

Randomized reduced basis methods for parameterized fractional elliptic PDEs

Harbir Antil^a, Arvind K. Saibaba^{b,*}

^a Department of Mathematical Sciences and Center for Mathematics and Artificial Intelligence, George Mason University, United States of America

^b Department of Mathematics, North Carolina State University, United States of America

ARTICLE INFO

Keywords:

Fractional elliptic PDEs
Reduced order models
Iterative methods
Randomization
Gaussian processes

ABSTRACT

This paper is interested in developing reduced order models (ROMs) for repeated simulation of fractional elliptic partial differential equations (PDEs) for multiple values of the parameters (e.g., diffusion coefficients or fractional exponent) governing these models. These problems arise in many applications including simulating Gaussian processes, geophysical electromagnetics. The approach uses the Kato integral formula to express the solution as an integral involving the solution of a parameterized elliptic PDE, which is discretized using finite elements in space and sinc quadrature for the fractional part. The offline stage of the ROM is accelerated using a solver for shifted linear systems, MPMRES-Sh, and using a randomized approach for compressing the snapshot matrix. Our approach is both computational and memory efficient. Numerical experiments on a range of model problems, including an application to Gaussian processes, show the benefits of our approach.

1. Introduction

We want to derive efficient methods for the parameterized fractional partial differential equation (PDE)

$$(\mathcal{A}(\mathbf{x}; \boldsymbol{\mu}))^\alpha y(\mathbf{x}; \boldsymbol{\mu}) = b(\mathbf{x}; \boldsymbol{\mu}), \quad \mathbf{x} \in \Omega \quad (1)$$

where $\mathcal{A}$ is an elliptic differential operator, parameterized by the set of parameters $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$. Here $0 < \alpha < 1$ is the fractional exponent. Throughout this paper, unless otherwise specified, we consider the boundary conditions to be homogeneous Neumann, but the approach can be modified slightly to accommodate other boundary conditions as well. We assume that the fractional power is defined in the spectral sense. Solving the fractional elliptic PDE is a computationally challenging problem in itself. Indeed, (1) is nonlocal and traditional ‘local basis’ approaches such as finite element method require dealing with dense numerical linear algebra. Therefore, the efficiency of traditional iterative methods becomes questionable.

In this paper, we want to address the more challenging problem of efficiently solving (1) for multiple instances of the parameter $\boldsymbol{\mu}$ and the exponent α . This is accomplished by the use of the reduced basis approach. These kinds of problems arise in imaging [1], geophysical electromagnetics [2], fractional Laplacian regularizers in optimization [3], harmonic maps [4]. In [5], the authors investigate the well-posedness of a Bayesian inverse problem with a fractional elliptic PDE forward model. In this case, the elliptic operator takes the form $\mathcal{A}(\mathbf{x}; \boldsymbol{\mu}) = -\nabla \cdot (D(\mathbf{x}; \boldsymbol{\mu}) \nabla)$, where $D(\mathbf{x}; \boldsymbol{\mu})$ is the parameterized diffusion coefficient. To estimate this diffusion coefficient from data, multiple instances of the fractional PDE have to be solved. A similar situation occurs in optimal control

* Corresponding author.

E-mail addresses: hantil@gmu.edu (H. Antil), asaibab@ncsu.edu (A.K. Saibaba).

problems where the fractional exponent is considered as a random variable [6], or optimization problems to identify the fractional exponent [7,8]. Another application of interest is the stochastic PDE (SPDE) approach to Gaussian processes, which is explored further in Section 4.1. In view of all these applications, there is a clear need for an efficient reduced order model (ROM).

In many applications of interest, high-fidelity models are governed by a single PDE or systems of PDEs that are expensive to simulate. Reduced order models refer to a collection of techniques for efficiently reducing the complexity of high-fidelity models (also known as full-order models or FOMs) with perhaps only a small sacrifice in accuracy compared to the FOMs. ROMs are especially beneficial for problems in which the full-order model has to be simulated for several parameter conditions as the “inner loop” for a variety of applications such as optimization, control, inversion, uncertainty quantification, design, or digital twins. This is known as parametric ROM and this is the setting in the present paper.

While there are several possible techniques for ROMs, we follow the so-called projection-based model reduction approach, in which there is a clear separation into two stages. The first stage is called the *offline stage* in which the FOM is simulated under a wide range of “representative” conditions of the parameters to collect snapshots of the system. This is then followed by a compression step to extract a basis that is used to then project the FOM onto a lower-dimensional subspace yielding a ROM. In the *online stage*, the lower-dimensional ROM is then simulated for a given parameter. The computational tradeoff comes from the fact that the offline costs may be large but upfront, and acceptable in certain situations, so long as the online costs are small and the accuracy is good.

While ROM for elliptic PDE-based problems is a well-established field, ROM for fractional elliptic PDEs is relatively less well-studied and is the main focus of this paper. We briefly review the literature in this area and summarize the main contributions of this paper.

Literature review A recent paper [9] considers a reduced basis approach for the spectral definition when realized using the Stinga-Torrea (Caffarelli-Silvestre) extension [10,11]. The articles [12,13] employ a reduced basis approach by interpolating operator norms. For another reduced ordering modeling approach for problems with nonlocal integral kernels, we refer to [14]. The paper [15] considers ROMs for time-fractional PDEs. The recent review paper [16] reviews numerical methods, including reduced order models, for a different definition of the nonlocal problem. Since these ROMs address different fractional/non-local models, we cannot offer a direct comparison to these works.

The following papers [17,18] are closest to our setting, in which $\mathcal{A}(\mathbf{x}; \boldsymbol{\mu})^\alpha y(\mathbf{x}; \boldsymbol{\mu}) = (-\Delta)^\alpha y(\mathbf{x})$ is the standard Laplacian and $g(\mathbf{x}; \boldsymbol{\mu}) = g(\mathbf{x})$. The reduced order model is constructed over the exponent α , but the parametric dependence of the PDE on other parameters $\boldsymbol{\mu}$ is not considered. We build on these approaches by considering the integral formulation but our proposed approach can deal with a much more general parameterization. Furthermore, these papers build on the traditional deterministic reduced order method; in our setting, we use randomized approaches, which offer additional benefits as highlighted below. In recent work [19], we have used the MPGMRES-Sh approach [20] for efficiently applying the fractional operator for a single parameter value $\boldsymbol{\mu}$. It was observed in this paper that the same basis can be used for multiple values of the exponent, which is important in our approach as well. This paper, therefore, considers the FOM and the ROM setting which is the focus of the present paper. However, the use of MPGMRES-Sh similar to [19] is an important aspect of this paper.

We also briefly mention that the idea of using randomization in ROMs has previously been explored; see, e.g., [21] and references within. Our approach is novel in two regards: the randomization method targets fractional PDEs which has not been considered previously, and the specific randomized compression technique we use in this paper is applicable to the streaming setting, in which several columns of the snapshot matrix are added sequentially. This approach allows the snapshots to be discarded after updating the sketches, therefore, lowering the storage costs involving snapshots.

Contributions In this paper, we derive reduced order models for solving (1). Our approach is based on the following features:

1. We use the spectral definition of the fractional elliptic operator and use the Kato integral formula to express the solution of (2) as an integral involving the solution of a parametric elliptic PDE with an additional parameter to compute the integral. The operator is then discretized using standard Lagrange finite elements, and the integral is discretized using the sinc quadrature approach.
2. We use a reduced basis approach for efficiently computing the solutions of the parametric elliptic PDE. It contains an offline stage in which a reduced basis is generated, and in the online stage, a projected system involving the reduced basis is solved.
3. The construction of the offline stage exploits the structure of the problem. We use an efficient iterative method MPGMRES-Sh to accelerate the convergence of shifted systems; this approach drastically reduces the number of PDE solves needed for generating the basis. Furthermore, we use randomized solvers to mitigate the cost of storing and compressing the snapshots, and an efficient method involving generalized eigendecomposition to accelerate the costs of the online stage.

As a part of future work, we plan to explore rigorous a posteriori error analysis [22,23] in the context of our randomized ROM approach.

2. Background

In this section, we discuss the integral approach to solving the fractional elliptic PDE. We also discuss the ingredients required for the ROM, namely the iterative solver MPGMRES-Sh (Section 2.2) and the randomized approach we use for compressing the snapshots (Section 2.3).

2.1. Integral approach

Here we follow the approach in [17, Section 2.2] that combines an integral formulation for the inverse fractional operator with a parameterized elliptic PDE. Using the Kato integral formula [24, Theorem 2], with a simplification $\lambda = 0$, the solution to (1) can be expressed as

$$y(\mathbf{x}; \boldsymbol{\mu}) = \frac{\sin(\alpha\pi)}{\pi} \int_{-\infty}^{\infty} e^{(1-\alpha)z} (e^z + \mathcal{A}(\mathbf{x}; \boldsymbol{\mu}))^{-1} b(\mathbf{x}; \boldsymbol{\mu}) dz.$$

After setting $\rho = (\boldsymbol{\mu}, z)$, we get the parametric PDE

$$(e^z + \mathcal{A}(\mathbf{x}; \boldsymbol{\mu}))u(\mathbf{x}; \rho) = b(\mathbf{x}; \boldsymbol{\mu}) \quad \mathbf{x} \in \Omega, \quad (2)$$

with appropriate boundary conditions, and arrive at the following integral formula for y

$$y(\mathbf{x}; \boldsymbol{\mu}) = \frac{\sin(\alpha\pi)}{\pi} \int_{-\infty}^{\infty} e^{(1-\alpha)z} u(\mathbf{x}; \rho) dz. \quad (3)$$

This integral approach is the foundation for the ROM that we construct. We now discuss the weak form of (2).

Weak form We aim to solve the parameterized PDE (2) for multiple instances of the parameters ρ . To this end, with an eye towards numerical approximation using finite elements, we consider the weak form of the problem, which reads: given $\rho \in \mathcal{P} \times \mathbb{R}_+$ find $u(\rho) \in V$ such that

$$a(u(\rho), v; \rho) = g(v; \rho) \quad \forall v \in V,$$

where the exact form of a depends on the form of $\mathcal{A}$ and the boundary conditions. We assume that the weak form has an affine parameter dependence as

$$a(u, v; \rho) := \sum_{t=1}^{n_a} f_t^a(\boldsymbol{\mu}) a_t(u, v) + e^z \int_{\Omega} u v d\mathbf{x}, \quad (4)$$

where $a_t : V \times V \rightarrow \mathbb{R}$ are independent of the parameters and the right-hand side also satisfies an affine parametric dependence

$$g(v; \rho) := \sum_{t=1}^{n_g} f_t^g(\boldsymbol{\mu}) g_t(v). \quad (5)$$

If the problem does not admit an affine parametric dependence, one can use techniques such as the empirical interpolation method [23, Chapter 10]. Following [25], we use finite element discretization in space and sinc quadrature for the integral, which we now discuss.

Discretization To perform the spatial discretization, we pick a finite-dimensional space $\mathcal{V}_h \subset H^1(\Omega)$ and then select a basis for this space denoted by $\{\phi_j\}_{j=1}^{N_h}$. This implies $\dim(\mathcal{V}_h) = N_h$ and any function $v_h \in \mathcal{V}_h$ can be represented by its expansion in the basis $v_h = \sum_{j=1}^{N_h} v_{h,j} \phi_j$. We denote the vector of its coefficients as $\mathbf{v} = [v_{h,1} \quad \dots \quad v_{h,N_h}]^T$. For the sinc quadrature to discretize the integral representation (3), let h represent the mesh parameter and define

$$Z_+ = \lceil \frac{\pi^2}{4\alpha\zeta^2} \rceil, \quad Z_- = \lceil \frac{\pi^2}{4(1-\alpha)\zeta^2} \rceil, \quad (6)$$

where $\zeta = 1/\log(1/h)$. We also define the quadrature weights and nodes as

$$w_j = \frac{\zeta \sin(\alpha\pi)}{\pi} e^{(1-\alpha)z_j}, \quad z_j = \zeta j, \quad -Z_- \leq j \leq Z_+. \quad (7)$$

Let $a(u, v; \boldsymbol{\mu})$ denote the weak form of the operator $\mathcal{A}(\mathbf{x}; \boldsymbol{\mu})$ with the corresponding boundary conditions; similarly, let $g(v; \boldsymbol{\mu})$ be the weak form of $b(\mathbf{x}; \boldsymbol{\mu})$. We seek a solution $u_{h,j} \in \mathcal{V}_h$ such that

$$a(u_{h,j}(\mathbf{x}; \boldsymbol{\mu}), v_h(\mathbf{x}); \boldsymbol{\mu}) + e^{z_j} \int_{\Omega} u_{h,j}(\mathbf{x}; \boldsymbol{\mu}) v_h(\mathbf{x}) d\mathbf{x} = g(v_h; \boldsymbol{\mu}), \quad \text{for all } v_h \in \mathcal{V}_h \quad (8)$$

and for $-Z_- \leq j \leq Z_+$. Having obtained the solutions $u_{h,j}$, the approximate solution y_h to (3) can be obtained using the sinc-quadrature formula as $y_h(\mathbf{x}; \boldsymbol{\mu}) = \sum_{j=-Z_-}^{Z_+} w_j u_{h,j}(\mathbf{x}; \boldsymbol{\mu})$. However, as can be seen, computing this quadrature formula requires the solution of many PDEs of the form (8). This can be computationally intensive if this process has to be repeated for multiple values of the exponent α and the parameter $\boldsymbol{\mu}$. This motivates the need for a ROM approach.

We proceed to discuss the matrix representation of this parametric PDE that will be the starting point of the ROMs. As before, let $\mathcal{V}_h \subset V$ be a finite-dimensional subspace of V with $\dim(\mathcal{V}_h) = N_h$ with a basis $\{\phi_1, \dots, \phi_{N_h}\}$. We denote the matrix representation of the bilinear forms as

$$[\mathbf{A}_t]_{ij} = a_t(\phi_i, \phi_j) \quad 1 \leq i, j \leq N_h, \quad 1 \leq t \leq n_a. \quad (9)$$

We denote by $\mathbf{M}$, the mass matrix $\mathbf{M} \in \mathbb{R}^{N_h \times N_h}$ with entries $[\mathbf{M}]_{ij} = \int_{\Omega} \phi_i \phi_j d\mathbf{x}$ for $1 \leq i, j \leq N_h$. We also define the vectors $\mathbf{g}_t$ with entries $[\mathbf{g}_t]_i = g_t(\phi_i)$ for $1 \leq i \leq N_h$, and $1 \leq t \leq n_g$. In summary, the discretized parametric system we aim to solve is

$$\left(\sum_{t=1}^{n_a} f_t^a(\boldsymbol{\mu}) \mathbf{A}_t + e^z \mathbf{M} \right) \mathbf{u} = \sum_{t=1}^{n_g} f_t^g(\boldsymbol{\mu}) \mathbf{g}_t, \quad (10)$$

for the set of parameters $\rho = (\mu, z)$.

The Ref. [25] establishes exponential convergence in quadrature and also derives appropriate error estimates with respect to spatial discretization. Provided that the so-called Kolmogorov n -width in the parameter space decays exponentially, which is indeed the case when the solution depends smoothly on parameters (cf. [26]), one can establish that reduced basis error also decays exponentially. In the case that the reduced bases are generated using a greedy method, we refer to [27] for such results. We now discuss an efficient method for solving the linear systems arising from the system (8).

2.2. MPGMRES-Sh

MPGMRES-Sh [20] is an iterative method for solving a sequence of shifted systems of the form

$$(\mathbf{C}_1 + \sigma_j \mathbf{C}_2) \mathbf{x}_j = \mathbf{b} \quad 1 \leq j \leq N_\sigma, \quad (11)$$

for a set of shifts $\{\sigma_j\}_{j=1}^{N_\sigma}$ and matrices $\mathbf{C}_1, \mathbf{C}_2 \in \mathbb{R}^{n \times n}$. The main idea of this approach is to use shift-and-invert preconditioners of the form $(\mathbf{C}_1 + \tau_j \mathbf{C}_2)_{j=1}^{n_p}$, where n_p is the number of preconditioners used. A simplified version of the algorithm is given in Algorithm 1.

Algorithm 1 Simplified version of MPGMRES-Sh

Require: Matrices $\mathbf{C}_1, \mathbf{C}_2$, Shifts $\{\sigma_j\}$ for $1 \leq j \leq N_\sigma$, Vector $\mathbf{b}$, Preconditioners $\mathbf{P}_j = \mathbf{C}_1 + \tau_j \mathbf{C}_2$, $1 \leq j \leq n_p$

- 1: Compute $\beta = \|\mathbf{b}\|_2$ and $\hat{\mathbf{V}}^{(1)} = \mathbf{b}/\beta$
 - 2: **for** $k = 1, \dots$, until convergence **do**
 - 3: $\hat{\mathbf{v}}_k = \hat{\mathbf{V}}^{(k)} \mathbf{e}_{n_p}$ and $\mathbf{W}^{(k)} = \begin{bmatrix} \mathbf{P}_1^{-1} \hat{\mathbf{v}}_k & \dots & \mathbf{P}_{n_p}^{-1} \hat{\mathbf{v}}_k \end{bmatrix}$
 - 4: $\hat{\mathbf{W}} = \mathbf{C}_2 \mathbf{W}^{(k)}$
 - 5: **for** $j = 1, \dots, k$ **do**
 - 6: $\mathbf{H}^{(j,k)} = (\hat{\mathbf{V}}^{(j)})^\top \hat{\mathbf{W}}$
 - 7: $\hat{\mathbf{W}} = \hat{\mathbf{W}} - \mathbf{V}^{(j)} \mathbf{H}^{(j,k)}$
 - 8: **end for**
 - 9: $\hat{\mathbf{W}} = \mathbf{QR}$ {Thin QR factorization}
 - 10: Construct $\hat{\mathbf{V}}^{(k+1)} = \mathbf{Q}$ and $\mathbf{H}^{(k+1,k)} = \mathbf{R}$
 - 11: Check for convergence for all $\{\sigma_j\}_{j=1}^{N_\sigma}$ and terminate, if converged, at iteration k_m .
 - 12: **end for**
 - 13: Consolidate matrices $\mathbf{Z}_{k_m} = [\mathbf{W}^{(1)} \quad \dots \quad \mathbf{W}^{(k_m)}]$
 - 14: **return** Basis $\mathbf{Z}_{k_m}$, where k_m is the number of iterations upon termination
-

The idea behind this approach is to form a basis for the search space $\mathbf{Z}_{k_m} \in \mathbb{R}^{n \times k_m n_p}$ where k_m is the number of MPGMRES-Sh iterations and at each iteration, we increment the dimension of the search space by n_p . Sometimes the basis fails to be incremented by n_p but this can be handled using rank-revealing QR factorizations; we do not consider this in the present work but the reader is referred to [20]. This search space is common to all the shifted systems and is used to solve a projected subproblem for each shift σ of size $(k_m n_p) \times (k_m n_p)$ to obtain a solution $\mathbf{p}(\sigma) \in \mathbb{R}^{(k_m n_p) \times 1}$. In a bit more detail, the approximate solution is computed as $\mathbf{x}_{k_m}(\sigma) = \mathbf{Z}_{k_m} \mathbf{p}_{k_m}(\sigma)$, where $\mathbf{p}_{k_m}$ solves the optimization problem

$$\min_{\mathbf{p} \in \mathbb{R}^{(k_m n_p) \times 1}} \|\mathbf{b} - (\mathbf{C}_1 + \sigma_j \mathbf{C}_2) \mathbf{Z}_{k_m} \mathbf{p}\|_2^2,$$

for each shift σ_j , $1 \leq j \leq N_\sigma$. In the application of interest in this present paper, we will focus on computing the basis $\mathbf{Z}_{k_m}$ rather than the solutions $\mathbf{x}(\sigma)$ as we now explain in this remark.

Remark 1. Since $\mathbf{Z}_{k_m}$ is the search space corresponding to the shifted systems (11), we can write $\mathbf{x}_j \approx \mathbf{Z}_{k_m} \mathbf{p}_{k_m}(\sigma_j)$. Therefore, the matrix

$$\mathbf{X} = [\mathbf{x}_1 \quad \dots \quad \mathbf{x}_{N_\sigma}] \approx \mathbf{Z}_{k_m} [\mathbf{p}_{k_m}(\sigma_1) \quad \dots \quad \mathbf{p}_{k_m}(\sigma_{N_\sigma})].$$

That is, the range of $\mathbf{X}$ is approximately captured by the matrix $\mathbf{Z}_{k_m}$. Since the leading left singular vectors approximate the range of $\mathbf{X}$, it is sufficient to store $\mathbf{Z}_{k_m}$. This key insight is important to reduce computational costs in the offline stage.

The number of iterations is determined when all the relative residual for the shifted systems is smaller than 10^{-8} . For more implementation details and theoretical justifications, the reader is referred to [20]. If the number of MPGMRES-Sh iterations, k_m , and the number of preconditioners, n_p , are both small, then the cost of solving the projected problem across multiple shifts is small and the cost of MPGMRES-Sh is dominated by the cost of forming the basis for the search space. In our applications, the number of preconditioners is 3 and the number of iterations is smaller than 30 yielding a basis $\mathbf{Z}_{k_m}$ of size at most 90. We now discuss the second main ingredient in our approach, the randomized low-rank approximations.

2.3. Randomized low-rank approximation

A reduced-order model requires compression of the matrix containing snapshots of the state. In this paper, we pursue a randomized approach since it has reduced computational and storage requirements compared to the full SVD. Furthermore, as we will explain, this approach is beneficial in our setting where the matrix is available in a streaming fashion.

Suppose we wanted to compute a low-rank approximation to the matrix $\mathbf{S} \in \mathbb{R}^{m \times n}$. We follow the approach from [28]. We draw two random matrices $\mathbf{\Omega} \in \mathbb{R}^{n \times \ell_1}$ and $\mathbf{\Psi} \in \mathbb{R}^{m \times \ell_2}$; we take them to be standard Gaussian random matrices, which means that the entries are independent and drawn from the Gaussian distribution with zero mean and unit variance. We form the sketches $\mathbf{Y}_1 = \mathbf{S}\mathbf{\Omega}$ and $\mathbf{Y}_2 = \mathbf{S}^\top \mathbf{\Psi}$, which approximate the column and row spaces of $\mathbf{S}$ respectively. Then compute the thin-QR factorization $\mathbf{Y}_1 = \mathbf{Q}\mathbf{R}$ and discard $\mathbf{R}$. The low-rank approximation of $\mathbf{S}$ can be computed as

$$\mathbf{S} \approx \mathbf{S}_{\text{rand}} \equiv \mathbf{Q}(\mathbf{\Psi}^\top \mathbf{Q})^\dagger \mathbf{Y}_2^\top,$$

where † is the Moore–Penrose inverse. The low-rank approximation can be converted into the SVD format to extract singular vectors with relatively less additional work. If a rank- r approximation of the matrix $\mathbf{S}$ is desired, assuming that the matrix is stored in dense form, the computational cost of this algorithm is $\mathcal{O}(rmn + r^2(m+n) + r^3)$ flops. The choice of the sketch sizes ℓ_1 and ℓ_2 is discussed in [28, Section 4.4] which we follow here as well. We choose $\ell_1 = 2r + 1$ and $\ell_2 = 2\ell_1 + 1$. Based on this sketch size, the error in expectation of the low-rank approximation satisfies [28, Theorem 4.3]

$$\mathbb{E} \|\mathbf{S} - \mathbf{S}_{\text{rand}}\|_F^2 \leq 4 \left(\sum_{j>r} \sigma_j^2 \right),$$

where $\{\sigma_j\}_{j=1}^{\min\{m,n\}}$ are the singular values of the snapshot matrix $\mathbf{S}$. The main message is that the error in the low-rank approximation due to the randomization is, in expectation, within a small factor of the “best” low-rank approximation (note that this approximation need not be unique). In particular, if the singular values of the snapshot matrix decay rapidly then the error in the low-rank approximation is very small.

Besides lower computational costs compared to the full SVD algorithm (i.e., $\mathcal{O}(mn \min\{m, n\})$), the proposed approach is amenable to the streaming setting where the matrix is updated sequentially. That is, if the matrix $\mathbf{S}$ is of the form

$$\mathbf{S} = \mathbf{S}_1 + \dots + \mathbf{S}_M,$$

where the summands $\mathbf{S}_j$ are available sequentially. As an example that is relevant to our application, consider the case, where the columns (or a block of columns) of a matrix are computed sequentially. It is easy to see that the sketches $\mathbf{Y}_1$ and $\mathbf{Y}_2$ can also be updated sequentially. Furthermore, once the sketches have been computed, the appropriate columns can be discarded since all the randomized algorithm only requires the sketches to compute the low-rank approximation. These insights will be relevant to the proposed algorithm resulting in an algorithm with lower storage and memory costs.

3. Reduced basis approaches

In this section, we present two approaches for ROM: a naïve approach (Section 3.1), and an efficient approach (Section 3.2). We conclude this section with a discussion on computational costs.

3.1. A naïve first approach

Before we present our efficient approach, we first discuss a naïve approach that will be the basis for comparison with our proposed approach.

We first draw a set of random instances of the parameters $\mathcal{M} = \{\mu^{(j)}\}$ for $1 \leq j \leq M$; strategies for sampling are discussed in [23, Section 6.6-6.7]. Similarly, using the definitions in Section 2.1 we set the parameters $\mathcal{Z} = \{z_j\}$ for $-Z_- \leq j \leq Z_+$ with $Z = Z_- + Z_+ + 1$ corresponding to the setting $\alpha = 0.5$. This is motivated by the observation in [19, Section 5.1] that the exponents e^{z_k} are completely independent of the exponent, except for the limits Z_- and Z_+ . As we will see in the numerical experiments, it is sufficient to generate the projection basis corresponding to a single exponent α , which is then applicable to a wider range of the exponent.

Thus, we now have the set of parameters $\mathcal{M} \times \mathcal{Z}$ for ρ . For each parameter value $\mu^{(j)}$ we first construct the matrices and vectors

$$\mathbf{K}_j = \sum_{i=1}^{n_g} f_i^a(\mu_j) \mathbf{A}_i, \quad \mathbf{f}_j = \sum_{i=1}^{n_g} f_i^g(\mu_j) \mathbf{g}_i, \quad 1 \leq j \leq M. \quad (12)$$

Then we solve the sequence of shifted linear equations

$$(\mathbf{K}_j + e^{z_k} \mathbf{M}) \mathbf{u}_{j,k} = \mathbf{f}_j \quad 1 \leq j \leq M, \quad -Z_- \leq k \leq Z_+, \quad (13)$$

to generate the snapshot matrix

$$\mathbf{S} = [\mathbf{U}^{(1)} \quad \dots \quad \mathbf{U}^{(M)}] \in \mathbb{R}^{N_h \times (MZ)}, \quad \mathbf{U}^{(j)} = [\mathbf{u}_{1,-Z_-} \quad \dots \quad \mathbf{u}_{j,Z_+}] \quad 1 \leq j \leq M. \quad (14)$$

The next step is to compute the dominant left singular vectors of $\mathbf{S}$ denoted $\mathbf{V} \in \mathbb{R}^{N_h \times K}$. We form the matrices $\hat{\mathbf{A}}_t = \mathbf{V}^T \mathbf{A}_t \mathbf{V}$ for $1 \leq t \leq n_a$, $\hat{\mathbf{M}} = \mathbf{V}^T \mathbf{M} \mathbf{V}$, and vectors $\hat{\mathbf{g}}_t = \mathbf{V}^T \mathbf{g}_t$ for $1 \leq t \leq n_g$.

This basis $\mathbf{V}$ can be used to approximate the solution space in the online stage. Given a new parameter value $\bar{\mu}$, and an exponent $\alpha \in (0, 1)$, we solve the projected system (see (6) for definitions of Z_+ , Z_-)

$$\left(\sum_{t=1}^{n_a} f_t^a(\bar{\mu}) \hat{\mathbf{A}}_t + e^{z_k} \hat{\mathbf{M}} \right) \hat{\mathbf{u}}_k = \left(\sum_{t=1}^{n_g} f_t^g(\bar{\mu}) \hat{\mathbf{g}}_t \right) \quad -Z_- \leq k \leq Z_+. \quad (15)$$

The approximation solution is $\hat{\mathbf{y}} = \mathbf{V} \left(\sum_{k=-Z_-}^{Z_+} w_k \hat{\mathbf{u}}_k \right)$. The details of this approach are given in Algorithm 2.

Algorithm 2 Reduced basis approach for fractional elliptic PDE: Online stage

Require: Sample parameter $\bar{\mu}$, Precomputed matrices $\{\hat{\mathbf{A}}_t\}_{t=1}^{n_a}$ and vectors $\{\hat{\mathbf{g}}_t\}_{t=1}^{n_g}$, exponent α

- 1: Compute quadrature nodes $\{z_k\}_{k=-Z_-}^{Z_+}$ and weights $\{w_k\}_{k=-Z_-}^{Z_+}$ corresponding to the exponent α ; see (6) and (7).
 - 2: Form and solve the projected system (15)
 - 3: Evaluate $\hat{\mathbf{w}} = \mathbf{V} \left(\sum_{k=-Z_-}^{Z_+} w_k \hat{\mathbf{u}}_k \right)$.
 - 4: **return** Approximate solution $\hat{\mathbf{w}}$
-

However, the cost of generating and compressing the snapshot matrix can be prohibitively expensive in practice as we now explain. This is because MZ linear systems need to be solved in total to compute the snapshot matrix. Note that M can range from 10–100 whereas Z can range from 100–1000, depending on the value of the exponent α and the discretization of the system. Furthermore, the size of $\mathbf{S}$ can be large so the cost of compressing this system is large; the cost of SVD is $\mathcal{O}(N_h M Z \min\{N_h, MZ\})$. Even the cost of storing the matrix may be expensive in practice. To address these issues we propose a new approach for efficiently generating and compressing the basis.

3.2. Proposed approach

There are two components to the proposed approach in the offline stage: the use of MPGMRES-Sh to solve the shifted systems and randomization to compress the snapshot matrix. In the online stage, we use a generalized eigendecomposition to speed up computational costs.

Offline stage First, we solve the systems (13) for each value of $\mu^{(j)}$ using the MPGMRES-Sh approach (Section 2.2). The algorithm is implemented with $\mathbf{C}_1 = \mathbf{M}$ and $\mathbf{C}_2 = \mathbf{A}_j$ and shifts $\sigma_k = 1/\exp(z_k)$ for $-Z_- \leq k \leq Z_+$. We get a basis $\mathbf{Z}^{(j)} \in \mathbb{R}^{N_h \times (n_m)}$ where n_m is the size of the MPGMRES-Sh basis (obtained as $n_m = k_m n_p$, i.e., the number of MPGMRES-Sh iterations k_m times the number of preconditioners n_p). For simplicity, we have assumed that the MPGMRES-Sh basis size is the same for every j but in practice, it may be different. We have also dropped the dependence of the basis on the number of iterations k_m . In all the experiments, for MPGMRES-Sh we take three preconditioners $n_p = 3$ with the corresponding values of $\tau \in \{10^{-8}, 10^{-4}, 10^{-2}\}$. This is the same setup as in [19] and we refer to this paper for additional implementation details. The preconditioners are factorized and their factorizations are stored.

After solving the j th set of systems, we have the basis $\mathbf{Z}^{(j)}$ with which we can approximate the solutions of $\mathbf{U}^{(j)}$; see (14). Therefore, rather than compute the matrix $\mathbf{U}^{(j)}$ and then compress it subsequently, following Remark 1, we directly compute the new “snapshot” matrix as $\hat{\mathbf{S}} = [\mathbf{Z}^{(1)} \quad \dots \quad \mathbf{Z}^{(M)}] \in \mathbb{R}^{N_h \times M n_m}$. This is because the range of $\mathbf{U}^{(j)}$ is approximately captured by the range of $\mathbf{Z}^{(j)}$, since $\mathbf{Z}^{(j)}$ is the search space for the set of shifted systems. Although it is not technically a matrix of snapshots, we still refer to $\hat{\mathbf{S}}$ as a snapshot matrix. Notice that the number of columns of $\hat{\mathbf{S}}$ is much smaller than that of $\mathbf{S}$ if $n_m \ll Z$, as is the case in numerical experiments.

Even with this reduction, the cost of storing and computing the dominant singular vectors can be expensive. To alleviate these costs, we use the single view randomized approach as described in Section 2.3. To this end, we consider two sketching matrices $\mathbf{\Omega} \in \mathbb{R}^{(M n_m) \times \ell_1}$ and $\mathbf{\Psi} \in \mathbb{R}^{N_h \times \ell_2}$ which are standard Gaussian random matrices. Given the target basis size K , we choose $\ell_1 = 2K + 1$ and $\ell_2 = 2\ell_1 + 1$. We observe that the matrix $\hat{\mathbf{S}}$ can be written as

$$\hat{\mathbf{S}} = \hat{\mathbf{S}}_1 + \dots + \hat{\mathbf{S}}_M \quad \hat{\mathbf{S}}_j = \mathbf{Z}^{(j)} \mathbf{E}_j^T,$$

where, using MATLAB notation, $\mathbf{E}_j = \mathbf{I}_{M n_m}(:, (j-1) * n_m + 1 : j * n_m)$ for $1 \leq j \leq M$ contains columns from the $(M n_m) \times (M n_m)$ identity matrix. Therefore, we can easily update the sketches $\mathbf{Y}_1$ and $\mathbf{Y}_2$ for each j and then discard the bases $\mathbf{Z}^{(j)}$. In fact, the matrix $\mathbf{\Omega}$ does not need to be formed all at once; this is beneficial since the basis size n_m may not be known in advance. Consider $\mathbf{\Omega} = [\mathbf{\Omega}_1^T \quad \dots \quad \mathbf{\Omega}_M^T]^T$, where $\mathbf{\Omega}_j \in \mathbb{R}^{n_m \times \ell_1}$. Then the sketch $\mathbf{Y}_1$ can be formed as

$$\mathbf{Y}_1 = \mathbf{Z}^{(1)} \mathbf{\Omega}_1 + \dots + \mathbf{Z}^{(M)} \mathbf{\Omega}_M.$$

The details of this approach are given in Algorithm 3. In numerical experiments, we use the decay of the singular values of the snapshot matrix to guide the choice of the target basis K . However, in practice, one can use adaptive randomized methods to estimate the basis size [29,30].

Online stage In the online stage, the steps involve solving the $K \times K$ systems for $Z = Z_+ + Z_- + 1$, see (15) which can be expensive. Rather than factorizing each linear system, we can use a generalized eigendecomposition. To this end, define $\hat{\mathbf{K}} = \sum_{j=1}^{n_a} f_t^a(\bar{\mu}) \hat{\mathbf{A}}_t$, $\hat{\mathbf{g}} = \sum_{j=1}^{n_g} f_t^g(\bar{\mu}) \hat{\mathbf{g}}_t$, and consider a generalized eigendecomposition of the pair $(\hat{\mathbf{K}}, \hat{\mathbf{M}})$; that is, compute a Cholesky factorization $\hat{\mathbf{M}} = \mathbf{L}\mathbf{L}^\top$ and compute an eigendecomposition $\mathbf{L}^{-1}\hat{\mathbf{K}}\mathbf{L}^\top = \mathbf{U}\mathbf{A}\mathbf{U}^\top$. We are guaranteed such an eigendecomposition since both $\mathbf{M}$ and $\hat{\mathbf{M}}$ are symmetric positive definite. Then, we can obtain the solution $\hat{\mathbf{y}}$ as

$$\hat{\mathbf{y}} = (\mathbf{V}\mathbf{L}^{-\top}\mathbf{U}) \left(\sum_{k=-Z_-}^{Z_+} w_k (\mathbf{A} + e^{z_k} \mathbf{I})^{-1} \right) \mathbf{U}^\top \mathbf{L}^{-1} \hat{\mathbf{g}}. \quad (16)$$

The modification to Algorithm 2 is straightforward and therefore omitted.

Algorithm 3 Reduced basis approach for fractional elliptic PDE: Offline stage

Require: Parameter samples $\mathcal{M} = \{\mu^{(1)}, \dots, \mu^{(M)}\}$, basis size K

Assemble the matrices $\mathbf{A}_t$ for $1 \leq t \leq n_a$ and vectors $\mathbf{g}_t$ for $1 \leq t \leq n_g$.

Compute the nodes $z_k = k\zeta$ where $\zeta = 1/\log(1/h)$ and $-Z_- \leq k \leq Z_+$ where $Z_+ = \lceil \frac{\pi^2}{2\zeta^2} \rceil$, $Z_- = \lceil \frac{\pi^2}{2\zeta^2} \rceil$

Draw random matrix $\Psi \in \mathbb{R}^{N_h \times \ell_2}$

Initialize sketches $\mathbf{Y}_1 = \mathbf{0} \in \mathbb{R}^{N_h \times \ell_1}$, $\mathbf{Y}_2 = []$

{Stage 1: Constructing sketches of the snapshot matrix}

for $j \in \{-Z_-, \dots, Z_+\}$ **do**

Form matrix $\mathbf{K}_j = \sum_{t=1}^{n_a} f_t^a(\mu_j) \mathbf{A}_t$ and vector $\mathbf{f}_j = \sum_{t=1}^{n_g} f_t^g(\mu_j) \mathbf{g}_t$

Solve the systems of linear equations (13) using MPGMRES-Sh to generate a basis $\mathbf{Z}^{(j)} \in \mathbb{R}^{N_h \times n_m}$.

Draw random matrix $\Omega_j \in \mathbb{R}^{n_m \times \ell_1}$

Update sketches $\mathbf{Y}_1 = \mathbf{Y}_1 + \mathbf{Z}^{(j)} \Omega_j$ and $\mathbf{Y}_2 = [\mathbf{Y}_2; (\mathbf{Z}^{(j)})^\top \Psi]$

end for

{Stage 2: Compression of snapshot matrix}

Compute thin QR $\mathbf{Y}_1 = \mathbf{Q}\mathbf{R}$

Compute $\mathbf{W} = (\Psi^\top \mathbf{Q})^\dagger \mathbf{Y}_2^\top$ and its thin SVD $\mathbf{U}_W \Sigma_W \mathbf{V}_W^\top$

Compute $\mathbf{V} = \mathbf{Q}\mathbf{U}_W(:, 1 : K)$ where K is the basis size

Precompute $\hat{\mathbf{A}}_t = \mathbf{V}^\top \mathbf{A}_t \mathbf{V}$ for $1 \leq t \leq n_a$, $\hat{\mathbf{M}} = \mathbf{V}^\top \mathbf{M} \mathbf{V}$, and vector $\hat{\mathbf{g}}_t = \mathbf{V}^\top \mathbf{M} \mathbf{w}$

return Basis vectors $\mathbf{V}$, matrices $\{\hat{\mathbf{A}}_t\}_{t=1}^{n_a}$ and vectors $\{\hat{\mathbf{g}}_t\}_{t=1}^{n_g}$

3.3. Computational cost

We now summarize the computational cost of the offline and online stages.

In the offline stage, there are two sources of computational cost: the snapshot matrix construction and compression. The construction essentially involves the cost of applying MPGMRES-Sh to M set of Z shifted systems. Let m_p be the number of preconditioners and k_m be the number of MPGMRES-Sh iterations and their product is $n_m = k_m n_p$; then the cost of one set of MPGMRES-Sh iterations is

$$T_{\text{MP}} = n_p T_{\text{PC}} + T_{\text{PA}} n_m + \mathcal{O}(N_h Z n_m^2 + Z n_m^3) \quad \text{flops},$$

where T_{PC} is the cost of constructing each preconditioner and T_{PA} is the cost of applying the combination of the matrix $\mathbf{C}_2$ and the preconditioner $\mathbf{P}_j$ (lines 3–4 of Algorithm 1). The total cost of forming the snapshot matrix is $T_S = M T_{\text{MP}}$. The cost of the randomized low-rank approximation is analyzed in Section 2.3; assuming the basis size K , the cost is $T_R = \mathcal{O}(N_h n_m M K + K^2(N_h + n_m) + K^3)$ flops. The total cost of the offline stage is then simply the sum of T_S and T_R .

For the online stage, the naïve approach requires us to solve Z systems with $\mathcal{O}(K^3)$ cost per system, with a total cost of $\mathcal{O}(Z K^3 + N_h K^2)$ flops. The proposed approach only requires $\mathcal{O}(K^3)$ cost for the eigendecomposition and $\mathcal{O}(K^2)$ flops per system solve for a total cost $\mathcal{O}(K^3 + K^2(Z + N_h))$ flops. If K is large, computing the eigendecomposition is computationally expensive or even prohibitive. Then one can use techniques from [31] to further reduce the computational cost. Another option is to view (15) through the lens of a generalized Sylvester equation; one can then use techniques from [32] to reduce the computational costs.

4. Numerical experiments

In this section, we explore the performance of the proposed ROMs on three illustrative applications. The relative error is measured in the L^2 norm with a reference finite element solution computed using MPGMRES-Sh. The finite element discretization was carried out using continuous piecewise linear elements. The MPGMRES-Sh solver has been validated in [19, Section 5.1]. The timing results were performed on a Mac Mini 2020, with 16 GB RAM and running MATLAB 2021a.

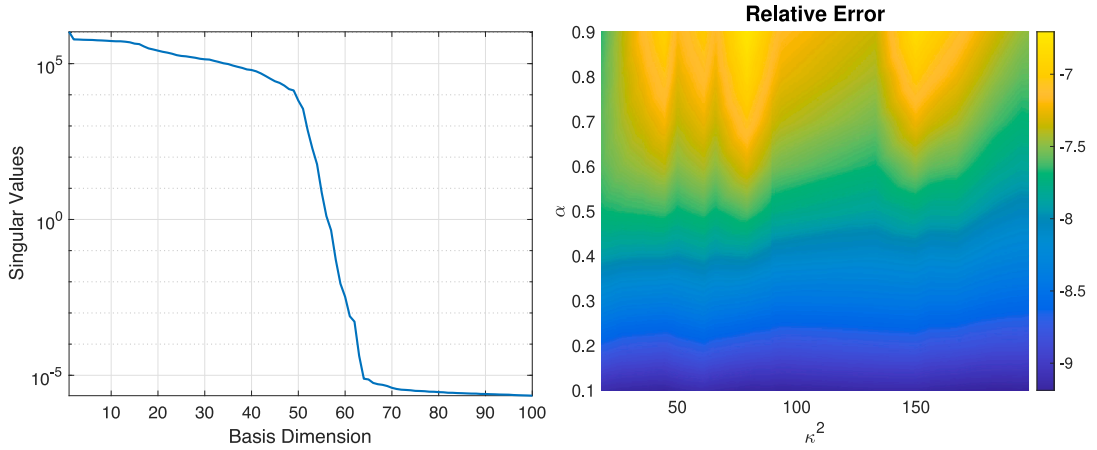


Fig. 1. (left) Singular values of the snapshot matrix $\hat{S}$. (right) The relative error ($\log_{10}$) in the test set.

Table 1

Wall-clock time (in seconds) of the time for computing the snapshot matrix $\hat{S}$ and the compression stages.

	Baseline	Proposed
Snapshot	47.03	46.66
Compression	18.69	0.59

4.1. Gaussian process

In this application, we consider simulation from the Gaussian process, by solving the stochastic PDE on the domain $\Omega \in \mathbb{R}^d$

$$(\kappa^2 - \Delta)^{(\nu+d/2)/2} u(\mathbf{x}) = \xi(\mathbf{x}), \quad \mathbf{x} \in \Omega \quad (17)$$

where ξ is the white noise process, and in the exponent ν is the smoothness parameter and d the spatial dimension. The parameter κ^2 is related to the correlation length of the process. We want to compute the solution of the SPDE for multiple values of the parameters κ^2 and $\alpha = (\nu + d/2)/2$. For this application, there is a single parameter $\mu = (\kappa^2)$. Therefore, $n_a = 2$ with $f_1^a(\mu) = 1$ and $f_2^a(\mu) = \kappa^2$; similarly, $n_g = 1$ with $f_1^g(\mu) = 1$. The weak forms are $a_1 = \int_{\Omega} \nabla u \cdot \nabla v d\mathbf{x}$, $a_2 = \int_{\Omega} u v d\mathbf{x}$, and $g_1 = \int_{\Omega} \xi v d\mathbf{x}$.

Experiment 1: Accuracy We take the domain $\Omega = (0, 1)^2$ and consider homogeneous Neumann boundary conditions for the SPDE. The domain is discretized using a grid of size $257 \times 257 = 66,049$ points. We use a basis size of $K = 100$ based on the decay of the singular values. We generate a training set of parameters $\kappa^2 \in [10, 200]$ in increments of 5, giving a training set of size $|\mathcal{M}| = 39$. We investigate the accuracy of the approach for several instances of κ^2 and α . Specifically, we generate a test set involving 20 values of $\kappa^2 \in [10, 200]$ and $\alpha \in [0.1, 0.9]$ in increments of 0.1 giving a test set of 180 candidates. As can be seen from Fig. 1, the relative error is small throughout the test set with errors ranging from 10^{-9} to 10^{-7} . The errors show a mild increase with the exponent α .

Experiment 2: Timings In this next experiment, we investigate the reduction in computational costs. We focus on the timing of the compression. The time of solving (13) for a single parameter sample $\mu^{(j)}$ using the naïve approach is 23.33 s whereas for MPGMRES-Sh the cost is 1.28 s (this time includes the time to factorize the preconditioners). Furthermore, in the naïve approach, the size of the snapshot matrix S is $66,049 \times 11895$, whereas in the proposed approach the size of the snapshot matrix is $66,049 \times 1872$. The size of the MPGMRES-Sh basis for each $\mu^{(j)}$ was 48. Since the cost of forming and compressing the snapshot matrix is enormous, we do not consider the naïve approach for timing purposes anymore. For the baseline approach, we use the approach in Section 3.2 but use SVD instead of the randomized approach.

We report both the time to construct the snapshot matrices $\hat{S}$ and the time for compression (SVD vs randomization). The results are reported in Table 1. Both methods give a comparable time for constructing the matrix. However, the baseline approach takes slightly more time (since the total number of MPGMRES-Sh iterations is not known in advance, and memory has to be allocated on-the-fly which makes it inefficient). Note that the proposed approach does not form the matrix $\hat{S}$ and the reported time includes the time to form the sketches Y_1 and Y_2 . However, when it comes to the compression costs, our approach shows a 30 $\times$ speedup over the SVD. This speedup will be more significant for larger-scale problems.

Finally, we report that the time for evaluating the ROM at the value $\alpha = 0.5$ is 0.01 s. In comparison the time for MPGMRES-Sh (1.28 s) and the naïve approach (23.33 s); it is, therefore, clear that the ROM has significant speedups over the FOMs.

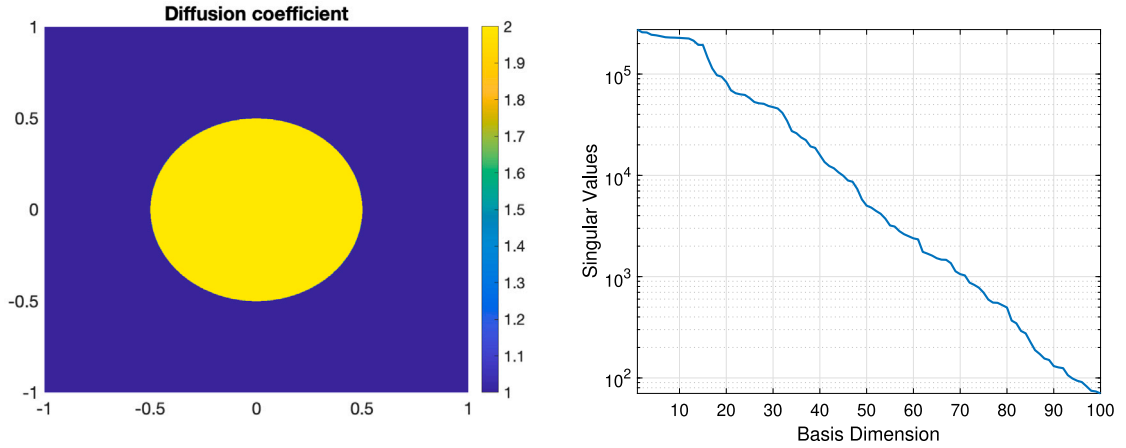


Fig. 2. (left) Visualization of the diffusion coefficient $D(\mathbf{x}; \mu)$; (right) first 100 singular values of the snapshot matrix $\hat{\mathbf{S}}$.

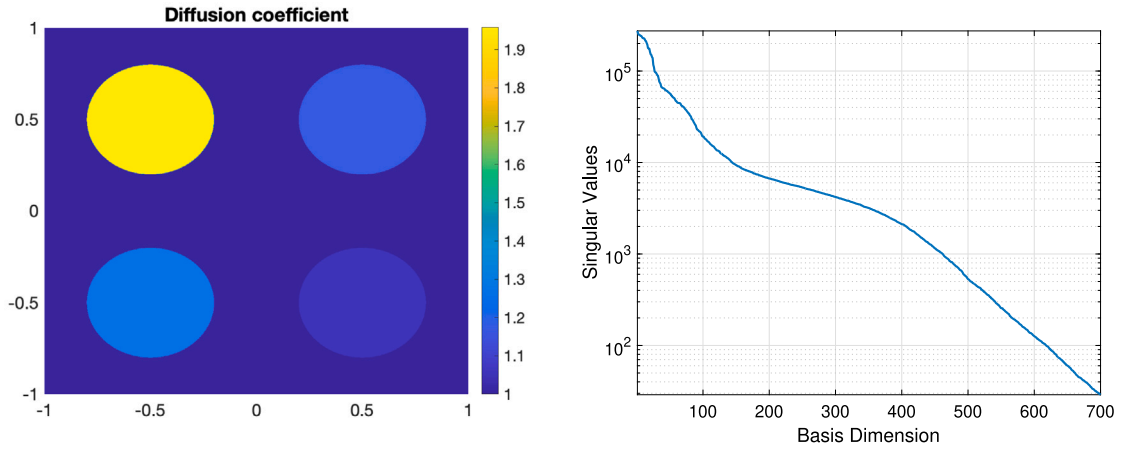


Fig. 3. (left) Visualization of the diffusion coefficient $D(\mathbf{x}; \mu)$ for Case B; (right) first 700 singular values of the snapshot matrix $\hat{\mathbf{S}}$.

4.2. Fractional cookies

In this experiment, inspired by the cookies test problem from [33], we consider the fractional parameterized PDE (2) with the domain $\Omega = (-1, 1)^2$ with Dirichlet boundary conditions, and the elliptic operator is given by $\mathcal{A}(\mathbf{x}; \mu) = -\nabla \cdot (D(\mathbf{x}; \mu) \nabla(\cdot))$ where

$$D(\mathbf{x}; \mu) = 1 + \sum_{i=1}^p \mu_i \chi_{\Omega_i}(\mathbf{x})$$

where χ_A is the characteristic function (that takes the value 1 if $\mathbf{x} \in A \subset \Omega$ and 0 otherwise), and p is the number of regions (“cookies”). We consider two different cases for the discs:

1. Case A: the number of discs $p = 1$ and Ω_1 is a disc of radius 0.5 and center $(0, 0)$.
2. Case B: the number of discs $p = 4$ and Ω_j are discs of radii 0.3 with centers $(-0.5, -0.5)$, $(-0.5, 0.5)$, $(0.5, 0.5)$, and $(0.5, -0.5)$.

See Figs. 2 and 3 for visualizations of the diffusion coefficient D .

We take the values of $\mu \in [0, 1]^p$. The fractional PDE is augmented with zero Dirichlet boundary conditions. For this problem, $n_a = p + 1$ with $a_t(u, v) = \int_{\Omega_t} \nabla u \cdot \nabla v d\mathbf{x}$ for $1 \leq t \leq p$ and $a_5 = \int_{\Omega} \nabla u \cdot \nabla v d\mathbf{x}$; similarly, $n_g = 1$ and $g_1(v) = \int_{\Omega} v d\mathbf{x}$. We discretize the domain using a grid of size 257^2 . To generate the training set, we pick $M = 100$ parameters $\mu^{(j)}$ from $[0, 1]^p$ sampled using Latin hypercube sampling. Based on the decay of the singular values, we take the reduced basis size to be $K = 100$ (Case A) and $K = 700$ (Case B). Compared to the test problem in Section 4.1, the singular values of the snapshot matrix do not decay as sharply in Case B (see Fig. 3). For the test set, similarly, we consider 10 parameters uniformly chosen at random from the range $\mu \in [0, 1]^p$ and for values of $\alpha \in [0.1, 0.9]$. Fig. 4 shows the distribution of relative errors for Case A and Case B. In Table 2, we display the computational time for the offline and online stages. As can be seen, for both cases, the online computational time outperforms the

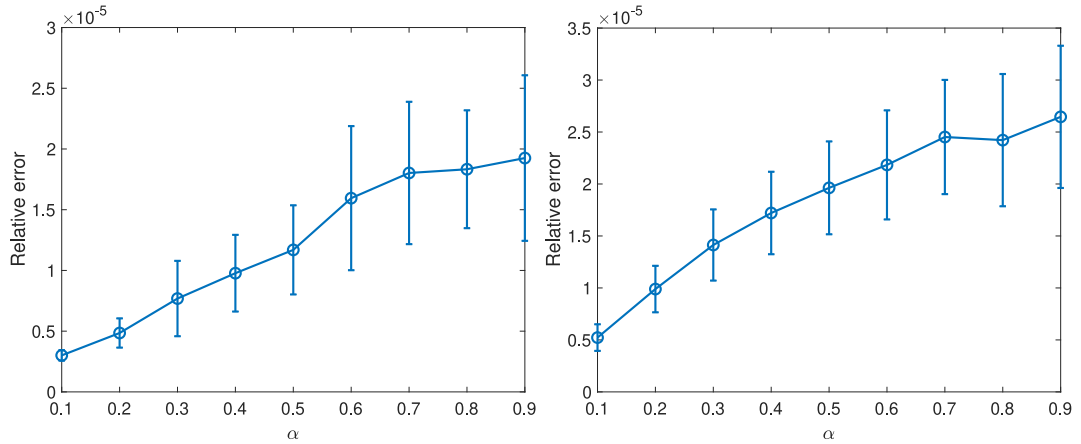


Fig. 4. Relative error in the test set (left) Case A, (right) Case B. The mean of the error over 10 random samples is plotted for different values of α , with the error bar denoting one standard deviation.

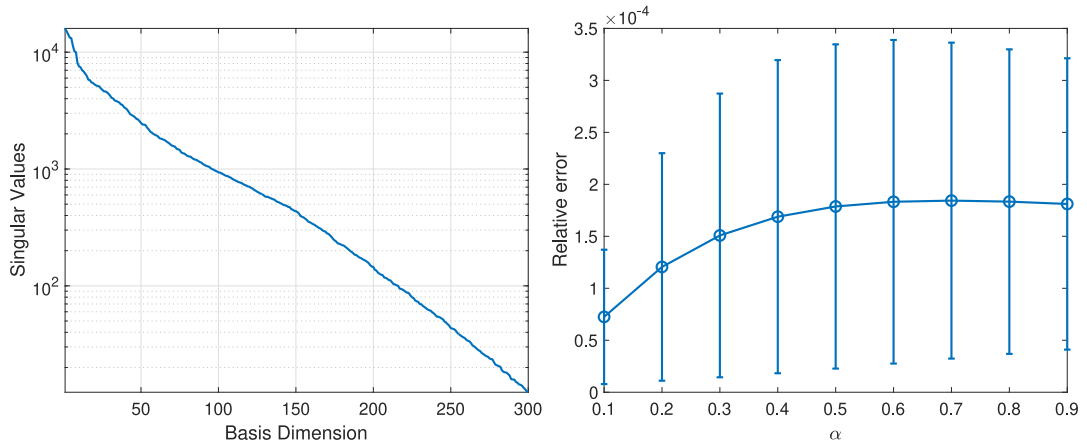


Fig. 5. (left) Singular values of the snapshot matrix $\hat{S}$, (right) the relative error in the test set. The mean of the error over 10 random samples is plotted for different values of α , with the error bar denoting one standard deviation.

Table 2

Wall-clock time (in seconds) for various components of the cookies test problem.

	Snapshot	Compression	Direct	MPGMRES-Sh	Online
Case A	143.72	0.67	71.83	1.48	0.006
Case B	236.49	20.51	71.71	1.93	0.37

Direct and the MPGMRES-Sh algorithms. However, the offline and online times of Case B are higher since the reduced basis has a higher rank K . Note that when the reduced basis rank K is high, the ROM may offer computational benefits over the conventional approaches only if N_h is sufficiently high.

4.3. Anisotropic diffusion

In this example, we consider the fractional PDE (1) with the anisotropic diffusion model $\mathcal{A}(\mathbf{x}; \boldsymbol{\mu}) = -\nabla \cdot (\boldsymbol{\Theta}(\boldsymbol{\mu}) \nabla(\cdot))$, where $\boldsymbol{\Theta}$ takes the form

$$\boldsymbol{\Theta}(\boldsymbol{\mu}) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} D_1 & \\ & D_2 \end{bmatrix} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}^T.$$

The fractional PDE is posed on the domain $\Omega = (0, 1)^2$ and is augmented with zero Dirichlet boundary conditions. The parameters $\boldsymbol{\mu} = (D_1, D_2, \theta)$ include the two diffusion coefficients and the angle θ .

We can write the weak form as $a(u, v; \rho) = \sum_{i=1}^3 f_i(\mu) a_i(u, v)$, where $a_1(u, v) = \int_{\Omega} \mathbf{e}_1 \mathbf{e}_1^T \nabla u \cdot \nabla v d\mathbf{x}$, $a_2(u, v) = \int_{\Omega} (\mathbf{e}_1 \mathbf{e}_2^T + \mathbf{e}_2 \mathbf{e}_1^T) \nabla u \cdot \nabla v d\mathbf{x}$, and $a_3(u, v) = \int_{\Omega} \mathbf{e}_2 \mathbf{e}_2^T \nabla u \cdot \nabla v d\mathbf{x}$. Here, $\mathbf{e}_1, \mathbf{e}_2$ are the two columns from the 2×2 identity matrix. Furthermore, $f_1^a(\mu) = (\cos^2 \theta) D_1 + (\sin^2 \theta) D_2$, $f_2^a(\mu) = \sin \theta \cos \theta (D_2 - D_1)$, and $f_3^a(\mu) = (\cos^2 \theta) D_2 + (\sin^2 \theta) D_1$. We take $b(\mathbf{x}; \mu) = 1$ so that $f_1^g(\mu) = 1$ and $g_1(v) = \int_{\Omega} v d\mathbf{x}$.

For the training set, we generate 100 samples generated uniformly at random from the ranges $(D_1, D_2, \theta) \in [0.5, 4.5]^2 \times [0, \pi/2]$. The basis size K is taken as 300. The test set is drawn randomly similarly to the training set and the maximum error over 90 samples is $\sim 2.5 \times 10^{-4}$. The distribution of the error for different values of the exponent α is given in the right panel of Fig. 5, whereas the left panel gives the decay of the singular values.

5. Conclusions

This paper proposes ROMs for parametric fractional elliptic PDEs. The recommended approach combines efficient solvers for shifted linear systems and randomization to dramatically reduce the computational and storage costs over a naïve version of the algorithm. Numerical results demonstrate the computational benefits and viability of the ROMs. Future work involves adaptive construction of the ROM to guarantee a certain user-specified error tolerance and the development of error indicators and metrics necessary to develop stopping criteria for the adaptive algorithms. Another line of future inquiry involves adaptively estimating the size of the reduced basis.

CRedit authorship contribution statement

Harbir Antil: Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing. **Arvind K. Saibaba:** Conceptualization, Methodology, Software, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article

Acknowledgments

AKS was supported in part by the National Science Foundation through the grant DMS-1845406. HA was partially supported by the National Science Foundation under grant DMS-2110263 and the Air Force Office of Scientific Research under Award NO: FA9550-22-1-0248. AKS would like to acknowledge Hussam Al Daas for helpful discussions.

References

- [1] H. Antil, S. Bartels, Spectral approximation of fractional PDEs in image processing and phase field modeling, *Comput. Methods Appl. Math.* 17 (4) (2017) 661–678.
- [2] C.J. Weiss, B.G. van Bloemen Waanders, H. Antil, Fractional operators applied to geophysical electromagnetics, *Geophys. J. Int.* 220 (2) (2020) 1242–1259.
- [3] H. Antil, Z.W. Di, R. Khatri, Bilevel optimization, deep learning and fractional Laplacian regularization with applications in tomography, *Inverse Problems* 36 (6) (2020) 064001.
- [4] H. Antil, S. Bartels, A. Schikorra, Approximation of fractional harmonic maps, *IMA J. Numer. Anal.* 43 (3) (2022) 1291–1323.
- [5] N.G. Trillos, D. Sanz-Alonso, The Bayesian formulation and well-posedness of fractional elliptic inverse problems, *Inverse Problems* 33 (6) (2017) 065006.
- [6] H. Antil, D.P. Kouri, J. Pfefferer, Risk-averse control of fractional diffusion with uncertain exponent, *SIAM J. Control Optim.* 59 (2) (2021) 1161–1187.
- [7] H. Antil, E. Otárola, A. Salgado, Optimization with respect to order in a fractional diffusion model: Analysis, approximation and algorithmic aspects, *J. Sci. Comput.* 77 (1) (2018) 204–224.
- [8] J. Sprekels, E. Valdinoci, A new type of identification problems: optimizing the fractional order in a nonlocal evolution equation, *SIAM J. Control Optim.* 55 (1) (2017) 70–93.
- [9] H. Antil, Y. Chen, A. Narayan, Reduced basis methods for fractional Laplace equations via extension, *SIAM J. Sci. Comput.* 41 (6) (2019) A3552–A3575.
- [10] L. Caffarelli, L. Silvestre, An extension problem related to the fractional Laplacian, *Comm. Partial Differential Equations* 32 (7–9) (2007) 1245–1260.
- [11] P. Stinga, J. Torrea, Extension problem and Harnack's inequality for some fractional operators, *Comm. Partial Differential Equations* 35 (11) (2010) 2092–2122.
- [12] T. Danczul, J. Schöberl, A reduced basis method for fractional diffusion operators II, *J. Numer. Math.* 29 (4) (2021) 269–287.
- [13] T. Danczul, J. Schöberl, A reduced basis method for fractional diffusion operators I, *Numer. Math.* 151 (2) (2022) 369–404.
- [14] Q. Guan, M. Gunzburger, C.G. Webster, G. Zhang, Reduced basis methods for nonlocal diffusion problems with random input data, *Comput. Methods Appl. Mech. Engrg.* 317 (2017) 746–770.
- [15] H. Fu, H. Wang, Z. Wang, POD/DEIM reduced-order modeling of time-fractional partial differential equations with applications in parameter identification, *J. Sci. Comput.* 74 (1) (2018) 220–243.
- [16] M. D'Elia, Q. Du, C. Glusa, M. Gunzburger, X. Tian, Z. Zhou, Numerical methods for nonlocal and fractional models, *Acta Numer.* 29 (2020) 1–124.
- [17] H. Dinh, H. Antil, Y. Chen, E. Cherkaev, A. Narayan, Model reduction for fractional elliptic problems using kato's formula, *Math. Control Relat. Fields* 12 (1) (2022) 115–146.
- [18] A. Bonito, D. Guignard, A.R. Zhang, Reduced basis approximations of the solutions to spectral fractional diffusion problems, *J. Numer. Math.* 28 (3) (2020) 147–160.

- [19] H. Antil, A.K. Saibaba, Efficient algorithms for Bayesian inverse problems with Whittle–Matérn priors, 2022, arXiv preprint [arXiv:2205.04417](https://arxiv.org/abs/2205.04417).
- [20] T. Bakhos, P.K. Kitanidis, S. Ladenheim, A.K. Saibaba, D.B. Szyld, Multipreconditioned GMRES for shifted systems, *SIAM J. Sci. Comput.* 39 (5) (2017) S222–S247.
- [21] A.K. Saibaba, Randomized discrete empirical interpolation method for nonlinear model reduction, *SIAM J. Sci. Comput.* 42 (3) (2020) A1582–A1608.
- [22] J.S. Hesthaven, G. Rozza, B. Stamm, Certified Reduced Basis Methods for Parametrized Partial Differential Equations, in: SpringerBriefs in Mathematics, Springer, Cham; BCAM Basque Center for Applied Mathematics, Bilbao, 2016, p. xiii+131, BCAM SpringerBriefs.
- [23] A. Quarteroni, A. Manzoni, F. Negri, Reduced Basis Methods for Partial Differential Equations, in: Unitext, vol. 92, Springer, Cham, 2016, p. xi+296, An introduction, *La Matematica per il 3+2*.
- [24] T. Kato, Note on fractional powers of linear operators, *Proc. Japan Acad.* 36 (3) (1960) 94–96.
- [25] A. Bonito, J. Pasciak, Numerical approximation of fractional powers of elliptic operators, *Math. Comp.* 84 (295) (2015) 2083–2110.
- [26] A. Pinkus, *N-Widths in Approximation Theory*, in: *Ergebnisse der Mathematik und ihrer Grenzgebiete (3) [Results in Mathematics and Related Areas (3)]*, vol. 7, Springer-Verlag, Berlin, 1985, p. x+291.
- [27] P. Binev, A. Cohen, W. Dahmen, R. DeVore, G. Petrova, P. Wojtaszczyk, Convergence rates for greedy algorithms in reduced basis methods, *SIAM J. Math. Anal.* 43 (3) (2011) 1457–1472.
- [28] J.A. Tropp, A. Yurtsever, M. Udell, V. Cevher, Practical sketching algorithms for low-rank matrix approximation, *SIAM J. Matrix Anal. Appl.* 38 (4) (2017) 1454–1485.
- [29] W. Yu, Y. Gu, Y. Li, Efficient randomized algorithms for the fixed-precision low-rank matrix approximation, *SIAM J. Matrix Anal. Appl.* 39 (3) (2018) 1339–1359.
- [30] P.-G. Martinsson, S. Voronin, A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices, *SIAM J. Sci. Comput.* 38 (5) (2016) S485–S507.
- [31] H.C. Elman, V. Forstall, Preconditioning techniques for reduced basis methods for parameterized elliptic partial differential equations, *SIAM J. Sci. Comput.* 37 (5) (2015) S177–S194.
- [32] V. Simoncini, Computational methods for linear matrix equations, *SIAM Rev.* 58 (3) (2016) 377–441.
- [33] D. Kressner, C. Tobler, Krylov subspace methods for linear systems with tensor product structure, *SIAM J. Matrix Anal. Appl.* 31 (4) (2010) 1688–1714.