a scheme that computes solutions using the spectral expansion of the associated discretized operators [27, 28, 45, 40]. The disadvantage of this approach is that the procedure is expensive, requiring a full eigendecomposition of a potentially very large matrix.
- A second approach uses an extension procedure to write the nonlocal d-dimensional PDE as a $(d+1)$-dimensional local PDE [34, 13, 41]. This latter PDE can be solved with existing methods, although some nontrivial tailoring of existing numerical methods is needed [35]. The challenge with this approach is that the spatial dimension is increased, and the extended PDE is degenerate, requiring specialized numerical methods. More recent efforts have significantly reduced the computational cost of this method. These methods exploit the tensor product structure that occurs as a result of the additional dimension [33, 1] but rely on discretizations that depend on the exponent s, i.e., essentially assume a non-zero lower bound for s.
- A final approach that we use as the starting point for the method proposed in this paper is an integral operator approach, which writes the solution (i.e., the inverse of the operator) as a type of Dunford-Taylor integral involving the resolvent of the local operator:

$$(-\Delta)^{-s} = \frac{\sin \pi s}{\pi} \int_0^\infty y^{-s} \left(y - \Delta\right)^{-1} \mathrm{d}y, \tag{1}$$

  See [29, Theorem 2 with $\lambda = 0$]. Numerical approaches for solving nonlocal PDEs using this formula discretize the integral in (1) with quadrature and require many local ("standard") PDE solves (equal to the number of quadrature points) in order to compute the solution to the fractional problem [11, 6, 9]. However, this results in an algorithm that can require, depending on the problem, many or hundreds of queries of an existing PDE solver to compute the solution to the fractional problem. The challenge with this approach is that often many local PDE solves may be necessary to ensure accuracy for a single

solution of the fractional PDE, making this approach quite expensive compared to traditional solvers. For certain operators these local PDE solves may be accomplished in parallel, but this does not diminish the overall cost. Other operators require coupling of these solves, e.g., in cases when a lifting strategy is used to solve non-homogenous nonlocal Poisson equations [43], making parallel approaches more difficult.

In this paper, we develop a novel model reduction algorithm for the third approach listed above to substantially alleviate the cost of traditional PDE solves. We concentrate on this approach for the spectral definition of the fractional Laplacian in this paper, but due to a similar integral formulation for the integral fractional Laplacian [10], our approach would extend to more general cases as well.

This paper is not the first strategy for model reduction for fractional elliptic problems. The authors in [4] provide a model reduction strategy, applied to the second (extension) approach listed above. More recently, the work in [18, 19] employs a reduced basis approach by interpolating operator norms. Reduced order methods for this problem using the integral operator approach are also investigated in [9]. However, low-rank structure in solution sets to fractional problems has been empirically noted even earlier [44]. For problems involving nonlocal integral kernels, the authors in [25] also proposed a reduced basis approach, but use a different strategy to perform model reduction. Our contributions in this article are as follows:

- We provide a rearrangement of the Dunford-Taylor integral considered in [11] that improves numerical stability. We first show that the analytical solution has s-independent L^2 stability bounds. This stability extends to the numerical discretization, independent of all discretization parameters. See Lemma 3.1 and Proposition 3.
- Our approach to discretize the Dunford-Taylor integral is a novel application of a global Gaussian quadrature rule. Our numerical results suggest that our quadrature choice is more efficient than previously proposed choices, cf. Figure 4. We cannot provide an analytical error bound in terms of the number of quadrature points, but we do provide a rigorous, computable error certificate; see Proposition 4. Previous work has required a number of quadrature points proportional to $\max(1/s, 1/(1-s))$ in order to obtain a specified level of accuracy. Our empirical results suggest that our approach also suffers from this limitation, see Figure 2.
- We employ the reduced basis method (RBM) to effect model reduction which, after a single *offline* computational investment, can accelerate subsequent computations of $(s, f) \mapsto u$ by at least two orders of magnitude. The offline portion of this algorithm requires approximately as much time as a single $(s, f) \mapsto u$ solve using the traditional Dunford-Taylor approach; see Algorithm 3.4.
- We provide a rigorous *a posteriori* error estimate for our solution computed via model reduction. This error estimate is computed as a by-product of the offline investment, and is therefore directly available; see Theorem 5.2.

Regarding alternative model reduction approaches to solve this problem: After initial dissemination of the first draft of this manuscript, the authors in [9] provided a complementary approach. For s_0 some fixed constant, [9] provides rigorous convergence analysis for energy norms for $s \geq s_0 > 0$, with estimates depending on $1/s_0$. The space interpolation-based approach in [18, 19] likewise provides convergence the L^2 norm for the discretized setting. Although comparably strong convergence

analysis is missing in our work (i.e., we do not yet have rigorous convergence analysis available), we do accomplish different goals: We establish L^2-stability of the fractional problem *uniformly* as $s \downarrow 0$ both at the discrete and continuous levels. In addition, we provide rigorous, computable error certification for the fractional problem (and not just for the auxilliary problem defined by the Dunford-Taylor integral); these results distinguish our study from the similar approach in [9]. In comparison to extension approaches, the additional reduced order modeling based approach considered in [4] requires solving a PDE in an additional (extended) dimension, and due to this requirement is less efficient, even if combined with efficiency gains in full-order models achieved in [33] and related work.

We remark that while we study PDEs with an operator of the form $(-\Delta)^s$, all our results extend to more general fractional elliptic operators. See Remarks 1 and 4.

This paper is structured as follows. Section 2 lays out our notation and describes the problem. Section 3 describes a new algorithm for expressing and computing the Dunford-Taylor solution that was first proposed in [11]; this new algorithm is a novel utilization of a Gaussian quadrature rule. The accuracy of this quadrature rule is empirically investigated in Section 4. Section 5 is our main algorithmic section that introduces an RBM algorithm for model reduction that computationally accelerates the fractional PDE algorithm from Section 3, and provides a computable error certificate for the model reduction. Finally, section 6 includes numerical results that demonstrate our new algorithms on a two-dimensional fractional Laplace problem and compares our algorithm against the predecessor in [11].

2. Notation and setup.

Vectors will be denoted in lowercase bold, and matrices in uppercase bold, e.g., $\boldsymbol{x}$ and $\boldsymbol{A}$, respectively. If $\boldsymbol{M}$ is a symmetric positive definite matrix, we define

$$\|\boldsymbol{x}\|_{\boldsymbol{M}}^2 \coloneqq \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x},$$

and $\|\boldsymbol{x}\|$ is the standard Euclidean norm. The matrix norm $\|\boldsymbol{A}\|$ is the standard induced ℓ^2 norm on matrices. If $\boldsymbol{A}$ is symmetric, then $\lambda_{\min}(\boldsymbol{A})$ denotes the smallest (real) eigenvalue of $\boldsymbol{A}$. If both $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric positive definite matrices in $\mathbb{R}^{N\times N}$, we define the smallest generalized eigenvalue of $(\boldsymbol{A}, \boldsymbol{B})$ as

$$\lambda_{\min}(\boldsymbol{A}, \boldsymbol{B}) \coloneqq \inf_{\boldsymbol{x}\in\mathbb{R}^N\setminus\{\boldsymbol{0}\}} \frac{\|\boldsymbol{x}\|_{\boldsymbol{A}}^2}{\|\boldsymbol{x}\|_{\boldsymbol{B}}^2}.$$

Note that under these assumptions on $\boldsymbol{A}$ and $\boldsymbol{B}$, the above expression is equal to the smallest λ such that $\boldsymbol{A}\boldsymbol{x} = \lambda \boldsymbol{B}\boldsymbol{x}$ has a nontrivial solution $\boldsymbol{x}$, and also we have that

$$\lambda_{\min}\left(\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2}\right) = \lambda_{\min}(\boldsymbol{A}, \boldsymbol{B}),$$

where $\boldsymbol{B}^{1/2}$ is the symmetric positive definite matrix square root of $\boldsymbol{B}$. Similarly, we use the notation $\lambda_{\max}(\cdot)$ and $\lambda_{\max}(\cdot,\cdot)$ to denote maximum eigenvalues.

Consider a bounded domain $\Omega \subset \mathbb{R}^d$ with Lipschitz boundary $\partial\Omega$; for spatial domains we are mainly concerned with $d \leq 3$ but this restriction is not necessary. We have

$$L^2(\Omega) \coloneqq \left\{v : \Omega \to \mathbb{R} \;\middle|\; \|v\|_{L^2} < \infty\right\}, \quad \|v\|_{L^2}^2 \coloneqq \langle v, v\rangle, \quad \langle v, w\rangle \coloneqq \int_\Omega v(x)w(x)\mathrm{d}x.$$

We will often write $L^2 = L^2(\Omega)$, and we define $\langle \nabla w, \nabla v \rangle = \sum_{j=1}^d \langle \frac{\partial}{\partial x_j} w, \frac{\partial}{\partial x_j} v \rangle$, which induces a definition for the L^2 norm $\|\nabla v\|_{L^2}$ of vector-valued functions. For brevity will also write $\|v\| = \|v\|_{L^2}$ and $\|\nabla v\| = \|\nabla v\|_{L^2}$. The standard Laplace eigenvalue problem on Ω with Dirichlet boundary conditions,

$$\begin{aligned} -\Delta u &= \lambda u, && x \in \Omega, \\ u(x) &= 0, && x \in \partial\Omega, \end{aligned} \tag{2}$$

yields an infinite sequence of eigenvalues $0 < \lambda_1 \leq \lambda_2 \cdots$ with associated eigenfunctions $\{\phi_n\}_{n=1}^\infty$. Here, and in all the following, the differential operator Δ operates on the x variable. The spectral theorem ensures that the eigenfunctions enjoy $L^2(\Omega)$-orthogonality and completeness, so that

$$u \in L^2(\Omega) \quad \Longrightarrow \quad u(x) = \sum_{n=1}^\infty u_n \phi_n(x), \qquad u_n = \langle u, \phi_n \rangle_{L^2(\Omega)}, \tag{3}$$

where we have further assumed that each ϕ_n has unit $L^2(\Omega)$ norm.

2.1. **The fractional Laplace problem.** In this section we provide the definition of the fractional operator $(-\Delta)^s$, and summarize some regularity properties of solutions. Let $s \in (0,1)$. In this section we describe the spectral definition of the fractional operator $(-\Delta)^s$, on bounded domains, supplemented with homogeneous Dirichlet boundary conditions, see [7] for the inhomogeneous case. Related definitions of similar nonlocal or fractional operators can be found in the literature [31]. If u has an expansion in eigenfunctions, then a formal definition for application of the fractional operator is,

$$u = \sum_{n=1}^\infty u_n \phi_n(x) \quad \Longrightarrow \quad (-\Delta)^s u = \sum_{n=1}^\infty \lambda_n^s u_n \phi_n(x). \tag{4}$$

We likewise use λ_n to define Sobolev spaces of fractional order. With $s \in (0,1)$:

$$\mathbb{H}^s(\Omega) := \left\{ u \in L^2(\Omega) \;\middle|\; (-\Delta)^{s/2} u \in L^2(\Omega) \right\}, \qquad \mathbb{H}^{-s}(\Omega) := \mathbb{H}^s(\Omega)^*,$$

where $\mathbb{H}^s(\Omega)^*$ denotes the dual space of $\mathbb{H}^s(\Omega)$. With these definitions, $(-\Delta)^s : \mathbb{H}^s(\Omega) \to \mathbb{H}^{-s}(\Omega)$. For the relation of $\mathbb{H}^s(\Omega)$ to the standard fractional order Sobolev space $H^s(\Omega)$, see [32, Chapter 1, Section 9] or [21] for a more modern survey.

Given data $f \in L^2(\Omega)$, our main goal is to compute the solution u to

$$\begin{aligned} (-\Delta)^s u &= f, && x \in \Omega \\ u &= 0, && x \in \partial\Omega \end{aligned} \tag{5}$$

for arbitrary $s \in (0,1)$. We emphasize that only for $s > 1/2$ can the boundary conditions in (5) be understood in the classical trace sense. This is due to the fact that, for smooth enough boundaries, $\mathbb{H}^s(\Omega) = H_0^s(\Omega)$ for $s > 1/2$. For $s < 1/2$, $\mathbb{H}^s(\Omega) = H_0^s(\Omega) = H^s(\Omega)$ and in this case the boundary conditions are not meaningful in the classical trace sense. One possible interpretation of the boundary conditions in this case is using the integration-by-parts formula, as described in [7, Theorem 3.1]. Finally, for $s = 1/2$ we have $\mathbb{H}^s(\Omega) = H_{00}^{\frac{1}{2}}(\Omega)$ i.e., the Lions-Magenes space, in this case, it follows from the definition of the space that the functions need to have certain decay at the boundary.

One can assume $f \in \mathbb{H}^{-s}$, yielding a solution u in $\mathbb{H}^s$. Note therefore that $f \in L^2$ is a stronger requirement than necessary for this fractional problem, and is even stronger than usual for local elliptic problems. We make this stronger

assumption to be assured, for fixed f, that $u \in L^2$ uniformly for $s \in (0,1)$. Note that this is a reasonable assumption in the optimal control context where controls are frequently assumed to be smoother than the explicit image of a differential operator. Notationally, we omit showing explicit dependence of u on the spatial variable x, and only show dependence on the fractional order s, which is a parameter. This convention will be used in the remainder of this paper when considering solutions to parameterized PDE's: notational dependence on parameters will be explicit, but that on the spatial variable x will be implicit. Therefore, we let $u(s) \in L^2$ denote the solution u to (5) for a fixed value of s. We will be interested in developing a computational algorithm for computing the *family* or *manifold* of solutions,

$$U := \left\{ u(s) \;\middle|\; s \in (0,1) \right\}.$$

If $f \in L^2$, then the solution u to (5) has $\mathbb{H}^{2s}$ membership. In order to ensure that u is also in the classical Sobolev space H^{2s}, we need to assume the domain to be more regular, for example, quasi-convex. Recall, that quasi-convexity ensures that

$$\|v\|_{H^2(\Omega)} \leq C \|(-\Delta)v\|_{L^2(\Omega)}, \quad \forall v \in H^2(\Omega) \cap H_0^1(\Omega),$$

see [23, Theorem 10.4]. See also [24] for a related discussion on solution regularity. Our goal is to study the manifold U so that the natural function space is $\cap_{s\in(0,1)}\mathbb{H}^{2s}$. Therefore, all of our investigations will assume $f \in L^2$ and study solutions $u(s)$ as elements of L^2.

In the remainder of this document we will consider the problem (5) with homogeneous Dirichlet boundary conditions. The inhomogeneous boundary case may be handled using the lifting technique in [7], which requires only a small modification of the algorithm proposed here. While the previous discussion has centered on the definition of the nonlocal operator and the associated PDE, our main analytical and computational tool will be an equivalent formulation via Kato's formula.

2.2. **Kato's integral solution of (5).** The following remarkable result provides an appealing formula for the solution u to (5):

$$u(s) = \beta(s) \int_{-\infty}^{\infty} e^{(1-s)y} \left(-\Delta + e^y\right)^{-1} f \mathrm{d}y, \qquad \beta(s) := \frac{\sin \pi s}{\pi}$$

which is a reformulation of Kato's formula (1) from [29, Theorem 2 with $\lambda = 0$]. This representation was first exploited in [11] for designing numerical algorithms, and is derived via a special kind of Dunford-Taylor integral. To write the above more explicitly, define $q(y)$, for fixed $y \in \mathbb{R}$, as the solution to the *local* y-parameterized PDE,

$$\begin{aligned} -\Delta q(y) + e^y q(y) &= f, && x \in \Omega \\ q(y) &= 0, && x \in \partial\Omega. \end{aligned} \tag{6a}$$

Then u is given by

$$u(s) = \beta(s) \int_{-\infty}^{\infty} q(y) e^{(1-s)y} \mathrm{d}y. \tag{6b}$$

This representation reveals that u is actually just an integral of solutions q to *local* Laplace-type problems. A solution method employing a discretization of the above formula then only requires solves of (classical) local PDE's in order to solve the nonlocal problem (5). The straightforward way to compute the solution via (6b) is to approximate the integral with a quadrature rule. This would require computing solutions $q(\cdot; y)$ to the PDE (6a) for many values of the parameter y.

More precisely, let $\{y_m, \tau_m\}_{m=1}^M$ be a quadrature rule for approximating the integral in (6b). We will give precise choices for this quadrature rule soon. Then we can approximate the solution $u(s)$ as

$$u(s) \approx u_M(s) := \sum_{m=1}^{M} \tau_m q(y_m) e^{(1-s)y_m}.$$

One then needs only to compute the ensemble of functions $\{q(y_m)\}_{m=1}^M$, which are solutions to local PDEs, in order to approximate the solution to the fractional problem. This is the particular approach adopted in [11], wherein a sinc quadrature rule is adopted, and associated error bounds are derived.

The observation we make in this paper is that the approach above requires approximately M times the work of solving a local problem; when M is large (which can be required when s is small), this may become computationally prohibitive, see for instance [42]. However, there are by-now standard model reduction approaches that allow one to efficiently compute solutions to parameterized PDE's when the number of queries M is very large; one such method that is directly applicable (to some extent) here is the reduced basis method, which we exploit in Section 5. The next section expresses (6) in a more computationally robust formulation, and proposes a new kind of quadrature for y-discretization. We observe in Section 6 that our new quadrature approach is much more efficient than the most efficient strategy considered in [11].

3. Fractional Laplace solutions via integral formulation.

Recall that given $s \in (0,1)$, we seek to evaluate (6b), which defines the solution $u(s)$ to the fractional Laplace problem (5). In this section we describe our algorithm for doing so, which discretizes the y variable using quadrature. The main components of this algorithm come in two stages: first we describe a partitioning formulation for the y variable, followed by a quadrature discretization of the y integral.

The approach described in this section augments the approach presented in [11]; our improvements include s-independent stability in both the continuous and discrete case. For small values of s and large y-quadrature rule size, the algorithm in [11] results in discrete operators whose norm becomes very large, which can be problematic for numerical implementation. In our reformulation, the discrete operators are uniformly bounded in s (for any quadrature rule). Second, we replace the sinc quadrature in [11] with a Gaussian quadrature rule. (The authors in [11] also propose using Gaussian quadrature rules, but theirs is a composite rule, whereas ours is a global rule and is designed differently.) Our results in section 6 indicate that this quadrature rule is substantially more efficient than sinc quadrature.

The sections below perform the following tasks:

- Section 3.1 rewrites the integral formulation (6b) in a form that we actually discretize. Lemma 3.1 is this rewritten expression. This rewritten expression allows one to easily observe L^2 stability results for solutions. This is Proposition 1. These stability results can naturally also be observed from (6b), to analyze the algorithm it is more convenient to use the rewritten form.
- Section 3.2 imposes a Galerkin-based spatial discretization on certain local auxilliary PDEs, which translates into a spatial discretization for the fractional solution u. Proposition 2 establishes that the discrete solution inherits the stability of the continuous problem.

- Section 3.3 introduces the particular global quadrature rule that we employ to discretize the y variable, and establishes a stability result for this fully discrete scheme in Proposition 3.

3.1. **Partitioning of the y variable.** To make computations numerically stable, we split our parameterized problem (6a) into regions $y \in (-\infty, 0]$ and $y \in [0, \infty)$. To accomplish this, we introduce a new parameterized PDE for a solution w that is closely related (but not identical) to q from (6a)

$$\begin{aligned} -\alpha\Delta w(\alpha,\beta) + \beta w(\alpha,\beta) &= f, && x \in \Omega \\ w(\alpha,\beta) &= 0, && x \in \partial\Omega. \end{aligned} \tag{7}$$

where $(\alpha, \beta) \in (0, \infty) \times [0, \infty)$ is the parameter. In our computational setting, we will only require $(\alpha, \beta) \in (0, 1]^2$. Comparing (7) with (6a), we see that $q(y) = w(1, e^y)$. We now define $w_\pm(y)$ as two specializations of w that will be used in the following:

$$w_-(y) \coloneqq w(1, e^{-y}) = q(-y), \quad w_+(y) \coloneqq w(e^{-y}, 1) = e^y q(y), \quad y \in [0, \infty). \tag{8a}$$

$$(-\Delta + e^{-y})w_-(y) = f, \qquad (-e^{-y}\Delta + 1)w_+(y) = f, \tag{8b}$$

for $x \in \Omega$, with boundary conditions $w_\pm(y) = 0$ for $x \in \partial\Omega$. We note that for $y \in [0, \infty)$, we have $q(y) = e^{-y} w_+(y)$. We can formulate the solution to (5) in terms of these new quantities.

Lemma 3.1. *The solution $u(s)$ to (5) is given by*

$$\begin{aligned} u(s) &= \sum_{\sigma \in \{-,+\}} \beta_0(s_\sigma) \int_0^\infty w_\sigma\left(\frac{y}{s_\sigma}\right) W(y)\mathrm{d}y \\ &= \beta_0(s_-) \int_0^\infty w_-\left(\frac{y}{s_-}\right) W(y)\mathrm{d}y + \beta_0(s_+) \int_0^\infty w_+\left(\frac{y}{s_+}\right) W(y)\mathrm{d}y, \end{aligned} \tag{9}$$

where β_0, $s_\pm$, and W are defined as

$$\beta_0(s) \coloneqq \beta(s)/s = \frac{\sin(\pi s)}{\pi s} = \operatorname{sinc}(s), \qquad s_\pm \coloneqq \frac{1}{2} \pm \left(s - \frac{1}{2}\right) \qquad W(y) \coloneqq e^{-y}.$$

Proof. Beginning with (6b), we have

$$\begin{aligned} u &= \beta(s) \int_{-\infty}^0 e^{(1-s)y} w(1, e^y)\mathrm{d}y + \beta(s) \int_0^\infty e^{(1-s)y} w(1, e^y)\mathrm{d}y \\ &= (1-s)\beta_0(1-s) \int_0^\infty w_-(y) \exp(-(1-s)y)\mathrm{d}y + s\beta_0(s) \int_0^\infty w_+(y) \exp(-sy)\mathrm{d}y \\ &= \beta_0(1-s) \int_0^\infty w_-\left(\frac{y}{1-s}\right) W(y)\mathrm{d}y + \beta_0(s) \int_0^\infty w_+\left(\frac{y}{s}\right) W(y)\mathrm{d}y, \end{aligned} \tag{10}$$

completing the proof. □

Given the domain Ω, we define C_Ω as the domain's Poincaré constant, i.e., the smallest constant such that for every $v \in H_0^1(\Omega)$,

$$\|v\| \le C_\Omega \|\nabla v\|. \tag{11}$$

Note that $C_\Omega = 1/\lambda_1^2$, where λ_1 is the minimal eigenvalue of the Laplacian from (2). The space $H_0^1(\Omega)$ is the standard Sobolev space of zero-trace $L^2(\Omega)$ functions

whose gradients are also in L^2. In what follows, we will also need the following quantity,

$$\widetilde{C}_\Omega^2 \coloneqq \max\left(1, C_\Omega^2\right), \tag{12}$$

which also depends only on Ω. Fixing (α, β), the weak formulation of (7) seeks a solution $w(\alpha,\beta) = w \in H_0^1(\Omega)$ as the unique function satisfying the Galerkin formulation,

$$a(w, v; \alpha, \beta) \coloneqq \langle f, v\rangle, \qquad \forall\, v \in H_0^1(\Omega), \tag{13}$$

with the bilinear form $a(\cdot,\cdot;\alpha,\beta)$ defined as

$$a(w, v; \alpha, \beta) \coloneqq \alpha\langle \nabla w, \nabla v\rangle + \beta\langle w, v\rangle. \tag{14}$$

With $\alpha > 0$ and $\beta \geq 0$, the Poincarè inequality (11) ensures that the coercivity property $a(v,v;y) \geq \alpha\|\nabla v\|^2_{L^2(\Omega)} \geq c\|v\|_{H_0^1(\Omega)}$ holds for some $c > 0$ uniformly in β, so that standard Lax-Milgram theory then yields a unique $H_0^1(\Omega)$ solution.

With this setup, we can demonstrate the utility of a formula like (9) by deriving an *s-independent* L^2 stability estimate for solutions to (5).

Proposition 1. *Assume $f \in L^2$. Then*

$$\sup_{s\in(0,1)} \|u(s)\| \leq \frac{4\widetilde{C}_\Omega^2}{\pi}\|f\|.$$

Proof. The function $w_+(y) \in H_0^1(\Omega)$ is the unique solution to

$$a\left(w_+(y), v; e^{-y}, 1\right) = \langle f, v\rangle, \qquad \forall\, v \in H_0^1(\Omega).$$

Taking $v = w_+(y)$ and using the Cauchy-Schwarz and Poincaré inequalities results in

$$\begin{aligned}\|f\|\|w_+(y)\| \geq \langle f, w_+(y)\rangle &= e^{-y}\langle \nabla w_+(y), \nabla w_+(y)\rangle + \langle w_+(y), w_+(y)\rangle \\ &\geq \left(1 + \frac{e^{-y}}{C_\Omega^2}\right)\|w_+(y)\|^2 \geq \|w_+(y)\|^2.\end{aligned}$$

We thus obtain

$$\sup_{y\geq 0}\|w_+(y)\| \leq \|f\|. \tag{15a}$$

A similar computation for w_- shows that

$$\sup_{y\geq 0}\|w_-(y)\| \leq C_\Omega^2\|f\|. \tag{15b}$$

Therefore, taking the L^2 norm in (9) and using the triangle inequality yields

$$\begin{aligned}\|u(s)\| &\leq \beta_0(s_-)\int_0^\infty \left\|w_-\left(\frac{y}{s_-}\right)\right\| W(y)\mathrm{d}y + \beta_0(s_+)\int_0^\infty \left\|w_+\left(\frac{y}{s_+}\right)\right\| W(y)\mathrm{d}y \\ &\overset{(15)}{\leq} \beta_0(s_-)C_\Omega^2\|f\|\int_0^\infty W(y)\mathrm{d}y + \beta_0(s_+)\|f\|\int_0^\infty W(y)\mathrm{d}y \\ &\leq \max(1, C_\Omega^2)\|f\|\left[\beta_0(s_-) + \beta_0(s_+)\right],\end{aligned}$$

where the third inequality uses the fact that W is a probability density on $[0,\infty)$. From the above, we immediately obtain the desired result by noting that

$$\beta_0(s_-) + \beta_0(s_+) = \frac{\sin(\pi s)}{\pi s(1-s)} \leq \frac{4}{\pi}, \qquad s \in (0,1).$$

The proof is complete. □

Note that formulas like (9) are not the only way to show such stability results, but Proposition 1 is meant to illustrate one possible use of these relations. Our main utility of such results will be to design a numerical scheme.

3.2. **Spatial discretization.** In this section we apply a spatial discretization to the result of Lemma 3.1. We proceed to discretize (14) using a finite element method. Let T_Ω be a conforming triangulation of Ω with K elements. We assume each element in the triangulation is isoparametrically equivalent to a standard canonical triangle/tetrahedron. For a fixed polynomial degree $k \geq 1$, we define the finite element space

$$V = \left\{ v \in C\left(\overline{\Omega}\right) \;\middle|\; v|_e \in P_k\left(e\right) \;\; \forall e \in T_\Omega, \; v|_{\partial\Omega} = 0 \right\},$$

where $P_k(e)$ is the space of polynomials of degree k or less over the element $e \in T_\Omega$. Let $\mathcal{N} = \dim V$. The finite element-discretized version of (13) is the Galerkin formulation seeking $w^{\mathcal{N}} \in V$ satisfying

$$a\left(w^{\mathcal{N}}, v; \alpha, \beta\right) = \langle f, v \rangle, \qquad \forall v \in V. \tag{16}$$

Let $w^{\mathcal{N}}(\alpha, \beta)$ be expressed as a linear expansion,

$$w^{\mathcal{N}}(\alpha, \beta) = \sum_{n=1}^{\mathcal{N}} w_n^{\mathcal{N}}(\alpha, \beta)\psi_n, \tag{17}$$

where $\{\psi_n\}_{n=1}^{\mathcal{N}}$ is a basis for V, e.g., a basis comprised of compactly supported piecewise polynomials. Collecting the linear degrees of freedom of $w^{\mathcal{N}}(y) \in V$ in the $\mathcal{N}$-dimensional vector $\boldsymbol{w}^{\mathcal{N}}$, then this vector satisfies the linear system,

$$\boldsymbol{A}(\alpha, \beta)\boldsymbol{w}^{\mathcal{N}}(\alpha, \beta) = \boldsymbol{f}, \qquad (\boldsymbol{f})_j = \langle f, \psi_j \rangle, \tag{18a}$$

where the $\mathcal{N} \times \mathcal{N}$ matrix $\boldsymbol{A}(\alpha, \beta)$ has entries,

$$(\boldsymbol{A}(\alpha, \beta))_{j,k} = a\left(\psi_k, \psi_j\right) = \alpha \left\langle \nabla\psi_k, \nabla\psi_j \right\rangle + \beta \left\langle \psi_k, \psi_j \right\rangle \coloneqq \alpha(\boldsymbol{S})_{j,k} + \beta(\boldsymbol{M})_{j,k}, \tag{18b}$$

for $j, k = 1, \ldots, \mathcal{N}$. Above we have defined the $\mathcal{N} \times \mathcal{N}$ y-independent stiffness and mass matrices $\boldsymbol{S}$ and $\boldsymbol{M}$, respectively. Both $\boldsymbol{S}$ and $\boldsymbol{M}$ are symmetric and positive-definite.

The matrices $\boldsymbol{S}$ and $\boldsymbol{M}$ can be used to define a "discretized" Poincaré constant $C_{\mathcal{N}}$ by using a standard Rayleigh quotient argument:

$$\frac{1}{C_\Omega^2} = \inf_{v \in H_0^1 \setminus \{0\}} \frac{\|\nabla v\|^2}{\|v\|^2} \leq \inf_{v \in V} \frac{\|\nabla v\|^2}{\|v\|^2} = \inf_{\boldsymbol{v} \in \mathbb{R}^{\mathcal{N}} \setminus \{\boldsymbol{0}\}} \frac{\boldsymbol{v}^T \boldsymbol{S} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{M} \boldsymbol{v}} = \lambda_{\min}\left(\boldsymbol{S}, \boldsymbol{M}\right) =: \frac{1}{C_{\mathcal{N}}^2}, \tag{19}$$

hence leading to the inequalities,

$$C_{\mathcal{N}} \leq C_\Omega, \qquad \max\left(1, C_{\mathcal{N}}^2\right) \overset{(12)}{\leq} \widetilde{C}_\Omega^2. \tag{20}$$

Our estimates for x-discrete quantities will involve $C_{\mathcal{N}}$, but we will sometimes use the above inequality to bound quantities in terms of C_Ω. Bounds involving C_Ω emphasize independence of the x-discretization. Bounds involving $C_{\mathcal{N}}$ emphasize the explicit computability of the bounds, since $C_{\mathcal{N}}$ is equal to an extremal eigenvalue of finite element matrices, which is computable with iterative eigenvalue solvers.

The maximum generalized eigenvalue of $(\boldsymbol{S}, \boldsymbol{M})$ will also play a small role in our estimates. In analogy with (19) we define

$$\frac{1}{K_{\mathcal{N}}^2} \coloneqq \lambda_{\max}(\boldsymbol{S}, \boldsymbol{M}). \tag{21}$$

Note that $K_{\mathcal{N}}$ in general tends to 0 as $\mathcal{N} \uparrow \infty$.

From the discretization of w we derive discretizations of $w_{\pm}(y)$ defined in (8). Thus, finite element discretizations for $w_{\pm}$ are specializations of those for w. In particular, we define

$$w_{-}^{\mathcal{N}}(y) \coloneqq w^{\mathcal{N}}\left(1, e^{-y}\right) \in V, \qquad w_{+}^{\mathcal{N}}(y) \coloneqq w^{\mathcal{N}}\left(e^{-y}, 1\right) \in V.$$

Denote by $\boldsymbol{w}_{\pm}^{\mathcal{N}}(y)$ the $\mathcal{N}$-dimensional vectors that are solutions to the linear systems,

$$\left(\boldsymbol{S} + e^{-y}\boldsymbol{M}\right)\boldsymbol{w}_{-}^{\mathcal{N}}(y) = \boldsymbol{f}, \qquad \left(e^{-y}\boldsymbol{S} + \boldsymbol{M}\right)\boldsymbol{w}_{+}^{\mathcal{N}}(y) = \boldsymbol{f}, \qquad y \in [0, \infty), \tag{22}$$

so that, akin to (17), we have

$$w_{\pm}^{\mathcal{N}}(y) = \sum_{j=1}^{\mathcal{N}} w_{j,\pm}^{\mathcal{N}}(y)\psi_j, \qquad \boldsymbol{w}_{\pm}^{\mathcal{N}}(y) = \left(w_{1,\pm}^{\mathcal{N}}(y), \ldots, w_{\mathcal{N},\pm}^{\mathcal{N}}(y)\right)^T.$$

We can now codify the fact that the solutions $w_{\pm}^{\mathcal{N}}(y)$ are L^2-stable *uniformly* in y.

Lemma 3.2. *Assume $f \in L^2(\Omega)$. Then*

$$\sup_{y \geq 0} \|w_{+}^{\mathcal{N}}(y)\| \leq \|f\|, \tag{23a}$$

$$\sup_{y \geq 0} \|w_{-}^{\mathcal{N}}(y)\| \leq C_{\mathcal{N}}^2 \|f\|. \tag{23b}$$

Proof. The result can be obtained by considering the discrete form (22). To begin we relate the L^2 norm of f to the Euclidean ℓ^2 norm of $\boldsymbol{f}$. Let P_V denote the L^2-orthogonal projector onto V. Then:

$$f_j = \langle f, \psi_j \rangle \implies \|P_V f\|^2 = \boldsymbol{f}^T \boldsymbol{M}^{-1} \boldsymbol{f}.$$

$$w_{+}^{\mathcal{N}}(y) = \sum_{j=1}^{\mathcal{N}} w_{j,+}^{\mathcal{N}}(y)\psi_j \implies \|w_{+}^{\mathcal{N}}(y)\|^2 = \left(\boldsymbol{w}_{+}^{\mathcal{N}}\right)^T \boldsymbol{M}\left(\boldsymbol{w}_{+}^{\mathcal{N}}\right).$$

Thus we have

$$\begin{aligned} \|\boldsymbol{f}\|_{\boldsymbol{M}^{-1}}^2 = \|P_V f\|^2 \leq \|f\|^2, \\ \|\boldsymbol{w}_{+}^{\mathcal{N}}\|_{\boldsymbol{M}}^2 = \|w_{+}^{\mathcal{N}}\|^2, \end{aligned} \tag{24}$$

Now since $\boldsymbol{M}$ is symmetric and positive-definite, it has a unique symmetric positive-definite square root $\boldsymbol{M}^{1/2}$. Thus:

$$\left(e^{-y}\boldsymbol{S} + \boldsymbol{M}\right)\boldsymbol{w}_{+}^{\mathcal{N}}(y) = \boldsymbol{f} \implies \left(e^{-y}\boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{M}^{-1/2} + \boldsymbol{I}\right)\left(\boldsymbol{M}^{1/2}\boldsymbol{w}_{+}^{\mathcal{N}}(y)\right) = \boldsymbol{M}^{-1/2}\boldsymbol{f}.$$

This in turn implies:

$$\|\boldsymbol{w}_{+}^{\mathcal{N}}\|_{\boldsymbol{M}} \leq \frac{1}{\lambda_{\min}\left(e^{-y}\boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{M}^{-1/2} + \boldsymbol{I}\right)} \|\boldsymbol{f}\|_{\boldsymbol{M}^{-1}}.$$

Since $\boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{M}^{-1/2}$ is symmetric and positive-definite, we have

$$\lambda_{\min}\left(e^{-y}\boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{M}^{-1/2} + \boldsymbol{I}\right) \geq \lambda_{\min}(\boldsymbol{I}) = 1.$$

Therefore,

$$\|\boldsymbol{w}_+^{\mathcal{N}}\|_{\boldsymbol{M}} \leq \frac{\|\boldsymbol{f}\|_{\boldsymbol{M}^{-1}}}{\lambda_{\min}\left(e^{-y}\boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{M}^{-1/2}+\boldsymbol{I}\right)} \leq \|\boldsymbol{f}\|_{\boldsymbol{M}^{-1}},$$

which, when combined with (24) yields (23a). A similar computation for $w_-(y)$ yields (23b) by using the definition of $C_{\mathcal{N}}$ in (19). □

The result above gives the stability of an algorithm that uses $w_\pm^{\mathcal{N}}(y)$ as a spatial discretization. In particular, consider the following semi-discrete approximation of $u(s)$,

$$\widetilde{\boldsymbol{u}}^{\mathcal{N}}(s) \coloneqq \sum_{\sigma\in\{+,-\}} \beta_0(s_\sigma)\int_0^\infty \boldsymbol{w}_\sigma^{\mathcal{N}}\left(\frac{y}{s_\sigma}\right)W(y)\mathrm{d}y, \quad \widetilde{u}^{\mathcal{N}}(s) \coloneqq \sum_{j=1}^{\mathcal{N}} \widetilde{u}_j^{\mathcal{N}}(s)\psi_j \in V. \tag{25}$$

A fully discrete scheme, introduced in the next section, would discretize the y variable. The following result mirrors the stability estimate of Proposition 1, showing s-uniform L^2 stability of the semi-discrete solution.

Proposition 2. *Assume $f \in L^2$. Then*

$$\sup_{s\in(0,1)} \|\widetilde{u}^{\mathcal{N}}(s)\| \leq \frac{4\widetilde{C}_\Omega^2}{\pi}\|f\|.$$

The proof, which we omit, is almost exactly the same as for Proposition 1 using the discrete stability estimates (23); the only novelty is the need to exercise the inequality (19).

Having described the spatial discretization, we now proceed to describe a discretization for the y-integrals in (9).

3.3. **Quadrature for the y-integral.** We will use an M-point W-Gaussian quadrature rule to discretize the integrals in (9). The weight function $W(y)$ is a weight function associated with a classical family of orthogonal polynomials: Laguerre polynomials. Let $\{p_n\}_{n\geq 0}$ denote the family of Laguerre polynomials, orthonormal under the weight W, i.e.,

$$\int_0^\infty p_n(y)p_m(y)W(y)\mathrm{d}y = \delta_{m,n}, \qquad m,n\in\mathbb{N}_0,$$

where $\delta_{m,n}$ is the Kronecker delta function. Like all orthogonal polynomials, the family $\{p_n\}_{n\geq 0}$ satisfies a three-term recurrence formula,

$$yp_n(y) = b_{n+1}p_{n+1}(y) + a_{n+1}p_n(y) + b_n p_{n-1}(y), \qquad n\geq 0,$$

with $p_0 \equiv 1$ and $p_{-1}\equiv 0$. The recurrence coefficients (a_n, b_n) are explicitly known:

$$b_0 = 1, \qquad b_n = n \qquad a_n = 2n-1, \qquad n\geq 1.$$

Among the properties of orthogonal polynomial families is the existence of a unique M-point quadrature rule with optimal polynomial exactness, the *Gaussian* quadrature rule. This rule has abscissae and weights, $(y_m,\tau_m)_{m=1}^M$, respectively, and integrates polynomials up to degree $2M-1$ exactly:

$$\int_0^\infty p(y)W(y)\mathrm{d}y = \sum_{j=1}^M \tau_j p(y_j), \qquad p\in\mathrm{span}\{1,y,\ldots,y^{2M-1}\}.$$

Although y_j and τ_j depend on the value of M, we omit this explicit notational dependence. This rule can also be easily computed with knowledge of the recurrence coefficients. Defining the $M \times M$ symmetric tridiagonal Jacobi matrix,

$$\boldsymbol{J}_M \coloneqq \begin{pmatrix} a_1 & b_1 & & & \\ b_1 & a_2 & b_2 & & \\ & \ddots & \ddots & \ddots & \\ & & & b_{M-1} & a_M \end{pmatrix},$$

consider its associated eigenvalue decomposition,

$$\boldsymbol{J}_M = \boldsymbol{V}\boldsymbol{L}\boldsymbol{V}^T, \quad \boldsymbol{L} = \operatorname{diag}(\ell_1, \ldots, \ell_M), \quad \boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \cdots & \boldsymbol{v}_M \end{bmatrix}, \tag{26a}$$

where $\boldsymbol{V}$ is unitary since $\boldsymbol{J}_M$ is symmetric. The Gaussian quadrature rule can be computed from these quantities. In particular,

$$y_j = \ell_j, \qquad \tau_j = v_{j,1}^2 b_0^2, \tag{26b}$$

where $v_{j,1}$ is the first component of the vector $\boldsymbol{v}_j$. To compute an M-point quadrature rule for W thus requires a size-M eigenvalue computation. Since W is a probability density function, then likewise $\sum_{j=1}^{M} \tau_j = 1$, and furthermore each τ_j is non-negative by (26).

The cost of computing the above M-point quadrature rule is dominated by the cost of computing the spectrum of symmetric tridiagonal matrix. For computing only the eigenvalues, many algorithms can compute this with $\mathcal{O}(M^2)$ cost [36]. Computation of the weights can also be accomplished in $\mathcal{O}(M^2)$ time by using an explicit formula for the weights in terms of the nodes (eigenvalues of $\boldsymbol{J}_M$). Thus, the entire computation can be completed with $\mathcal{O}(M^2)$ effort.

Now let M_- and M_+ be the number of quadrature points used to approximate the "$-$" and "$+$" integrals in (9), respectively.[1] This results in two sets of W-Gaussian quadrature rules,

$$(y_{j,-}, \tau_{j,-})_{j=1}^{M_-}, \qquad (y_{j,+}, \tau_{j,+})_{j=1}^{M_+}.$$

We apply these rules to the integrals in (25), resulting in the fully discrete approximation,

$$\boldsymbol{u}^{\mathcal{N}}(s) \coloneqq \sum_{\sigma \in \{+,-\}} \beta_0(s_\sigma) \sum_{j=1}^{M_\sigma} \boldsymbol{w}_\sigma^{\mathcal{N}} \left(\frac{y_{j,\sigma}}{s_\sigma} \right) W(y), \quad u^{\mathcal{N}}(s) \coloneqq \sum_{j=1}^{\mathcal{N}} u_j^{\mathcal{N}}(s) \psi_j \in V \tag{27}$$

We emphasize that the y-discretization does not suffer from any numerical instabilities as $s \uparrow 1$ or $s \downarrow 0$: the weights $\tau_{j,\pm}$ are positive, no larger than 1, and independent of s, $\beta_0(1-s)$ and $\beta_0(s)$ are just sinc functions, and $w_\pm(y)$ has bounded L^2 norm for all $y \geq 0$, i.e., for all inputs. The following codifies this stability.

Proposition 3. *Assume $f \in L^2(\Omega)$. Then*

$$\sup_{s \in (0,1)} \|u^{\mathcal{N}}(s)\| \leq \frac{4\widetilde{C}_\Omega^2}{\pi} \|f\|, \tag{28}$$

Proof. The proof is very similar to the proof for proposition (1). Take the $\|\cdot\|_{\boldsymbol{M}}$ norm on both sides of (27), use the triangle inequality and (23), and note that the quadrature weights $\tau_{j,\pm}$ all satisfy $0 \leq \tau_{j,\pm} \leq 1$ since they are all positive and W is

[1] We describe in section 6 how these values are chosen.

a probability density. Note that (28) holds for any quadrature rule for the y variable if augmented by a multiplicative constant equal to the quadrature condition number (sum of absolute value of weights). □

Just as in [11], assuming we have the ability to compute $\boldsymbol{w}_{\pm}^{\mathcal{N}}$, then the formulation above immediately yields an algorithm to compute $u^{\mathcal{N}}(s)$ via (27).

Remark 1. All of our results extend to the case when we solve (5) but replacing $-\Delta$ with a general elliptic operator $\mathcal{E}$ that (i) satisfies the coercivity condition $\langle \mathcal{E}v, v\rangle \geq \alpha \|v\|_{H_0^1}$ for $\alpha > 0$ and (ii) can be associated with a symmetric variational bilinear form. In this more general case, we need only replace all the instances of $\widetilde{C}_{\Omega}^2$ with $\widetilde{C}_{\Omega}^2/\alpha$. The matrix $\boldsymbol{S}$ should likewise be replaced with the associated matrix defined from the bilinear form of $\mathcal{E}$.

3.4. **Algorithm summary.** The sections above identify an algorithm for computing $u^{\mathcal{N}}(s)$ in (27). We summarize the procedure in Algorithm 1. The discrete solution adheres to the stability bound in Proposition 3. Note that we have not yet described how one should decide on values for $M_{\pm}$. We provide a concrete computational strategy for accomplishing this in the next section. However, one of our goals in our numerical results section is to compare our algorithm to existing ones, which determine $M_{\pm}$ using (48).

Algorithm 1 GQ algorithm: Produces solution to the fractional Laplace problem (5).

Precondition: Availability of a discrete solution $\boldsymbol{w}^{\mathcal{N}}(\alpha, \beta)$ from the formulation (16).

function FracLapGQ(s)
  Determine $M_{\pm}$, e.g., via (48).
  Generate quadrature rules $(y_{j,\pm}, \tau_{j,\pm})_{j=1}^{M_{\pm}}$ using (26).
  for $j \leftarrow 1$ to M_- **do**
    Compute $\boldsymbol{w}_-^{\mathcal{N}}\left(\frac{y_{j,-}}{1-s}\right) = \boldsymbol{w}^{\mathcal{N}}\left(1, \exp\left(-\frac{y_{j,-}}{1-s}\right)\right)$ from (22) or (18).
  for $j \leftarrow 1$ to M_+ **do**
    Compute $\boldsymbol{w}_+^{\mathcal{N}}\left(\frac{y_{j,+}}{s}\right) = \boldsymbol{w}^{\mathcal{N}}\left(\exp\left(-\frac{y_{j,+}}{s}\right), 1\right)$ from (22) or (18).
  Compute $\boldsymbol{u}^{\mathcal{N}}(s)$ from (27).
  return $\boldsymbol{u}^{\mathcal{N}}(s)$

4. **Error due to quadrature discretization.** The formulation (27) is our numerical approximation to compute solutions to (5). This formulation is a discretization over both the x and y variables (via a finite element formulation and a quadrature rule, respectively). To understand the error that the y discretization contributes, we analyze the discrepancy between $\widetilde{u}^{\mathcal{N}}(s)$ and $u^{\mathcal{N}}(s)$.

To proceed, we need an auxiliary function that measures the absolute error between a size-M quadrature rule $(y_j, \tau_j)_{j=1}^{M}$ and the exact integral applied to a particular function:

$$g_M(a,b) \coloneqq \left| \int_0^{\infty} \frac{W(y)}{1+ae^{-by}} \mathrm{d}y - \sum_{j=1}^{M} \frac{\tau_j}{1+ae^{-by_j}} \right|, \quad (a,b) \in (0,\infty) \times (1,\infty). \tag{29}$$

We also need to define intervals on the real line enclosing the spectrum of some discretized operators. Recalling the definitions of $C_{\mathcal{N}}$ and $K_{\mathcal{N}}$ in (19) and (21), respectively, define intervals I and $I_{\pm}$ as

$$I_{-} = \left[K_{\mathcal{N}}^2, C_{\mathcal{N}}^2\right] \subset (0,\infty), \qquad I_{+} = \left[\frac{1}{C_{\mathcal{N}}^2}, \frac{1}{K_{\mathcal{N}}^2}\right] \subset (0,\infty), \qquad I = I_{-} \bigcup I_{+}$$

The error committed by the quadrature rule can be understood by studying the quantity,

$$G_{\pm}(M,s) \coloneqq \beta_0\left(s_{\pm}\right) \sup_{a \in I_{\pm}} g_M\left(a, \frac{1}{s_{\pm}}\right). \tag{30}$$

The precise statement is as follows.

Proposition 4. *Assume $f \in L^2$. Then for each $s \in (0,1)$,*

$$\left\|u^{\mathcal{N}}(s) - \widetilde{u}^{\mathcal{N}}(s)\right\| \leq \widetilde{C}_{\Omega}^2 \|f\| \sum_{\sigma \in \{+,-\}} G_{\sigma}\left(M_{\sigma}, s\right) \tag{31}$$

and therefore,

$$\sup_{s \in (0,1)} \left\|u^{\mathcal{N}}(s) - \widetilde{u}^{\mathcal{N}}(s)\right\| \leq \frac{4\widetilde{C}_{\Omega}^2 \|f\|}{\pi} \max_{\sigma \in \{+,-\}} \sup_{a \in I_{\sigma}, b \in (1,\infty)} g_{M_{\sigma}}(a,b). \tag{32}$$

Proof. The same argument that produces the relations (24) implies that

$$\left\|u^{\mathcal{N}}(s) - \widetilde{u}^{\mathcal{N}}(s)\right\|_{L^2} = \left\|\boldsymbol{u}^{\mathcal{N}}(s) - \widetilde{\boldsymbol{u}}^{\mathcal{N}}(s)\right\|_{\boldsymbol{M}},$$

so we proceed to study the quantity on the right-hand side. The difference between the "$-$" integral contributions in $\boldsymbol{u}^{\mathcal{N}}(s) - \widetilde{\boldsymbol{u}}^{\mathcal{N}}(s)$ is proportional to

$$\int_0^{\infty} \boldsymbol{w}_{-}^{\mathcal{N}}\left(\frac{y}{s_{-}}\right) W(y)\mathrm{d}y - \sum_{j=1}^{M_{-}} \boldsymbol{w}_{-}^{\mathcal{N}}\left(\frac{y_{j,-}}{s_{-}}\right) \tau_{j,-}.$$

We can express the solution $\boldsymbol{w}_{-}^{\mathcal{N}}(y)$ as

$$\boldsymbol{w}_{-}^{\mathcal{N}}(y) = \left[\boldsymbol{S} + e^{-y/s_{-}}\boldsymbol{M}\right]^{-1} \boldsymbol{f} = \boldsymbol{M}^{-1/2}\left[\boldsymbol{A} + e^{-y/s_{-}}\boldsymbol{I}\right]^{-1} \boldsymbol{M}^{-1/2}\boldsymbol{f},$$

where we have defined $\boldsymbol{A} \coloneqq \boldsymbol{M}^{-1/2}\boldsymbol{S}\boldsymbol{M}^{-1/2}$. The matrix $\boldsymbol{A}$ is symmetric and positive definite, and thus has an eigenvalue decomposition

$$\boldsymbol{A} = \boldsymbol{W}\boldsymbol{T}\boldsymbol{W}^T, \qquad \boldsymbol{W}\boldsymbol{W}^T = \boldsymbol{I}, \qquad \boldsymbol{T} = \mathrm{diag}(t_1, \ldots, t_{\mathcal{N}}).$$

Then further manipulation of the $\boldsymbol{w}_{-}^{\mathcal{N}}(y)$ expression yields

$$\boldsymbol{w}_{-}^{\mathcal{N}}(y) = \boldsymbol{M}^{-1/2}\boldsymbol{W}\boldsymbol{H}\left(\frac{y}{s_{-}}\right)\boldsymbol{W}^T\boldsymbol{M}^{-1/2}\boldsymbol{f},$$

where $\boldsymbol{H}(y)$ is a diagonal matrix having entries

$$\left(\boldsymbol{H}(y)\right)_{j,j} = \frac{1}{t_j + e^{-y}} = \frac{1}{t_j}\left[1 + \frac{1}{t_j}e^{-y}\right]^{-1}.$$

Thus,

$$\begin{aligned}
&\left\| \int_0^\infty \boldsymbol{w}_-^{\mathcal{N}}\left(\frac{y}{s_-}\right) \mathrm{d}y - \sum_{j=1}^{M_-} \boldsymbol{w}_-^{\mathcal{N}}\left(\frac{y_{j,-}}{s_-}\right) \tau_{j,-} \right\|_{\boldsymbol{M}} \\
&\quad \leq \left\| \boldsymbol{W} \left[\int_0^\infty \boldsymbol{H}(y/s_-) W(y) \mathrm{d}y - \sum_{j=1}^{M_-} \tau_{j,-} \boldsymbol{H}\left(\frac{y_{j,-}}{s_-}\right) \right] \boldsymbol{W}^T \right\| \left\| \boldsymbol{M}^{-1/2} \boldsymbol{f} \right\| \\
&\quad = \left\| \left[\int_0^\infty \boldsymbol{H}(y/s_-) W(y) \mathrm{d}y - \sum_{j=1}^{M_-} \tau_{j,-} \boldsymbol{H}\left(\frac{y_{j,-}}{s_-}\right) \right] \right\| \|\boldsymbol{f}\|_{\boldsymbol{M}^{-1}} \\
&\quad \overset{(24)}{\leq} \left\| \left[\int_0^\infty \boldsymbol{H}(y/s_-) W(y) \mathrm{d}y - \sum_{j=1}^{M_-} \tau_{j,-} \boldsymbol{H}\left(\frac{y_{j,-}}{s_-}\right) \right] \right\| \|f\|_{L^2} \\
&\quad = \|f\|_{L^2} \max_{j=1,\dots,\mathcal{N}} \frac{1}{t_j} g_M\left(\frac{1}{t_j}, \frac{1}{s_-}\right),
\end{aligned}$$

The first inequality is sub-multiplicativity of the $\|\cdot\|$ matrix norm, and the first equality uses the invariance of the same norm under unitary transformations. A similar computation for the "+" quantities yields

$$\left\| \int_0^\infty \boldsymbol{w}_+^{\mathcal{N}}\left(\frac{y}{s_+}\right) \mathrm{d}y - \sum_{j=1}^{M_+} \boldsymbol{w}_+^{\mathcal{N}}\left(\frac{y_{j,+}}{s_+}\right) \tau_{j,+} \right\|_{\boldsymbol{M}} \leq \|f\|_{L^2} \max_{j=1,\dots,\mathcal{N}} g\left(t_j, \frac{1}{s_+}\right).$$

The combination of these results implies

$$\begin{aligned}
\left\|u^{\mathcal{N}}(s) - \widetilde{u}^{\mathcal{N}}(s)\right\|_{L^2} &\leq \beta_0(s_-) \left\| \int_0^\infty \boldsymbol{w}_-^{\mathcal{N}}\left(\frac{y}{s_-}\right) \mathrm{d}y - \sum_{j=1}^{M_-} \boldsymbol{w}_-^{\mathcal{N}}\left(\frac{y_{j,-}}{s_-}\right) \tau_{j,-} \right\|_{\boldsymbol{M}} \\
&\quad + \beta_0(s_+) \left\| \int_0^\infty \boldsymbol{w}_-^{\mathcal{N}}\left(\frac{y}{s_+}\right) \mathrm{d}y - \sum_{j=1}^{M_+} \boldsymbol{w}_+^{\mathcal{N}}\left(\frac{y_{j,+}}{s_+}\right) \tau_{j,+} \right\|_{\boldsymbol{M}} \\
&\leq \frac{\|f\|_{L^2}}{\lambda_{\min}(\boldsymbol{S}, \boldsymbol{M})} \sup_{a \in I_-} g_{M_-}\left(a, \frac{1}{s_-}\right) + \|f\|_{L^2} \sup_{a \in I_+} g_{M_+}\left(a, \frac{1}{s_+}\right).
\end{aligned}$$

Using the inequality (20) yields the result. □

The summation on the right-hand side of (31) can be computed independent of the data f, and requires only knowledge of the extremal eigenvalues of the discrete operator, cf. (19) and (21). While we cannot at present provide a theoretical estimate of this error, we numerically investigate the behavior of this error on $M_\pm$ in our numerical results section.

Remark 2. Comparing (31) with the stability bound (28) suggests that many of the factors in (31) appear due to bounding the error relative to $\|\widetilde{u}^{\mathcal{N}}\|_{L^2}$. Thus the supremum over g is the factor that arises due to the quadrature error.

Remark 3. The result (31) also shows that the error between $u^{\mathcal{N}}(s)$ and $\widetilde{u}^{\mathcal{N}}(s)$ is stable independent of s since

$$g_M(a,b) \leq \left| \int_0^\infty W(y) \right| + \left| \sum_{j=1}^{M} \tau_j \right| = 2,$$

uniformly in a, b, and M. Thus,

$$\sup_{s\in(0,1)} \left\| u^{\mathcal{N}}(s) - \widetilde{u}^{\mathcal{N}}(s) \right\|_{L^2} \leq \frac{8\widetilde{C}_\Omega^2}{\pi} \|f\|_{L^2}.$$

This again suggests that, independent of all discretization parameters, our numerical algorithm is stable. However, a rigorous convergence analysis for our quadrature rule is not yet available.

4.1. **Empirical behavior of quadrature error.** The main result from Proposition 4 is that the error in the fully discrete approximation (27) that is due to the y-quadrature discretization is computable without solving any PDE's, assuming that the extremal generalized eigenvalues of $(\boldsymbol{S}, \boldsymbol{M})$, coded in the quantities $C_{\mathcal{N}}$ and $K_{\mathcal{N}}$, are known. In particular, this implies that most details of the spatial discretization need not be utilized to understand the quadrature error; we only require extremal eigenvalues of discretized operators.

We empirically investigate the accuracy of the quadrature rule in this section. Throughout our tests, we will use the following values:

$$C_{\mathcal{N}}^2 = 2 \iff \lambda_{\min}(\boldsymbol{S}, \boldsymbol{M}) = \frac{1}{2}, \qquad K_{\mathcal{N}}^2 = \frac{1}{10^6} \iff \lambda_{\max}(\boldsymbol{S}, \boldsymbol{M}) = 10^6.$$

Since $C_{\mathcal{N}}$ is bounded above by the analytical Poincaré constant of the domain C_Ω, then choosing this $\mathcal{O}(1)$ quantity for $C_{\mathcal{N}}$ is reasonable. The choices above make the intervals $I_\pm$ defined in Proposition 4 explicit. The finite element discretization from our numerical experiments in Section 6 results in values $C_{\mathcal{N}}^2 = 0.0506$ and $K_{\mathcal{N}}^2 = 2.36 \times 10^{-6}$.

We note that $G_\pm$ in (30) can be numerically approximated for each (M, s) by replacing the supremum over a with the maximum over a discrete mesh. We compute the supremums in $G_\pm$ by discretizing the intervals $I_\pm$ with 200 logarithmically spaced points. (I.e., $\log I_\pm$ is replaced with 200 equispaced points.) This discretization then allows us to compute $G_\pm$, and hence allows us to compute approximations to the bound in (31). In Figure 1, we show the behavior of $G_\pm$ as a function of (M, s). We note that ensuring small values of G_+ requires more quadrature points when s is close to 0. In contrast, controlling G_- requires more quadrature points when s is close to 1. However, the behavior of G_+ for small $s_+ = s$ is more restrictive than the behavior of G_- for small $s_- = 1 - s$. Thus, we expect that G_+ is the term that requires more computational investment to guarantee a certain error level.

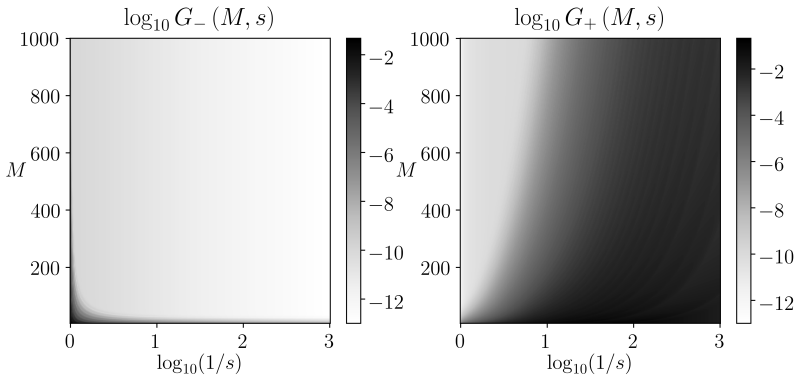

Figure 1. Values of $\log_{10} G_\pm$ defined in (30) as a function of (M, s). We show s-dependence as $\log_{10}(1/s)$ since the error behavior for small s is the most restrictive. We observe that, for fixed M, $G_\pm$ has large values when $s_\pm$ is small.

To explore this further, we define the smallest value[2] of M needed to assure a given error level δ:

$$\widetilde{M}(\delta, s) \coloneqq \widetilde{M}_- (\delta, s) + \widetilde{M}_+ (\delta, s) , \tag{33}$$

$$\widetilde{M}_\pm (\delta, s) \coloneqq \min \left\{ M \in \mathbb{N} \;\middle|\; G_\pm (M, s) \leq \frac{\delta}{2} \right\} . \tag{34}$$

For $\delta = 10^{-2}$, 10^{-4}, and 10^{-6}, we display the values of these $\widetilde{M}$ quantities in Figure 2. We see that for small values of s, the requisite number of points $\widetilde{M}$ scales like $1/s$. In particular, for small s, more effort (quadrature points) is allocated to $\widetilde{M}_+$, but for small $1 - s$, comparatively more effort is allocated to $\widetilde{M}_-$. Therefore, the

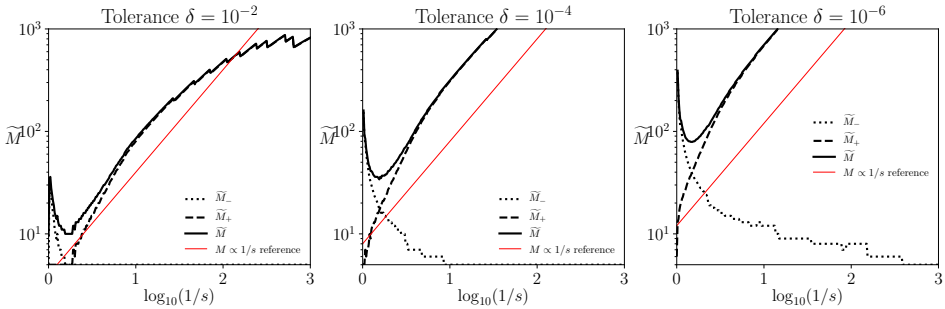

Figure 2. Values of $\widetilde{M}$ and $\widetilde{M}_\pm$ defined in (33) for various values of the tolerance δ. We show s-dependence as $\log_{10}(1/s)$ since the error behavior for small s is most restrictive. For visual reference, a $1/s$ curve is also plotted. We see that for small values of $s_\pm$, the corresponding value of $\widetilde{M}_\pm$ is large.

number of quadrature queries $M_- + M_+$ in the fully discrete scheme (27) can be

[2]To avoid computational effects of oscillating errors due to, e.g., even/odd parity of the quadrature rule, we actually compute the smallest value of M so that M, $M+1$, $M+2$, and $(M+3)$-point quadrature rules all achieve the stated accuracy requirement.

quite large when the fractional order s is very small. This general observation, including the $1/s$-type behavior shown in Figure 2, is consistent with earlier work [6]. Thus, the number of local PDE solutions $M = M_- + M_+$ needed to compute an accurate solution is large. This motivates a need to make these solves more efficient; we achieve this in the next section via model reduction.

5. **Model reduction for the integral formulation.** This section proposes an augmentation of the algorithm in the previous section. The cost of computing the fully discrete solution (27) is essentially $M_- + M_+$ queries of finite element solvers for $\boldsymbol{w}_\pm^{\mathcal{N}}$. In practice one can require $M_- + M_+ \sim 100$, cf. Figure 2 and earlier work [11, 42], resulting in a substantial computational cost if the cost of computing $\boldsymbol{w}_\pm^{\mathcal{N}}$ is high.

We observe that the formulations (8) for $w_\pm$ (and also (22) for the discrete counterparts $\boldsymbol{w}_\pm^{\mathcal{N}}$) are quintessential examples of parameterized PDE's where RBM algorithms are used to accelerate solution queries. Thus, RBM can be used to ameliorate the cost of performing $M_- + M_+$ queries of these PDE's. In RBM terminology, an available expensive discrete solution is called a *truth* solution. Thus, our truth solutions for the auxiliary PDE problem for $w_\pm$ are $w_\pm^{\mathcal{N}}$ defined in (22). The associated truth solution for $u(s)$ is (27). The purpose of RBM procedures is to diminish the cost of evaluating the truth solution.

5.1. **Reduced basis methods.** Let $\mathcal{L}$ be a (local) differential operator, and consider the following PDE parameterized by a Euclidean parameter $y \in D \subset \mathbb{R}^p$:

$$\mathcal{L}(w; x; y) = f(x; y), \qquad (x, y) \in \Omega \times D \subset \mathbb{R}^d \times \mathbb{R}^p, \tag{35}$$

where x is the spatial variable and y is a parameter. The operator $\mathcal{L}$ is differential in the x variable. For example, the PDE defining $w(y)$ from (7) can be written as (35) with the operator,

$$\mathcal{L} = -\alpha\Delta + \beta I, \qquad p = 2, \qquad y = (\alpha, \beta) \in D = (0, 1]^2. \tag{36}$$

We assume that (35) is well-posed for each $y \in D$. For a fixed y, one usually develops an x-discretization with $\mathcal{N} \gg 1$ degrees of freedom yielding a solution $w^{\mathcal{N}}$ with membership in an $\mathcal{N}$-dimensional subspace. For us, this is the discretization defined in section 3.2. We assume that $\mathcal{N}$ is large enough so that

$$\sup_{y \in D} \|w^{\mathcal{N}}(y) - w(y)\|_{L^2} \leq \epsilon,$$

where ϵ is a user-prescribed tolerance. Thus, the map $y \mapsto w^{\mathcal{N}}(y) \approx w(y)$ requires algorithms whose complexity is dependent on $\mathcal{N}$.[3] Such an algorithm that performs the operation $y \mapsto w^{\mathcal{N}}(y)$ is called a *truth* approximation or solver.

The reduced basis method (RBM) is a thematic collection of model reduction strategies for parameterized PDEs that compute an emulator $y \mapsto w_N(y) \approx w^{\mathcal{N}}(y)$, whose complexity behaves like $\mathcal{O}(N^3)$ or $\mathcal{O}(N^2)$, where $N \ll \mathcal{N}$. For $N/\mathcal{N}$ sufficiently small, this can result in an emulator w_N whose evaluation is substantially cheaper than the truth approximation $w^{\mathcal{N}}$. The RBM emulator takes the form,

$$w_N(y) \coloneqq \sum_{k=1}^{N} c_{N,k}(y)\phi_k, \qquad \phi_k \in V_N \coloneqq \operatorname{span}\left\{w^{\mathcal{N}}(y_1), \ldots, w^{\mathcal{N}}(y_N)\right\} \subset V, \tag{37}$$

[3]For linear elliptic operators $\mathcal{L}$, this complexity can in principle scale like $\mathcal{O}(\mathcal{N}\log\mathcal{N})$, but frequently is $\mathcal{O}(\mathcal{N}^2)$, or even $\mathcal{O}(\mathcal{N}^3)$ depending on the details of the employed numerical solver.

where $\{y_k\}_{k=1}^N$ are particular parameter values that are chosen during the RBM construction procedure. The success of RBM algorithms rely on three main components:

- The condition that the *manifold* of solutions,
$$W(D) \coloneqq \left\{ w(y) \;\middle|\; y \in D \right\} \subset L^2(\Omega),$$
is "low rank". The mathematically precise statement of this is that the Kolmogorov N-width of the manifold,
$$d_N(W) \coloneqq \inf_{\substack{V \subset L^2 \\ \dim V = N}} \sup_{y \in D} \inf_{v \in V} \left\| w(y) - v \right\|_{L^2(\Omega)},$$
decays quickly with N. "Quickly" ideally means exponentially, but high algebraic rates of decay are also suitable. This condition ensures an RBM emulator w_N can achieve L^2-proximity to the truth approximation $w^{\mathcal{N}}$ when $N/\mathcal{N}$ is very small. We provide empirical evidence in this paper that this condition is true. Note that for our particular problem (36), exponential decay of the N width is known if we can assume that α is bounded away from 0 [17]. However Kato's formula requires α to take values arbitrarily close to 0. In a follow-up paper in preparation, we present N width estimates uniformly for $\alpha \in (0, 1]$ [3].
- The condition that the truth approximation $w^{\mathcal{N}}(y)$ comes with a practically computable *a posteriori* error estimate $\Delta(y)$, satisfying,
$$\Delta(y) \gtrsim \left\| w(y) - w^{\mathcal{N}}(y) \right\|_{L^2}.$$
This usually comes in the form of *a posteriori* finite element estimates, and in practice in the algorithm are actually used to measure $\|w^N(y) - w^{\mathcal{N}}(y)\|$. This condition ensures that the parameter values $\{y_n\}_{n=1}^N$ in (37) can be chosen in a computationally tractable manner. In our case the PDE's we consider are linear so that efficient residual-based error indicators Δ can be derived.
- The condition that the operator $\mathcal{L}$ and right-hand side f have *affine* dependence on the parameter y. This means that one has the expressions,
$$\mathcal{L} = \sum_{q=1}^{Q_{\mathcal{L}}} \gamma_q(y) \mathcal{L}_q, \qquad\qquad f(x; y) = \sum_{q=1}^{Q_f} \sigma_q(y) f_q(x),$$
where we have introduced (i) y-independent differential operators $\mathcal{L}_q$, (ii) y-independent functions $f_q(x)$, (iii) x-independent functions $\gamma_q(y)$, and (iv) x-independent functions $\sigma_q(y)$. More precisely, one requires the weak (variational) form of $\mathcal{L}$ to have such a decomposition. This condition is needed so that evaluation of the RBM emulator map $y \mapsto u^N(y)$ can be accomplished using operations that are independent of the truth approximation discretization parameter $\mathcal{N}$. We will briefly justify this for our situation in the next section.

Our next goal is to apply the RBM algorithm to the truth discretizations of $w_\pm(y)$ that define the fractional solution $u(s)$. We discuss this in the next section.

5.2. **RBM formulation.** We describe here the RBM procedure for approximating $\boldsymbol{w}_-^{\mathcal{N}}$ via a reduced basis emulator; the procedure for $\boldsymbol{w}_+^{\mathcal{N}}$ is nearly identical. We recall the discrete truth approximation formulation that defines $\boldsymbol{w}_-^{\mathcal{N}}$:
$$\left(\boldsymbol{S} + e^{-y} \boldsymbol{M} \right) \boldsymbol{w}_-^{\mathcal{N}}(y) = \boldsymbol{f}, \tag{38}$$

where $\boldsymbol{S}$, $\boldsymbol{M}$, and $\boldsymbol{f}$ are defined in (18). Detailed exposition of application of the RBM algorithm to this parameterized PDE (and to much more general cases) can already be found in existing textbook literature [37, 26, 39]. Here we give a only brief synopsis of the major steps in the algorithm for completeness, but refer to the previously-mentioned references for details and motivating explanation of the algorithm. In particular, in what follows we describe the algorithm using vectors and matrices instead of more common functional-analytic mathematical statements; this choice is made for simplicity of exposition since the algorithm itself is not new and we instead focus on the application of the algorithm.

As indicated by relation (37), an RBM algorithm produces an emulator $w_{n,-}$ given by

$$w_{n,-}(y) = \sum_{k=1}^{n} c_{n,k}(y) w_-^{\mathcal{N}}(y_k), \qquad \boldsymbol{c_n} = (c_{n,1}, \ldots, c_{n,n})^T, \tag{39}$$

where we have made a particular choice of the basis ϕ_k appearing in (37).[4] Also, since the RBM procedure builds w_N sequentially by first building $w_1, w_2, \ldots$, we label the RBM dimension as n, satisfying $1 \le n \le N$, in this section. We must specify the parameter values $\{y_k\}_{k=1}^n$ and the coefficients $\{c_{n,k}\}_{k=1}^n$, which is the focus of the following discussion.

5.3. **Computing the $c_{n,k}$.** We assume that $y_1, \ldots, y_n$ have been chosen and are known, and now seek to define the coefficients $\{c_{n,k}\}_{k=1}^n$, equivalently the vector $\boldsymbol{c}_n$, whose computation allow evaluation of $y \mapsto w_n(y)$. To proceed we define a new matrix $\boldsymbol{U} \in \mathbb{R}^{\mathcal{N} \times n}$, having entries

$$\boldsymbol{U}_{n,-} = \left[\boldsymbol{w}^{\mathcal{N}}(y_1) \;\; \boldsymbol{w}^{\mathcal{N}}(y_2) \;\; \cdots \;\; \boldsymbol{w}^{\mathcal{N}}(y_n)\right], \qquad (U_{n,-})_{j,k} = w_j^{\mathcal{N}}(y_k),$$

where the vector $\boldsymbol{w}^{\mathcal{N}}$ and its entries $w_j^{\mathcal{N}}$ are expansion coefficients for the truth approximation solution, see (17).

Then for each y, the coefficients $c_{n,k}$ of the RBM solution are defined by seeking the vector $\boldsymbol{c}_n(y) \in \mathbb{R}^n$ satisfying

$$\boldsymbol{U}^T \left(\boldsymbol{S} + e^{-y}\boldsymbol{M}\right) \boldsymbol{U} \boldsymbol{c}_n(y) = \boldsymbol{U}^T \boldsymbol{f}. \tag{40}$$

Assuming $\boldsymbol{U}$ has linearly independent columns (which is assured by the choice of y_k discussed in the next section), then this uniquely defines $\boldsymbol{c}_n(y)$ for each y, and prescribes the RBM solution w_n via (39). One final point of interest is that our truth variational form (38) exhibits *affine* dependence on the parameter y, making it possible to compute $\boldsymbol{c}_n$ very efficiently. We may rearrange computations in (40) so that

$$\left(\boldsymbol{B} + e^{-y}\boldsymbol{C}\right) \boldsymbol{c}(y) = \boldsymbol{g},$$

where

$$\boldsymbol{B} \coloneqq \boldsymbol{U}^T \boldsymbol{S} \boldsymbol{U} \in \mathbb{R}^{n \times n}, \qquad \boldsymbol{C} \coloneqq \boldsymbol{U}^T \boldsymbol{M} \boldsymbol{U} \in \mathbb{R}^{n \times n}, \qquad \boldsymbol{g} \coloneqq \boldsymbol{U}^T \boldsymbol{f} \in \mathbb{R}^n,$$

so that the quantities $\boldsymbol{B}$, $\boldsymbol{C}$, and $\boldsymbol{g}$, once computed, are all *independent* of *both* the truth discretization dimension $\mathcal{N}$ and the parameter y. Thus, for each y, the coefficients $\boldsymbol{c}_n$ (i.e., the RBM solution w_n) can be computed with complexity that depends *only* on n and *not* on $\mathcal{N}$. Since in practice $n \ll \mathcal{N}$ this can result in

[4] The parameter values y_k and coefficients $c_{n,k}$ should be labeled $y_{k,-}$ and $c_{n,k,-}$, respectively, to differentiate them from the analogous quantities resulting from applying RBM to $w_+^{\mathcal{N}}$. However, we omit this notational dependence for more clarity in exposition.

computational savings, especially if we wish to query $y \mapsto w_-^n(y)$ numerous times. This is one of the major attractions of model reduction with RBM.

5.4. **Choosing y_k.** The ingredient we are left to provide to complete our description of the RBM algorithm is the choice of parameter values y_k in (39). Given an RBM approximation w_n, we focus on the choice of y_{n+1}. We accomplish this via the standard greedy procedure in RBM algorithms. Ideally, this choice is given by

$$y_{n+1} = \operatorname*{argmax}_{y \geq 0} \left\| w^{\mathcal{N}}(y) - w_n(y) \right\|. \tag{41}$$

Unfortunately, this explicit form requires computing the full solution $w^{\mathcal{N}}(y)$ at all parameter values y, which RBM seeks to avoid. To circumvent this restriction, the standard strategy is to resort to residual-based error indicators. The following lemma identifies one such computable residual-based error indicator $\Delta_n(y)$.

Lemma 5.1. *Define the residual vector $\boldsymbol{r}_n(y) \in \mathbb{R}^{\mathcal{N}}$ as*

$$\begin{aligned} \boldsymbol{r}_{n,-}(y) &\coloneqq \boldsymbol{f} - \left(\boldsymbol{S} + e^{-y}\boldsymbol{M}\right) \boldsymbol{U}_{n,-}\boldsymbol{c}_{n,-}(y), \\ \boldsymbol{r}_{n,+}(y) &\coloneqq \boldsymbol{f} - \left(\boldsymbol{S}e^{-y} + \boldsymbol{M}\right) \boldsymbol{U}_{n,+}\boldsymbol{c}_{n,-}(y), \end{aligned} \tag{42}$$

and the indicator

$$\begin{aligned} \Delta_{n,-}(y) &\coloneqq \frac{C_{\mathcal{N}}^2}{\sqrt{\lambda_{\min}(\boldsymbol{M})}\left(C_{\mathcal{N}}^2 e^{-y} + 1\right)} \|\boldsymbol{r}_n(y)\|, \\ \Delta_{n,+}(y) &\coloneqq \frac{C_{\mathcal{N}}^2}{\sqrt{\lambda_{\min}(\boldsymbol{M})}\left(e^{-y} + C_{\mathcal{N}}^2\right)} \|\boldsymbol{r}_n(y)\|, \end{aligned} \tag{43}$$

Then

$$\Delta_{n,\pm}(y) \geq \left\| w_{\pm}^{\mathcal{N}}(y) - w_{n,\pm}(y) \right\| \tag{44}$$

Proof. We show the result for the "$-$" quantities; a similar proof works for the "$+$" quantities. The residual $\boldsymbol{r}_{n,-}$ satisfies

$$\left(\boldsymbol{S} + e^{-y}\boldsymbol{M}\right)\left(\boldsymbol{w}_-^{\mathcal{N}}(y) - \boldsymbol{w}_{n,-}(y)\right) = \boldsymbol{r}_{n,-},$$

so that

$$\left\| w_-^{\mathcal{N}}(y) - w_{n,-}(y) \right\|_{L^2} = \left\| \boldsymbol{w}_-^{\mathcal{N}}(y) - \boldsymbol{w}_{n,-}(y) \right\|_{\boldsymbol{M}} \leq \frac{1}{\lambda_{\min}(\boldsymbol{S}+e^{-y}\boldsymbol{M},\boldsymbol{M})} \left\| \boldsymbol{M}^{-1/2}\boldsymbol{r}_{n,-} \right\|,$$

with

$$\lambda_{\min}\left(\boldsymbol{S} + e^{-y}\boldsymbol{M}, \boldsymbol{M}\right) \coloneqq \inf_{\boldsymbol{v}\in\mathbb{R}^{\mathcal{N}}} \frac{\boldsymbol{v}^T(\boldsymbol{S}+e^{-y}\boldsymbol{M})\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}} = e^{-y} + \inf_{\boldsymbol{v}\in\mathbb{R}^{\mathcal{N}}} \frac{\boldsymbol{v}^T\boldsymbol{S}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v}} = e^{-y} + \lambda_{\min}(\boldsymbol{S},\boldsymbol{M}).$$

To summarize, we have the estimate

$$\begin{aligned} \left\| w_-^{\mathcal{N}}(y) - w_{n,-}(y) \right\| &\leq \frac{1}{e^{-y} + \lambda_{\min}(\boldsymbol{S},\boldsymbol{M})} \left\| \boldsymbol{M}^{-1/2}\boldsymbol{r}_{n,-} \right\| \\ &\leq \frac{1}{\sqrt{\lambda_{\min}(\boldsymbol{M})}\left(e^{-y} + \lambda_{\min}(\boldsymbol{S},\boldsymbol{M})\right)} \|\boldsymbol{r}_{n,-}\|, \end{aligned}$$

which is the desired result by using the definition of $C_{\mathcal{N}}$ in (19). □

The residual vectors $\boldsymbol{r}_{n,\pm}$ can be efficiently computed for many values of y. We illustrate this $\boldsymbol{r}_{n,-}$. We have:

$$\begin{aligned} \boldsymbol{r}_{n,-}(y) &\coloneqq \boldsymbol{f} - \left(\boldsymbol{S} + e^{-y}\boldsymbol{M}\right)\boldsymbol{U}_{n,-}\boldsymbol{c}_{n,-}(y) \\ &= P_{\mathcal{R}(\boldsymbol{R}_n)^\perp}\boldsymbol{f} + P_{\mathcal{R}(\boldsymbol{R}_n)}\left(\boldsymbol{f} - \left(\boldsymbol{S} + e^{-y}\boldsymbol{M}\right)\boldsymbol{U}_{n,-}\boldsymbol{c}_{n,-}(y)\right), \end{aligned} \tag{45}$$

where $P_{\mathcal{R}(\boldsymbol{A})} : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^{\mathcal{N}}$ is the $\mathbb{R}^{\mathcal{N}}$-orthogonal projector onto the column space of a matrix $\boldsymbol{A}$, and

$$\boldsymbol{R}_n \coloneqq [\boldsymbol{S}\boldsymbol{U}_{n,-},\ \boldsymbol{M}\boldsymbol{U}_{n,-}] \in \mathbb{R}^{\mathcal{N}\times(2n)}.$$

The orthogonal decomposition in (45) shows that the Pythagorean theorem can be used to compute the Euclidean vector norm $\|\boldsymbol{r}_{n,-}(y)\|_2$ in an efficient way for several values of y:

- $\|P_{\mathcal{R}(\boldsymbol{R}_n)^\perp}\boldsymbol{f}\|_2$ is y-independent, so that it can be computed once and stored.
- $P_{\mathcal{R}(\boldsymbol{R}_n)}\left(\boldsymbol{f} - (\boldsymbol{S} + e^{-y}\boldsymbol{M})\,\boldsymbol{U}_{n,-}\boldsymbol{c}_{n,-}(y)\right)$ is a vector in a $2n$-dimensional, y-independent vector space. Thus, the norm can be computed with only n-dependent complexity. The fact that $\boldsymbol{c}_n(y)$ appears with linear behavior in this expression ensures that we can rearrange computations so that, for each $\boldsymbol{c}_n(y)$, the norm of this quantity can be computed using complexity that is dependent only on n.

In summary, while the right-hand side of (44) is not efficiently computable for many values of y, the left-hand side is efficiently computable for several values of y since $\|\boldsymbol{r}_n\|_2$ is efficient to compute, and $\lambda_{\min}(\boldsymbol{S},\boldsymbol{M})$ does not depend on y and can be computed either directly or iteratively with generalized eigenvalue solvers once and subsequently stored.

Standard greedy algorithms for RBM methods require a computable quantity satisfying (44), and replace the essentially un-computable maximization (41) with the computable maximization

$$y_{n+1} = \operatorname*{argmax}_{y\geq 0} \Delta_n(y). \tag{46}$$

The above maximization has an objective function that is efficiently computable, and the inequality (44) ensures that the maximization (46) is a *weak* greedy algorithm. Weak greedy algorithms in turn ensure that the set of chosen parameters $\{y_1,\ldots,y_N\}$ defines an RBM subspace V_N in (37) whose best approximation to the truth solution $\boldsymbol{w}_-^{\mathcal{N}}$ is comparable to the Kolmogorov N-width [8, 20].

We have completed the basic description of the RBM algorithm: the emulators $w_{N,\pm}(y)$ are defined by computing coefficients $\boldsymbol{c}_{N,\pm}(y)$ as described in the previous section, and the parameter values y_k are chosen according to (46) by computing the estimators $\Delta_{n,\pm}(y)$ for $n = 1,2,\ldots,$. One usually sequentially computes y_k until $\sup_{y\geq 0}\Delta_{n,\pm}(y)$ is smaller than some specified tolerance, so that one can rigorously certify the error committed by the RBM emulators. This tolerance condition is usually how the terminal RBM dimension N is computationally specified.

One final observation we make is a major theoretical result of this paper:

Theorem 5.2. *Let $N_\pm$ denote the RBM dimensions formed for the emulators $w_{N_\pm,\pm}(y)$. Then*

$$\sup_{s\in(0,1)} \left\|u^{\mathcal{N}}(s) - u_N(s)\right\| \leq \frac{4}{\pi}\max\left\{\sup_{y\geq 0}\Delta_{N_+,+}(y), \sup_{y\geq 0}\Delta_{N_-,-}(y)\right\}. \tag{47}$$

Proof. We have

$$u^{\mathcal{N}}(s) - u_N(s) = \beta_0(s_-) \sum_{j=1}^{N_-} \tau_{j,-} \left[w_-^{\mathcal{N}} \left(\frac{y_{j,-}}{s_-} \right) - w_{N_-,-} \left(\frac{y_{j,-}}{s_-} \right) \right] + \beta_0(s_+) \sum_{j=1}^{N_+} \tau_{j,+} \left[w_+^{\mathcal{N}} \left(\frac{y_{j,+}}{s_+} \right) - w_{N_+,+} \left(\frac{y_{j,+}}{s_+} \right) \right].$$

Taking L^2 norms of both sides, and using the triangle inequality with (44) yields the result. □

We note that the quantities $\Delta_{N_\pm,\pm}(y)$ are computed during the RBM construction phase, so that these estimators are available. Therefore, (47) provides a computable error bound that can be used to certify error committed by using the RBM algorithm.

Remark 4. Just as with Remark 4, all our results above extend to a more general elliptic operator $\mathcal{E}$ satisfying the assumptions outlined in Remark 4. In this case, all of our formulas in this section carry over.

5.5. **Algorithm summary.** The full algorithm of this section first uses the RBM algorithm to perform model reduction on the parameterized PDE solutions $w_\pm^{\mathcal{N}}(y)$.[5] Subsequently, the efficient RBM emulators $w_{N,\pm}(y)$ are used in the GQ algorithm from Section 3.4. We describe the full algorithm in Algorithm 2.

The algorithm we have described in this section is a skeleton version of a modern RBM algorithm. We summarize various improvements that should be implemented in order for an RBM algorithm to be efficient and accurate:

- One does not usually solve (46) by maximizing over the parameter continuum, and instead maximizes over a discrete set. For bounded parameter domains, it is common to use a uniform grid and subsequently adaptively (e.g., dyadically) refine the grid to ensure that no local maxima are skipped. Over the unbounded domain, we employ a logarithmic map; see section 6.4 for details.
- Our RBM ansatz (39) uses solution *snapshots* $w_\pm^{\mathcal{N}}(y)$ as basis functions. This is known to generally lead to ill-conditioning in the formulation (40) even for small n (since in practice the columns of $\boldsymbol{U}_{n,\pm}$ are "nearly" linearly dependent). A better prescription is to build the RBM basis functions by orthogonalizing snapshots.
- Some naive implementations of the decomposition (45) via quadratic forms leads to numerical roundoff error that results in stagnation of the error indicators $\Delta_{n,\pm}(y)$ near root-machine precision. (E.g., for double precision, stagnation occurs when $\Delta_{n,\pm}(y)$ takes values around 10^{-8}. More careful computations allow one to overcome this limitation [14, 15, 12, 16].

We again refer to [37, 26, 39] for a more complete description of important but standard RBM algorithm details.

Finally, we note that one can combine the error estimates in (47) and (31) via the triangle inequality to create a computable bound for $\|u^N - \widetilde{u}^{\mathcal{N}}\|$. This error would estimate the error committed by the two novel innovations of this paper: our Gaussian quadrature approach and the model reduction procedure.

[5] In the previous sections we have describe this process only for $w_-^{\mathcal{N}}$, but the process for $w_+^{\mathcal{N}}$ results in almost the same procedure with only minor differences stemming from the location of the e^{-y} factor.

Algorithm 2 RBM algorithm: Produces solution to the fractional Laplace problem (5). The function OfflineFracLapRBM needs to be completed only once and has complexity dependent on $\mathcal{N} = \dim V$. Afterwards, OnlineFracLapRBM can be called for arbitrary values of $s \in (0,1)$ with a computational cost dependent only on $\dim V_N = N \ll \mathcal{N}$.

Precondition: Availability of a discrete solution $\boldsymbol{w}^{\mathcal{N}}(\alpha,\beta)$ from the formulation (16).

function OfflineFracLapRBM
  $n \leftarrow 1$. Randomly choose y_1, compute and store $w_-^{\mathcal{N}}(y_1)$.
  Assemble RBM emulator $w_1(\cdot)$.
  for $n \leftarrow 2$ to N_- **do**
    Compute y_n from (46).
    Compute and store $w_-^{\mathcal{N}}(y_n)$.
    Assemble RBM emulator $w_n(\cdot)$.
  Repeat above computations to assemble $w_{N_+,+}(\cdot)$
  return RBM emulators $w_{N_\pm,\pm}(\cdot)$

Precondition: Availability of RBM emulators $w_{N_\pm,\pm}(\cdot)$.

function OnlineFracLapRBM(s)
  Call FracLapGQ(s), Algorithm 1, replacing $w_\pm^{\mathcal{N}}(\cdot)$ with RBM emulators $w_{N_\pm,\pm}(\cdot)$.
  return $\boldsymbol{u}_N(s)$.

6. **Numerical examples.** In these experiments, we compare the effectiveness of our improvements to current methods. We first describe the setup of the test problems that we consider in our simulations. We solve (5) on $\Omega = [0,1]^2$ with homogeneous Dirichlet boundary conditions. We test a total of 3 algorithms:

- "SQ" — The sinc quadrature approach from [11].
- "GQ" — Algorithm 1 detailed in section 3, utilizing a modified version of the integral formulation in [11] along with Gaussian quadrature.
- "RBM" — The approach detailed in section 5.5, leveraging the reduced basis method to accelerate the GQ algorithm.

To compare the three methods, we will use the same number of y-quadrature points $M = M_- + M_+$ in each approach. This number represents the total number of local PDE solves needed to compute an approximation to $u(s)$. In particular, we make the choices given in [11, 6], which depend on the spatial mesh and fractional order:

$$M_+ = \left\lceil \frac{\pi^2}{4sr^2} \right\rceil, \qquad M_- = \left\lceil \frac{\pi^2}{4(1-s)r^2} \right\rceil, \qquad r = \frac{1}{\log(\sqrt{\mathcal{N}})} \tag{48}$$

The finite element discretization is accomplished with linear quadrilateral finite elements on a Cartesian tessellation of Ω. The one-dimensional grids that define this Cartesian tessellation are isotropic with respect to the two dimensions, and are defined as equidistant meshes with 2^K points. We will use various values of K.

6.1. **Manufactured Solutions on $[0,1]^2$.** Consider the physical domain $\Omega = [0,1]^2$. In this case, an explicit family of eigenfunctions for the Laplacian with

homogeneous Dirichlet boundary conditions is available:

$$\phi_{n,m}(x) = \sin(n\pi x_1)\sin(m\pi x_2), \qquad n, m \in \mathbb{N}, \qquad x = (x_1, x_2) \in \Omega,$$

which satisfy

$$-\Delta\phi_{n,m}(x) = \lambda_{n,m}\phi_{n,m}(x), \qquad \lambda_{n,m} = \pi^2(n^2 + m^2).$$

With this in hand, and using the inverse of the relation (4), we can easily construct explicit solutions for testing using eigenfunction expansions. We explore the effectiveness of our algorithms through three manufactured solutions:

- "SINE" — The function u and data f are, in this case,

$$u(x) = \frac{1}{(2\pi^2)^s}\sin(\pi x_1)\sin(\pi x_2), \qquad f(x) = \sin(\pi x_1)\sin(\pi x_2)$$

- "MIXED MODES" — The function u and data f are, in this case,

$$u(x) = \frac{1}{(116\pi^2)^s}\sin(4\pi x_1)\sin(10\pi x_2), \qquad f(x) = \sin(4\pi x_1)\sin(10\pi x_2)$$

- "SQUARE BUMP" — The data f is the indicator function

$$f(x) = \mathbb{1}_{[0.25,0.75]^2}(x).$$

  While an analytical solution is available as an infinite sum of eigenfunctions, we instead numerically compute a solution on a grid formed by $2^9 + 1$ equally spaced nodes in both directions and consider this the "exact" solution. In comparison, the finest mesh used to evaluate the accuracy of our solver has $2^6 + 1$ nodes in both directions.

6.2. **Spatial convergence.** Our first test verifies that we recover spatial convergence in terms of the finite element mesh size. Since we are primarily interested in accuracy and not efficiency, this section compares the SQ and GQ methods, with the results summarized in Figure 3.

We can see that the proposed GQ algorithm performs slightly better than the existing SQ algorithm for the same number of quadrature points M. We also remark that the GQ implementation allows us to generate solutions for small fractional parameters with less numerical difficulty. With the SQ approach, the difficulty arises when application of the quadrature rule results in large values of a term involving e^y appearing in operators that must be inverted.

6.3. **Quadrature rule efficiency.** In this section we compute errors committed by the GQ and SQ algorithms for different values of the quadrature rule size M. The purpose of this test is to understand the efficiency of the quadrature rule, i.e., the number of solutions of $w_{\pm}^{\mathcal{N}}$ required. Figure 4 illustrates errors for the three test cases as a function of the total number of quadrature nodes. The test in this section does not fix $M_{\pm}$ as given by (48). Instead, given a number of quadrature points M (the abscissa in Figure 4), we generate M-point quadrature rules for the Gaussian quadrature and sinc approaches. Thus, the quadrature rule for each M is generated anew.

The results indicate that the GQ algorithm converges faster than the SQ with respect to the number of PDE solves. This shows that the GQ algorithm appears to be far more efficient than sinc quadrature for computing solutions to these fractional problems.

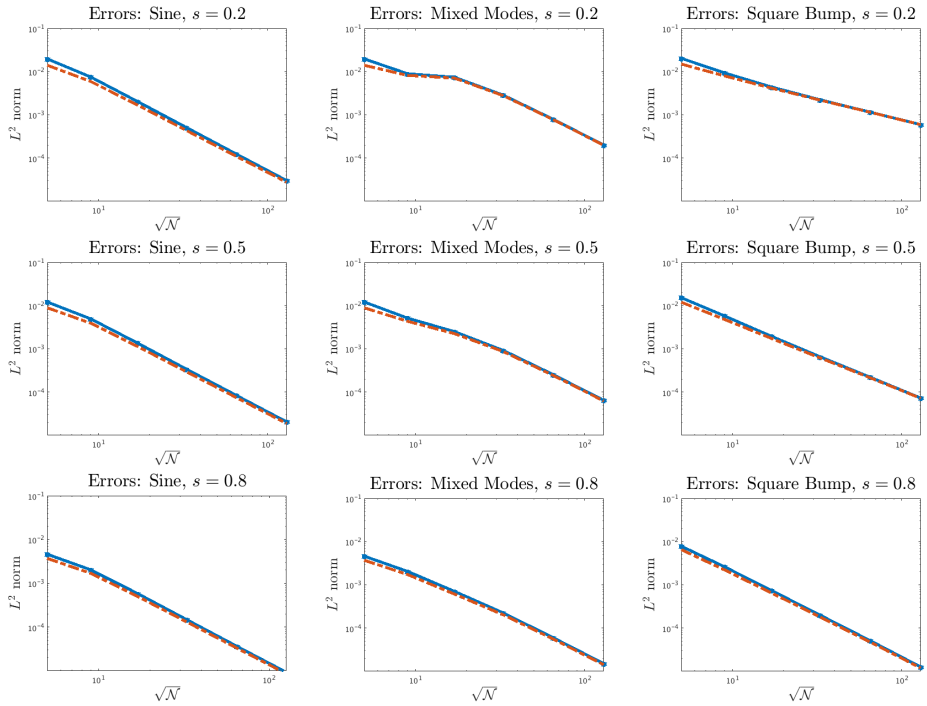

Figure 3. Convergence of SQ (solid blue) and GQ methods (red dashed) as the spatial mesh is refined. Each used a dyadic mesh along the spatial variable with increasing resolution. A parameter value of $s = 0.2$ was used and similar results were seen for value of s between 0.1 and 0.9. We used the number of quadrature points for the integral suggested by current methods [6].

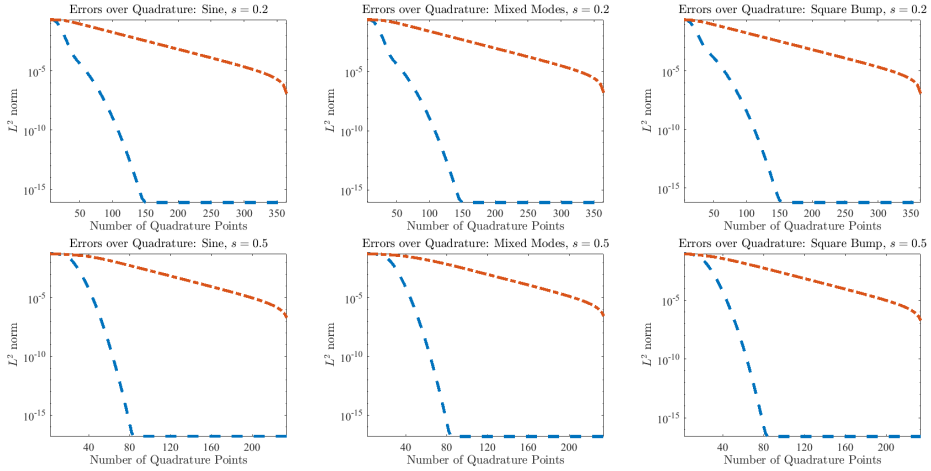

Figure 4. Accuracy comparison of the SQ (red, dotted and dashed) and GQ methods (blue, dashed), with fractional order $s = 0.2$ (top plots) and $s = 0.5$ (bottom plots). Similar results where observed for value of s between 0.1 and 0.9.

6.4. **RBM offline efficiency.** This section investigates the RBM algorithm. For now, we restrict our attention to one-time queries of $u(s)$, i.e., to situations when, given Ω, f, and s, we seek to compute *only* $u(s)$ for this given s. For the GQ algorithm this involves a single run of the routine FracLapGQ in Algorithm 1. For the RBM algorithm, this entails a single run of the OfflineFracLapRBM routine in Algorithm 2, followed by a single run of OnlineFracLapRB routine.

For the RBM algorithm, we solve (46) by discretizing the $y \in [0, \infty)$ domain in a uniform way under a logarithmic map. Precisely: we set $z = e^{-y}$ for $y \in [0, \infty)$ and proceed to discretize $z \in (0, 1]$. We take 128 equispaced points in the z variable and map back to y-space with $z \mapsto -\log z = y$. We subsequently perform a discrete maximization over this set instead of the continuous optimization (46).

In figure 5 we compare the GQ algorithm to the accelerated RBM algorithm, including the offline construction time. We see that the initial investment of the RBM algorithm in the offline phase is substantial, accumulating to the time required for the direct GQ method with 150-200 quadrature points. However, we see that after this initial offline investment, subsequent evaluations of the RBM surrogate are extremely efficient, so that the effort required to evaluate $M \gg 1$ quadrature point is essentially the same as that required to evaluate at a single quadrature point.

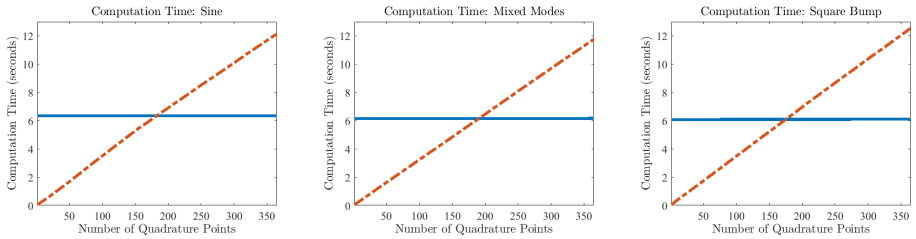

Figure 5. "Offline" (i.e., one-time) computational investment for a single solve of (5) with a fixed value of $s = 0.2$. These experiments compare both the direct (dotted and dashed) and reduced basis methods (solid) using the gaussian quadrature. Each used a dyadic mesh with 7 levels. Similar results where seen for value of s between 0.1 and 0.9.

6.5. **RBM accuracy.** We now investigate the accuracy delivered by the RBM algorithm in the construction of reduced order models for $w_{\pm}^{\mathcal{N}}$. Our rigorous error certificate for $u(s)$ using the reduced order model is (47), but we here consider a finer estimate using the proof of Theorem 5.2. We define the error estimator,

$$\Delta_N(s) := \sum_{\sigma \in \{+,-\}} \beta_0(s_\sigma) \sum_{j=1}^{M_\sigma} \Delta_{N,\sigma}(s_{j,\sigma}) \tau_{j,\sigma},$$

where $\Delta_{N,\pm}$ are the computable error indicators defined in (43), and we again choose $M_\pm$ as in (48). One can see from the proof of Theorem 5.2 that this quantity bounds the error committed by the RBM procedure. We plot Δ_N in Figure 6 as a function of N, and observe that it decays exponentially.

Finally, we remark that Δ_N only certifies the error committed by the model reduction RBM algorithm; the error committed by the y-quadrature rule is not certified by this quantity.

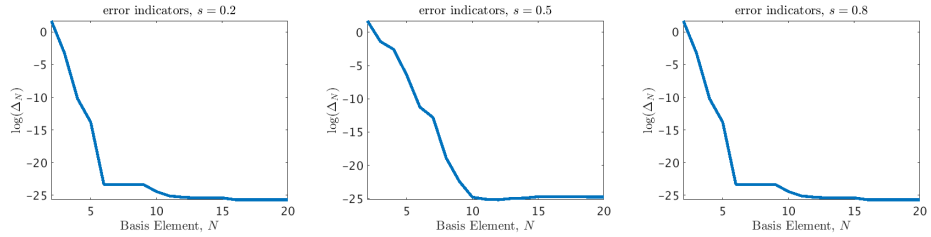

Figure 6. Error indicators $\Delta_N(s)$ as a function of N, for $s = 0.2, 0.5, 0.8$.

6.6. **RBM accuracy and efficiency.** Finally, we explore the accuracy afforded by the RBM procedure for computing $s \mapsto u(s)$, and also verify the computational efficiency of the procedure. In Figure 7 left and center, we demonstrate that the error committed by the RBM algorithm is stable, even for relatively small values of the parameter $0.01 < s < 0.1$. Furthermore, the right pane of this figure demonstrates that if we wish to *repeatedly* query the map $s \mapsto u(s)$ for several values of s, the RBM algorithm is undeniably more efficient by an order of magnitude even for just one query, and by three orders of magnitude if 1000 queries are needed. This is an especially salient point for optimization or control applications involving forward models with fractional PDEs of varying fractional order s, where repeated evaluation of such PDE solutions is needed.

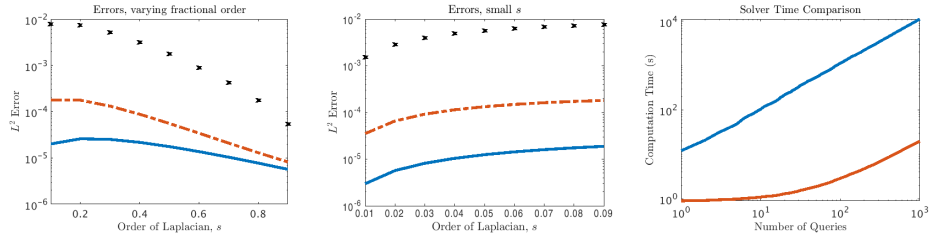

Figure 7. Accuracy of the RBM algorithm over a range of values of s (left and center). The Sine example is plotted in a red dot-dashed, the Mixed Modes in a blue solid line, and the Square Bump case in black crosses. In the right pane we show the *cumulative* computational time required by the GQ algorithm (blue) versus the RBM algorithm (red). Each query refers to an evaluation of the map $s \mapsto u(s)$. In particular this cumulative time for the RBM solver includes the one-time offline cost required by OfflineFracLapRBM in Algorithm 2.

7. **Conclusion.** We propose a novel model reduction strategy for computing solutions to fractional Laplace PDE's, in particular (5). Our algorithm builds on the ideas introduced in [11], improving accuracy and stability, and accelerating that algorithm considerably. Our model reduction strategy hinges on the fact that the solution to the fractional problem can be written in terms of classical, local elliptic PDE's, for which RBM-based model reduction is known to be efficient.

We provide novel stability bounds for both the continuous and discrete problems, and our numerical experiments suggest that our Gaussian quadrature approach is more efficient than alternative quadrature methods. All of our algorithmic and theoretical results apply to solutions to differential equations involving fractional

powers of general elliptic operators. A rigorous proof of the convergence for our quadrature rule is the subject of ongoing study.

Acknowledgments. H. Antil was partially supported by National Science Foundation awards DMS-1818772 and DMS-1913004 and Air Force Office of Scientific Research (AFOSR) under Award NO: FA9550-19-1-0036. Y. Chen's research was supported by NSF awards DMS-1719698 and AFOSR award FA9550-18-1-0383. E. Cherkaev was supported by NSF awards DMS-0940249 and DMS-1413454. A. Narayan was partially supported by NSF award DMS-1848508, AFOSR award FA9550-15-1-0467, and DARPA EQUiPS contract N660011524053. This material is based upon work supported by the National Science Foundation under Grant No. DMS-1439786 and by the Simons Foundation Grant No. 50736 while Y. Chen and A. Narayan were in residence at the Institute for Computational and Experimental Research in Mathematics in Providence, RI, during the "Model and dimension reduction in uncertain and dynamic systems" program.

REFERENCES

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

[2] H. Antil and S. Bartels, Spectral approximation of fractional PDEs in image processing and phase field modeling, *Comput. Methods Appl. Math.*, **17** (2017), 661–678.

[3] H. Antil, Y. Chen and A. Narayan, Kolmogorov widths and reduced order modeling for fractional elliptic operators, preprint, (2019).

[4] H. Antil, Y. Chen and A. Narayan, Reduced basis methods for fractional Laplace equations via extension, *SIAM J. Sci. Comput.*, **41** (2019), A3552–A3575.

[5] H. Antil, R. Khatri and M. Warma, External optimal control of nonlocal PDEs, *Inverse Problems*, **35** (2019), 084003, 35 pp.

[6] H. Antil and J. Pfefferer, *A short Matlab implementation of fractional Poisson equation with nonzero boundary conditions*, Technical report, 2017.

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

[8] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova and P. Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods, *SIAM J. Math. Anal.*, **43** (2011), 1457–1472.

[9] A. Bonito, D. Guignard and A. R. Zhang, Reduced basis approximations of the solutions to spectral fractional diffusion problems, *J. Numer. Math.*, **28** (2020).

[10] A. Bonito, W. Lei and J. E. Pasciak, Numerical approximation of the integral fractional Laplacian, *Numer. Math.*, **142** (2019), 235–278.

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

[12] A. Buhr, C. Engwer, M. Ohlberger and S. Rave, A numerically stable a posteriori error estimator for reduced basis approximations of elliptic equations, preprint, (2014), arXiv:1407.8005.

[13] L. Caffarelli and L. Silvestre, An extension problem related to the fractional Laplacian, *Comm. Partial Differential Equations*, **32** (2007), 1245–1260.

[14] F. Casenave, Accurate a posteriori error evaluation in the reduced basis method, *C. R. Math. Acad. Sci. Paris*, **350** (2012), 539–542.

[15] F. Casenave, A. Ern and T. Lelièvre, Accurate and online-efficient evaluation of the a posteriori error bound in the reduced basis method, *ESAIM Math. Model. Numer. Anal.*, **48** (2014), 207–229.

[16] Y. Chen, J. Jiang and A. Narayan, A robust error estimator and a residual-free error indicator for reduced basis methods, *Comput. Math. Appl.*, **77** (2019), 1963–1979.

[17] A. Cohen and R. DeVore, Approximation of high-dimensional parametric PDEs, *Acta Numer.*, **24** (2015), 1–159.

[18] T. Danczul and J. Schöberl, A Reduced Basis Method For Fractional Diffusion Operators I, preprint, (2019), arXiv:1904.05599.

[19] T. Danczul and J. Schöberl, A Reduced Basis Method For Fractional Diffusion Operators II, preprint, (2020), arXiv:2005.03574.
[20] R. DeVore, G. Petrova and P. Wojtaszczyk, Greedy algorithms for reduced bases in Banach spaces, *Constr. Approx.*, **37** (2013), 455–466.
[21] E. Di Nezza, G. Palatucci and E. Valdinoci, Hitchhiker's guide to the fractional Sobolev spaces, *Bull. Sci. Math.*, **136** (2012), 521–573.
[22] M. E. Farquhar, T. J. Moroney, Q. Yang, I. W. Turner and K. Burrage, Computational modelling of cardiac ischaemia using a variable-order fractional Laplacian, preprint, (2018), arXiv:1809.07936.
[23] F. Gesztesy and M. Mitrea, A description of all self-adjoint extensions of the Laplacian and Krein-type resolvent formulas on non-smooth domains, *J. Anal. Math.*, **113** (2011), 53–172.
[24] G. Grubb, Regularity of spectral fractional Dirichlet and Neumann problems, *Math. Nachr.*, **289** (2016), 831–844.
[25] Q. Guan, M. Gunzburger, C. G. Webster and G. Zhang, Reduced basis methods for nonlocal diffusion problems with random input data, *Comput. Methods Appl. Mech. Engrg.*, **317** (2017), 746–770.
[26] J. S. Hesthaven, G. Rozza and B. Stamm, *Certified Reduced Basis Methods for Parametrized Partial Differential Equations*, SpringerBriefs in Mathematics, Springer International Publishing, Cham, 2016.
[27] M. Ilic, F. Liu, I. Turner and V. Anh, Numerical Approximation of a fractional-in-space diffusion equation, I, *Fract. Calc. Appl. Anal.*, **8** (2005), 323–341. https://eudml.org/doc/11303
[28] M. Ilic, F. Liu, I. Turner and V. Anh, Numerical approximation of a fractional-in-space diffusion equation (II) - with nonhomogeneous boundary conditions, *Fract. Calc. Appl. Anal.*, **9** (2006), 333–349. http://www.diogenes.bg/fcaa/
[29] T. Kato, Note on fractional powers of linear operators, *Proc. Japan Acad.*, **36** (1960), 94–96.
[30] D. Kumar, J. Singh and S. Kumar, A fractional model of Navier–Stokes equation arising in unsteady flow of a viscous fluid, *Journal of the Association of Arab Universities for Basic and Applied Sciences*, **17** (2015), 14–19.
[31] M. Kwaśnicki, Ten equivalent definitions of the fractional laplace operator, *Fract. Calc. Appl. Anal.*, **20** (2017), 7–51.
[32] J. L. Lions and E. Magenes, *Non-Homogeneous Boundary Value Problems and Applications. Vol. 1*, Die Grundlehren der mathematischen Wissenschaften, Springer-Verlag, New York-Heidelberg, 1972. https://www.springer.com/gp/book/9783642651632
[33] D. Meidner, J. Pfefferer, K. Schürholz and B. Vexler, hp-finite elements for fractional diffusion, *SIAM J. Numer. Anal.*, **56** (2018), 2345–2374.
[34] S. Molchanov and E. Ostrovskii, Symmetric stable processes as traces of degenerate diffusion processes, *Theory of Probability & Its Applications*, **14** (1969), 127–130. https://epubs.siam.org/doi/abs/10.1137/1114012
[35] 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), 733–791.
[36] B. N. Parlett, *The Symmetric Eigenvalue Problem*, Classics in Applied Mathematics, vol. 20, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1998.
[37] A. T. Patera and G. Rozza, *Reduced basis approximation and a posteriori error estimation for parametrized partial differential equations*, MIT Press, 2007.
[38] P. Perdikaris and G. E. Karniadakis, Fractional-order viscoelasticity in one-dimensional blood flow models, *Annals of Biomedical Engineering*, **42** (2014), 1012–1023.
[39] A. Quarteroni, A. Manzoni and F. Negri, *Reduced Basis Methods for Partial Differential Equations*, UNITEXT, vol. 92, La Matematica per il 3+2. Springer, Cham, 2016.
[40] F. Song, C. Xu and G. E. Karniadakis, Computing fractional Laplacians on complex-geometry domains: Algorithms and simulations, *SIAM J. Sci. Comput.*, **39** (2017), A1320–A1344.
[41] P. R. Stinga and J. L. Torrea, Extension problem and Harnack's inequality for some fractional operators, *Comm. Partial Differential Equations*, **35** (2010), 2092–2122.
[42] C. J. Weiss, B. G. van Bloemen Waanders and H. Antil, Fractional operators applied to geophysical electromagnetics, *Geophysical Journal International*, **220** (2020), 1242–1259.
[43] C. J. Weiss, B. G. v. B. Waanders and H. Antil, Fractional Operators Applied to Geophysical Electromagnetics, preprint, arXiv:1902.05096.
[44] D. R. Witman, M. Gunzburger and J. Peterson, Reduced-order modeling for nonlocal diffusion problems, *Internat. J. Numer. Methods Fluids*, **83** (2017), 307–327.

[45] Q. Yang, I. Turner, F. Liu and M. Ilić, Novel numerical methods for solving the time-space fractional diffusion equation in two dimensions, *SIAM J. Sci. Comput.*, **33** (2011), 1159–1180.

Received August 2020; revised December 2020.

E-mail address: hd2238@nyu.edu
E-mail address: hantil@gmu.edu
E-mail address: yanlai.chen@umassd.edu
E-mail address: elena@math.utah.edu
E-mail address: akil@sci.utah.edu