Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 58-12 (2024) 312–317

An Easy-To-Use Tool to Solve Differential Equations with the Fractional Laplacian ⋆

Fabio V. Difonzo * **Roberto Garrappa** **

* *Istituto per le Applicazioni del Calcolo "Mauro Picone", Consiglio Nazionale delle Ricerche (IAC-CNR). Via Giovanni Amendola 122/I, 70126 - Bari, Italy (e-mail: fabiovito.difonzo@cnr.it).*

** *Dipartimento di Matematica, Università degli Studi di Bari, Via E.Orabona 4, 70125 Bari, ITALY (e-mail: roberto.garrappa@uniba.it)*

Abstract: Numerical simulation of fractional-order partial differential equations is a challenging task and the majority of computing environments does not provide support for these problems. In this paper we describe how to exploit some of the Matlab features (a programming language not supporting fractional calculus in a naive way) to solve partial differential equations with the spectral fractional Laplacian. For shortness we focus on fractional Poisson equations but the proposed approach can be extended, with just some technical difficulties, to more involved problems. This approach cannot be considered as a highly efficient and accurate way to solve fractional partial differential equations, but as an easy-to-use tool for non specialists in numerical computation to obtain solutions without having to produce sophisticated numerical codes.

Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license (https://creativecommons.org/licenses/by-nc-nd/4.0/)

Keywords: Fractional Laplacian, spectral definition, numerical computation, Matlab, eigenfunctions and eigenvalues.

1. INTRODUCTION

Nonlocal models are attracting a great attention in the simulation of real-world phenomena. Complex systems, indeed, usually exhibit some kind of nonlocality and fractional calculus is one of the main tools for incorporating nonlocality in mathematical models. We refer, for instance, to Berardi et al. (2023), D'Elia et al. (2020) and Suzuki et al. (2023) for some recent contributions in this field

One of the most interesting tools to represent nonlocality in space is the fractional Laplacian (i.e., a Laplacian operator of non-integer order). This operator does not have a unique definition and different approaches to generalize the usual integer-order Laplacian to arbitrary real orders have been proposed (see, for instance Bonito et al. (2018); Kwaśnicki (2017); Lischke et al. (2020); Stinga (2019)).

The different approaches are in general equivalent when applied on the whole $\mathbb{R}^d$ domain, but they show substantial differences and peculiarities on bounded domains. From the perspective of applications, however, simulations on bounded domains are of more interest.

Numerical simulation of fractional-order systems is more challenging than integer-order ones (e.g, see Garrappa and Popolizio (2022)): it is necessary to take into account a persistent memory, and the lack of regularity of solutions makes difficult to obtain accurate approximations.

Although most of the computing tools provides some utilities for solving partial differential equations (PDEs) of integer order, such utilities does not longer apply to PDEs with the fractional Laplacian.

The aim of this paper is to show how to exploit some of the Matlab capabilities to handle integer-order PDEs in order to solve similar problems with the fractional Laplacian. In particular, we focus on a Poisson equation subject to homogeneous boundary conditions of Dirichlet type

$$\begin{cases} \left(-\Delta\right)^s u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x} \in \Omega \\ u(\mathbf{x}) = 0 & \mathbf{x} \in \partial\Omega \end{cases}, \tag{1}$$

where $0 < s < 1$ and $\Omega \subset \mathbb{R}^2$ is any sufficiently smooth bi-demensional domain. The fractional Laplacian $\left(-\Delta\right)^s$ is intended according to the spectral definition.

Although for shortness and simplicity we focus on (1), the same approach can be extended to other problems, as for instance with different elliptic operators or when homogeneous boundary conditions of Neumann type are imposed. And, even some non-trivial difficulties arise, non-homogenous boundary conditions can be considered as well (see Antil et al. (2018)). Finally, we mention that, as discussed by Difonzo and Garrappa (2023), a similar approach can be exploited for time-space fractional PDEs.

The structure of this paper is the following. Section 2 is devoted to present the spectral fractional Laplacian. In Section 3 we describe how to compute, by means of the Matlab PDE toolbox, eigenfunctions and eigenvalues of the indeger-order Laplacian which forms the basis for defining and computing the fractional Laplacian and its action on any source function. In Section 4 we present some numerical experiments and in Section 5 we provide some final remarks.

⋆ This work is funded by Italian MUR under the PRIN-PNRR project P2022M7JZW - "SAFER MESH" and by the GNCS-INdAM project CUP_E53C23001670001.

2405-8963 Copyright © 2024 The Authors. This is an open access article under the CC BY-NC-ND license.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2024.08.208

2. THE SPECTRAL FRACTIONAL LAPLACIAN

In order to present the spectral fractional Laplacian, it is useful to first spend a few words to describe the eigendecomposition (decomposition in terms of eigenfunctions and eigenvalues) of the usual integer-order Laplacian, illustrate how this eigendecomposition can be used to solve the Poisson equation of integer order and hence extend these results to the fractional order and, in particular, to the Poisson equation with the spectral fractional Laplacian.

2.1 Eigendecomposition of the Laplacian

We consider a set of eigenfunctions $\{\varphi_k\}_{k\in\mathbb{N}}$ of the negative Laplacian $-\Delta$ on Ω, coupled with homogeneous Dirichlet conditions, together with the associated eigenvalues $\{\lambda_k\}_{k\in\mathbb{N}}$, i.e.

$$-\Delta\varphi_k(\mathbf{x}) = \lambda_k\varphi_k(\mathbf{x}), \quad \forall \mathbf{x}\in\Omega, \tag{2}$$

with $\varphi_k(\mathbf{x}) = 0$ for $\mathbf{x}\in\partial\Omega$.

Moreover, to form a set of eigenfunctions, the functions $\{\varphi_k\}_{k\in\mathbb{N}}$ are assumed to be orthogonal, i.e.

$$\langle\varphi_k,\varphi_j\rangle = 0, \text{ for } k\neq j.$$

As usual, for real functions $f, g\in L^2(\Omega)$, the inner product and the corresponding norm on Ω are

$$\langle f,g\rangle = \int_\Omega f(\mathbf{x})g(\mathbf{x})\mathrm{d}\mathbf{x}, \quad \|f\| = \sqrt{\langle f,f\rangle}.$$

The eigenvalues λ_k are positive since the the negative Laplacian with homogeneous boundary conditions is a positive definite operator, i.e. $\langle -\Delta u, u\rangle \geq 0$. Moreover, the sequence $\{\lambda_k\}_{k\in\mathbb{N}}$ diverges, namely $\lim_{k\to\infty}\lambda_k = +\infty$. The eigenfunctions $\{\varphi_k\}_{k\in\mathbb{N}}$ belong to the space

$$H_0^2(\Omega) = \Big\{u:\Omega\to\mathbb{R}\,|\,u,\nabla u\in L^2(\Omega),\, u\big|_{\partial\Omega}=0\Big\}.$$

Since the eigenfunctions $\{\varphi_k\}_{k\in\mathbb{N}}$ form a complete set in $L^2(\Omega)$, for any function $f\in L^2(\Omega)$ it is possible to consider the Fourier expansion

$$f(\mathbf{x}) = \sum_{k=0}^{\infty}\hat{f}_k\varphi_k(\mathbf{x}), \quad \hat{f}_k = \frac{\langle f,\varphi_k\rangle}{\|\varphi_k\|^2}. \tag{3}$$

2.2 Spectral solution of the Poisson equation

Whenever $f\in L^2(\Omega)$, thanks to the Lax-Milgram Lemma the solution $u(\mathbf{x})$ to the integer-order Poisson equation

$$\begin{cases} -\Delta u(\mathbf{x}) = f(\mathbf{x}) & \mathbf{x}\in\Omega \\ u(\mathbf{x}) = 0 & \mathbf{x}\in\partial\Omega \end{cases}, \tag{4}$$

belongs to $L^2(\Omega)$ as well. Therefore, $u(\mathbf{x})$ possesses a Fourier expansion

$$u(\mathbf{x}) = \sum_{k=0}^{\infty}\hat{u}_k\varphi_k(\mathbf{x}), \tag{5}$$

with unknown Fourier coefficients $\hat{u}_k$. Thanks to (2) the action of the Laplacian can be described in terms of eigenfunctions and eigenvalues according to

$$-\Delta u(\mathbf{x}) = -\sum_{k=0}^{\infty}\hat{u}_k\Delta\varphi_k(\mathbf{x}) = \sum_{k=0}^{\infty}\hat{u}_k\lambda_k\varphi_k(\mathbf{x}).$$

By replacing the Fourier representations of $u(\mathbf{x})$ and $f(\mathbf{x})$ in the Poisson equation (4)

$$\sum_{k=0}^{\infty}\hat{u}_k\lambda_k\varphi_k(\mathbf{x}) = \sum_{k=0}^{\infty}\hat{f}_k\varphi_k(\mathbf{x}),$$

it is possible to determine the Fourier coefficients $\hat{u}_k$ since

$$\sum_{k=0}^{\infty}(\hat{u}_k\lambda_k - \hat{f}_k)\varphi_k(\mathbf{x}) = 0 \quad\Longleftrightarrow\quad \hat{u}_k = \frac{\hat{f}_k}{\lambda_k},\ k=0,1,\dots.$$

2.3 The spectral fractional Laplacian and the fractional Poisson equation

The eigendecomposition $\{\varphi_k,\lambda_k\}_{k\in\mathbb{N}}$ of $-\Delta$ allows to extend the standard integer-order Laplacian operator to a fractional-order $0<s<1$ according to the following definition for which we refer to Bonito et al. (2018); Caffarelli and Stinga (2016); Lischke et al. (2020).

Definition 1. Let $\Omega\subset\mathbb{R}^2$, $u\in L^2(\Omega)$ and assume that an eigendecomposition $\{\varphi_k,\lambda_k\}_{k\in\mathbb{N}}$ of the negative Laplacian $-\Delta$, subject to homogeneous Dirichlet boundary conditions on $\partial\Omega$, is available. The *spectral fractional Laplacian* $\big(-\Delta\big)^s$ of non-integer order $0<s<1$ is hence defined as

$$\big(-\Delta\big)^s u(\mathbf{x}) = \sum_{k=0}^{\infty}\hat{u}_k\lambda_k^s\varphi_k(\mathbf{x}). \tag{6}$$

It is useful to observe that when $s=0$ Eq. (6) represents the identity operator; moreover, when $s=1$ the integer-order Laplacian is obtained form (6).

The utility of the Definition 1 of the fractional Laplacian relies on the possibility of easily obtaining the solution of the fractional Laplacian (1) once an eigendecomposition $\{\varphi_k,\lambda_k\}_{k\in\mathbb{N}}$ of the integer-order Laplacian is available.

Indeed, after replacing (3) and (6) in (1) we observe that

$$\sum_{k=0}^{\infty}\hat{u}_k\lambda_k^s\varphi_k(\mathbf{x}) = \sum_{k=0}^{\infty}\hat{f}_k\varphi_k(\mathbf{x})$$

from which one easily obtains the uknown Fourier coefficients of the solution $u(\mathbf{x})$ to (1) as

$$\hat{u}_k = \frac{\hat{f}_k}{\lambda_k^s} = \frac{\langle f,\varphi_k\rangle}{\lambda_k^s\|\varphi_k\|^2}.$$

Therefore, in order to solve the fractional Poisson equation (1) the following main tasks can be accomplished:

- computing a finite subset $\{\varphi_k,\lambda_k\}_{k=0,K}$ of the eigendecomposition with respect to the specific domain Ω and the selected boundary conditions;
- computing the inner products $\langle f,\varphi_k\rangle$ and the norms $\|\varphi_k\|$;
- determining the number K of eigenfunctions and eigenvalues to be used to obtain an acceptable accuracy (the Fourier expansion must be necessarily truncated).

Finding an optimal value for the number K of terms in the Fourier expansion is not an easy task. Indeed, K must be small enough to avoid a too much expensive computation (with maybe strong accumulation of round-off error) and large enough to avoid inaccuracy. The presence of the factor λ_k^s suggests that the computation

becomes difficult with small s since it requires to evaluate a large decomposition with λ_k in a sufficiently large interval.

Clearly, one could set the interval $[0, L]$ in which to find eigenvalues λ_k in dependence of s, thus to make L^{-s} small enough. The number of eigenvalues in such interval, however, depends on the characteristic of the domain. Moreover, determining the appropriate L for the sought eigenvalues, up to a given tolerance, would require a sharp analysis on the truncation of the infinite series (3) of the Fourier expansion of the function f. In order to do so, one could leverage classical results in Grafakos (2004), that say that the decay of Fourier coefficients of f depends on its smoothness, by adapting them when the basis of $L^2(\Omega)$ is given by the eigenfunctions of $-\Delta$ on Ω, for specific domains and boundary conditions. However, the purpose of this paper is to just provide a practical way to obtain solutions to (1) and therefore we avoid such detailed analysis.

3. COMPUTATION OF EIGENFUNCTIONS AND EIGENVALUES

In some simple cases, analytical representations of eigenfunctions and eigenvalues of the Laplacian (which depends on the domain and the boundary conditions) are available.

This is the case, for instance, of the unit square $[0,1]\times[0,1]$ subject to homogeneous Dirichlet boundary conditions for which eigenfunctions and eigevalues are

$$\lambda_{j,k} = (j\pi)^2 + (k\pi)^2,$$
$$\varphi_{j,k}(x,y) = 2\sin(j\pi x)\sin(k\pi y)$$

for $j,k = 1,2,\dots$ and $\mathbf{x} = (x,y) \in \Omega$.

For more general domains, eigenfunctions and eigenvalues can just be numerically approximated. It is beyond the scope of this paper to describe methods for the numerical computation of eigenfunctions and eigenvalues, a topic for which we refer to the existing literature, as for instance the interesting review by Boffi (2010).

We instead prefer to exploit the potentials of the Matlab PDE toolbox which not only allows to solve a wide range of PDEs, but it also provides some tools to determine suitable approximations of eigenfunctions and eigenvalues.

It is indeed possible to provide a numerical approximation of a certain number, say K, of eigenfunctions and eigenvalues and thus approximate the solution to (1) as

$$u^K(\mathbf{x}) = \sum_{k=0}^{K} \hat{u}_k \varphi_k(\mathbf{x}). \tag{7}$$

It is true that procedures for solving PDEs based on the numerical approximation of the eigendecomposition are not efficient. Indeed, computing eigenfunctions and eigenvalues is an expensive task, especially in the presence of non trivial domains. However, in absence of any built-in function for fractional-order PDEs, the procedure we are going to describe may be used as a way to easily obtain a first approximation or for testing more involved methods.

3.1 Computing eigenfunctions and eigenvalues by the Matlab PDE toolbox

The set of instructions for computing eigenfunctions and eigenvalues is quite simple but requires a certain attention to properly define the main parameters. After creating the PDE model by means of

```
model = createpde ;
```

we observe that the general form of a PDE in the Matlab PDE toolbox is

$$m\frac{\partial^2}{\partial t^2}u + d\frac{\partial}{\partial t}u - \nabla\cdot\left(c\nabla u\right) + au = f$$

where m, d, c, a and f can be functions of space and time other than, in some cases, of the solution u itself and of some of its derivatives. Moreover the general eigenvalue problem in the Matlab PDE toolbox is

$$-\nabla\cdot\left(c\nabla u\right) + au = \lambda du$$

and, since $\Delta u = \nabla\cdot\nabla u$, and all parameters must be assigned, it is sufficient to introduce $m = 0$, $d = 1$, $c = 0$, $a = 0$, $f = 0$ to define the eigenvalue problem $-\Delta u = \lambda u$ by means of

```
specifyCoefficients(model, ...
    'm',0,'d',1'c',1,'a',0,'f',0);
```

The description of the geometry of the domain Ω is a more subtle task for which the Matlab PDE toolbox offers different tools. The choice of the most suitable tool depends on several aspects and we will be illustrated later by means of some examples.

To impose boundary conditions of Dirichlet type it is sufficient to provide the following instruction and, clearly, it is sufficient to replace the keyword `'dirichlet'` with `'neumann'` to switch to homogeneous Neumann boundary conditions

```
  applyBoundaryCondition(model,'dirichlet', ...
  'Edge',1:model.Geometry.NumEdges,'u',0) ;
```

The Matlab PDE toolbox makes use of finite element methods; the eigenfunctions in two-dimensional domains are approximated by piecewise polynomials in each of the triangle in which the domain is decomposed. Eigenfunctions are given in terms of their values on some points on each triangle. In particular, if linear polynomials are used, just values on vertices are given; if quadratic polynomials are instead used, values on mid-points are provided as well. In Figure 1 we show, for an L-shaped domain, the vertices on which the values of the eigenfunctions are given when linear polynomials are used (left plot) and the vertices and mid-points for the quadratic case (right plot). These meshes are created respectively by

```
generateMesh(model,'Hmax',h, ...
    'GeometricOrder','linear');
```

and

```
generateMesh(model,'Hmax',h, ...
    'GeometricOrder','quadratic');
```

with h the size of the triangular mesh on Ω.

Hence, once the desired interval $[0, L]$ in which to find the eigenvalues λ_k is defined, thus limiting the number of

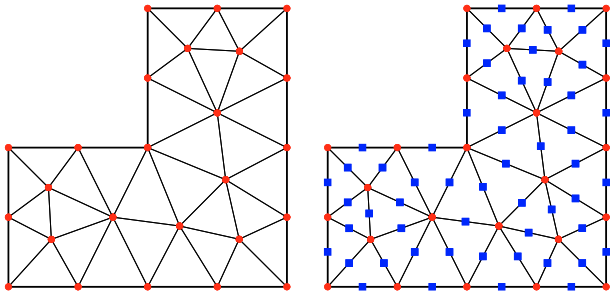

Fig. 1. An L-shaped domain with just vertices for linear polynomials (left) and with vertices and mid-points for quadratic polynomials (right).

eigenvalues (and corresponding eigenfunctions) to be computed, their evaluation is made by the simple instruction

```
eigdecomp = solvepdeeig(model,[0,L]) ;
```

which returns a structure array with three fields: the field `edigdecomp.Eigenvectors` contains values of the eigenfunctions in each point of the triangulation (see Figure 1), `edigdecomp.Eigenvalues` collects the eigenvalues and `edigdecomp.Mesh` the mesh (this is, in turn, a further structure array with coordinates of each node in `edigdecomp.Mesh.Nodes` and indexes of the nodes of each triangle in `edigdecomp.Mesh.Nodes`).

3.2 Evaluation of the solution of the fractional Poisson equation

The inner products $\langle f, \varphi_k \rangle$ and $\langle \varphi_k, \varphi_k \rangle$, necessary to determine the coefficients in the Fourier expansion of $f(\mathbf{x})$, are evaluated by integrating on each triangle and then summing all the sub-integrals.

When using linear polynomials as nodal basis in each triangle it is sufficient to use

$$\langle f, \varphi_k \rangle \approx \sum_{\mathcal{K} \in \mathcal{T}} A_{\mathcal{K}} \sum_{i=1}^{3} f(\mathbf{x}^{\mathrm{v}}_{\mathcal{K},i}) \varphi_k(\mathbf{x}^{\mathrm{v}}_{\mathcal{K},i})$$

while with quadratic polynomials the mid-point rule appears the most suitable rule

$$\langle f, \varphi_k \rangle \approx \sum_{\mathcal{K} \in \mathcal{T}} A_{\mathcal{K}} \sum_{i=1}^{3} f(\mathbf{x}^{\mathrm{m}}_{\mathcal{K},i}) \varphi_k(\mathbf{x}^{\mathrm{m}}_{\mathcal{K},i})$$

with $\mathbf{x}^{\mathrm{v}}_{\mathcal{K},i}$ and $\mathbf{x}^{\mathrm{m}}_{\mathcal{K},i}$ respectively vertices and mid-points of each edge the triangle $\mathcal{K}$ in the triangulation $\mathcal{T}$, and $A_{\mathcal{K}}$ the area of the triangle $\mathcal{K}$ which is computed as

$$A_{\mathcal{K}} = \frac{1}{2} \det \begin{pmatrix} x^{\mathrm{v}}_{\mathcal{K},2} - x^{\mathrm{v}}_{\mathcal{K},1} & x^{\mathrm{v}}_{\mathcal{K},3} - x^{\mathrm{v}}_{\mathcal{K},1} \\ y^{\mathrm{v}}_{\mathcal{K},2} - y^{\mathrm{v}}_{\mathcal{K},1} & y^{\mathrm{v}}_{\mathcal{K},3} - y^{\mathrm{v}}_{\mathcal{K},1} \end{pmatrix}$$

with $\mathbf{x}^{\mathrm{v}}_{\mathcal{K},i} = (x^{\mathrm{v}}_{\mathcal{K},2}, y^{\mathrm{v}}_{\mathcal{K},2}) \in \mathcal{K}$.

The above procedure will be repeated for each element and each eigenfunction, and for computing $\langle \varphi_k, \varphi_k \rangle$ too, as sketched in the code below.

```
for k = 1 : K_eig
for ne = 1 : N_ele
i_el = eig_dec.Mesh.Elements(:,ne) ;
x_ver = eig_dec.Mesh.Nodes(1,i_el(1:3)) ;
y_ver = eig_dec.Mesh.Nodes(2,i_el(1:3)) ;
x_mid = eig_dec.Mesh.Nodes(1,i_el(4:6)) ;
y_mid = eig_dec.Mesh.Nodes(2,i_el(4:6)) ;
phi_mid = eig_dec.Eigenvectors(i_el(4:6),k) ;
BE = [ ...
x_ver(2)-x_ver(1) , x_ver(3)-x_ver(1) ; ...
y_ver(2)-y_ver(1) , y_ver(3)-y_ver(1) ] ;
Area_T = det(BE)/2 ;
loc.x = x_mid ; loc.y = y_mid ;
phi_norm(k) = phi_norm(k) + ...
Area_T/3*(phi_mid.'*phi_mid) ;
f_hat(k) = f_hat(k) + ...
Area_T/3*(f_fun(loc,0)*phi_mid) ;
end
phi_norm(k) = sqrt(phi_norm(k)) ;
f_hat(k) = f_hat(k)/phi_norm(k)^2 ;
end
```

4. NUMERICAL EXAMPLES

In this Section we present some examples with two different domains (the unitary square domain and an annulus) and we show a few instances of the first eigenfunctions and eigenvalues, together with the solution of the corresponding fractional Poisson equation (1) for some values of the fractional order $0 < s < 1$.

The spatial variable $\mathbf{x} \in \Omega$ will be often identified by its two components, namely $\mathbf{x} = (x, y) \in \Omega$ and quadratic polynomials are used as basis functions. For all experiments we have selected a mesh-size $h = 0.005$ and the interval $[0, 200]$ for the eigenvalues.

4.1 Unitary square domain

We start with the simple unitary square domain $\Omega = [0,1] \times [0,1]$ whose geometry can be described in the Matlab PDE toolbox by means of

```
a = 0 ; b = 1 ; c = 0 ; d = 1 ;
R = [2 4 a b b a c c d d]' ;
[dl,bt] = decsg(R,'R',char('R')) ;
[dl,bt] = csgdel(dl,bt) ;
geometryFromEdges(model,dl) ;
```

In Table 1 we show the complete sequence of eigenvalues computed in the interval $[0, 200]$.

Table 1. Eigenvalues for the square domain.

λ_1	19.739	λ_8	128.305
λ_2	49.348	λ_9	167.783
λ_3	49.348	λ_{10}	167.783
λ_4	78.957	λ_{11}	177.653
λ_5	98.696	λ_{12}	197.392
λ_6	98.696	λ_{13}	197.392
λ_7	128.305		

In this specific case it is possible to check the accuracy of the obtained approximations by comparison with the exact values provided at the beginning of Section 3.

This is a quite regular domain whose eigenvalues diverge in a fast way, thus we can cover the selected interval $[0, 200]$ with a small number of eigenvalues. Hence, good enough approximations are obtained with a limited number of eigenfunctions. In Figure 2 we plot the first 4 eigenfunctions.

Solutions obtained for $s \in \{0.5, 0.7, 0.9\}$ are presented in Figure 3. As an attempt to check the accuracy of computed eigenfunctions and eigenvalues, in the fourth plot of Figure

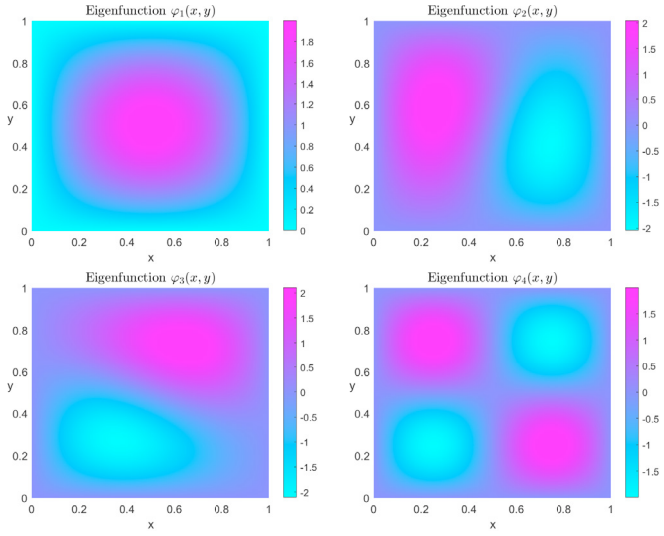

Fig. 2. First 4 evaluated eigenfunctions and eigenvalues for the unitary square domain $\Omega = [0,1] \times [0,1]$

3 we show the difference between the solution for $s = 1.0$ (i.e., the standard integer-order Poisson equation) with the solution obtained by means of the standard quadratic finite element method.

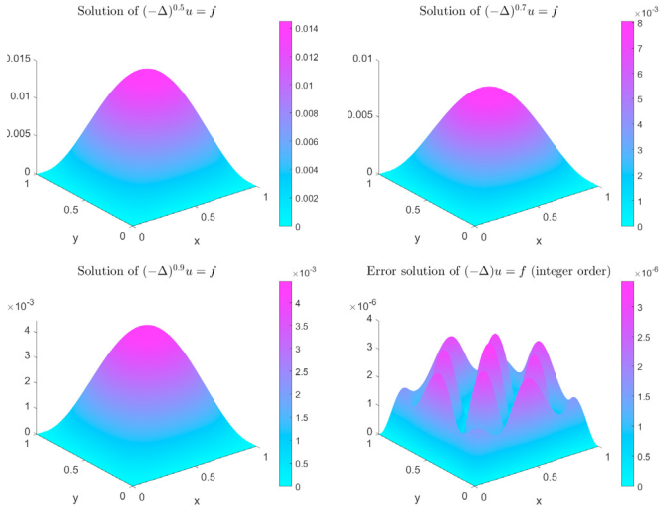

Fig. 3. Solutions of $(-\Delta)^s u = f$ for different values of s in the unitary square domain. The fourth plot is the difference between the solution of the integer-order Poisson equation evaluated by truncation of (5) and that obtained by a quadratic finite element method.

This last plot clearly does not validate results for the fractional-order case. However, they seem to suggest the accuracy of the computed eigendecomposition.

4.2 Annulus

We now consider the domain Ω consisting of an annulus with internal radius $r_0 = 0.6$ and external radius $r_1 = 1.0$, defined by means of

```
r0 = 0.6 ; r1 = 1.0 ;
C1 = [ 1 0 0 r1 0 0 0 0 ]' ;
C2 = [ 1 0 0 r0 0 0 0 0 ]' ;
[dl,bt] = decsg([C1,C2],'C1-C2',char('C1','C2')
  ')
[dl,bt] = csgdel(dl,bt) ;
geometryFromEdges(model,dl) ;
```

together with the source function

$$f(x,y) = -4\left[\frac{x^2+y^2-(r_0+r_1)\sqrt{x^2+y^2}+r_0 r_1}{(r_1-r_0)^2}\right]$$

whose plot is presented in Figure 4.

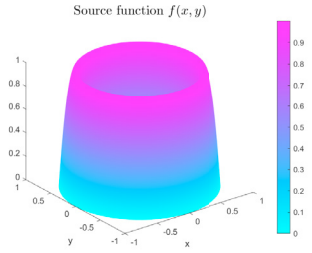

Fig. 4. Source function $f(x,y)$ used for the example in the annulus domain.

In Table 2 we list the eigenvalues in the interval $[0, 200]$, while the plots of the first 4 eigenfunctions are given in Figure 5.

Table 2. Eigenvalues for the annulus domain.

λ_1	61.284	λ_{11}	101.054
λ_2	62.886	λ_{12}	118.373
λ_3	62.886	λ_{13}	118.373
λ_4	67.686	λ_{14}	138.705
λ_5	67.686	λ_{15}	138.705
λ_6	75.668	λ_{16}	161.984
λ_7	75.668	λ_{17}	161.984
λ_8	86.803	λ_{18}	188.143
λ_9	86.803	λ_{19}	188.143
λ_{10}	101.054		

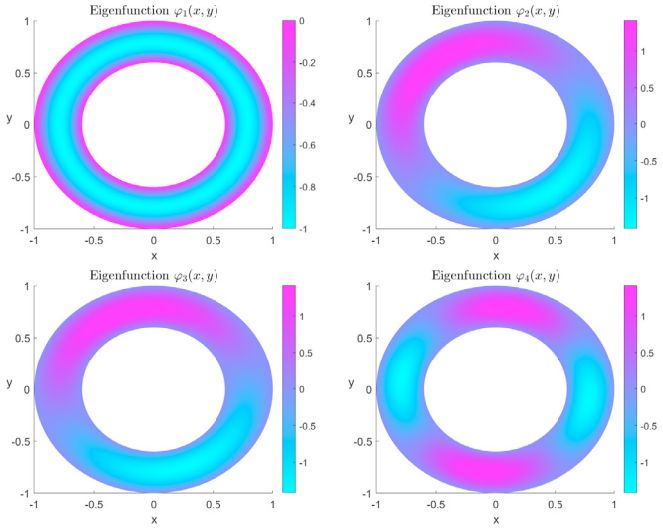

Fig. 5. First 4 evaluated eigenfunctions and eigenvalues for the annulus domain.

Again, the solutions for fractional orders $s \in \{0.5, 0.7, 0.9\}$ are presented in Figure 6.

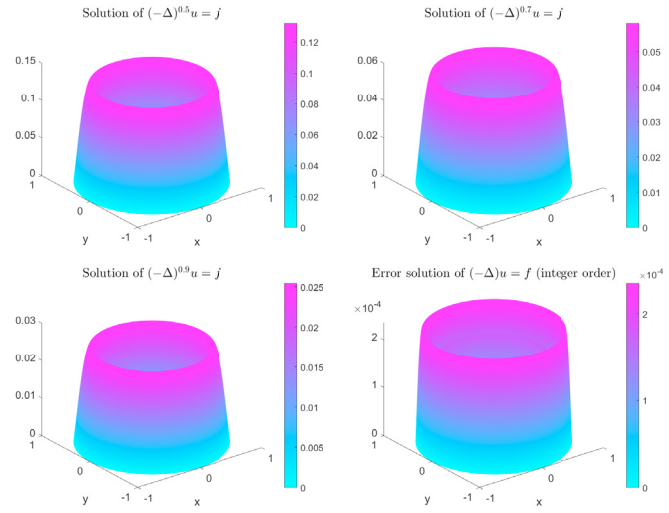

Fig. 6. Solutions of $(-\Delta)^s u = f$ for different values of s in the Annulus domain. The fourth plot is the difference between the solution of the integer-order Poisson equation evaluated by truncation of (5) and that obtained by a quadratic finite element method.

5. CONCLUDING REMARKS

In this work we have discussed an approach to exploit Matlab features, originally devised to solve integer-order problems, in order to solve fractional Poisson equations with the spectral fractional Laplacian.

The proposed approach exploits the capability of the Matlab PDE toolbox of approximating eigenfunctions and eigenvalues of the Laplacian operator by means of finite element methods. The toolbox appears therefore suitable for the spectral fractional Laplacian since its definition is based on the generalization of the eigendecomposition of the standard Laplacian.

We mention that this approach does not constitute a highly effective way to solve fractional Poisson equations, but it can be used, with no particular difficulties, by non specialists in numerical analysis to solve these and other similar problems.

REFERENCES

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

Berardi, M., Difonzo, F.V., and Pellegrino, S.F. (2023). A numerical method for a nonlocal form of Richards' equation based on peridynamic theory. *Comput. Math. Appl.*, 143, 23–32.

Boffi, D. (2010). Finite element approximation of eigenvalue problems. *Acta Numer.*, 19, 1–120.

Bonito, A., Borthagaray, J.P., Nochetto, R.H., Otárola, E., and Salgado, A.J. (2018). Numerical methods for fractional diffusion. *Comput. Vis. Sci.*, 19(5-6), 19–46.

Caffarelli, L.A. and Stinga, P.R. (2016). Fractional elliptic equations, Caccioppoli estimates and regularity. *Ann. Inst. H. Poincaré Anal. Non Linéaire*, 33(3), 767–807.

D'Elia, M., Du, Q., Glusa, C., Gunzburger, M., Tian, X., and Zhou, Z. (2020). Numerical methods for nonlocal and fractional models. *Acta Numer.*, 29, 1–124.

Difonzo, F.V. and Garrappa, R. (2023). A numerical procedure for fractional-time-space differential equations with the spectral fractional Laplacian. In *Fractional differential equations—modeling, discretization, and numerical solvers*, volume 50 of *Springer INdAM Ser.*, 29–51. Springer, Singapore.

Garrappa, R. and Popolizio, M. (2022). A computationally efficient strategy for time-fractional diffusion-reaction equations. *Comput. Math. Appl.*, 116, 181–193.

Grafakos, L. (2004). *Classical and modern Fourier analysis*. Pearson Education, Inc., Upper Saddle River, NJ.

Kwaśnicki, M. (2017). Ten equivalent definitions of the fractional Laplace operator. *Fract. Calc. Appl. Anal.*, 20(1), 7–51.

Lischke, A., Pang, G., Gulian, M., Song, F., Glusa, C., Zheng, X., Mao, Z., Cai, W., Meerschaert, M.M., Ainsworth, M., and Karniadakis, G.E. (2020). What is the fractional Laplacian? A comparative review with new results. *J. Comput. Phys.*, 404, 109009, 62.

Stinga, P.R. (2019). User's guide to the fractional Laplacian and the method of semigroups. In *Handbook of fractional calculus with applications. Vol. 2*, 235–265. De Gruyter, Berlin.

Suzuki, J.L., Gulian, M., Zayernouri, M., and D'Elia, M. (2023). Fractional modeling in action: a survey of nonlocal models for subsurface transport, turbulent flows, and anomalous materials. *J. Peridyn. Nonlocal Model.*, 5(3), 392–459.