

# **Iterative Low-rank Approximation Solvers for the Extension Method for Fractional Diffusion**

**I. Georgieva, S. Harizanov, C. Hofreither**

**RICAM-Report 2019-14**

# Iterative Low-rank Approximation Solvers for the Extension Method for Fractional Diffusion

Irina Georgieva

*Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,  
Acad. G. Bonchev Str., Bl. 8, 1113, Sofia, Bulgaria, irina@math.bas.bg*

Stanislav Harizanov

*Institute of Mathematics and Informatics, Bulgarian Academy of Sciences,  
Acad. G. Bonchev Str., Bl. 8, 1113, Sofia, Bulgaria, sharizanov@parallel.bas.bg*

Clemens Hofreither

*Johann Radon Institute for Computational and Applied Mathematics (RICAM), Austrian Academy of Sciences,  
Altenberger Str. 69, 4040 Linz, Austria, clemens.hofreither@ricam.oeaw.ac.at*

---

## Abstract

We consider the numerical method for fractional diffusion problems which is based on an extension to a mixed boundary value problem for a local operator in a higher dimensional space. We observe that, when this problem is discretized using tensor product spaces as is commonly done, the solution can be very well approximated by low-rank tensors. This motivates us to apply iterative low-rank approximation algorithms in order to efficiently solve this extended problem. In particular, we employ a recently proposed greedy Tucker approximation method as well as a more classical greedy rank one update method. Throughout, all objects of interest are kept in suitable low-rank approximations, which dramatically reduces the required amount of memory compared to the full formulation of the extended problem.

Our approach can be used for general, non-structured space discretizations. If the space discretization itself has tensor product structure, we can further decompose the problem in order to deal with even lower-dimensional objects. We also note that the approach can be directly applied to higher-order discretizations both in space and the extended variable.

In several numerical examples, we demonstrate the convergence behaviour of the proposed methods. In particular, the Tucker approximation approach requires only a few iterations in order to reach the discretization error in all tested settings.

*Keywords:* fractional diffusion, fractional Laplace, numerical methods, low-rank tensor approximation, Tucker tensors

---

## 1. Introduction

In recent years fractional diffusion models rapidly increased their popularity [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]. They are found to accurately describe various non-local physical processes, whose better understanding is important for different research fields.

On the other hand, the theoretical setup and the efficient solving of such application-driven problems, involving non-local operators is rather complicated. There are at least two non-equivalent definitions for the fractional diffusion operator. The first one is based on Riesz potential, while the second one involves the spectral decomposition of the corresponding non-fractional elliptic operator. The presence of highly singular kernels and the necessity to deal with an unbounded region of integration make the former approach numerically expensive even in low spatial dimensions. The density of the stiffness matrix of the finite element

discretization of the latter approach and the need of fine computational meshes, due to low FEM accuracy for spectral fractional diffusion problems, create serious numerical problems, when directly attacking it. There are four general alternatives, based on transformation of the original spectral problem to some auxiliary local problems, that allow for solving real life applications of fractional diffusion models in computational domains with general geometry. In the first one the original problem is extended to a mixed boundary value problem in the semi-infinite cylinder [13, 14]. In the second one the original problem is transformed to a pseudo-parabolic one [15]. For the third one an exponentially convergent quadrature scheme for the Dunford-Taylor integral representation is applied for approximating the fractional diffusion operator [16]. Different types of quadrature rules, such as Gauss-Jacobi, have also been considered in the literature [17, 18, 19]. Here, the fractional power of the elliptic operator is approximated using Padé techniques. For the fourth one, best uniform rational approximation techniques are used in order to approximate the action of the dense fractional FEM matrix by solving several linear systems with sparse matrices, that are positive diagonal shifts of the non-fractional FEM matrix [20].

Some further developments of these approaches can be found, e.g., in [21, 22, 23, 24, 25], as well as in [26, 27], where first efficient parallel algorithms are presented.

In this work, we use the first approach mentioned above, the extension method to a mixed boundary-value problem, as a starting point. The characterization of an arbitrary fractional power of the Laplacian as a Dirichlet-to-Neumann mapping of an extended problem in a semi-infinite cylinder in a space of one higher dimension was rigorously studied by Caffarelli and Silvestre [28] for problems posed on the full space  $\mathbb{R}^d$ , and in the later publications [29, 30, 31] for bounded domains  $\Omega \subset \mathbb{R}^d$ . A discretization method based on this idea was proposed in [13], and an efficient multilevel solver for the resulting  $d + 1$ -dimensional problem was described in [14]. A hybrid FEM-spectral method based on the same idea was recently proposed in [32].

We observe that the discrete solution of the  $d + 1$ -dimensional elliptic partial differential equation used in this extension method admits a highly efficient approximation by low-rank matrices. This observation motivates the reformulation of this extended problem as a linear system on tensors where both the operator on the left-hand side and the given right-hand side have low rank in a suitable sense, and also the solution tensor can be well approximated with low rank. We then apply iterative solution methods for such tensor linear systems to this problem which operate directly on such low-rank representations. In particular, we consider a greedy rank one update (GROU) algorithm (see also [33]) and a greedy Tucker approximation (GTA) algorithm recently proposed in [34]. In both cases, we exploit the symmetric positive definiteness of the operator in order to find best rank one approximations to the solution. In the GROU method, these rank one approximations are simply added to the previous iterate. In the GTA method, we use a more sophisticated approach where we build a tensor product subspace which contains all rank one approximations and then perform Galerkin projection of the solution into this small space.

Both algorithms are very memory efficient, never requiring the construction of the full stiffness matrix for the extended problem, but only certain lower-dimensional objects. Also, the solution field is not stored as a full  $R^{d+1}$ -dimensional object, but rather as a low-rank tensor represented either in the canonical format in the GROU method or as a Tucker tensor in the GTA method. Therefore, these approaches allow us to solve problems which would not be tractable with classical solvers due to memory limitations.

In several numerical examples, we demonstrate that the GTA method reaches the discretization error after only few iterations in all tested problems. The simpler GROU method has similar good performance in some examples, but performs significantly worse in 2D problems whose extended formulations are fully decomposed into 3D tensor equations. This behavior matches earlier numerical experiments for the Poisson equation [34].

A further benefit of these tensor-based iterative methods is that they are essentially fully algebraic and immediately generalize, e.g., to higher-order discretizations both in the spatial and the extended variable as well as to different discretization schemes such as Isogeometric Analysis, as we demonstrate in our tests.

There have been previous efforts to apply low-rank methods to the solution of fractional partial differential equations. In particular, we point out the work [35], which treats problems in tensor product domains and uses a tensor train (TT) format in order to approximate the solution. An optimal control problem with fractional elliptic constraint has been treated using tensor methods in [36], again relying on tensor product spaces. We also point out the earlier work [37] which uses hierarchical Kronecker tensor product

approximation for nonlocal problems in high-dimensional tensor product spaces. An advantage of our method over all these previous approaches is that it works for arbitrary space discretizations, whereas the previous approaches require the problem to be posed in a tensor product space.

The remainder of the paper is outlined below. First we introduce the fractional diffusion problem as well as the extension method for its solution in Section 2. In Section 3, we introduce some fundamental notions of tensors and tensor approximation and then describe the two main algorithms which we use to solve low-rank tensor linear systems. In Section 4, we describe how to apply these algorithms to our discretization of the extension method and discuss some practical aspects. Finally, we present several numerical examples in Section 5.

## 2. The extension method for fractional diffusion

### 2.1. The fractional diffusion problem

Given some open and bounded domain  $\Omega \subset \mathbb{R}^d$ ,  $s \in (0, 1)$ , and a suitable right-hand side  $f$  defined on  $\Omega$ , we consider the fractional diffusion equation

$$\begin{aligned} \mathcal{L}^s u &= f && \text{in } \Omega, \\ u &= 0 && \text{on } \Gamma = \partial\Omega, \end{aligned} \tag{1}$$

where  $\mathcal{L}u = -\operatorname{div}(A\nabla u)$  is a self-adjoint, elliptic diffusion operator with  $A(x) \in \mathbb{R}^{d \times d}$  symmetric and uniformly positive definite. It is a nontrivial question how to define the fractional operator power  $\mathcal{L}^s$ , and we refer the reader to [38, 39] and the references therein for various approaches to the definition of fractional derivatives and fractional operator powers.

Herein, we consider only bounded domains with homogeneous Dirichlet boundary conditions, which allows a relatively simple spectral definition of the fractional operators. Under the above assumptions,  $\mathcal{L}$  admits a system of eigenfunctions  $u_j$  with corresponding eigenvalues  $\lambda_j > 0$  such that

$$\mathcal{L}u_j = \lambda_j u_j \quad \forall j = 1, 2, \dots$$

and  $(u_i, u_j) = \delta_{ij}$ , where  $(\cdot, \cdot)$  denotes the  $L_2$ -inner product in  $\Omega$ . A fractional power of  $\mathcal{L}$  can then be defined as

$$\mathcal{L}^s u = \sum_{j=1}^{\infty} \lambda_j^s (u, u_j) u_j. \tag{2}$$

As outlined in the introduction, various methods have been proposed in order to numerically solve this problem. In the following, we recall a method based on extension of the non-local fractional diffusion problem to a Neumann-to-Dirichlet mapping for a local, but higher-dimensional problem.

### 2.2. The extension method

We summarize here the approach of Nochetto, Otárola and Salgado [13]. We let  $\alpha = 1 - 2s \in (-1, 1)$  and introduce an extended boundary value problem for a function  $U(x, y)$ ,  $x \in \Omega$ ,  $y \in (0, \infty)$  given by

$$\begin{aligned} -\operatorname{div}(y^\alpha \tilde{A} \nabla U) &= 0 && \text{in } \Omega \times (0, \infty), \\ \lim_{y \rightarrow \infty} U(x, y) &= 0 && \forall x \in \Omega, \\ U(x, y) &= 0 && \forall x \in \partial\Omega, y \in (0, \infty), \\ -(\lim_{y \rightarrow 0} y^\alpha \partial_y U(x, y)) &= d_s f(x) && \forall x \in \Omega, \end{aligned}$$

where  $\tilde{A}(x, y) = \operatorname{diag}(A(x), 1) \in \mathbb{R}^{(d+1) \times (d+1)}$  and

$$d_s = 2^{1-2s} \frac{\Gamma(1-s)}{\Gamma(s)}$$

is a dimension-independent constant involving the gamma function  $\Gamma$ . The solution  $u$  of problem (1) is then given by the Dirichlet trace

$$u(x) = U(x, 0) \quad \forall x \in \Omega.$$

Since  $U$  decays exponentially with  $y$ , it is justified to truncate the above problem to a finite cylinder  $\Omega \times (0, Y)$  with some  $Y > 0$  and enforce homogeneous Dirichlet boundary conditions on  $\Omega \times \{Y\}$  instead of the decay condition; cf. [13] for the corresponding analysis. With suitable test functions  $V(x, y)$  which vanish at the Dirichlet boundaries  $\partial\Omega \times (0, Y)$  and  $\Omega \times \{Y\}$ , the variational formulation reads

$$a(U, V) := \int_{\Omega \times (0, Y)} y^\alpha \tilde{A} \nabla U \cdot \nabla V \, d(x, y) = d_s \int_{\Omega} f(x) V(x, 0) \, dx. \quad (3)$$

In order to discretize this problem, we introduce a finite-dimensional space  $V_h \subset H_0^1(\Omega)$  which satisfies the homogeneous Dirichlet boundary conditions and which is spanned by a basis  $(\varphi_j)_{j=1}^n$ . Typically, this space will be a standard finite element space, say, piecewise linear functions over a triangulation composed of simplices. Furthermore, we introduce a discrete space for the extended direction  $W_h = \text{span}\{\psi_j : j = 1, \dots, m\} \subset C(0, Y)$  with the Dirichlet boundary condition  $\psi_j(Y) = 0$ . The problem (3) is then discretized using tensor product basis functions

$$\varphi_i(x) \psi_j(y) \quad \forall i = 1, \dots, n, j = 1, \dots, m, x \in \Omega, y \in (0, Y).$$

In order to compute the stiffness matrix, we choose two arbitrary tensor product basis functions  $U(x, y) = \varphi_{i_1}(x) \psi_{i_2}(y)$  and  $V(x, y) = \varphi_{j_1}(x) \psi_{j_2}(y)$  and find that

$$\begin{aligned} a(U, V) &= \int_{\Omega \times (0, Y)} y^\alpha \begin{pmatrix} A \nabla_x \varphi_{i_1}(x) \psi_{i_2}(y) \\ (\varphi_{i_1}(x) \psi'_{i_2}(y)) \end{pmatrix} \cdot \begin{pmatrix} \nabla_x \varphi_{j_1}(x) \psi_{j_2}(y) \\ (\varphi_{j_1}(x) \psi'_{j_2}(y)) \end{pmatrix} d(x, y) \\ &= \left( \int_0^Y y^\alpha \psi_{i_2} \psi_{j_2} dy \right) \left( \int_{\Omega} A \nabla_x \varphi_{i_1} \cdot \nabla_x \varphi_{j_1} dx \right) + \left( \int_0^Y y^\alpha \psi'_{i_2} \psi'_{j_2} dy \right) \left( \int_{\Omega} \varphi_{i_1} \varphi_{j_1} dx \right). \end{aligned}$$

Thus the stiffness matrix for the extended problem (3) is given as the sum of the two Kronecker products

$$A = M_y^{(\alpha)} \otimes K + K_y^{(\alpha)} \otimes M, \quad (4)$$

where  $K, M \in \mathbb{R}^{n \times n}$  are the standard symmetric and positive definite stiffness and mass matrices

$$K_{ij} = (A \nabla \phi_j, \nabla \phi_i), \quad M_{ij} = (\phi_j, \phi_i) \quad \forall i, j = 1, \dots, n,$$

and  $M_y^{(\alpha)} \in \mathbb{R}^{m \times m}$  and  $K_y^{(\alpha)} \in \mathbb{R}^{m \times m}$  are the weighted mass and stiffness matrices

$$[M_y^{(\alpha)}]_{ij} = \int_0^Y y^\alpha \psi_j(y) \psi_i(y) dy, \quad [K_y^{(\alpha)}]_{ij} = \int_0^Y y^\alpha \psi'_j(y) \psi'_i(y) dy.$$

Formula (4) provides a simple way of computing the stiffness matrix for the extended problem. We also point out that the formula holds for arbitrary Galerkin discretizations of the spatial problem and as such can also be used for higher-order finite element methods or Isogeometric Analysis discretizations [40].

For the discrete space  $W_h$ , we chose linear splines over a graded mesh as suggested in [13]. That is, we choose a mesh grading parameter  $\gamma \geq 1$  and construct the nodes of our mesh as

$$y_j = Y \left( \frac{j}{m} \right)^\gamma, \quad j = 0, \dots, m. \quad (5)$$

Over this mesh, we build a standard piecewise linear spline space with a basis of hat functions  $(\psi_j)_{j=1}^m$ . In [13], it is suggested to use  $\gamma > 3/(1-s)$ .

### 3. Low-rank tensor approximation methods for the solution of linear systems

#### 3.1. Tensors, rank and low-rank approximation

We first introduce some general notions about tensors and low-rank approximation. We denote a tensor of order  $d$  and dimensions  $N_1, \dots, N_d$  by  $B \in \mathbb{R}^{\mathcal{I}} := \mathbb{R}^{N_1 \times \dots \times N_d}$  with  $\mathcal{I} = \{1, \dots, N_1\} \times \dots \times \{1, \dots, N_d\}$ .

**Definition 1.** The  $d$ -dimensional *tensor product* of vectors  $x^j \in \mathbb{R}^{N_j}$ ,  $j = 1, \dots, d$  is the tensor  $x^1 \otimes \dots \otimes x^d \in \mathbb{R}^{\mathcal{I}}$  with the entries

$$[x^1 \otimes \dots \otimes x^d]_{i_1, \dots, i_d} = x_{i_1}^1 \dots x_{i_d}^d.$$

**Definition 2** ([41]). The *rank* of  $B$  is the smallest  $R \in \mathbb{N}_0$  such that there exist vectors  $x_j^r \in \mathbb{R}^{N_j}$ ,  $j = 1, \dots, d$ ,  $r = 1, \dots, R$  with

$$B = \sum_{r=1}^R x_1^r \otimes \dots \otimes x_d^r.$$

Often, we are interested in low-rank approximations to tensors of higher rank. As a norm to measure the distance between tensors, we will always use the Frobenius norm

$$\|B\|^2 = \sum_{(i_1, \dots, i_d) \in \mathcal{I}} B_{i_1, \dots, i_d}^2 = \langle B, B \rangle,$$

where  $\langle \cdot, \cdot \rangle$  denotes the Frobenius inner product.

If  $X^*$  minimizes the error  $\|B - X\|$  among all tensors  $X \in \mathbb{R}^{\mathcal{I}}$  of rank at most  $R \in \mathbb{N}_0$ , we call  $X^*$  a *best rank  $R$  approximation* to  $B$ . Note that, for  $d > 2$ , such a best approximation may not be unique or not even exist [42].

A popular method for computing best rank  $R$  approximations is the Alternating Least Squares (ALS) algorithm. It is based on sequentially fixing all but the  $k$ -th factor and minimizing the error of the resulting approximation by optimizing only the  $k$ -th factor, which means solving a linear least squares problem. For the sake of completeness, we give a variant of this algorithm for the best rank one approximation problem in Algorithm 1. Possible stopping criteria for Algorithm 1 include reaching a fixed number of iterations, the reduction of the target functional to a desired level, or stagnation of the changes in the factors  $x_j$ .

---

#### Algorithm 1 Alternating least squares for rank 1 approximation (ALS)

---

```

function ALS( $B \in \mathbb{R}^{\mathcal{I}}$ )
  choose nonzero starting vectors  $x_j \in \mathbb{R}^{N_j}$ ,  $j = 1, \dots, d$ 
  while not converged do
    for  $k = 1, \dots, d$  do
      solve the linear least squares problem

$$x_k \leftarrow \arg \min_{y \in \mathbb{R}^{N_k}} \|B - x_1 \otimes \dots \otimes x_{k-1} \otimes y \otimes x_{k+1} \otimes \dots \otimes x_d\|^2$$

    end for
  end while
  return  $(x_1, \dots, x_d)$ 
end function

```

---

For details on how to solve the linear least squares problems which occur within Algorithm 1, we refer to the literature, e.g., [43].

### 3.2. Multiway tensor product and Tucker tensor format

Above we noted that, for  $d > 2$ , there are certain difficulties associated with the best rank  $R$  approximation. In order to circumvent these problems, different low-rank tensor representations have been developed. In the following, we introduce the so-called *Tucker* or *tensor subspace representation*.

**Definition 3.** Given  $d$  matrices  $U_j$  and a tensor  $X$  of order  $d$ ,

$$U_j \in \mathbb{R}^{N_j \times r_j}, \quad j = 1, \dots, d, \quad X \in \mathbb{R}^{r_1 \times \dots \times r_d},$$

where  $N_j, r_j \in \mathbb{N}$ , the *multiway tensor product* is the tensor

$$T = (U_1, \dots, U_d) \cdot X \in \mathbb{R}^{N_1 \times \dots \times N_d} \quad (6)$$

with the entries

$$T_{i_1, \dots, i_d} = \sum_{\alpha_1=1}^{r_1} \dots \sum_{\alpha_d=1}^{r_d} [U_1]_{i_1, \alpha_1} \dots [U_d]_{i_d, \alpha_d} X_{\alpha_1, \dots, \alpha_d}, \quad (i_1, \dots, i_d) \in \mathcal{I}.$$

The following two two properties of this product follow from the definition.

- For arbitrary matrices  $Q_j, U_j$  of compatible sizes, we have

$$(Q_1, \dots, Q_d) \cdot ((U_1, \dots, U_d) \cdot X) = (Q_1 U_1, \dots, Q_d U_d) \cdot X. \quad (7)$$

- Let  $\mathcal{A} = A_1 \otimes \dots \otimes A_d : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}^{\mathcal{I}}$  be a *rank one tensor linear operator*, where  $A_j : \mathbb{R}^{N_j} \rightarrow \mathbb{R}^{N_j}$  are matrices. By this we mean that the application of  $\mathcal{A}$  to a rank one tensor  $x_1 \otimes \dots \otimes x_d$  is given by

$$\mathcal{A}(x_1 \otimes \dots \otimes x_d) = (A_1 x_1) \otimes \dots \otimes (A_d x_d).$$

In practice, such an operator is often represented as the Kronecker product of the matrices  $A_j$ . The application of  $\mathcal{A}$  to a tensor  $B \in \mathbb{R}^{\mathcal{I}}$  is then given by

$$\mathcal{A}B = (A_1, \dots, A_d) \cdot B. \quad (8)$$

We say that a tensor  $T \in \mathbb{R}^{\mathcal{I}}$  is in *Tucker format* [44] or *tensor subspace representation* [45] if it has the form (6), and we call the tuple  $(r_1, \dots, r_d)$  its *Tucker rank* or *multilinear rank* assuming that it is the minimal tuple which allows such a representation.

A Tucker tensor can be represented on a computer by storing only the matrices  $U_j$  and the core tensor  $X$ . If  $N_j = N$  and  $r_j = r$ , this requires the storage of  $dNr + r^d$  floating point numbers, rather than  $N^d$  for the same tensor in full tensor format. The storage  $r^d$  required for the core tensor  $X$  still grows exponentially with the order  $d$ , which makes other representations such as the hierarchical Tucker format [46, 45] or the tensor train format [47] more desirable for high-dimensional problems. Nevertheless, the simplicity of the Tucker format makes it attractive in low-dimensional settings.

### 3.3. Rank one approximation of solutions of tensor linear systems

We now turn our attention to the approximation of a tensor  $Z \in \mathbb{R}^{\mathcal{I}}$  which is given as the solution of a tensor linear equation

$$\mathcal{A}Z = F,$$

where  $\mathcal{A} : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}^{\mathcal{I}}$  is symmetric positive definite linear operator on tensors and  $F \in \mathbb{R}^{\mathcal{I}}$ . We will always assume  $\mathcal{A}$  to have low Kronecker rank. By this we mean that there exists a rank  $R_{\mathcal{A}} \in \mathbb{N}$  and (possibly sparse) matrices  $A_j^\rho \in \mathbb{R}^{N_j \times N_j}$ ,  $j = 1, \dots, d$ ,  $\rho = 1, \dots, R_{\mathcal{A}}$ , such that for all  $x_j \in \mathbb{R}^{N_j}$ ,  $j = 1, \dots, d$ , we have

$$\mathcal{A}(x_1 \otimes \dots \otimes x_d) = \sum_{\rho=1}^{R_{\mathcal{A}}} (A_1^\rho x_1) \otimes \dots \otimes (A_d^\rho x_d). \quad (9)$$

The so-called “greedy Tucker approximation” (GTA) algorithm, first introduced in [34], for the solution of such low-rank tensor linear systems is an iterative method based on the following ideas:

1. the successive rank one approximation to the solution of the linear system with the current residual,
2. the greedy construction of a low-dimensional tensor subspace which is spanned by this sequence of rank one approximations,
3. the Galerkin projection of the unknown solution  $Z$  into this low-dimensional space.

First, we discuss how to obtain rank one approximations to the solution. The symmetric positive definite operator  $\mathcal{A}$  induces the energy norm  $\|X\|_{\mathcal{A}} := \sqrt{\langle \mathcal{A}X, X \rangle}$ . The minimization of the error in the energy norm

$$\|Z - X\|_{\mathcal{A}}^2 = \|Z\|_{\mathcal{A}}^2 + \|X\|_{\mathcal{A}}^2 - 2\langle \mathcal{A}Z, X \rangle = \|Z\|_{\mathcal{A}}^2 + \|X\|_{\mathcal{A}}^2 - 2\langle F, X \rangle$$

is thus equivalent to the minimization problem  $\|X\|_{\mathcal{A}}^2 - 2\langle F, X \rangle \rightarrow \min$ , which can be treated by an Alternating Least Squares method analogous to Algorithm 1 in order to obtain a best rank one approximation  $x_1 \otimes \dots \otimes x_d$  to the current error in the energy norm. The update rule for the vector  $x_k$  is obtained by taking the gradient of  $\|X\|_{\mathcal{A}}^2 - 2\langle F, X \rangle$  with respect to  $x_k$  and setting it to 0. The resulting algorithm is shown in Algorithm 2 below.

---

**Algorithm 2** Alternating least squares for s.p.d. linear systems (**ALS-LS-SPD**)

---

**function** ALS-LS-SPD( $\mathcal{A} : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}^{\mathcal{I}}, F \in \mathbb{R}^{\mathcal{I}}$ )  
 choose nonzero starting vectors  $x_j \in \mathbb{R}^{N_j}, j = 1, \dots, d$   
**while** not converged **do**  
   **for**  $k = 1, \dots, d$  **do**  
     update  $x_k$  by solving the linear equation

$$\sum_{\rho=1}^{R_{\mathcal{A}}} \left( \sum_{\substack{1 \leq j \leq d \\ j \neq k}} x_j^T A_j^{\rho} x_j \right) A_k^{\rho} x_k = (x_1^T, \dots, x_{k-1}^T, I_{N_k}, x_{k+1}^T, \dots, x_d^T) \cdot F$$

**end for**  
**end while**  
**return**  $(x_1, \dots, x_d)$   
**end function**

---

### 3.4. The algorithm GTA-LS-SPD

We now describe the algorithm **GTA-LS-SPD** for approximating solutions of tensor linear systems in Tucker format, first given in [34].

We start with an empty Tucker tensor  $T = 0$  and empty matrices (i.e., with zero columns)  $U_j, j = 1, \dots, d$ , which represent the empty tensor product subspace before the first iteration. In each iteration, we compute the rank one tensor  $x_1 \otimes \dots \otimes x_d$  which minimizes the residual  $\|F - \mathcal{A}(T + x_1 \otimes \dots \otimes x_d)\|$  by Algorithm 2 (Step 1). We then enrich the tensor product subspace such as to include this new rank one approximation by extending each matrix  $U_j$  with the vector  $x_j$  and orthonormalizing by a modified Gram-Schmidt procedure. We denote the result of this Step 2 by

$$\text{orth}(U_j, x_j),$$

which is a matrix with the same number of rows and one more column than  $U_j$  such that  $\text{span}(\text{orth}(U_j, x_j)) = \text{span}(U_j) + \text{span}\{x_j\}$  and  $\text{orth}(U_j, x_j)$  has orthonormal columns.

The final Step 3 is to approximately satisfy the linear equation  $\mathcal{A}Z = F$  in the tensor subspace spanned by the bases  $(U_1, \dots, U_d)$ . To this end, we introduce the linear operators

$$\begin{aligned} \mathcal{U} : \mathbb{R}^{r_1 \times \dots \times r_d} &\rightarrow \mathbb{R}^{N_1 \times \dots \times N_d}, & \mathcal{U}X &= (U_1, \dots, U_d) \cdot X, \\ \mathcal{U}^T : \mathbb{R}^{N_1 \times \dots \times N_d} &\rightarrow \mathbb{R}^{r_1 \times \dots \times r_d}, & \mathcal{U}^T Y &= (U_1^T, \dots, U_d^T) \cdot Y, \end{aligned}$$

and seek a coefficient tensor  $X \in \mathbb{R}^{r_1 \times \dots \times r_d}$  such that, in some sense,

$$\mathcal{A}(\mathcal{U}X) \approx F.$$

Since the operator  $\mathcal{A}$  stems from a Galerkin discretization of a partial differential equation, it seems natural to perform this approximation by another Galerkin projection of the variational interpretation

$$\langle \mathcal{A}Z, \psi \rangle = \langle F, \psi \rangle \quad \forall \psi \in \mathbb{R}^{\mathcal{I}},$$

to the tensor subspace spanned by  $\mathcal{U}$ . This results in the problem: find  $\varphi \in \text{range}(\mathcal{U}) \subset \mathbb{R}^{\mathcal{I}}$  such that

$$\langle \mathcal{A}\varphi, \psi \rangle = \langle F, \psi \rangle \quad \forall \psi \in \text{range}(\mathcal{U}), \quad (10)$$

or equivalently, finding  $X \in \mathbb{R}^{r_1 \times \dots \times r_d}$  such that

$$\langle \mathcal{A}\mathcal{U}X, \mathcal{U}Y \rangle = \langle F, \mathcal{U}Y \rangle \quad \forall Y \in \mathbb{R}^{r_1 \times \dots \times r_d}.$$

Thus, (10) is solved by  $\varphi = \mathcal{U}X$ , where  $X$  is the solution of the linear system

$$(\mathcal{U}^T \mathcal{A} \mathcal{U})X = \mathcal{U}^T F. \quad (11)$$

Since in our case  $\mathcal{A}$  stems from the Galerkin discretization of a coercive and bounded bilinear form, then, by Céa's lemma (see, e.g., [48]), the unique solution  $\varphi$  is a quasi-optimal approximation to  $Z$  in the space spanned by  $\mathcal{U}$ . The constant of quasi-optimality depends only on the underlying variational problem (namely, on the constants in the coercivity and boundedness estimates), but not on  $\mathcal{U}$ .

The algorithm **GTA-LS-SPD** which combines Steps 1–3 is given in Algorithm 3. We can terminate the algorithm either after a fixed number of iterations or once the initial residual has been reduced by a desired tolerance factor.

---

**Algorithm 3** Greedy Tucker approximation for linear systems (**GTA-LS-SPD**)

---

```

function GTA-LS-SPD( $\mathcal{A} : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}^{\mathcal{I}}, F \in \mathbb{R}^{\mathcal{I}}$ )
  let  $U_j = \{\}$  for  $j = 1, \dots, d, X = 0$ 
  for  $k = 1, 2, \dots, R$  do
     $T \leftarrow (U_1, \dots, U_d) \cdot X$ 
     $(x_1, \dots, x_d) = \text{ALS-LS-SPD}(\mathcal{A}, F - \mathcal{A}T)$ 
    for  $j = 1, \dots, d$  do
       $U_j \leftarrow \text{orth}(U_j, x_j)$ 
    end for
     $X \leftarrow (\mathcal{U}^T \mathcal{A} \mathcal{U})^{-1} \mathcal{U}^T F$ 
  end for
  return  $(U_1, \dots, U_d), X$ 
end function

```

---

If  $F$  is given in low-rank Tucker format, we also obtain the residual  $F - \mathcal{A}T$  in low-rank Tucker format by combining properties (8) and (7). In order to add Tucker tensors, we use the procedure described in detail in [45]. The trivial addition of Tucker tensors also increases the resulting Tucker rank additively, which may make the application of **ALS-LS-SPD** too expensive. Therefore, we apply HOSVD projection as described in [45] to reduce the Tucker rank of the residual before invoking **ALS-LS-SPD**.

In each iteration, we obtain  $X$  by solving the smaller ( $r_1 \cdots r_d$  unknowns) linear system (11). The matrix to be inverted here generally is dense and has size  $k^d \times k^d$  in the  $k$ -th iteration. For the problems we consider here, this linear system usually remains small enough so that it can be solved efficiently by a direct solver. If this is not the case, the solution of this inner problem may be sped up by means of Gauss-Seidel iteration as described in [34].

### 3.5. The Greedy Rank One Update algorithm for linear systems

For the sake of comparison, we give a simple greedy rank one update algorithm for symmetric positive definite systems, **GROU-LS-SPD**, Algorithm 4. Like **GTA-LS-SPD**, this method is based on successive best rank one approximations, but simply adds them up, resulting in an approximation in canonical form and does not consider optimization in tensor product subspaces.

---

#### Algorithm 4 Greedy rank one updates for linear systems (**GROU-LS-SPD**)

---

```

function GROU-LS-SPD( $\mathcal{A} : \mathbb{R}^{\mathcal{I}} \rightarrow \mathbb{R}^{\mathcal{I}}, F \in \mathbb{R}^{\mathcal{I}}$ )
  let  $X_0 = 0$ 
  for  $k = 1, 2, \dots, R$  do
     $(x_1, \dots, x_d) = \text{ALS-LS-SPD}(\mathcal{A}, F - \mathcal{A}X_{k-1})$ 
     $X_k = X_{k-1} + x_1 \otimes \dots \otimes x_d$ 
  end for
  return  $X_R$ 
end function

```

---

## 4. Application of tensor algorithms to the extension method

### 4.1. Formulation as a tensor linear system

The solution of the discretized extended problem (3) has a coefficient vector of length  $mn$  which may be interpreted as a matrix  $Z \in \mathbb{R}^{m \times n}$ . Our goal is to exploit the low-rank structure of this matrix in order to construct fast solvers for (3). Of course, this can only be successful if this solution matrix can be well approximated with a matrix of low rank. One way to ascertain this is by exploiting knowledge about the smoothness of the solution; see, e.g., [49]. For the present study, we limit ourselves to numerical evidence that the solution can be approximated with low rank. To this end, we compute the discrete solution of the extended problem (3) for a range of one-dimensional discretizations  $V_h$  and with the right-hand side  $f \equiv 1$ , reshape the result into a  $m \times n$  matrix and compute its singular values. The result is shown in Figure 1 for uniform space discretizations  $V_h$  with  $n = 100, 200$  and  $400$  degrees of freedom with the choice  $m = n/2$ . Only the first 40 of the  $n/2$  singular values are shown.

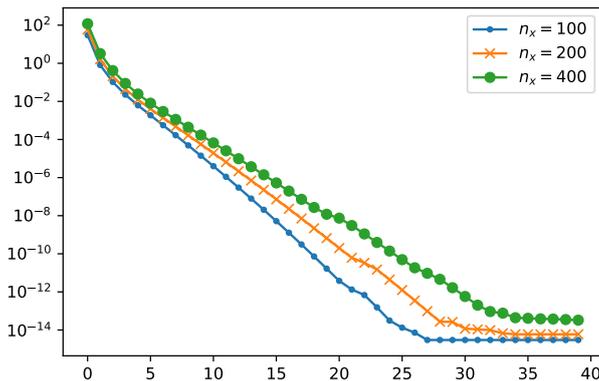


Figure 1: First singular values of solution matrix of the extended problem, where  $\Omega = (-1, 1)$ .

The figure shows that (1) the singular values decay very rapidly, meaning that the solution matrix can be very well approximated by a matrix of low rank, and (2) the decay rate depends only very mildly on the number of degrees of freedom, meaning that the required rank grows only slowly with  $n$ .

We represent the linear mapping represented by the stiffness matrix  $A$  from (4) as  $\mathcal{A} : \mathbb{R}^{m \times n} \rightarrow \mathbb{R}^{m \times n}$  by setting, for vectors  $v \in \mathbb{R}^m, u \in \mathbb{R}^n$ ,

$$\mathcal{A}(v \otimes u) = (M_y^{(\alpha)} v) \otimes (K u) + (K_y^{(\alpha)} v) \otimes (M u), \quad (12)$$

where  $v \otimes u = v u^T \in \mathbb{R}^{m \times n}$  is the rank one outer product of vectors. The definition of  $\mathcal{A}$  for matrices of higher rank follows by linearity. Thus, the operator  $\mathcal{A}$  falls into the class of admissible operators described in (9) with  $R_{\mathcal{A}} = 2$ . We also note that by linearity, property (8) describes how to apply a low-rank operator such the one in (12) to an arbitrary tensor  $B$ .

Then problem (3) is equivalent to finding the matrix solution  $Z \in \mathbb{R}^{m \times n}$  of the linear equation

$$\mathcal{A}Z = F, \quad (13)$$

where  $F \in \mathbb{R}^{m \times n}$  is the discretized right-hand side of (3) reshaped into a matrix. If we denote by  $\mathbf{f} \in \mathbb{R}^n$  the coefficient vector of the  $L_2$ -projection of  $f$  with respect to the basis of  $V_h$ , then

$$d_s \int_{\Omega} f(x) \phi_i(x) \psi_j(0) dx = d_s \psi_j(0) [M \mathbf{f}]_i \quad \forall i = 1, \dots, n, j = 1, \dots, m,$$

and thus

$$F = d_s \mathbf{c} \otimes M \mathbf{f}, \quad (14)$$

where the vector  $\mathbf{c} \in \mathbb{R}^m$  is given by  $c_j = \psi_j(0)$ . This makes it clear that  $F$  has rank one.

Thus, (13) is a linear equation where the right-hand side has rank one, the operator  $\mathcal{A}$  has Kronecker rank  $R_{\mathcal{A}} = 2$ , and the solution  $Z$  can be well approximated with low rank according to our numerical observations above. These facts allow us to apply the solvers described in Section 3 in order to exploit this low-rank structure and approximate the solution  $Z$  by means of low-rank matrices.

#### 4.2. Full and partial tensor decomposition

In the previous subsection, we described a general decomposition of the discretized extended problem which is valid for arbitrary space discretizations  $V_h$ . Irrespective of dimension, the solution tensor  $Z$  has order 2 in this decomposition, i.e., is a matrix.

A further decomposition is possible if the problem (1) is posed in a  $d$ -dimensional tensor product domain and has separable coefficient  $A$ . Then, by using a tensor product discretization of  $\Omega$  and applying the same arguments that lead to the derivation of (4), we can further decompose  $K$  and  $M$  into Kronecker products. For instance, it is classical that the discrete Laplace and mass operators in a three-dimensional tensor product space can be written as

$$K = K_1 \otimes M_2 \otimes M_3 + M_1 \otimes K_2 \otimes M_3 + M_1 \otimes M_2 \otimes K_3, \quad M = M_1 \otimes M_2 \otimes M_3,$$

where  $K_j, M_j$  are the stiffness and mass matrices associated with the  $j$ -th discrete univariate space. Inserting these formulae into (4), we obtain

$$\begin{aligned} A &= M_y^{(\alpha)} \otimes K_1 \otimes M_2 \otimes M_3 + M_y^{(\alpha)} \otimes M_1 \otimes K_2 \otimes M_3 \\ &\quad + M_y^{(\alpha)} \otimes M_1 \otimes M_2 \otimes K_3 + K_y^{(\alpha)} \otimes M_1 \otimes M_2 \otimes M_3. \end{aligned} \quad (15)$$

Thus, in this case the tensor equation (13) can be written as an equation for a tensor  $Z \in \mathbb{R}^{m \times n_1 \times \dots \times n_d}$  of order  $d + 1$ , and the operator  $\mathcal{A}$  has Kronecker rank  $R_{\mathcal{A}} = d + 1$ . We refer to this case as *full tensor decomposition*, and the original formulation in two dimensions as *partial tensor decomposition*.

### 4.3. Solution of the fractional diffusion problem by low-rank methods

We can now summarize our approach for solving (1) as follows. First, we choose the space discretization  $V_h$  and the discretization of the extended dimension  $W_h$ . Then we set up the operator  $\mathcal{A}$ , either using the partial decomposition (12) or, if  $V_h$  itself has tensor product structure, using the full tensor decomposition formula (15). Here we do not store  $\mathcal{A}$  itself, but only the individual matrices that make up its low-rank representation, which leads to a significant reduction in memory usage. The application of  $\mathcal{A}$  to a Tucker tensor can be efficiently implemented by combining properties (7) and (8). Also the right-hand side  $F$  has to be represented in low-rank Tucker format, which is trivial in the partial decomposition case due to (14) and can be achieved using any black-box method for Tucker approximation (e.g., the direct approximation variant of the GTA method described in [34]) in the full decomposition case. We then apply either **GTA-LS-SPD** or **GROU-LS-SPD** to the low-rank tensor linear system (13) in order to obtain a low-rank approximation to  $Z$ . Finally, we extract the degrees of freedom from  $Z$  which correspond to the boundary  $\Omega \times \{0\}$  of the extended problem described in Section 2.2, which are the coefficients of our approximate solution of (1) in  $V_h$ .

Throughout, all objects are kept in low-rank representations, realizing the reduction in storage described in Section 3.

The main computational effort in each iteration of our low-rank approximation algorithm is spent on

- the invocation of **ALS-LS-SPD** (for both GTA and GROU),
- the solution of the dense linear system (11) (GTA only).

The linear system (11) to be solved in the  $k$ -th iteration has  $k^2$  degrees of freedom in the partial tensor decomposition case and  $k^{d+1}$  degrees of freedom in the full tensor decomposition case. As long as it remains small enough (which is the case in our examples in the next section), it can simply be solved by standard dense linear algebra routines without dominating the overall runtime. Should this system grow too large, we can apply the Gauss-Seidel iteration process described in [34] in order to accelerate this step.

Thus, the main computational effort is spent in approximating the rank one approximations via ALS. From Algorithm 2, we see that each iteration requires the solution of certain linear systems whose matrices are linear combinations of the matrices  $A_k^\rho$ ,  $\rho = 1, \dots, R_A$ . Since the factors  $x_j^T A_j^\rho x_j$  are positive due to the positive definiteness of all involved matrices, also these linear combinations have positive coefficients and thus remain positive definite. In fact, in our application, they are linear combinations of mass and stiffness matrices and as such are spectrally equivalent to the stiffness matrices. A robust preconditioner for  $K$  can thus be used also to solve these shifted problems.

In the extended dimension, as well as in the case of full tensor decomposition, these matrices are even smaller and usually banded as they arise from discretizations of one-dimensional problems. Thus, these problems can be directly solved by standard routines for banded matrices in linear time, assuming the bandwidth is small.

Thus, for the partial tensor decomposition case, one iteration of **ALS-LS-SPD** requires roughly  $\mathcal{O}(n)$  operations (assuming that  $n > m$ ). For the full tensor decomposition case where  $n = n_1 \cdots n_d$ , it requires roughly  $\mathcal{O}((d+1)\bar{n})$  operations, where  $\bar{n} = \max\{m, n_1, \dots, n_d\}$ , which is usually faster than in the partial decomposition case. To our knowledge, there are no known theoretical bounds on the number of ALS iterations required to reach a prescribed accuracy.

## 5. Numerical results

To illustrate the proposed low-rank tensor approximation methods for the solution of the fractional diffusion problem, we present several numerical examples. The used implementation of the low-rank approximation algorithms in the Python programming language can be found on the third author's homepage<sup>1</sup>.

---

<sup>1</sup>see <https://orcid.org/0000-0002-6616-5081> for an up to date link

In all examples, we used the cutoff  $Y = 5$ , and for the extended direction  $W_h$  we use a graded mesh according to (5). Details for the space discretization  $V_h$  are mentioned in the respective examples.

In all settings, we consider the fractional parameters  $s = 0.25$  and  $s = 0.5$ . We could not solve the problems in the case  $s = 0.75$  as the stiffness matrix for the extended problem became extremely ill-conditioned, leading to a loss of accuracy. This effect was not specific to our low-rank approach, but also appeared when attempting to solve the extended problem with a sparse direct solver.

In all examples, we show the decay of the residual norm as a function of the number of iterations of the low-rank approximation algorithms, both for the GTA and the GROU algorithm. Wherever possible, we also show the  $\ell_2$ -errors of the solution vector compared to the true solution of the discrete problem. Finally, in all examples, we computed a reference solution of the fractional diffusion problem on a sufficiently fine grid and computed the  $L_2$ -errors in  $\Omega$  of the GTA approach with respect to this reference solution in order to estimate the true  $L_2$ -error. Of course, these errors are bounded from below by the approximation error of the true solution in the space  $V_h$ . For instance, for Example 1, it is known that the discretization error behaves like  $\mathcal{O}(h^{\min\{2, 2s+0.5\}})$  (see [16]). We will see that the GTA method requires usually only few iterations to reach this approximation error.

### 5.1. Example 1: 1D

We consider fractional power  $s$  of the negative second derivative operator equation on the interval  $\Omega = (-1, 1)$  with homogeneous Dirichlet boundary conditions,

$$\left(-\frac{d^2}{dx^2}\right)^s u(x) = 1 \quad \forall x \in (-1, 1), \quad u(-1) = u(1) = 0.$$

We consider three uniform space discretizations  $V_h$  with  $n = 100, 200$  and  $400$  degrees of freedom and as a basis for this space we use standard piecewise linear Lagrange hat functions. For the extended direction  $W_h$  we use linear splines over a graded mesh according to (5).

In Figure 2 are presented the residuals and  $\ell_2$ -errors for the full solution of the extended problem using Algorithm 3 and Algorithm 4 plotted over the iterations for the three space discretizations. In addition, we show the  $\ell_2$  errors for the best rank  $k$  approximation, which was computed from the full discrete solution via truncation of the singular value decomposition.

The  $L_2$  errors in  $\Omega$  for the GTA method for the three space discretizations and for  $s = 0.25$  and  $s = 0.5$  are shown in Figure 3, plotted over the number of iterations.

### 5.2. Example 2: 2D checkerboard problem

We consider fractional power  $s$  of the Poisson's equation on  $\Omega = (-1, 1)^2$  with Dirichlet boundary conditions,

$$(-\Delta)^s u(x) = f \quad \forall x \in \Omega, \quad u(x) = 0 \quad \forall x \in \partial\Omega$$

where the right-hand side is the checkerboard function

$$f(x, y) = \text{sign}(x) \text{sign}(y).$$

We consider bilinear functions over uniform quadrilateral meshes in the spatial domain, and as a basis for the space  $V_h$  we use tensor products of standard Lagrange hat functions. For the extended direction  $W_h$  we use linear splines over a graded mesh according to (5). Since this problem is posed in a tensor product domain, we use the full tensor decomposition as described in Section 4.2, that is, we determine the solution as a tensor  $Z$  of order 3.

The degrees of freedom for  $V_h$  are roughly  $n^2$ , and for  $W_h$  roughly  $n/2$ . The discretization of the extended problem thus has roughly  $n^3/2$  degrees of freedom.

On Figure 4 are presented the residuals and the  $\ell_2$ -errors for the full extended problem using Algorithm 3 and Algorithm 4 plotted over the iterations for the three space discretizations.

The  $L_2$  errors in  $\Omega$  for the GTA method for the three space discretizations and for the cases  $s = 0.25$  and  $s = 0.5$  are shown in Figure 5, plotted over the number of iterations.

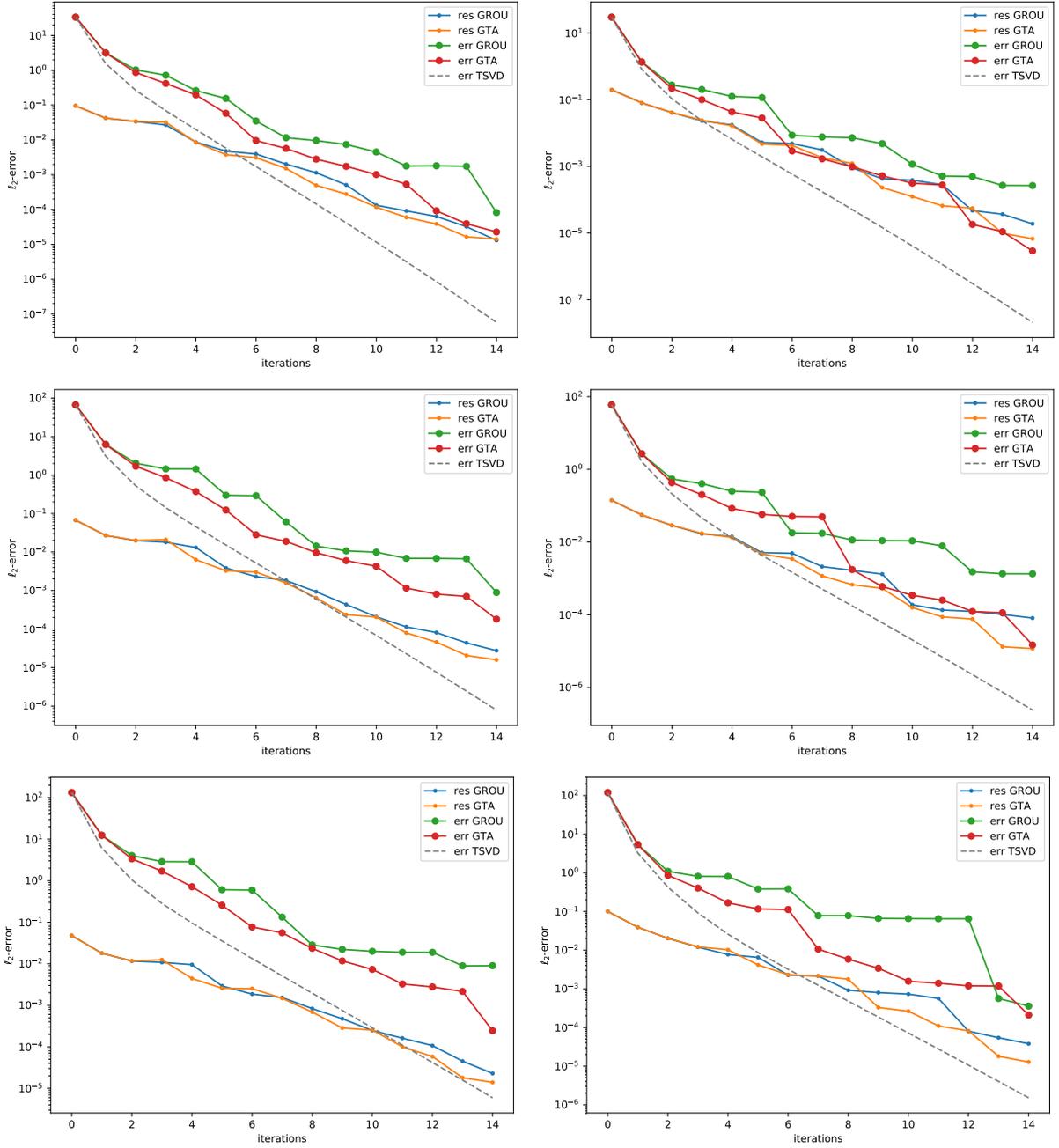


Figure 2: Example 1; left column:  $s = 0.25$ , right column:  $s = 0.5$ . Top to bottom:  $n = 100, 200, 400$ . Errors and residuals plotted over iterations for algorithms GTA and GROU. The dashed grey line denotes the best approximation error computed by truncated SVD.

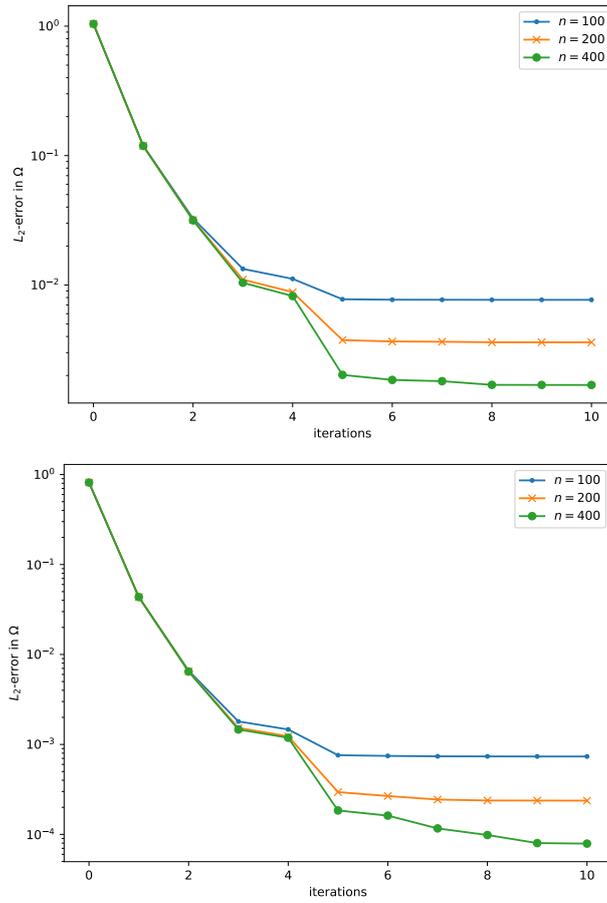


Figure 3: Example 1; top:  $s = 0.25$ , bottom:  $s = 0.5$ .  $L_2$  errors in  $\Omega$  plotted over the number of iterations for the GTA method

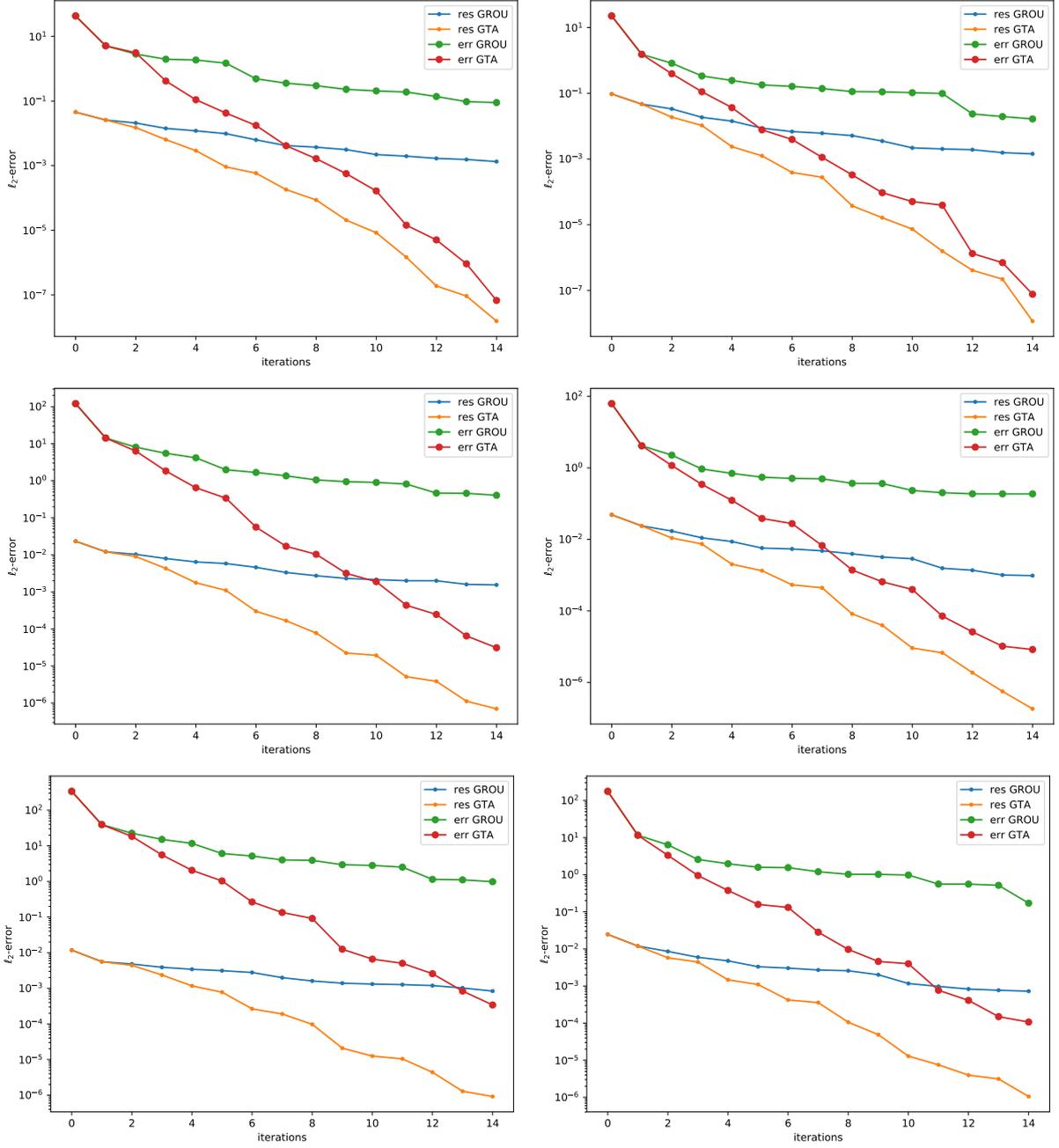


Figure 4: Example 2; left column:  $s = 0.25$ , right column:  $s = 0.5$ . Top to bottom:  $n = 40, 80, 160$ .  $\ell_2$ -errors and residuals plotted over iterations for algorithms GTA and GROU.

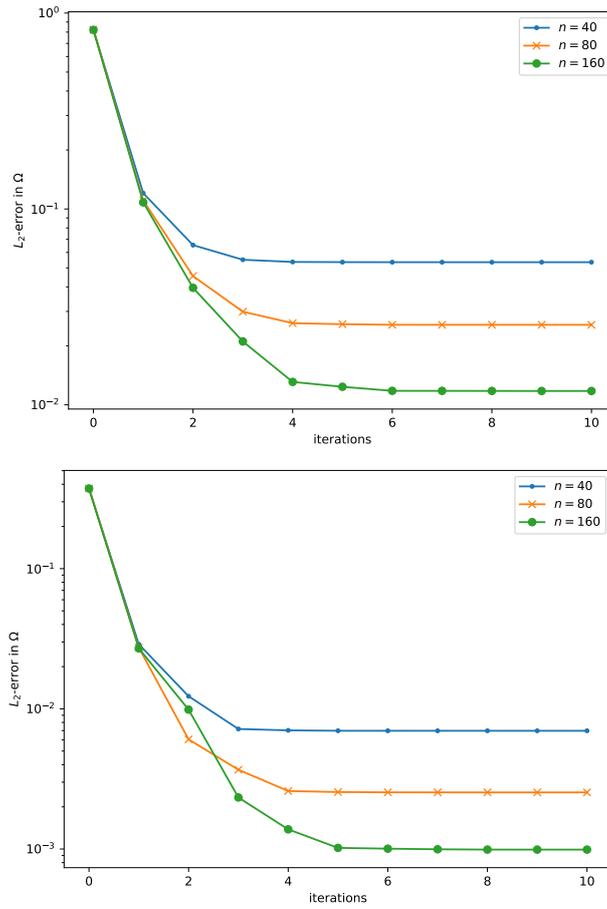


Figure 5: Example 2; top:  $s = 0.25$ , bottom:  $s = 0.5$ .  $L_2$  errors in  $\Omega$  plotted over the number of iterations for the GTA method

### 5.3. Example 3: 2D example in a nontrivial domain

In order to demonstrate the applicability of our methods to problems in nontrivial domains as well as higher-order discretizations, we consider the quarter annulus domain  $\Omega = \{(x, y) \in \mathbb{R}^2 : x > 0, y > 0, 1 < \sqrt{x^2 + y^2} < 2\}$  and solve the problem

$$(-\Delta)^s u(x, y) = f(x, y) = \begin{cases} 1, & |(x, y) - (0.4, 1.5)| \leq 0.2, \\ 0, & \text{otherwise} \end{cases} \quad \forall (x, y) \in \Omega,$$

with homogeneous Dirichlet boundary conditions. The domain and right-hand side are illustrated, together with an approximate solution for  $s = 0.5$ , in Figure 6.

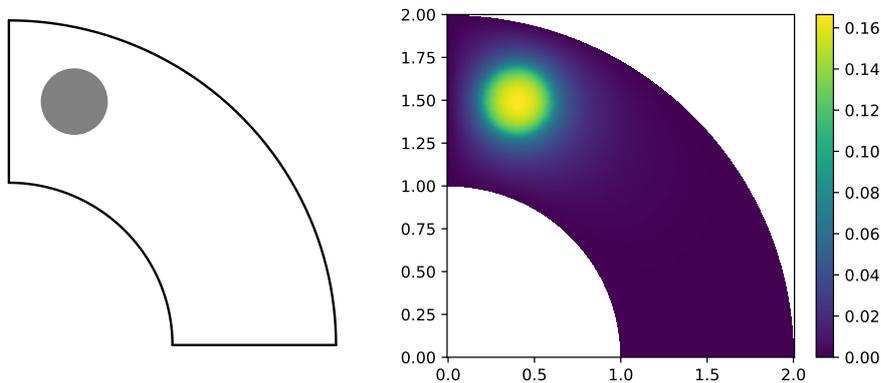


Figure 6: Example 3; left: domain and right-hand side (white=0, gray=1); right: approximate solution for  $s = 0.5$

In order to discretize this problem, we use an Isogeometric Analysis approach where we first construct a uniform tensor product mesh over a reference domain  $(0, 1)^2$  and construct a cubic tensor product spline space over this mesh. The computational domain is represented by a geometry mapping  $\mathcal{G} : (0, 1)^2 \rightarrow \Omega$  given in terms of NURBS. The basis functions on the domain  $\Omega$  are then given by the concatenation of  $\mathcal{G}^{-1}$  with the tensor product B-splines on the reference domain. We refer to [40] for further details on the isogeometric approach.

Also for the extended direction  $W_h$ , we use cubic splines over a graded mesh according to (5).

All used spline spaces have maximum continuity  $C^2$ . Thus, the degrees of freedom for  $V_h$  are again roughly  $n^2$ , and for  $W_h$  roughly  $n/2$ . The discretization of the extended problem again has roughly  $n^3/2$  degrees of freedom.

Since this problem is not posed in a tensor product domain, we use only partial tensor decomposition as described in Section 4.2, that is, the solution is described as a low-rank matrix. We note that it would be possible to perform full tensor decomposition in the reference domain, however this would require to approximate the operator  $\mathcal{A}$  with increased Kronecker rank.

On Figure 7 we present the residuals for the case of  $s = 0.25$  and  $s = 0.5$  using Algorithm 3 and Algorithm 4 plotted over the iterations for the three space discretizations. Note that in this example we could not compute the  $\ell_2$  errors for the extended problem as even the computation of the stiffness matrix (4) exceeded the available memory on a machine with 12GB RAM in the case  $n = 160$ . The low-rank methods are much more memory efficient and can deal with the problem of this size without difficulties.

The  $L_2$  errors in  $\Omega$  for the GTA method for the three space discretizations and for the cases  $s = 0.25$  and  $s = 0.5$  are shown in Figure 8, plotted over the number of iterations.

### 5.4. Discussion

In all examples, the GTA method achieved an  $L_2$ -error close to the approximation error with less than 10 iterations, often significantly less.

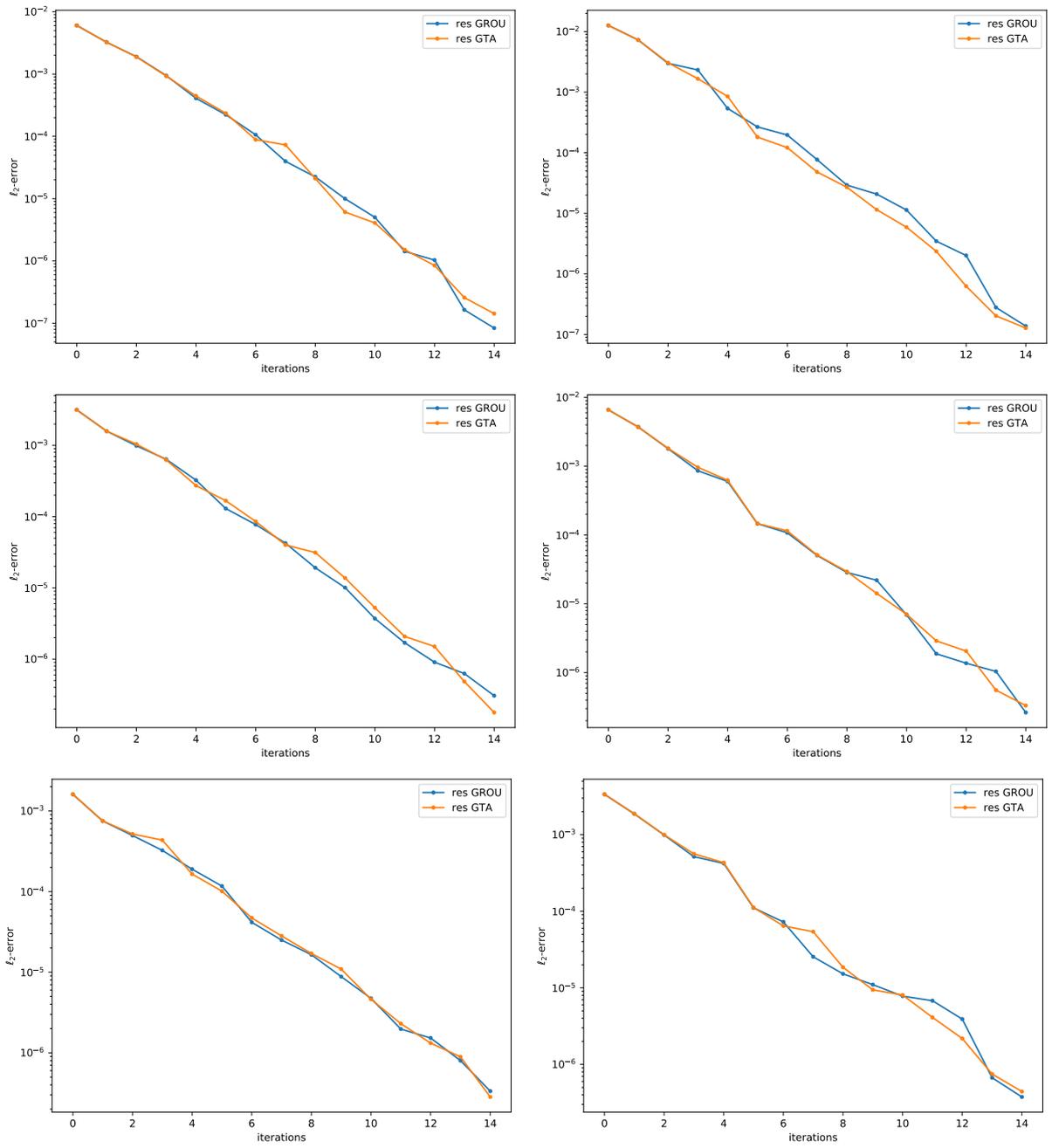


Figure 7: Example 3: 2D example in quarter annulus domain. Left column:  $s = 0.25$ , right column:  $s = 0.5$ . Top to bottom:  $n = 40, 80, 160$ . Residuals plotted over iterations for algorithms GTA and GROU.

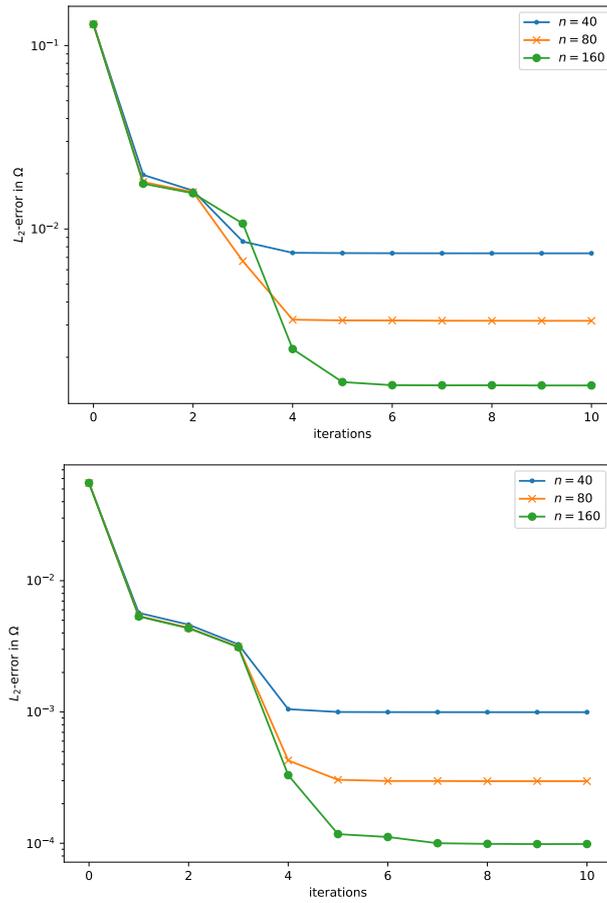


Figure 8: Example 3; 2D example in quarter annulus domain. Top:  $s = 0.25$ , bottom:  $s = 0.5$ .  $L_2$  errors in  $\Omega$  plotted over the number of iterations for the GTA method

The simpler GROU method has comparable, if slightly worse performance for the 1D example and almost identical performance for the nontrivial 2D example, where partial tensor decomposition had to be used. For both of these examples, the solution tensor has order 2, i.e., is a matrix.

In the 2D example in the tensor product domain where full tensor decomposition could be used, the solution tensor has order three and the GTA method performs significantly better than the GROU method. This observation is in agreement with the results from [34], where this difference in performance was observed for a 3D Poisson problem.

Generally, computational cost per iteration is quite similar for the GTA and the GROU algorithms. The major additional cost for GTA is the solution of the dense core linear system given in (11). In the  $k$ -th iteration, this problem has  $k^2$  unknowns in the partial tensor decomposition case and  $k^{d+1}$  unknowns in the full tensor decomposition case (cf. Section 4.2). Since all problems required less than 10 iterations to approach the discretization error, the problem was always smaller than 1000 degrees of freedom and could be solved very quickly with modern dense linear algebra routines. Note, in particular, that the size of this problem does not depend on the discretization spaces  $V_h$  or  $W_h$ .

## Acknowledgments

This work was supported by the bilateral project DNTS-Austria 01/3/2017 (WTZ BG 03/2017), funded by Bulgarian National Science Fund and OeAD (Austria). The first and the third authors were supported also by Grant DN 02/14/2016, funded by the Bulgarian National Science Fund.

## References

- [1] G. M. Zaslavsky, Chaos, fractional kinetics, and anomalous transport, *Physics Reports* 371 (6) (2002) 461–580 (2002).
- [2] A. C. Eringen, *Nonlocal Continuum Field Theories*, Springer, 2002 (2002).
- [3] S. Silling, Reformulation of elasticity theory for discontinuities and long-range forces, *Journal of the Mechanics and Physics of Solids* 48 (1) (2000) 175–209 (2000).
- [4] O. G. Bakunin, *Turbulence and Diffusion: Scaling Versus Equations*, Springer Science & Business Media, 2008 (2008).
- [5] B. McCay, M. Narasimhan, Theory of nonlocal electromagnetic fluids, *Archives of Mechanics* 33 (3) (1981) 365–384 (1981).
- [6] M. Ainsworth, Z. Mao, Analysis and approximation of a fractional Cahn-Hilliard equation, *SIAM J. Numer. Anal.* 55 (4) (2017) 1689–1718 (2017).
- [7] J. Bear, A. H.-D. Cheng, *Modeling Groundwater Flow and Contaminant Transport*, Vol. 23, Springer, 2010 (2010).
- [8] P. Bates, On some nonlocal evolution equations arising in materials science. *nonlinear dynamics and evolution equations*, *Fluid Institute Communications* 48 (1-40) (2006) 10 (2006).
- [9] E. Zijlstra, A. Kalitsov, T. Zier, M. Garcia, Fractional diffusion in silicon, *Advanced Materials* 25 (39) (2013) 5605–5608 (2013).
- [10] A. Bueno-Orovio, D. Kay, V. Grau, B. Rodriguez, K. Burrage, Fractional diffusion models of cardiac electrical propagation: role of structural heterogeneity in dispersion of repolarization, *Journal of the Royal Society Interface* 11 (97) (2014). doi:10.1098/rsif.2014.0352.
- [11] G. Gilboa, S. Osher, Nonlocal operators with applications to image processing, *Multiscale Modeling & Simulation* 7 (3) (2008) 1005–1028 (2008).
- [12] A. Abirami, P. Prakash, K. Thangavel, Fractional diffusion equation-based image denoising model using cngl scheme, *Int. J. Computer Mathematics* (2017) 1–18 (2017).
- [13] R. H. Nochetto, E. Otárola, A. J. Salgado, A PDE approach to fractional diffusion in general domains: A priori error analysis, *Foundations of Computational Mathematics* 15 (3) (2015) 733–791 (2015). doi:10.1007/s10208-014-9208-x.
- [14] L. Chen, R. H. Nochetto, E. Otárola, A. J. Salgado, Multilevel methods for nonuniformly elliptic operators and fractional diffusion, *Mathematics of Computation* 85 (302) (2016) 2583–2607 (2016). doi:10.1090/mcom/3089.
- [15] P. N. Vabishchevich, Numerically solving an equation for fractional powers of elliptic operators, *Journal of Computational Physics* 282 (2015) 289–302 (2015). doi:10.1016/j.jcp.2014.11.022.
- [16] A. Bonito, J. E. Pasciak, Numerical approximation of fractional powers of elliptic operators, *Mathematics of Computation* 84 (295) (2015) 2083–2110 (2015). doi:10.1090/s0025-5718-2015-02937-8.
- [17] L. Aceto, P. Novati, Rational approximation to the fractional Laplacian operator in reaction-diffusion problems, *SIAM Journal on Scientific Computing* 39 (1) (2017) A214–A228 (2017).
- [18] A. Frommer, S. Güttel, M. Schweitzer, Efficient and stable Arnoldi restarts for matrix functions based on quadratures, *SIAM Journal on Scientific Computing* 35 (2) (2014) 661–683 (2014).
- [19] P. Vabishchevich, Numerical solution of time-dependent problems with fractional power elliptic operator, *Computational Methods in Applied Mathematics* 18 (1) (2017) 111–128 (2017).
- [20] S. Harizanov, R. Lazarov, S. Margenov, P. Marinov, Y. Vutov, Optimal solvers for linear systems with fractional powers of sparse SPD matrices, *Numer Linear Algebra Appl.* (2018) e2167 (2018).

- [21] D. Bolin, K. Kirchner, The rational SPDE approach for Gaussian random fields with general smoothness, submitted, posted as arXiv:1711.04333 (November 2017).
- [22] A. Bonito, J. P. Borthagaray, R. H. Nochetto, E. Otárola, A. J. Salgado, Numerical methods for fractional diffusion, *Computing and Visualization in Science* 19 (5-6) (2018) 19–46 (2018). doi:10.1007/s00791-018-0289-y.
- [23] A. Bonito, J. Pasciak, Numerical approximation of fractional powers of regularly accretive operators, *IMA J Numer Anal* 37 (3) (2016) 1245–1273 (2016).
- [24] S. Harizanov, S. Margenov, Positive approximations of the inverse of fractional powers of SPD M-matrices, in: *Control Systems and Mathematical Methods in Economics*, Vol. 687 of *Lecture Notes in Economics and Mathematical Systems*, Springer International Publishing AG, 2018, pp. 147–163 (2018).
- [25] R. Lazarov, P. Vabishchevich, A numerical study of the homogeneous elliptic equation with fractional order boundary conditions, *Fractional Calculus and Applied Analysis* 20 (2) (2017) 337–351 (2017).
- [26] R. Ciegis, V. Starikovicus, S. Margenov, R. Kriauziene, Parallel solvers for fractional power diffusion problems, *Concurrency and Computation: Practice and Experience* 29 (24) (2017). doi:10.1002/cpe.4216.
- [27] N. Kosturski, S. Margenov, Y. Vutov, Performance analysis of mg preconditioning on intel xeon phi: Towards scalability for extreme scale problems with fractional laplacians, in: *Large-Scale Scientific Computing*, Vol. LNCS 10665, Springer, 2017, pp. 304–312 (2017).
- [28] L. Caffarelli, L. Silvestre, An extension problem related to the fractional Laplacian, *Communications in Partial Differential Equations* 32 (8) (2007) 1245–1260 (2007). doi:10.1080/03605300600987306.
- [29] C. Brändle, E. Colorado, A. de Pablo, U. Sánchez, A concave-convex elliptic problem involving the fractional Laplacian, *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 143 (01) (2013) 39–71 (2013). doi:10.1017/s0308210511000175.
- [30] A. Capella, J. Dávila, L. Dupaigne, Y. Sire, Regularity of radial extremal solutions for some non-local semilinear equations, *Communications in Partial Differential Equations* 36 (8) (2011) 1353–1384 (2011). doi:10.1080/03605302.2011.562954.
- [31] P. R. Stinga, J. L. Torrea, Extension problem and harnack’s inequality for some fractional operators, *Communications in Partial Differential Equations* 35 (11) (2010) 2092–2122 (2010). doi:10.1080/03605301003735680.
- [32] M. Ainsworth, C. Glusa, Hybrid finite element–spectral method for the fractional Laplacian: Approximation theory and efficient solver, *SIAM Journal on Scientific Computing* 40 (4) (2018) A2383–A2405 (2018). doi:10.1137/17m1144696.
- [33] A. Ammar, F. Chinesta, A. Falcó, On the convergence of a greedy rank-one update algorithm for a class of linear systems, *Archives of Computational Methods in Engineering* 17 (4) (2010) 473–486 (2010). doi:10.1007/s11831-010-9048-z.
- [34] I. Georgieva, C. Hofreither, Greedy low-rank approximation in Tucker format of tensors and solutions of tensor linear systems, NuMa Report 2018-02, Institute of Computational Mathematics, Johannes Kepler University Linz, Austria, to appear in *Journal of Computational and Applied Mathematics* (2018).  
URL <http://www.numa.uni-linz.ac.at/publications/List/2018/2018-02.pdf>
- [35] T. Breiten, V. Simoncini, M. Stoll, Low-rank solvers for fractional differential equations, *Electronic Transactions on Numerical Analysis* 45 (2016) 107–132 (2016).
- [36] G. Heide, V. Khoromskaia, B. N. Khoromskij, V. Schulz, Tensor approach to optimal control problems with fractional  $d$ -dimensional elliptic operator in constraints (2018). arXiv:1809.01971.
- [37] W. Hackbusch, B. N. Khoromski, Hierarchical Kronecker tensor-product approximation to a class of nonlocal operators in high dimension, *Tech. Rep. 16*, Max-Planck-Institut für Mathematik in den Naturwissenschaften Leipzig (2004).  
URL <https://www.mis.mpg.de/de/publications/preprints/2004/prepr2004-16.html>
- [38] M. D. Ortigueira, J. A. T. Machado, What is a fractional derivative?, *Journal of Computational Physics* 293 (2015) 4–13 (2015). doi:10.1016/j.jcp.2014.07.019.
- [39] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. M. Meerschaert, M. Ainsworth, G. E. Karniadakis, What is the fractional Laplacian?, arXiv e-prints (2018). arXiv:1801.09767.
- [40] T. J. R. Hughes, J. A. Cottrell, Y. Bazilevs, Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement, *Computer Methods in Applied Mechanics and Engineering* 194 (39-41) (2005) 4135–4195 (Oct. 2005). doi:10.1016/j.cma.2004.10.008.
- [41] J. B. Kruskal, Rank, decomposition, and uniqueness for 3-way and  $n$ -way arrays, in: R. Coppi, S. Bolasco (Eds.), *Multiway data analysis*, Elsevier, North-Holland, Amsterdam, 1989, pp. 7–18 (1989).
- [42] V. de Silva, L.-H. Lim, Tensor rank and the ill-posedness of the best low-rank approximation problem, *SIAM Journal on Matrix Analysis and Applications* 30 (3) (2008) 1084–1127 (2008). doi:10.1137/06066518x.
- [43] T. G. Kolda, B. W. Bader, Tensor decompositions and applications, *SIAM Review* 51 (3) (2009) 455–500 (2009). doi:10.1137/07070111X.
- [44] L. R. Tucker, Some mathematical notes on three-mode factor analysis, *Psychometrika* 31 (3) (1966) 279–311 (1966). doi:10.1007/BF02289464.
- [45] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer Series in Computational Mathematics, Springer Berlin Heidelberg, 2012 (2012).
- [46] W. Hackbusch, S. Kühn, A new scheme for the tensor representation, *Journal of Fourier Analysis and Applications* 15 (5) (2009) 706–722 (2009). doi:10.1007/s00041-009-9094-9.
- [47] I. V. Oseledets, Tensor-train decomposition, *SIAM Journal on Scientific Computing* 33 (5) (2011) 2295–2317 (2011). doi:10.1137/090752286.
- [48] S. Brenner, L. Scott, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, 1994 (1994).
- [49] M. Griebel, H. Harbrecht, Approximation of bi-variate functions: singular value decomposition versus sparse grids, *IMA Journal of Numerical Analysis* 34 (1) (2014) 28–54 (2014). doi:10.1093/imanum/drs047.