Numerical Algorithms
https://doi.org/10.1007/s11075-026-02359-y

RESEARCH

Exploring rational approximations of fractional power operators for preconditioning

Lidia Aceto[1] · **Mariarosa Mazza**[2]

Received: 7 July 2025 / Accepted: 10 March 2026

Abstract
We define and analyze preconditioners for the Riesz operator $-(-\Delta)^{\frac{\alpha}{2}}$, $\alpha \in (1, 2]$ commonly used in fractional models, such as anomalous diffusion. For α close to 2 there are various effective preconditioners at disposal with linear computational cost. Seminal results on treatment of the case α near 1 that still maintains linear computational complexity has been obtained approximating the Riesz operator as a fractional power of a discretized Laplacian, using Gauss-Jacobi formula. In this work, we extend this rational preconditioning approach by leveraging additional quadrature rules with exponential convergence. More precisely, we investigate both sinc and Gauss-Laguerre quadratures and show that, after an opportune choice of the involved parameters, both allow us to construct preconditioners based on a sum of a few shifted Laplacian inverses, and achieve high computational efficiency, ensuring numerical optimality. Several numerical results show that the sinc-based preconditioner is more versatile than the Gauss-Laguerre preconditioner, and that both outperform the Gauss-Jacobi one.

Keywords Fractional power · Rational approximations · Preconditioners · Riesz operator

Mathematics Subject Classification (2010) 335R11 · 65F08 · 65F60 · 15B05 · 65D32

Lidia Aceto and Mariarosa Mazza contributed equally to this work.

✉ Mariarosa Mazza
mariarosa.mazza@uniroma2.it

Lidia Aceto
lidia.aceto@uniupo.it

[1] Dipartimento di Scienze e Innovazione Tecnologica, Università del Piemonte Orientale, Viale T. Michel, 11, 15121 Alessandria, Italy

[2] Dipartimento di Matematica, Università di Roma Tor Vergata, Via della Ricerca Scientifica, 1, 00133 Rome, Italy

Published online: 30 March 2026

1 Introduction

In this paper we concentrate on the non-local operator defined as

$$-(-\Delta)^{\frac{\alpha}{2}} u := -\frac{1}{2\cos(\alpha\pi/2)} \sum_{i=1}^{k} \left({}_{a_i}D^{\alpha}_{x_i} u + {}_{x_i}D^{\alpha}_{b_i} u \right), \qquad \alpha \in (1,2],$$

on $\Omega = \prod_{i=1}^{k} [a_i, b_i]$, with ${}_{a_i}D^{\alpha}_{x_i} u$, ${}_{x_i}D^{\alpha}_{b_i} u$ the left and right Riemann-Liouville fractional derivatives. This operator, known as the Riesz operator, frequently arises in mathematical models that describe processes with memory or hereditary properties, such as anomalous diffusion and fractional quantum mechanics.

The numerical treatment of these models involves discretizing the Riesz operator, typically using finite difference or finite element methods. However, this often leads to dense and ill-conditioned linear systems. To address the computational challenges associated with these linear systems, various preconditioning techniques have been investigated, each with distinct strengths and limitations. Multigrid methods offer robustness across the range of α values with a cost of $O(N \log N)$, with N the matrix-size [1, 2]. Circulant preconditioners [3] achieve the same computational cost, though they cannot ensure convergence when the problem involves higher spatial dimensions [4]. Among algebraic preconditioners specifically designed for coefficient matrices with Toeplitz structure, the τ -preconditioner analyzed in Huang et al. [5] is proven to be convergent while maintaining an $O(N \log N)$ complexity. For values of α near 2, simple Laplacian-based preconditioners [1] and in general banded preconditioners [6, 7] obtained as a truncation of the coefficient matrix have proven effective with a cost of $O(N)$. On the other hand, banded preconditioners are not expected to properly work in case α is near 1, due to the low decay of the coefficient matrix entries. In [8] we have proposed a rational preconditioner that still maintains linear computational complexity and that is efficient also for α close to 1. Here we investigate the performances of new rational alternatives valid again for $\alpha \approx 1$.

The core idea of a rational preconditioning builds on approximating $-(-\Delta)^{\frac{\alpha}{2}}$ as a fractional power of a discretized Laplacian using the Matrix Transfer Technique [9].

A widely used approach to define the fractional powers of a self-adjoint positive definite operator $\mathcal{L}$ acting in a Hilbert space $\mathcal{H}$ involves an integral representation. For $\beta > 0$, the function $\mathcal{L}^{-\beta}$ can be represented in the Dunford-Taylor integral form as follows

$$\mathcal{L}^{-\beta} = \frac{1}{2\pi i} \int_{\gamma_{r,\theta}} z^{-\beta} (z\mathcal{I} - \mathcal{L})^{-1} dz, \tag{1}$$

where the contour $\gamma_{r,\theta}$ encircles the negative real axis clockwise and is the union of two half-lines connected with an open arc of a circle as shown in Fig. 1. As $r \to 0$ and $\theta \to \pi$, the integral (1) can be written as

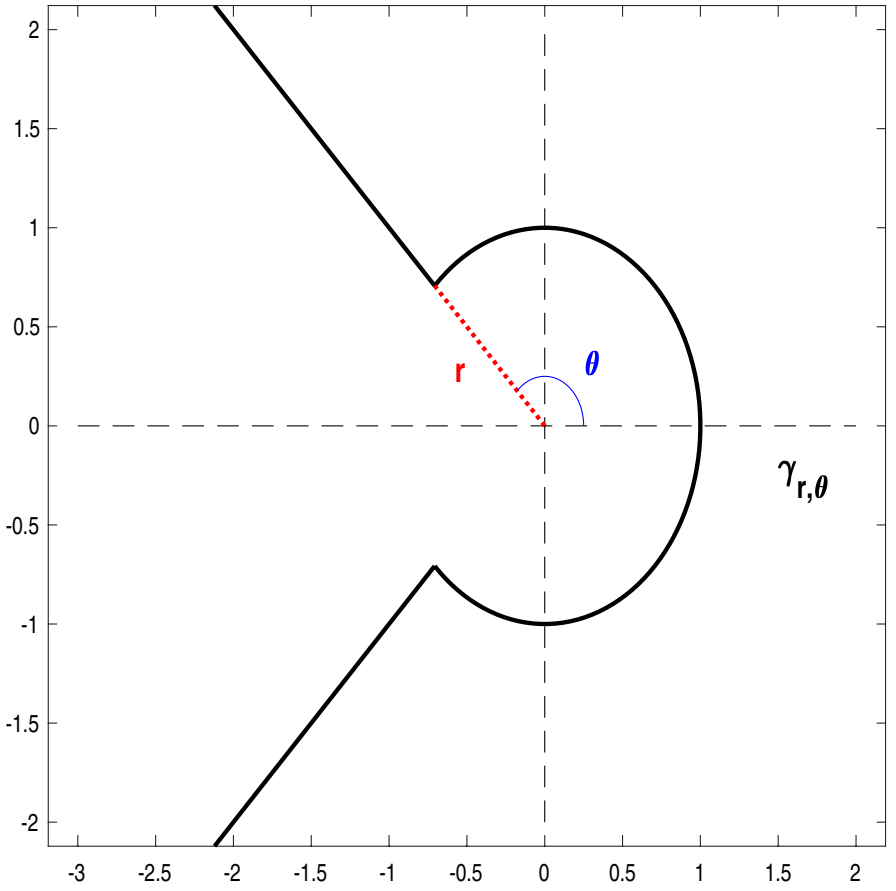

Fig. 1 Shape of the integration contour $\gamma_{r,\theta}$

$$\mathcal{L}^{-\beta} = \frac{\sin(\beta\pi)}{\pi} \int_0^{+\infty} s^{-\beta}(s\mathcal{I} + \mathcal{L})^{-1} ds, \qquad \beta \in (0,1), \tag{2}$$

cfr. [10, Eqs. (1.1)-(1.2)].

Rational approximations of this operator have been approached through various methods in the literature. For example, in the context of unbounded operators, methods based on the best uniform rational approximation (BURA) of functions related to $\lambda^{-\beta}$ have been studied in [11–14], using a modified version of the Remez algorithm.

More recently, the time-stepping methods introduced in Vabishchevich [15] for the parabolic reformulation of fractional diffusion equations were interpreted by Hofreither in [16] as rational approximations of $\lambda^{-\beta}$.

Among the techniques that approximate the action of matrix fractional powers, we mention Krylov-type methods like those in [17] and Massei and Robol [18], and the reduced conjugate gradient basis method in Li et al. [19].

After a suitable change of variable, alternative and effective methods for the numerical approximation of (2) are quadrature rules; see, e.g., [17, 20–29]. This approach typically yields rational approximations $\mathcal{R}_{m-1,m}(\mathcal{L})$ of (2), where

$$\mathcal{R}_{m-1,m}(\lambda) = \frac{p_{m-1}(\lambda)}{q_m(\lambda)}, \qquad p_{m-1}(\lambda) \in \Pi_{m-1},\ q_m(\lambda) \in \Pi_m,$$

with Π_ℓ representing the set of polynomials of degree at most ℓ, and m being either equal to or closely related to the number of points of the quadrature formula.

In [8] we have investigated the Gauss-Jabobi quadrature approach given in [22] to propose a preconditioner that results in the sum of *m* inverses of shifted Laplacian matrices, where *m* is equal to the number of points of the quadrature formula. Here with Laplacian matrix is intended the one corresponding to the chosen discretization. Whenever we are equipped with an efficient solver for it, solving a linear system associated with the preconditioner in [8] can be done applying that same algorithm *m*-times, which makes this strategy independent of the given discretization and of the structure of the resulting coefficient matrix. A moderately small *m* yields good results, and increasing *m* with mesh refinement helps achieve optimal iteration counts. The need of increasing *m* with the mesh refinement is motivated by the fact that, although exponential, the Gauss-Jacobi quadrature convergence depends in the exponent also on the spectrum of the Laplacian. However, higher *m* values raise the computational cost of solving the linear system associated with the preconditioner.

In this work, we extend the rational preconditioning approach in [8] by leveraging additional quadrature rules with exponential convergence, specifically the sinc quadrature [25, 26] and Gauss-Laguerre quadrature [24]. Both methods allow us to form preconditioners that result again in the sum of *m* inverses of shifted Laplacian matrices, where *m* is closely related to the quadrature points number, and therefore is structure-independent.

Concerning the sinc quadrature, we unify and generalize the results in [25, 26] by introducing a parameter μ in the change of variable $s = \mathrm{e}^{-\mu x}$, that reduces to the one in Bonito and Pasciak [26] for $\mu = 2$ and to the one in Bonito et al. [25] for $\mu = -1$. We show that μ can be selected independently of the approximation goals and discuss a strategy to opportunely truncate the sinc formula. As a result, the sinc preconditioner refines and enhances the performance of the Gauss-Jacobi for α close to 1 keeping the number of inverses small and ensuring numerical optimality.

Regarding the Gauss-Laguerre proposal, since the weights of the Gauss-Laguerre rule decay exponentially, it is possible to truncate the formula in a suitable way without loss of accuracy. However, the truncation parameter may vary in a large range corresponding to a large number of inversions which makes this preconditioning proposal less versatile than the sinc one, although better performing than Gauss-Jacobi in more than one dimension.

The rest of the paper is structured as follows. In Section 2 we recall the Gauss-Jacobi quadrature rule and related error estimates. In Section 3 we generalize the approach associated with sinc quadrature and recall the Gauss-Laguerre one. In Section 4 we discuss our preconditioning proposals. In Section 5 we present numerical

experiments in which we compare the performance of the different approaches discussed in the previous sections, including a comparison with state-of-the-art methods. Finally, in Section 6 some conclusions are drawn.

Notation Throughout the work, the symbol $\approx$ denotes a generic approximation, and the symbols $\sim$ and $\lesssim$ are used to denote asymptotic equivalence and less than or asymptotically equal to, respectively. The symbol $\mathcal{I}$ refers to either the identity operator or the identity matrix, with its size being understood from the context.

2 Guass-Jacobi quadrature rule

In this section, we review some key results of the Gauss-Jacobi quadrature approach developed in [20–22], relevant to our analysis.

Consider the Cayley-type transform

$$s = \tau \frac{1-t}{1+t}, \qquad \tau > 0,$$

which allows us to rewrite the integral (2) as follows

$$\mathcal{L}^{-\beta} = \frac{2\sin(\beta\pi)\tau^{1-\beta}}{\pi} \int_{-1}^{1} (1-t)^{-\beta} (1+t)^{\beta-2} \left(\tau \frac{1-t}{1+t}\mathcal{I} + \mathcal{L}\right)^{-1} dt.$$

Approximating this integral using the m-point Gauss-Jacobi rule with weight function $\omega(t) = (1-t)^{-\beta}(1+t)^{\beta-1}$, and corresponding weights w_j and nodes ϑ_j, we obtain

$$\mathcal{L}^{-\beta} \approx \sum_{j=1}^{m} \frac{2\sin(\beta\pi)\tau^{1-\beta}}{\pi} \frac{w_j}{1+\vartheta_j} \left(\tau \frac{1-\vartheta_j}{1+\vartheta_j}\mathcal{I} + \mathcal{L}\right)^{-1} =: \tau^{-\beta} \mathcal{R}^{GJ}_{m-1,m}\left(\frac{\mathcal{L}}{\tau}\right).$$

The error analysis can be conducted by first considering the scalar case with a generic λ, and then extending the result to the operator case by taking the maximum.

For a bounded operator $\mathcal{L}_N$ with spectrum contained in the interval $[\lambda_1, \lambda_N]$, where $\lambda_1 = \lambda_{\min}(\mathcal{L}_N)$ and $\lambda_N = \lambda_{\max}(\mathcal{L}_N)$, the following bound for the absolute error holds

$$\left\| \mathcal{L}_N^{-\beta} - \tau^{-\beta} \mathcal{R}^{GJ}_{m-1,m}\left(\frac{\mathcal{L}_N}{\tau}\right) \right\|_2 \leq \max_{\lambda \in [\lambda_1, \lambda_N]} \tau^{-\beta} \left| E_{m-1,m}\left(\frac{\lambda}{\tau}\right) \right|, \tag{3}$$

and for the relative error we have

$$\left\| \mathcal{I} - \mathcal{L}_N^{\beta} \left(\tau^{-\beta} R^{GJ}_{m-1,m}\left(\frac{\mathcal{L}_N}{\tau}\right)\right) \right\|_2 \leq \max_{\lambda \in [\lambda_1, \lambda_N]} \left(\frac{\lambda}{\tau}\right)^{\beta} \left| E_{m-1,m}\left(\frac{\lambda}{\tau}\right) \right|. \tag{4}$$

Here,

$$E_{m-1,m}\left(\lambda/\tau\right) := \left(\lambda/\tau\right)^{-\beta} - \mathcal{R}^{GJ}_{m-1,m}(\lambda/\tau).$$

In this case, the order of convergence is given by

$$\mathcal{O}(\exp(-4m\sqrt[4]{\lambda_1/\lambda_N})).$$

A suitable choice for τ can be obtained by minimizing both bounds in (3) and (4). In the first case, a reliable value is

$$\tau = \tau_m := \begin{cases} \lambda_1 \left(\frac{\beta}{2me}\right)^2 \exp\left(2W\left(\frac{4m^2e}{\beta^2}\right)\right), & \text{if } m \leq \overline{m}, \\ \left(\tilde{\sigma}_m + \sqrt{\tilde{\sigma}_m^2 + (\lambda_1\lambda_N)^{1/2}}\right)^2, & \text{if } m > \overline{m}, \end{cases} \tag{5}$$

where $W(\cdot)$ is the Lambert W function,

$$\overline{m} = \frac{\beta}{2\sqrt{2}} \left(\frac{\lambda_N}{\lambda_1}\right)^{1/4} \left[\ln\left(\frac{\lambda_N}{\lambda_1}\right) + 2\right]^{1/2},$$

and

$$\tilde{\sigma}_m := -\frac{\beta}{8m} \ln\left(\frac{\lambda_N}{\lambda_1}\right) \lambda_N^{1/2}.$$

We refer the reader to [22, Propositions 3, 4, and eqs. (37)-(38)] for additional details. In the second case, following [8, Theorem 3], we set

$$\tau = \sqrt{\lambda_1\lambda_N}. \tag{6}$$

For $\beta \in (\frac{1}{2}, 1]$, a finer selection of τ, based on the spectral properties of the discretization matrices, has been proposed in [8]. Interpolating between these values, for a uni-dimensional problem, we define

$$\tau = \frac{1}{N^{2\beta}}, \tag{7}$$

ensuring a smooth transition between the behavior of τ in (6), suitable for β close to $\frac{1}{2}$, and in (5), which is more appropriate for β near 1. In the multi-dimensional case N is replaced by the number of discretization points in each direction.

3 Exponential convergent quadrature rules

This section revisits and generalizes the sinc approximation previously developed in [25, 26] and reviews key results of the Gauss-Laguerre approach discussed in [24].

3.1 Sinc quadrature

Using the change of variable

$$s = e^{-\mu x}, \qquad \mu \in \mathbb{R} \setminus \{0\},$$

the integral representation (2) of $\mathcal{L}^{-\beta}$ is converted into an integral defined over the entire real line, i.e,

$$\begin{aligned}\mathcal{L}^{-\beta} =& \frac{\mu \sin(\beta\pi)}{\pi} \int_{-\infty}^{+\infty} e^{\mu\beta x} \left(\mathcal{I} + e^{\mu x}\mathcal{L}\right)^{-1} dx \\ =:& \frac{\mu \sin(\beta\pi)}{\pi} \int_{-\infty}^{+\infty} g(x;\mathcal{L}) dx.\end{aligned}$$

By approximating this integral using the infinite trapezoidal rule (also known as sinc quadrature or single exponential quadrature) with stepsize h we obtain

$$\mathcal{L}^{-\beta} \approx \frac{\mu \sin(\beta\pi)}{\pi} h \sum_{\ell=-\infty}^{+\infty} g(\ell h;\mathcal{L})$$

which, after truncation, becomes

$$\mathcal{L}^{-\beta} \approx \frac{\mu \sin(\beta\pi)}{\pi} h \sum_{\ell=-M_1}^{M_2} g(\ell h;\mathcal{L}) =: \mathcal{R}^{SE}_{m-1,m}(\mathcal{L}), \qquad m = M_1 + M_2 + 1 \quad (8)$$

where M_1, M_2 are positive integers to be chosen appropriately.

We now apply classical error analysis to this type of quadrature. In this light, we define

$$E_{M_1,M_2,h}(\mathcal{L}) := \left\| \mathcal{L}^{-\beta} - \mathcal{R}^{SE}_{m-1,m}(\mathcal{L}) \right\|$$

where $\|\cdot\|$ is the operator norm. To estimate this error, we follow a scalar approach based on the results of Crouzeix and Palencia [30]. For the scalar argument, we have

$$E_{M_1,M_2,h}(\lambda) := \left| \lambda^{-\beta} - \mathcal{R}^{SE}_{m-1,m}(\lambda) \right| \leq \frac{\mu \sin(\beta\pi)}{\pi} \left(\mathcal{E}_D + \mathcal{E}_{T_L} + \mathcal{E}_{T_R}\right),$$

where, up to a constant,

$$\mathcal{E}_D := \left| \int_{-\infty}^{+\infty} g(x;\lambda)dx - h \sum_{\ell=-\infty}^{+\infty} g(\ell h;\lambda) \right| \tag{9}$$

is the discretization error and

$$\mathcal{E}_{T_L} := h \sum_{\ell=-\infty}^{-M_1-1} |g(\ell h;\lambda)|\,, \qquad \mathcal{E}_{T_R} := h \sum_{\ell=M_2+1}^{+\infty} |g(\ell h;\lambda)|$$

are the left and right truncation errors, respectively. For the sake of simplicity, we omit their dependence on M_1, M_2, h. To determine an estimation for the discretization error, we can apply the following result given in [31, Theorem 2.20].

Theorem 1 *Let $f : \mathbb{R} \to \mathbb{R}$ be an integrable function that is analytic in the infinite strip domain*

$$\mathcal{D}_d := \{\zeta \in \mathbb{C} : |Im(\zeta)| < d\}, \qquad d > 0,$$

and which satisfies the following conditions:

- $\displaystyle\int_{-d}^{d} |f(x+i\eta)|d\eta = \mathcal{O}(|x|^a), \quad x \to \pm\infty, \quad 0 \le a < 1,$
- $\displaystyle\mathcal{N}(f,d) := \lim_{\eta \to d^-} \left\{ \int_{-\infty}^{+\infty} |f(x+i\eta)|dx + \int_{-\infty}^{+\infty} |f(x-i\eta)|dx \right\} < +\infty.$

Then,

$$\begin{aligned} \left| \int_{-\infty}^{+\infty} f(x)dx - h \sum_{\ell=-\infty}^{+\infty} f(\ell h) \right| &\le \frac{\mathcal{N}(f,d)}{\sinh(2\pi d/h)} e^{-\pi d/h} \\ &\lesssim \mathcal{N}(f,d)\, e^{-2\pi d/h}, \qquad h \to 0. \end{aligned} \tag{10}$$

As for the poles of the function $g(x;\lambda)$, by solving

$$1 + e^{\mu x}\lambda = 0$$

we find that they are given by

$$x_k = -\frac{1}{\mu}\log|\lambda| - i\left(\frac{\pi}{\mu} + 2k\frac{\pi}{\mu}\right), \qquad k \in \mathbb{Z}.$$

Consequently, the function $g(x;\lambda)$ is analytic in $\mathcal{D}_{\pi/|\mu|}$, that is in the strip domain with $d = \pi/|\mu|$.

To maintain clarity in the presentation, we will focus on the case $\mu > 0$ from here on, as the case $\mu < 0$ can be treated in a similar manner.

Now, we observe that, for $\eta \in \mathbb{R}$, $|\eta| < \pi/\mu$ and $\lambda \geq \lambda_0 > 0$

$$\left|(1 + e^{\mu(x+i\eta)}\lambda)^{-1}\right| \leq \begin{cases} 1, & \text{for} \quad x \leq 0, \\ e^{-\mu x}\lambda_0^{-1}, & \text{for} \quad x > 0. \end{cases}$$

Therefore,

$$|g(x + i\eta; \lambda)| \leq \begin{cases} e^{\mu\beta x}, & \text{for} \quad x \leq 0, \\ e^{-\mu(1-\beta)x}\lambda_0^{-1}, & \text{for} \quad x > 0. \end{cases}$$

This implies that

$$\begin{aligned} \mathcal{N}(g, \pi/\mu) &= \lim_{\eta \to (\pi/\mu)^-} \left\{ \int_{-\infty}^{+\infty} |g(x + i\eta; \lambda)|\, dx + \int_{-\infty}^{+\infty} |g(x - i\eta; \lambda)|\, dx \right\} \\ &\leq \frac{2}{\mu}\left(\frac{1}{\beta} + \frac{1}{(1-\beta)\lambda_0}\right), \end{aligned}$$

and also that

$$\int_{-\pi/\mu}^{\pi/\mu} |g(x + i\eta; \lambda)| d\eta = \mathcal{O}(1), \quad x \to +\infty.$$

Finally, for the discretization error in (9) we have the estimate (cf. (10))

$$\mathcal{E}_D \leq \frac{2}{\mu}\left(\frac{1}{\beta} + \frac{1}{(1-\beta)\lambda_0}\right) e^{-2\pi d/h}, \qquad d = \frac{\pi}{\mu}.$$

To obtain an estimate of $\mathcal{E}_{T_L}$ and $\mathcal{E}_{T_R}$ we first note that

$$|g(x; \lambda)| \leq \begin{cases} e^{\mu\beta x}, & \text{for} \quad x \leq 0, \\ e^{-\mu(1-\beta)x}\lambda_0^{-1}, & \text{for} \quad x > 0. \end{cases}$$

Then we immediately have

$$\mathcal{E}_{T_L} \leq \frac{1}{\mu\beta} e^{-\mu\beta M_1 h}, \qquad \mathcal{E}_{T_R} \leq \frac{1}{\mu(1-\beta)\lambda_0} e^{-\mu(1-\beta)M_2 h}.$$

We are now in the position to state the result which bounds the quadrature error.

Theorem 2 *Let $\mathcal{R}^{SE}_{m-1,m}(\mathcal{L})$ be the truncated sinc quadrature defined by (8) with $\mathcal{L}$ replaced by $\lambda \geq \lambda_0$. Then, for $\mu > 0$*

$$\left|\lambda^{-\beta} - \mathcal{R}^{SE}_{m-1,m}(\lambda)\right| \leq \frac{\mu \sin(\beta\pi)}{\pi} \left(\frac{2(1-\beta)\lambda_0 + 2\beta}{\mu\beta(1-\beta)\lambda_0} e^{-\frac{2\pi^2}{\mu h}} + \frac{e^{-\mu\beta M_1 h}}{\mu\beta} + \frac{e^{-\mu(1-\beta)M_2 h}}{\mu(1-\beta)\lambda_0} \right). \quad (11)$$

In this case, the convergence order is

$$\mathcal{O}(\exp(-\pi\sqrt{2\beta(1-\beta)}\sqrt{m})).$$

Remark 1 As mentioned at the beginning of this section, it is important to emphasize that this error analysis has already been addressed in the literature for specific values of μ. For instance, in Bonito and Pasciak [26], Bonito and Pasciak develop a sinc approximation based on the integral representation (12) under the transformation $x = \log(t)$. This corresponds to the case of $\mu = 2$ in our analysis. However, they determine the parameters M_1, M_2, h corresponding to the strip with $d = \pi/4$. The estimate obtained in this way can be improved by selecting the maximum allowable value for d, i.e., $d = \pi/2$. In [25], the authors instead consider the fractional powers of operators given by the integral (2) (referred to as the Balakrishnan integral in this context) and apply the change of variable corresponding to the case $\mu = -1$ in our analysis. Therein, the strip domain is correctly set with $d = \pi/2$ at the beginning of Section 3.1, where the error analysis is performed. However, when equalizing the three exponentials in the estimate, the value of d is set equal to $\pi/4$ (see [25, Remark 3.1]), leading to an exponential decay that is less efficient than the one obtained with the correct value of d.

3.2 Gauss-Laguerre approach

The substitution $s = t^{-2}$ allows us to rewrite the integral in equation (2) as

$$\mathcal{L}^{-\beta} = \frac{2\sin(\beta\pi)}{\pi} \int_0^{+\infty} t^{2\beta-1} \left(\mathcal{I} + t^2\mathcal{L}\right)^{-1} dt. \quad (12)$$

This can be separated into two integrals

$$\mathcal{L}^{-\beta} = \frac{\sin(\beta\pi)}{\pi} \left[2\int_0^1 t^{2\beta-1} \left(\mathcal{I} + t^2\mathcal{L}\right)^{-1} dt + 2\int_1^{+\infty} t^{2\beta-1} \left(\mathcal{I} + t^2\mathcal{L}\right)^{-1} dt \right].$$

Next, we perform the following transformations

$$t^{2\beta} = e^{-y}, \qquad t^{2(\beta-1)} = e^{-y},$$

which leads to the following two integrals

$$I^{(1)}(\mathcal{L}) := \int_0^{+\infty} e^{-y}(\mathcal{I} + e^{-y/\beta}\mathcal{L})^{-1}dy,$$
$$I^{(2)}(\mathcal{L}) := \int_0^{+\infty} e^{-y}(e^{-y/(1-\beta)}\mathcal{I} + \mathcal{L})^{-1}dy.$$

By applying the n-point Gauss-Laguerre rule to both integrals with respect to the weight function $\omega(y) = e^{-y}$, with the weights $\omega_j^{(n)}$ and the nodes $\zeta_j^{(n)}$ (in ascending order) we obtain the following rational approximation

$$\begin{aligned}\mathcal{L}^{-\beta} =& \frac{\sin(\beta\pi)}{\beta\pi} I^{(1)}(\mathcal{L}) + \frac{\sin(\beta\pi)}{(1-\beta)\pi} I^{(2)}(\mathcal{L}) \\ \approx& \frac{\sin(\beta\pi)}{\beta\pi} \mathcal{R}^{(1)}_{n-1,n}(\mathcal{L}) + \frac{\sin(\beta\pi)}{(1-\beta)\pi} \mathcal{R}^{(2)}_{n-1,n}(\mathcal{L}) \\ =:& \mathcal{R}^{GL}_{m-1,m}(\mathcal{L}), \qquad m = 2n,\end{aligned} \tag{13}$$

where

$$\mathcal{R}^{(1)}_{n-1,n}(\mathcal{L}) := \sum_{j=0}^{n} \omega_j^{(n)} \left(\mathcal{I} + e^{-\zeta_j^{(n)}/\beta}\mathcal{L}\right)^{-1},$$
$$\mathcal{R}^{(2)}_{n-1,n}(\mathcal{L}) := \sum_{j=0}^{n} \omega_j^{(n)} \left(e^{-\zeta_j^{(n)}/(1-\beta)}\mathcal{I} + \mathcal{L}\right)^{-1}.$$

The error analysis is provided in [24, Section 5], and here we present only the key results. The absolute error is bounded as follows

$$\left\|\mathcal{L}^{-\beta} - \mathcal{R}^{GL}_{m-1,m}(\mathcal{L})\right\| \approx 4\sin(\beta\pi) S\left(\frac{m}{2}, \beta\right),$$

where for each even m,

$$S\left(\frac{m}{2}, \beta\right) = \begin{cases} e^{-3(\frac{m}{2}\beta^2\pi^2)^{1/3}}, & (\forall\, m \wedge \beta \le 1/2) \vee (m > m^* \wedge \beta > 1/2), \\ e^{-(8\pi(1-\beta)\frac{m}{2})^{1/2}}, & (m \le m^* \wedge \beta > 1/2), \end{cases} \tag{14}$$

with

$$m^* \approx \frac{9\beta^4}{(1-\beta)^3}.$$

Since the weights of the Gauss-Laguerre quadrature decay exponentially, the formula can be truncated to reduce the number of inversions without compromising accuracy, cf. [24, Sec. 6]. For any even positive integer m, the truncation involves $m_{tr} = 2n_{tr}$ terms instead of m, where

$$n_{tr} = \begin{cases} \left\lfloor 2\sqrt{3}\left(\frac{\beta m^2}{4\pi^2}\right)^{1/3} \right\rfloor, & (\forall\, m \wedge \beta \leq 1/2) \vee (m > m^* \wedge \beta > 1/2), \\ \left\lfloor 2(1-\beta)^{1/4}\left(\frac{m}{\pi}\right)^{3/4} \right\rfloor, & (m \leq m^* \wedge \beta > 1/2), \end{cases} \tag{15}$$

see [24, Eqs. (6.8) and (6.11)]. Truncation is advantageous for approximation, especially when many inversions are needed, as it reduces computational cost.

The quadrature rules discussed in this section provide efficient rational approximations of the fractional power operator. In the next section, we examine how these approximations can be utilized to construct effective preconditioners that enhance both computational efficiency and robustness in solving fractional differential equations.

4 Rational preconditioning

We now extend the rational preconditioning approach proposed in [8] by leveraging boh sinc quadrature and Gauss-Laguerre quadrature for solving linear systems with coefficient matrix a discretization of $-(-\Delta)^{\frac{\alpha}{2}}$, $\alpha \in (1, 2]$. While existing preconditioning techniques often struggle when α is close to 1, by leveraging these quadrature approximations with exponential convergence, we propose new preconditioners that significantly improve numerical performance while maintaining computational efficiency.

Le us start from the following boundary value problem

$$\begin{aligned} D(\boldsymbol{x})(-\Delta)^{\alpha/2} u(\boldsymbol{x}) &= f(\boldsymbol{x}), & \boldsymbol{x} &:= (x_1, \ldots, x_k) \in \Omega \subset \mathbb{R}^k \\ u(\boldsymbol{x})|_{\hat{\Omega}} &= 0, & \hat{\Omega} &= \mathbb{R}^k \backslash \Omega \end{aligned} \tag{16}$$

with $D(\boldsymbol{x}) \geq 0$, $f(\boldsymbol{x})$ the source function. For the sake of simplicity, here we focus on a uniform centered finite difference discretization, but other discretazions could be taken into account as well. Given $\boldsymbol{n} = (n_1, \ldots, n_k)$ a multi-index, let us define the uniform partition of the domain $\Omega = \prod_{i=1}^{k} [a_i, b_i]$ as $\boldsymbol{\xi_j} = (\xi_{j_1}^{(1)}, \ldots, \xi_{j_k}^{(k)})$ with

$$\xi_{j_i}^{(i)} = a_i + j_i \nu_i, \quad \nu_i = \frac{b_i - a_i}{n_i + 1}, \quad j_i = 0, 1, \ldots, n_i + 1, \quad i = 1, 2, \ldots, k.$$

The centered finite difference discretization of $-(-\Delta)^{\alpha/2}$ is then

$$\mathcal{A}_{\alpha,N}^{(k)} := \sum_{i=1}^{k} I_{n_i^-} \otimes A_{n_i} \otimes I_{n_i^+}, \tag{17}$$

where $N = \prod_{j=1}^{k} n_j$, $n_i^- = \prod_{j=1}^{i-1} n_j$, $n_i^+ = \prod_{j=i+1}^{k} n_j$, and $A_{n_i} = \nu_i^{-\alpha} T_{n_i}(f_\alpha)$ with $\nu_i = 1/(n_i + 1)$

$$f_\alpha(\theta_i) := (\,2 - 2\cos(\theta_i)\,)^{\alpha/2},$$

and

$$T_{n_i}(f_\alpha) = \begin{pmatrix} \ell_0 & \ell_{-1} & \dots & \dots & \ell_{1-n_i} \\ \ell_1 & \ell_0 & \ell_{-1} & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ell_1 & \ell_0 & \ell_{-1} \\ \ell_{n_i-1} & \dots & \dots & \ell_1 & \ell_0 \end{pmatrix},$$

the Toeplitz matrix related to f_α, where

$$\ell_j = \frac{(-1)^j \Gamma(\alpha+1)}{\Gamma(\alpha/2-j+1)\Gamma(\alpha/2+j+1)}, \quad j = 0, \pm 1, \pm 2, \dots, \pm(n_i-1),$$

and $\Gamma(\cdot)$ the Euler gamma function. Without loss of generality, we focus only on the case $\boldsymbol{n} = (\tilde{n}, \dots, \tilde{n})$ and on the domain $\Omega = \prod_{i=1}^k [0,1]$. This leads to rewriting (17) as follows

$$\mathcal{A}^{(k)}_{\alpha,N} = \nu^{-\alpha} T^{(k)}_N(\psi_\alpha),$$

with $\nu = 1/(\tilde{n}+1)$, $\psi_\alpha(\boldsymbol{\theta}) := \sum_{i=1}^k (2 - 2\cos\theta_i)^{\alpha/2}$, and $T^{(k)}_N(\psi_\alpha)$ the k-level Toeplitz generated by ψ_α. When $k = 1$, $\psi_\alpha = f_\alpha$ and $T^{(1)}_N(\psi_\alpha) = T_N(f_\alpha)$.

Note that for $\alpha = 2$, $\mathcal{A}^{(k)}_{\alpha,N}$ corresponds to the finite-difference discretization of the k-dimensional Laplace operator with mesh-width ν in each direction.

Thanks to the k-level Toeplitz structure of $\mathcal{A}^{(k)}_{\alpha,N}$ and in light of the results obtained in [32], we know that the function $\psi_\alpha(\boldsymbol{\theta})$ corresponds to the so-called symbol and provides an estimate of the eigenvalues of the matrices $\nu^\alpha \mathcal{A}^{(k)}_{\alpha,N}$ in the sense of [33, Definition 3.1] when $\tilde{n} \to \infty$. Moreover, it holds that

$$\kappa_2(\nu^\alpha \mathcal{A}^{(k)}_{\alpha,N}) \sim \frac{1}{\tilde{n}^\alpha}, \tag{18}$$

with $\kappa_2(\cdot)$ the spectral condition number of a matrix. Note that in presence of other discretizations, for instance when B-splines are used [34], or when sufficiently regular nonuniform grid are employed, the coefficient matrix is not a pure multi-level Toeplitz anymore, but reduces to the more general GLT class [33] and similar results can be derived as well.

The spectral knowledge in (18) has been leveraged in [8] to define the following preconditioner based on the Gauss-Jacobi quadrature

$$\left(P_{m,N}^{GJ}\right)^{-1} := \tau^{-\frac{\alpha}{2}} \mathcal{R}_{m-1,m}^{GJ}\left(\frac{\mathcal{L}_N}{\tau}\right) = \frac{2\sin(\frac{\alpha}{2}\pi)\tau^{1-\frac{\alpha}{2}}}{\pi} \sum_{j=1}^{m} \frac{w_j}{1+\vartheta_j}\left(\tau\frac{1-\vartheta_j}{1+\vartheta_j}\mathcal{I} + \mathcal{L}_N\right)^{-1},$$

with τ as in (7) for $\beta = \alpha/2$, and with $\mathcal{L}_N = \mathcal{A}_{2,N}^{(k)}$, a finite difference discretization of the Laplacian on a grid with mesh-width $\nu = 1/(\tilde{n}+1)$ in each direction. The proposed preconditioner is then the inverse of a sum of *m* inverses of shifted *k*-level Toeplitz matrices. As shown in [8], thanks to the theory systematically presented in Garoni and Serra-Capizzano [33], it can also be endowed with a symbol that provides an estimate of its eigenvalues, and is defined as follows

$$p_{m,\alpha}^{GJ}(\boldsymbol{\theta}) := \frac{2\sin(\frac{\alpha}{2}\pi)\tau^{\frac{\alpha}{2}}}{\pi}\psi_2(\boldsymbol{\theta}) \sum_{j=1}^{m} \frac{\tilde{w}_j}{1+\tilde{\vartheta}_j}\left(\tau\frac{1-\tilde{\vartheta}_j}{1+\tilde{\vartheta}_j} + \psi_2(\boldsymbol{\theta})\right)^{-1}, \quad (19)$$

with $\tilde{w}_j$ and $\tilde{\vartheta}_j$ the weights and nodes of the Gauss–Jacobi rule with weight function $(1-t)^{\frac{\alpha}{2}-1}(1+t)^{-\frac{\alpha}{2}}$. Roughly speaking, according again to the theory in Garoni and Serra-Capizzano [33], if $p_{m,\alpha}^{GJ}$ approximates ψ_α well enough, the eigenvalues of the preconditioned matrix (weakly) clusters at 1.

In line with the results in [8], Figure 2 shows that in the one-dimensional case, when $\alpha \approx 1$, the accuracy of the approximation provided by $p_{m,\alpha}^{GJ}$ improves significantly as m increases. On the other hand, for $\alpha \approx 2$, $p_{m,\alpha}^{GJ}$ is already accurate for $m = 1$. This means that $P_{m,N}^{GJ}$ behaves as a (shifted) Laplacian preconditioner, which is known to be highly effective when $\alpha \approx 2$; see Donatelli et al. [4].

Remark 2 In the general case, the *k*-dimensional Laplacian used in the definition of the preconditioner should be interpreted as the one associated with the chosen discretization. Whenever we are equipped with an efficient solver for the *k*-dimensional Laplacian, solving a linear system associated with a preconditioner consisting of a

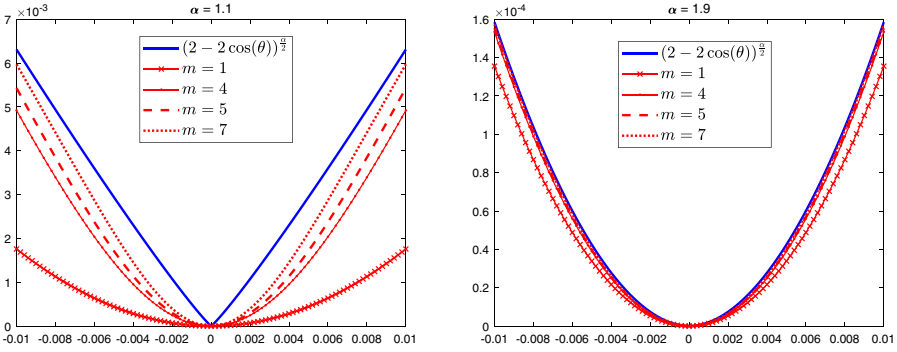

Fig. 2 Comparison of $(2-2\cos(\theta))^{\frac{\alpha}{2}}$ (blue line) and $p_{m,\alpha}^{GJ}$ for $\alpha = 1.1$ (left) and $\alpha = 1.9$ (right)

sum of inverses of shifted Laplacians (such as $P^{GJ}_{m,N}$) can be done by applying the same algorithm m times, which makes this strategy independent of the chosen discretization and of the structure of the resulting coefficient matrix. The overall cost is therefore m times the cost of solving a linear system with a shifted Laplacian, so the effectiveness of these preconditioners depends on how small m can be kept and on how m should scale with N.

In the next sections we extend this same approach of defining a preconditioner to the case of sinc quadrature and the Gauss-Laguerre quadrature described in Section 3.1 and Section 3.2.

4.1 Sinc preconditioning

Starting from (8), let us define the preconditioner based on the sinc quadrature

$$\begin{aligned}\left(P^{SE}_{m,N}\right)^{-1} &:= \mathcal{R}^{SE}_{m-1,m}(\mathcal{L}_N)\\ &= \mu\,\frac{\sin(\frac{\alpha}{2}\pi)}{\pi} h \sum_{\ell=-M_1}^{M_2} e^{\mu\frac{\alpha}{2}\ell h}\left(\mathcal{I}+e^{\mu\ell h}\mathcal{L}_N\right)^{-1}, \qquad m = M_1+M_2+1,\end{aligned} \tag{20}$$

with $\mathcal{L}_N = \mathcal{A}^{(k)}_{2,N}$. Of course, similar computational considerations to the ones done for $P^{GJ}_{m,N}$ in Remark 2 apply to this preconditioner. In what follows we discuss the choice of M_1, M_2, μ, h and search for a way to keep m small.

4.1.1 Parameters estimation considerations

As m represents the number of inversions, in light of preconditioning purposes, M_1, M_2 should be kept small so to perform few inversions. The goal is to choose M_1 and M_2 such that the three exponential terms in the error bound (11) decay at a similar rate. With this aim, we define

$$\delta := \frac{2\pi^2}{\mu h}, \tag{21}$$

and, setting $\beta = \frac{\alpha}{2}$, we compute

$$M_1 = \left\lceil \frac{\delta^2}{\alpha\pi^2} \right\rceil, \qquad M_2 = \left\lceil \frac{\delta^2}{(2-\alpha)\pi^2} \right\rceil. \tag{22}$$

Choosing $\delta \approx \pi$ and computing M_1 and M_2 by (22) we expect that the discretization and the truncation errors have similar exponential behavior for $\alpha \leq 1.5$. This is confirmed in Fig. 3 that shows the three exponentials on the right-hand side of (11) with $\beta = \frac{\alpha}{2}$. The left side of the figure corresponds to $\delta = 1$, while the right side corresponds to $\delta = \pi$. For values of α close to 1, M_2 is small. In contrast, for $\alpha \to 2$, M_2 tends to infinity. On the other hand, M_1 remains constant at 1. This behavior is

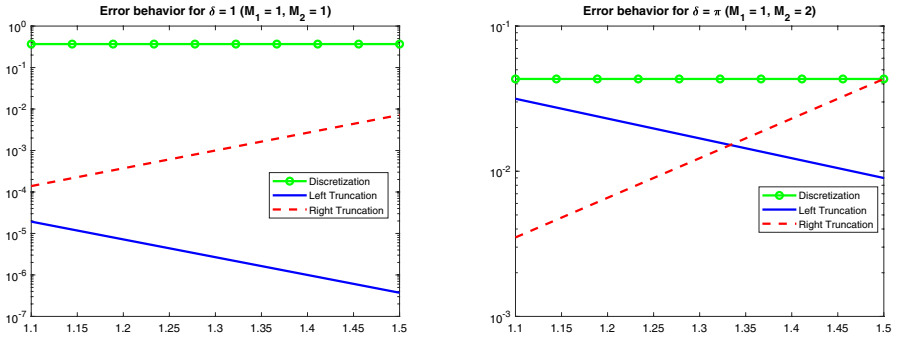

Fig. 3 Errors behavior for $\delta = 1$ (left) and $\delta = \pi$ (right)

illustrated in Fig. 4 where $\alpha \in [1.1, 1.9]$ and $\delta = \pi$. As a consequence, we expect that $P^{SE}_{m,N}$ involves few inversions and is competitive for values of α close to 1, while it requires an increasing number of inversions and becomes expensive, as α tends to 2.

About the choice of μ and h, let us first note that the two appear in (20) only as a product, meaning we can adjust one while keeping the other fixed such that their product does not change. Setting $\delta \approx \pi$, from (21) we get $\mu h \approx 2\pi$. A practical choice is then to fix $\mu = 2$ and set $h \approx \pi$. However, any other choice of μ would simply require an inversely proportional adjustment in h, making μ arbitrary in this sense.

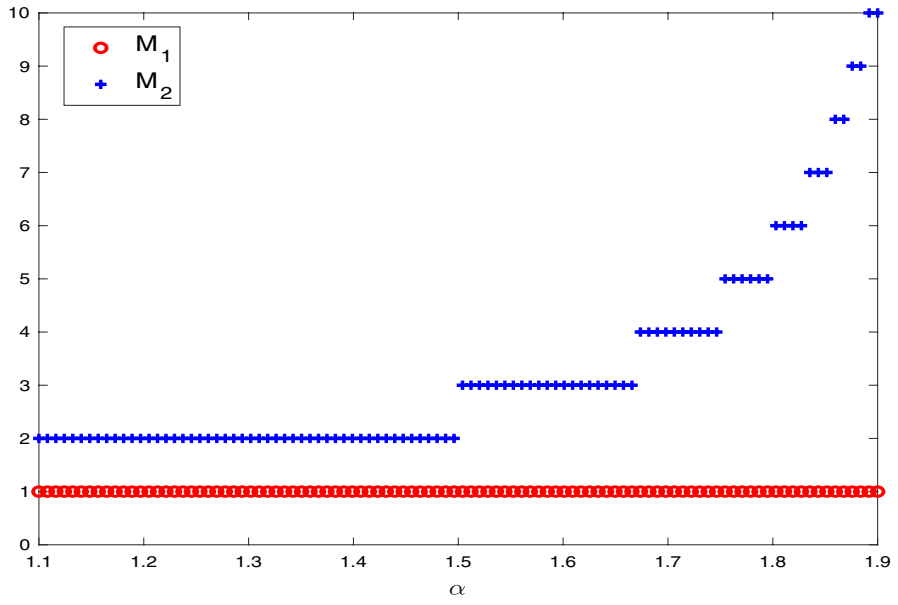

Fig. 4 Values of M_1 and M_2, computed by (22), vs α for $\delta = \pi$

4.1.2 Spectral properties

By following the same steps done in [8], we can associate to the inverse of (20) the function

$$p_{m,\alpha}^{SE}(\boldsymbol{\theta}) := \mu \frac{\sin(\frac{\alpha}{2}\pi)}{\pi} h\, \psi_2(\boldsymbol{\theta}) \sum_{\ell=-M_2}^{M_1} e^{\mu(1-\frac{\alpha}{2})\ell h} \left(1 + e^{\mu \ell h} \psi_2(\boldsymbol{\theta})\right)^{-1} \tag{23}$$

that, as $\tilde{n} \to +\infty$, provides an estimate for the eigenvalues of $P_{m,N}^{SE}$.

Figure 5 shows that, in the one-dimensional case, for $\alpha \approx 1$, the accuracy of the approximation of ψ_α provided by $p_{m,\alpha}^{SE}$ improves with m and that small values of m corresponding to δ in the neighbor of π give satisfactory results. However, this approach fails to be competitive for $\alpha \approx 2$ because m increases rapidly, as already evident from Figure 4.

Comparing Fig. 5 with Fig. 2, we observe that for $\alpha \approx 1$, $p_{m,\alpha}^{SE}$ provides a more accurate approximation of ψ_α near zero than $p_{m,\alpha}^{GJ}$, which retains a visible quadratic behavior at zero even for large values of m. This suggests that $P_{m,N}^{SE}$ is the better choice where accuracy is most needed, namely for $\alpha \approx 1$. On the other hand, for $\alpha \approx 2$, as previously noted, the most effective option remains the Laplacian preconditioner proposed in Donatelli et al. [4].

4.2 Gauss-Laguerre preconditioning

Starting from (13) and fixed n, we define the preconditioner based on the Gauss-Laguerre quadrature

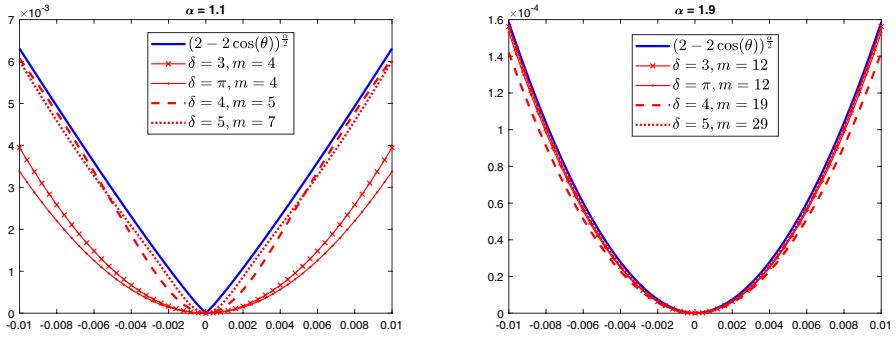

Fig. 5 Comparison of $(2 - 2\cos(\theta))^{\frac{\alpha}{2}}$ (blue line) and $p_{m,\alpha}^{SE}$ for $\alpha = 1.1$ (left) and $\alpha = 1.9$ (right)

$$\begin{aligned}\left(P_{m,N}^{GL}\right)^{-1} :=& \mathcal{R}_{m-1,m}^{GL}(\mathcal{L}_N) \\ =& \frac{\sin(\frac{\alpha}{2}\pi)}{\frac{\alpha}{2}\pi} \sum_{j=0}^{n} \omega_j^{(n)} \left(\mathcal{I} + e^{-2\zeta_j^{(n)}/\alpha}\mathcal{L}_N\right)^{-1} \\ &+ \frac{\sin(\frac{\alpha}{2}\pi)}{(1-\frac{\alpha}{2})\pi} \sum_{j=0}^{n} \omega_j^{(n)} \left(e^{-\zeta_j^{(n)}/(1-\frac{\alpha}{2})}\mathcal{I} + \mathcal{L}_N\right)^{-1}, \qquad m = 2n,\end{aligned} \tag{24}$$

where $\mathcal{L}_N = \mathcal{A}_{2,N}^{(k)}$. Moreover, mimicking the approximation approach, we also consider m_{tr} in place of m and define the preconditioner based on the truncated Gauss-Laguerre quadrature

$$\begin{aligned}\left(P_{m_{tr},N}^{GL,tr}\right)^{-1} :=& \frac{\sin(\frac{\alpha}{2}\pi)}{\frac{\alpha}{2}\pi} \sum_{j=0}^{n_{tr}} \omega_j^{(n)} \left(\mathcal{I} + e^{-2\zeta_j^{(n)}/\alpha}\mathcal{L}_N\right)^{-1} \\ &+ \frac{\sin(\frac{\alpha}{2}\pi)}{(1-\frac{\alpha}{2})\pi} \sum_{j=0}^{n_{tr}} \omega_j^{(n)} \left(e^{-\zeta_j^{(n)}/(1-\frac{\alpha}{2})}\mathcal{I} + \mathcal{L}_N\right)^{-1}, \qquad m_{tr} = 2n_{tr},\end{aligned} \tag{25}$$

where n_{tr} is defined in (15) with $\beta = \alpha/2$. The considerations in Remark 2 apply to these preconditioners as well. Of course, considered the same set of Gauss-Laguerre nodes, $P_{m_{tr},N}^{GL,tr}$ is expected to be less expensive than $P_{m,N}^{GL}$.

4.2.1 Parameter estimation considerations

In principle, also for Gauss-Laguerre approach we could use the error estimates in Section 3.2 to choose the parameter m. In this light, if we impose that the error behaves as $\mathrm{e}^{-\pi}$ according to the findings in Section 4.1.1 for the sinc quadrature, we obtain $m = 2$. This is also confirmed by Fig. 6 where we can compare the number of inversions needed by the sinc and the Gauss-Laguerre quadrature rules, fixed the error estimate as $e^{-\delta}$. However, since the Gauss-Laguerre approach involves two distinct integrals, we expect that choosing $m = 2$ is not enough to define a good preconditioner. The numerical experiments presented in the next section corroborate this behavior. Therefore, for $P_{m,N}^{GL}$ in (24) there is no clear criterion for selecting the value of m. The only observation we can make is that since m^* is already large near $\alpha = 1$, to maintain m sufficiently small, the estimate we use should rely on the second term in the formula (14). Thus, we can only provide a range of variation for m between 2 and m^* that is far too large, and limits the use of the Gauss-Laguerre preconditioner to a "by trial-and-error", progressively increasing m.

Hence, it could be convenient to use the truncated formula which, for the same level of error, allows us to reduce the number of inversions; see Fig. 6. We then consider $P_{m_{tr},N}^{GL,tr}$ from (25) with n_{tr} as defined in (15), where $\beta = \alpha/2$. Table 1 lists the first ten values of n_{tr} for $\alpha = 1.1, 1.2, 1.5$. Since multiple values of m could result in the same n_{tr}, to maximize approximation accuracy, one could select the largest pos-

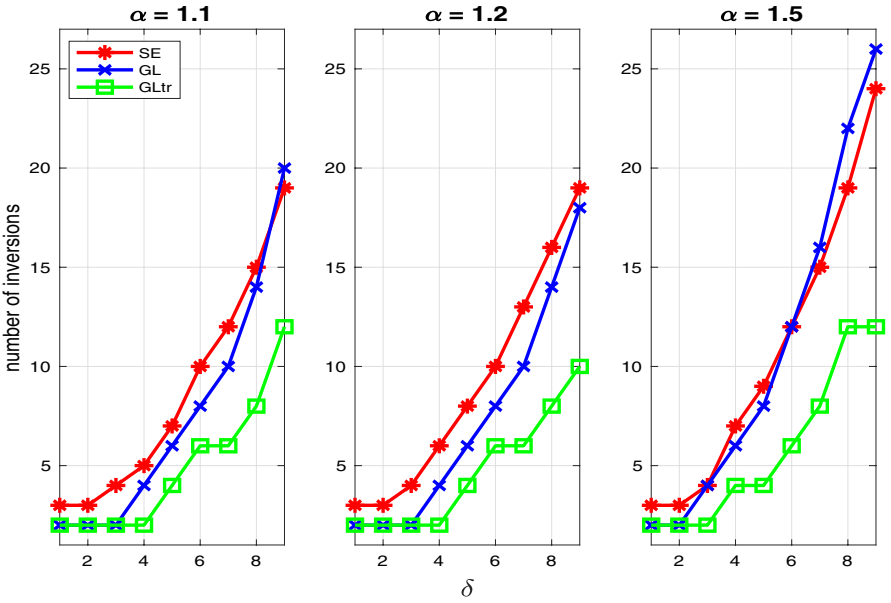

Fig. 6 Number of inversions vs the error estimate $e^{-\delta}$ for $\alpha = 2\beta$, with $\beta = 0.55, 0.6, 0.75$. for sinc (SE), Gauss-Laguerre (GL), truncated Gauss-Laguerre (GL,tr) quadratures

Table 1 Values of the quantity n_{tr} given in (15) for $m = 2, 4, 6, \ldots, 20$ with $\alpha = 2\beta$

α \ m	2	4	6	8	10	12	14	16	18	20	m^*
1.1	1	**1**	**2**	3	**3**	4	4	5	5	6	9.0377
1.2	1	**1**	**2**	3	**3**	4	4	5	5	6	18.2250
1.5	1	**1**	2	**2**	3	**3**	4	4	5	5	182.2500

The bold values are the ones used in the numeral tests shown in Table 3

sible m, as highlighted in bold in Table 1. However, this approach also corresponds to proceed by trial-and-error, without precise method to determine the optimal m in advance. On the other hand, if we reverse the procedure and select n_{tr} corresponding to m^*, we may end up with large values of m_{tr}. For example, when $\alpha = 1.1, 1.2, 1.5$, we obtain $m_{tr} = 6, 10, 58$, respectively. In other words, truncating the formula is advantageous because it reduces the computational cost, but the truncation parameter may still vary in a large range corresponding to a large number of inversions. As a result, in all the numerical tests presented in Section 5, we proceed by progressively increasing n_{tr}.

4.2.2 Spectral properties

Starting from (24), we can compute the symbol

$$p_{m,\alpha}^{GL}(\boldsymbol{\theta}) := \frac{\sin(\frac{\alpha}{2}\pi)}{\frac{\alpha}{2}\pi}\psi_2(\boldsymbol{\theta})\sum_{j=0}^{n}\omega_j^{(n)}\left(e^{-2\zeta_j^{(n)}/\alpha}+\psi_2(\boldsymbol{\theta})\right)^{-1} + \frac{\sin(\frac{\alpha}{2}\pi)}{(1-\frac{\alpha}{2})\pi}\psi_2(\boldsymbol{\theta})\sum_{j=0}^{n}\omega_j^{(n)}\left(1+e^{-\zeta_j^{(n)}/(1-\frac{\alpha}{2})}\psi_2(\boldsymbol{\theta})\right)^{-1}, \tag{26}$$

that, as $\tilde{n} \to +\infty$, provides an estimate for the eigenvalues of $P_{m,N}^{GL}$. The symbol $p_{m_{tr},\alpha}^{GL,tr}$ for the truncated case is similarly obtained replacing n with n_{tr}.

For $k = 1$, Figure 7 illustrates the accuracy of the approximation of ψ_α provided by $p_{m,\alpha}^{GL}$ and $p_{m_{tr},\alpha}^{GL,tr}$ when $\alpha = 1.1$. The behavior of the two symbols is not particularly satisfactory for the considered values of m and m_{tr}. This can be attributed to the fact that, in the Gauss-Laguerre approach, we perform two separate quadratures on the positive and negative semi-axes, using only a few points for each when m is small. For this same reason truncation does not lead to a real improvement, and may even worsen the results.

4.3 Spectral comparisons

A comparison of the three symbols in (19), (23) and (26) provides insights into the performance of the corresponding preconditioners. In this context, we first recall that all three symbols, for a fixed number of addends, exhibit a zero at the origin of order two due to the presence of $\psi_2(\boldsymbol{\theta})$. Meanwhile, the symbol of the coefficient matrix $\psi_\alpha(\boldsymbol{\theta})$ has a zero at the origin of order α. We then follow this reasoning: we first examine the behavior of ratio $p_{m,\alpha}^{quad}(\boldsymbol{\theta})/\psi_2(\boldsymbol{\theta})$, $quad = GJ, SE, GL$ at the origin and as the number of addends increases; we then establish a hierarchy among the symbols at the origin based on this behavior. Let us define the following quantities

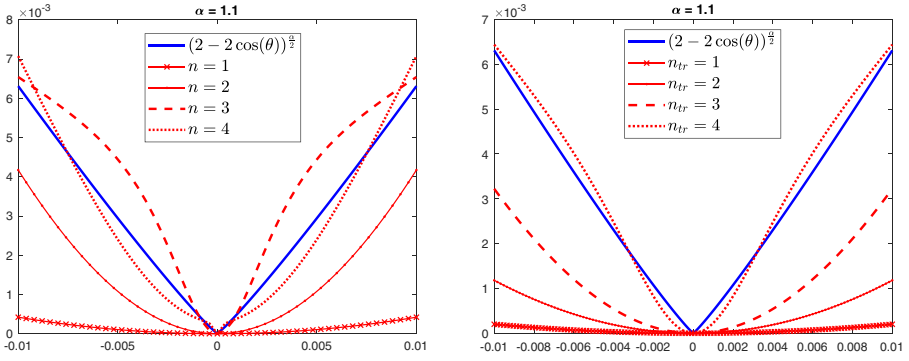

Fig. 7 Fixed $\alpha = 1.1$, comparison of $(2-2\cos(\theta))^{\frac{\alpha}{2}}$ (blue line) with $p_{m,\alpha}^{GL}$ for $m = 2n = 2, 4, 6, 8$ (left) and with $p_{m_{tr},\alpha}^{GL,tr}$ for $m_{tr} = 2n_{tr} = 2, 4, 6, 8$, corresponding to $m = 4, 6, 10, 12$

$$p_0^{GJ} := \frac{2\sin(\frac{\alpha}{2}\pi)}{\pi}\tau^{\frac{\alpha}{2}-1}\sum_{j=1}^{m}\frac{\tilde{w}_j}{1-\tilde{\vartheta}_j}, \tag{27}$$

$$p_0^{SE} \overset{\mu=2}{:=} 2\frac{\sin(\frac{\alpha}{2}\pi)}{\pi}h\sum_{\ell=-M_2}^{M_1} e^{2(1-\frac{\alpha}{2})\ell h}, \tag{28}$$

$$p_0^{GL} := \frac{\sin(\frac{\alpha}{2}\pi)}{\frac{\alpha}{2}\pi}\sum_{j=0}^{n}\omega_j^{(n)}e^{2\zeta_j^{(n)}/\alpha} + \frac{\sin(\frac{\alpha}{2}\pi)}{(1-\frac{\alpha}{2})\pi}\sum_{j=0}^{n}\omega_j^{(n)}. \tag{29}$$

We first note that the sum in (27) is the *m*-nodes Gauss-Jacobi quadrature of the improper integral

$$\int_{-1}^{1}(1-x)^{\alpha/2-1}(1+x)^{\alpha/2}\frac{1}{1-x}\mathrm{d}x = \lim_{t\to 1^-}\frac{1}{2-\alpha}\left(\frac{1+t}{1-t}\right)^{1-\alpha/2}$$
$$\underset{y=\frac{1}{1-t}}{=} \lim_{y\to+\infty}\frac{1}{2-\alpha}(2y-1)^{1-\alpha/2}.$$

As a consequence, we can assert that

$$p_0^{GJ} \sim \frac{2\sin(\frac{\alpha}{2}\pi)}{\pi}\frac{1}{2-\alpha}\left(\frac{2y-1}{\tau}\right)^{1-\alpha/2}, \quad y\to+\infty.$$

Similarly, *h* times the sum in (28) is the sinc-quadrature of the improper integral

$$\int_{-\infty}^{+\infty}\mathrm{e}^{(2-\alpha)x}\mathrm{d}x = \lim_{y\to+\infty}\frac{\mathrm{e}^{(2-\alpha)y}-\mathrm{e}^{-(2-\alpha)y}}{2-\alpha}$$
$$= \frac{2}{2-\alpha}\lim_{y\to+\infty}\sinh(2-\alpha)y$$

that then implies

$$p_0^{SE} \sim \frac{2\sin(\frac{\alpha}{2}\pi)}{\pi}\frac{2}{2-\alpha}\sinh(2-\alpha)y, \quad y\to+\infty.$$

Finally, the two sums in (29) correspond to the *n*-nodes Gauss-Laguerre quadratures of $\mathrm{e}^{2x/\alpha}$ and of 1, that is

$$\int_0^{+\infty}\mathrm{e}^{(2/\alpha-1)x}\mathrm{d}x = \frac{\alpha}{2-\alpha}\lim_{y\to+\infty}\mathrm{e}^{(2/\alpha-1)y}-1$$
$$\int_0^{+\infty}\mathrm{e}^{-x}\mathrm{d}x = \lim_{y\to+\infty}1-\mathrm{e}^{-y}$$

leading to

$$p_0^{GL} \sim \frac{2\sin(\frac{\alpha}{2}\pi)}{\pi}\frac{1}{2-\alpha}\left(\mathrm{e}^{(2/\alpha-1)y}-\mathrm{e}^{-y}\right), \quad y\to+\infty.$$

Now, for the case of interest, that is $\alpha \to 1$, we have that for $y \to +\infty$

$$p_0^{GJ} \sim \frac{2}{\pi}\sqrt{\frac{2y-1}{\tau}}, \qquad p_0^{SE} \sim \frac{4}{\pi}\sinh y, \qquad p_0^{GL} \sim \frac{4}{\pi}\sinh y. \tag{30}$$

This analysis suggests that for a fixed number of addends and consequently for a fixed y, all the symbols behave near the origin as $\boldsymbol{\theta}^2 \cdot c_y$, where c_y is a constant that increases positively with y according to the hierarchy in (30), that is as a square-root for Gauss-Jacobi quadrature, and as an hyperbolic sine for sinc and Gauss-Laguerre quadratures. Notably, in the case of Gauss-Jacobi quadrature, the behavior can be modulated via the parameter τ.

From these results, we expect that both $P_{m,N}^{SE}$ and $P_{m,N}^{GL}$ exhibit superior convergence properties compared to $P_{m,N}^{GJ}$, whose performance is significantly influenced by the choice of τ. This aligns with the findings in previous studies [8].

5 Numerical experiments

In this section, we assess the performance of the newly defined preconditioners $P_{m,N}^{SE}$, $P_{m,N}^{GL}$, $P_{m_{tr},N}^{GL,tr}$ (referred to as "SE", "GL", and "truncated GL", respectively) and compare them with $P_{m,N}^{GJ}$ (briefly, "GJ") for solving the linear system using preconditioned conjugate gradient (PCG)

$$\mathcal{A}_{\alpha,N}^{(k)}\boldsymbol{u} = D_N^{-1}\boldsymbol{b}, \quad k=1,2,$$

where D_N is the diagonal matrix that discretizes $D(\boldsymbol{x})$ in (16), and $\boldsymbol{b}$ is the vector that discretizes the source term $f(\boldsymbol{x})$.

The weight and the nodes of both Gauss-Jacobi and Gauss-Laguerre quadratures are computed by means of the Chebfun functions `jacpts.m` and `lagpts.m`; see Hale and Trefethen [35].

All preconditioners are implemented by solving the linear systems associated to the shifted Laplacian matrices by the Thomas algorithm from Donatelli et al. [4] for the one-dimensional case ($k=1$), and by the multigrid method from Moghaderi et al. [1] in the two-dimensional case ($k=2$). This is the same choice made in [8], motivated by the aim of enabling a direct comparison between the new rational proposals and the Gauss-Jacobi preconditioner introduced therein. Both methods achieve linear computational cost. Of course, other methods to solve the 2D-Laplacian matrices exist and can be combined with our preconditioners as well.

For all tests, the tolerance of the PCG method is fixed at 10^{-7}, and the initial guess is set to the zero vector.

The numerical results are presented separately for the 1D and 2D cases in the following subsections.

5.1 1D test

Let $D(x) = \Gamma(3-\alpha)(1+x)^{\alpha}$ and $f(x) = 1$, $x \in [0, 10]$. For this 1D example we compare the performance of different preconditioners in terms of clustering of the eigenvalues for the resulting preconditioned matrices and in terms of CPU times. Concerning the parameters choice, based on the discussion in Section 4.1.1 we fix $m = 4$ for SE ensuring for GJ, GL, and truncated GL the same number of inversions. A detailed discussion on how the GJ and GL preconditioners behave with respect to the related parameters is postponed to the 2D tests.

Figure 8 illustrates the eigenvalues of the preconditioned matrices for all the considered preconditioners. SE and GJ preconditioners show an almost α-independent clustering compared to GL preconditioners that exhibit less clustered and possibly dangerously close to zero eigenvalues for truncated GL preconditioner. The cluster provided by the SE preconditioner is tighter compared to those given by the GJ and GL preconditioners. This behavior suggests better convergence properties rates for the SE preconditioner, as confimed in Fig. 9 where we compare the computational time required by the different preconditioners as a function of the problem size N, with $N = 2^{10} - 1, \ldots, 2^{14} - 1$. The truncated GL preconditioner is omitted due to its poor clustering properties. The left panel displays the performance of SE and GJ for $m = 4$, showing that SE reduces computation time for small α, particularly at $\alpha \approx 1$. The right panel compares GL and GJ for $m = 4$, indicating that GL does not provide a substantial computational benefit over GJ in 1D problems.

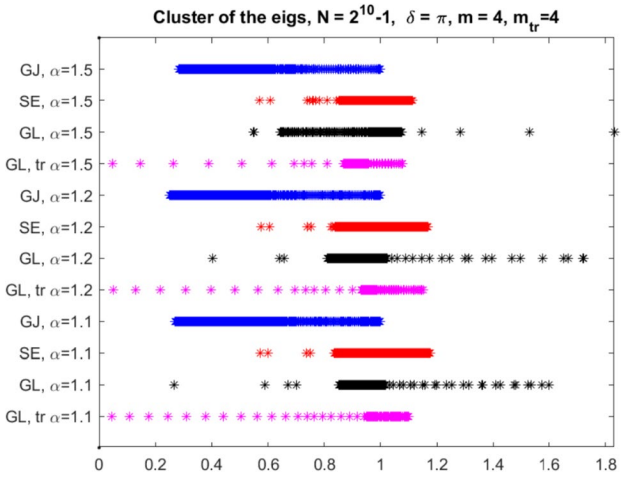

Fig. 8 Cluster of the eigenvalues for the preconditioned matrices with GJ, SE, GL and truncated GL preconditioners

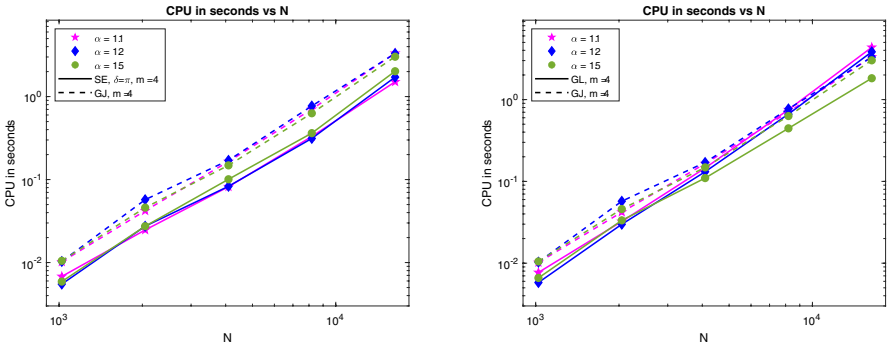

Fig. 9 1D test: CPU time in seconds vs N for SE and GJ preconditioner (left), and for GL and GJ preconditioners (right) with $m = 4$

Overall, these numerical results confirm that in 1D, SE is the most efficient preconditioner for small α, while GL does not offer significant advantages.

We conclude this section by comparing the SE preconditioner with the τ-preconditioner proposed in Huang et al. [5] when $\alpha = 1.1$. In the case of the finite difference scheme on a uniform grid described in Section 4, the SE preconditioner requires slightly more iterations than the τ-preconditioner. However, the overall CPU times remain comparable; see Fig. 10 for a 1D example. This is expected as the cost per iteration of our method is $O(N)$ with a constant depending on m (here $m = 4$), while the τ-preconditioner involves a sine transform of cost $O(N \log N)$, but the lower iteration count of the τ-preconditioner compensates for its higher per-iteration cost, leading to similar total running times.

On the other hand, and consistently with the discussion in Remark 2, the main advantage of our preconditioners is that they remain effective even when fast transforms are no longer applicable. This occurs, for instance, when the discretization of the Riesz operator is obtained via finite differences on a nonuniform grid obtained by mapping the uniform grid through the function $1 - \cos(\frac{\pi}{2}x)$. Crucially, although the Toeplitz structure of the coefficient matrix and the diagonalizability of the precon-

N	Iterations	
	SE	τ
127	9	6
255	9	7
511	10	8
1023	10	8
2047	10	8
4095	11	9
8191	11	9
16383	13	9

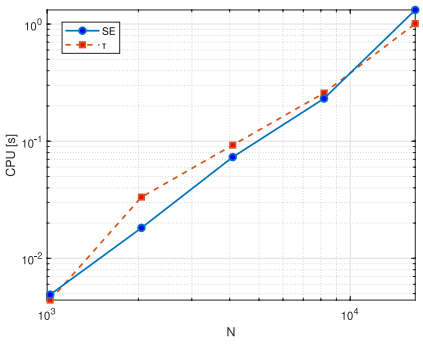

Fig. 10 1D test: iterations (left panel) and CPU time in seconds (right panel) vs N for SE ($m = 4$) and τ preconditioners in case of the finite difference on a uniform grid. Here $\alpha = 1.1$

ditioner via the sine transform are lost, the shifted Laplacian matrices that must be solved within the SE preconditioner remain tridiagonal, so the cost per iteration stays linear in N. Figure 11 reports the performance of the SE preconditioner compared with the τ-preconditioner from Huang et al. [5] in terms of both iteration counts and CPU times. These results clearly show that the τ-preconditioner loses effectiveness when the matrix no longer has a Toeplitz structure, whereas our approach is structure-independent. Overall, the take-home message is that, whenever an efficient solver for the shifted Laplacian systems is available, the SE preconditioner inherits the same favorable performance.

5.2 2D test

Let $D(\boldsymbol{x}) = \Gamma(3-\alpha)(1+x_1)^{\alpha}(1+x_2)^2$ and $f(\boldsymbol{x}) = 1$, $\boldsymbol{x} \in [0,10]^2$. The performance of the preconditioners in 2D both in terms of iteration number and CPU times is summarized in Tables 2, 3 and 4.

Table 2 reports the number of iterations and CPU times for the PCG method using the Gauss-Jacobi preconditioner with different values of m. The results confirm that increasing m reduces iterations, but with diminishing returns beyond $m = 4$.

Table 3 presents results for the GL preconditioner with different values of m and its truncated version with n_{tr} as in Table 1. Unlike in 1D, GL achieves better performance in 2D compared to GJ, demonstrating lower CPU times for $m > 2$. This is consistent with the fact that the condition number behaves as in (18), i.e., independently of the dimensionality of the problem, which then means that in 2D the matrices are expected to be better conditioned than in 1D. About the truncated GL variant, for $n_{tr} > 1$ it slightly improves the number of iterations and CPU times of GL, making it a viable alternative for reducing computational cost compared to GJ preconditioner although we do not know how to choose n_{tr} a priori.

Table 4 compares the best of GJ and GL preconditioners with SE and Laplacian preconditioners. In light of these numerical results, we can conclude that in 2D, the GL preconditioner becomes a valid alternative to the GJ preconditioner, especially in its truncated form, where it maintains efficiency while reducing computational over-

N	Iterations	
	SE	τ
127	21	41
255	22	61
511	23	86
1023	26	123
2047	28	188
4095	31	306
8191	36	403
16383	43	643

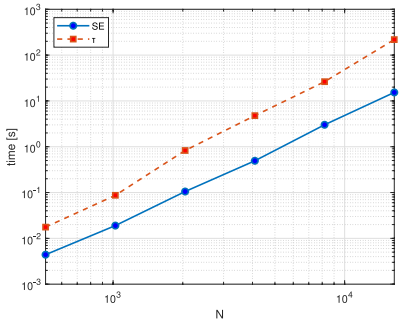

Fig. 11 1D test: iterations (left panel) and CPU time in seconds (right panel) vs N for SE ($m = 4$) and τ-preconditioners in case of finite differences on a nonuniform grid obtained as a mapping of the uniform one by means of the function $1 - \cos(\frac{\pi}{2}x)$. Here $\alpha = 1.1$

Table 2 2D test: Number of iterations, execution time (in seconds), by PCG with GJ preconditioner with $m = 1, 2, \ldots, 6$

α	N	GJ $m=1$		GJ $m=2$		GJ $m=3$		GJ $m=4$		GJ $m=5$		GJ $m=6$	
		It	CPU	It	CPU	It	CPU	It	CPU	It	CPU	It	CPU
1.1	961	15	0.0015	11	**0.0014**	10	0.0025	10	0.0029	10	0.0038	10	0.0041
	3969	18	**0.0178**	14	0.0240	12	0.0316	11	0.0387	10	0.0387	10	0.0465
	16129	22	**0.1272**	16	0.1401	14	0.1660	12	0.1708	11	0.1931	11	0.2214
	65025	27	0.9645	20	0.8725	16	**0.7867**	15	0.8590	13	0.8713	12	0.8748
	261121	33	7.7421	24	6.5803	20	5.6520	17	5.6043	16	5.5972	15	**5.5209**
1.2	961	15	**0.0012**	11	0.0015	10	0.0020	10	0.0027	10	0.0038	10	0.0042
	3969	18	**0.0186**	14	0.0258	12	0.0328	11	0.0355	10	0.0390	10	0.0462
	16129	22	**0.1247**	17	0.1443	14	0.1559	13	0.1902	12	0.2076	11	0.2259
	65025	27	0.9252	20	0.8825	17	**0.8440**	15	0.8865	14	0.9295	13	0.9938
	261121	32	7.9145	24	6.5466	20	**5.5719**	18	6.0585	17	5.9036	15	5.6876
1.5	961	13	**0.0011**	11	0.0014	10	0.0021	10	0.0028	9	0.0034	9	0.0050
	3969	16	**0.0179**	13	0.0233	12	0.0304	11	0.0348	10	0.0397	10	0.0583
	16129	19	**0.1123**	15	0.1318	14	0.1671	13	0.2010	12	0.2202	12	0.2803
	65025	22	**0.7508**	18	0.7902	16	0.8641	15	0.8825	14	0.9877	14	1.2475
	261121	25	6.1013	21	**5.7565**	19	5.8604	18	6.0080	17	6.9005	16	6.8828

Table 3 2D test: Number of iterations, execution time (in seconds), by PCG with GL preconditioner with $m = 2n = 2, 4, 6$ and truncated GL preconditioner with $n_{tr} = 1, 2, 3$ that is $m = 4, 6, 10$ for $\alpha = 1.1, 1.2$, and $m = 4, 8, 12$ for $\alpha = 1.5$

α	N	GL $m = 2$		GL $m = 4$		GL $m = 6$		GL $n_{tr} = 1$		GL $n_{tr} = 2$		GL $n_{tr} = 3$	
		It	CPU	It	CPU	It	CPU	It	CPU	It	CPU	It	CPU
1.1	961	10	**0.0017**	10	0.0028	10	0.0045	11	0.0018	10	0.0025	10	0.0035
	3969	12	0.0219	10	0.0345	10	0.0460	16	**0.0096**	10	0.0112	10	0.0143
	16129	17	0.1427	12	0.1723	11	0.2102	23	0.1085	11	**0.0735**	11	0.0892
	65025	25	1.0536	14	0.7925	12	0.8874	35	1.1206	13	**0.5476**	11	0.5601
	261121	38	10.5970	15	4.8881	13	4.8601	52	13.9977	17	5.4344	12	**4.4529**
1.2	961	10	**0.0014**	10	0.0027	10	0.0042	11	0.0016	10	0.0025	10	0.0038
	3969	12	0.0212	10	0.0315	10	0.0458	16	0.0103	10	**0.0122**	10	0.0160
	16129	17	0.1457	11	0.1543	11	0.2104	25	0.1203	11	**0.0742**	11	0.0973
	65025	26	1.0884	13	0.7714	12	0.8688	39	1.3201	12	**0.5198**	11	0.5736
	261121	40	10.3741	16	5.3927	13	5.2240	59	16.0663	15	4.7376	12	**4.3941**
1.5	961	11	**0.0016**	10	0.0028	10	0.0064	11	0.0018	9	0.0022	10	0.0037
	3969	12	0.0209	10	0.0315	10	0.0464	14	0.0097	10	**0.0122**	10	0.0162
	16129	14	0.1246	11	0.1552	11	0.2129	24	0.1098	10	**0.0668**	10	0.0883
	65025	21	0.8982	12	0.6815	12	0.8687	40	1.4000	11	**0.5273**	11	0.5528
	261121	35	9.6134	13	4.2183	13	5.0956	69	18.2878	13	4.3241	11	**4.1266**

Table 4 2D test: Number of iterations, execution time (in seconds), by PCG with the best of GJ ($m = 6$), GL ($m = 6$), truncated GL ($n_{tr} = 3$), SE preconditioner with $\delta = \pi$, $M_1 = 1, M_2 = 2, m = 4$ and Laplacian

α	N	GJ $m = 6$		GL $n_{tr} = 3$		GL $m = 6$		SE $m = 4$		Lapl	
		It	CPU	It	CPU	It	CPU	It	CPU	It	CPU
1.1	961	10	0.0041	10	0.0035	10	0.0045	10	0.0046	22	**0.0023**
	3969	10	0.0465	10	**0.0143**	10	0.0460	11	0.0445	29	0.0154
	16129	11	0.2214	11	**0.0892**	11	0.2102	11	0.1928	41	0.1802
	65025	12	0.8748	11	**0.5601**	12	0.8874	12	0.8034	54	1.7093
	261121	15	5.5209	12	4.4529	13	4.8601	12	**4.2697**	74	17.0342
1.2	961	10	0.0042	10	0.0038	10	0.0042	10	0.0031	20	**0.0016**
	3969	10	0.0462	10	0.0160	10	0.0458	11	0.0353	27	**0.0158**
	16129	11	0.2259	11	**0.0973**	11	0.2104	11	0.1509	35	0.1571
	65025	13	0.9938	11	**0.5736**	12	0.8688	12	0.7237	47	1.4527
	261121	15	5.6876	12	4.3941	13	5.2240	12	**3.9573**	61	14.9287
1.5	961	9	0.0050	10	0.0037	10	0.0064	10	0.0028	16	**0.0013**
	3969	10	0.0583	10	0.0162	10	0.0464	10	0.0324	19	**0.0113**
	16129	12	0.2803	10	**0.0883**	11	0.2129	11	0.1574	23	0.1042
	65025	14	1.2475	11	**0.5528**	12	0.8687	11	0.6620	28	0.9479
	261121	16	6.8828	11	4.1266	13	5.0956	11	**3.7210**	33	8.1894

head, and that the SE preconditioner at the largest considered dimension is the most competitive option for small α. In this case, unlike for the truncated GL variant, we do know how to choose m based on the error analysis.

A key observation from these experiments is that the SE preconditioner ensures optimality in the sense that the number of iterations does not increase with problem size. A similar, albeit less stable, trend is observed for GL, where iteration counts remain relatively stable as the dimension increases. In contrast, for the GJ preconditioner, it is evident that achieving optimality requires increasing m as the problem size grows. Finally, note that, as expected, for the considered values of α, all rational preconditioners outperform the Laplacian preconditioner.

Similar to the 1D case, also in this 2D test the performances of the SE preconditioner ($m = 4$) are comparable to the one of the τ preconditioner in Huang et al. [5] on a uniform grid in terms of CPU times while it requires a slight higher number of iterations; see Fig. 12 where $\alpha = 1.1$. As a by-product of the versatility of the SE preconditioner, Figure 13 shows that its performances are satisfactory also in the case of non-uniform meshes where the τ preconditioner use is intrinsically limited by its multi-level Toeplitz nature. The obtained results look promising and a more in-depth investigation of the non-uniform case will be subject of future research.

We conclude our numerical tests in the 2D case, comparing, again in case of a uniform mesh, the SE preconditioner with the preconditioner obtained from the $(m - 1, m)$ min-max rational approximation, where the poles are computed using the `minimax.m` function from the Chebfun library. The choice of the $(m - 1, m)$ approximant ensures consistency with the one obtained through SE. The results for $m = 4$, shown in Fig. 14, illustrate that the two preconditioners require nearly the same number of iterations. Since the iteration counts are very close for both methods, further comparisons, such as CPU time measurements, are unnecessary. It is worth

N	Iterations	
	SE	τ
961	10	7
3969	11	7
16129	11	7
65025	12	8
261121	12	8

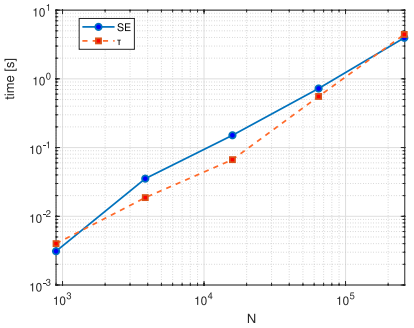

Fig. 12 2D test: Iterations (left panel) and CPU time in seconds (right panel) vs N for SE ($m = 4$) and τ preconditioners in case of finite differences on a uniform grid. Here $\alpha = 1.1$

N	Iterations	
	SE	τ
961	23	43
3969	25	67
16129	27	104
65025	31	162
261121	40	247

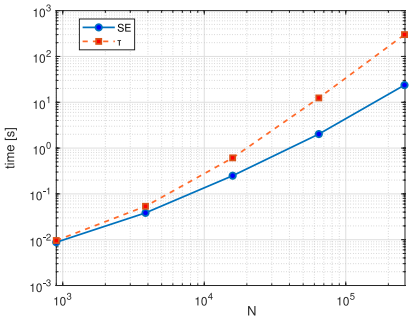

Fig. 13 2D test: Iterations (left panel) and CPU time in seconds (right panel) vs N for SE ($m = 4$) and τ preconditioners in case of finite differences on a non-uniform cartesian grid obtained in each direction as a mapping of the uniform one by means of the function $1 - \cos(\frac{\pi}{2}x)$. Here $\alpha = 1.1$

noting that the min-max preconditioner relies on an accurate approximation of the spectrum of the matrix $\mathcal{L}_N$. While this is straightforward in this particular case, it may increase the computational cost in general. On the other hand, SE achieves similar performance without requiring explicit spectral knowledge.

A potential alternative to our approach is given by rational Krylov-type methods [36] like the one in Massei and Robol [18], which employs a rational Krylov technique to solve linear systems where the coefficient matrix is a fractional power of a Laplacian. This method is particularly effective in its intended setting, where the system matrix is derived as a function of a Kronecker product structure. However, a direct comparison with our preconditioning approach is not entirely feasible. In our case, the coefficient matrix is not a fractional power of a Laplacian; rather, we use rational approximations of such operators as preconditioners for more general systems. Comparing the two approaches would therefore imply using the rational Krylov method itself as a preconditioner, which might be pointless from a computational cost perspective. Similar considerations apply when taking into account the reduced conjugate gradient basis method proposed in Li et al. [19].

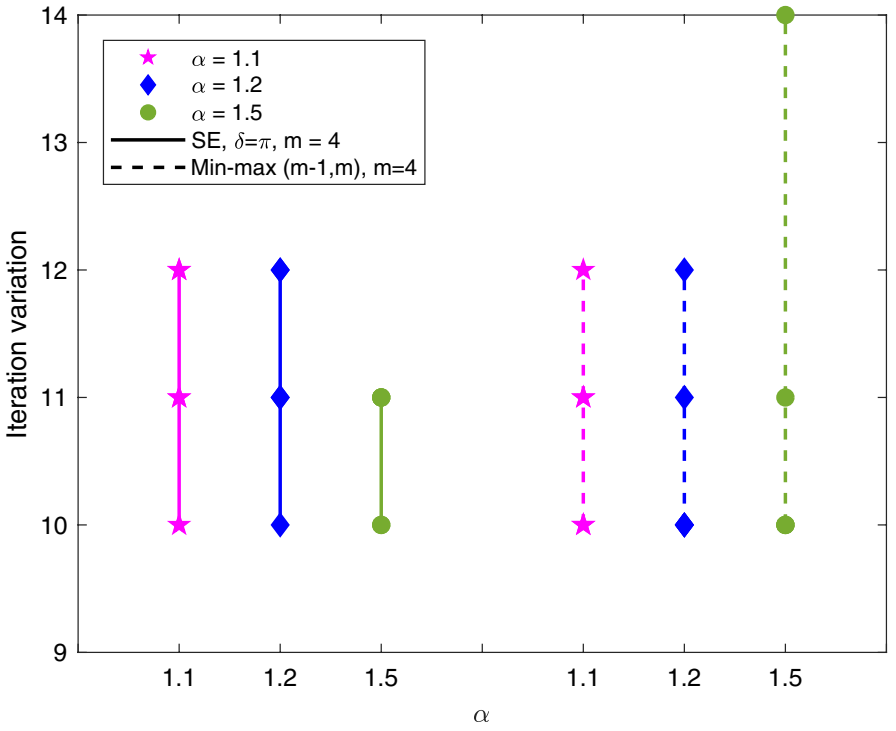

Fig. 14 2D test: Iteration variation for SE preconditioner with $\delta = \pi$, $m = 4$ (solid line) and for min-max rational approximation with degree $(m-1, m)$, $m = 4$ preconditioner (dashed line) when $N = (2^k - 1)^2$, $k = 5, \ldots, 9$

In conclusion, these findings highlight the effectiveness of rational preconditioners, particularly the sinc-based approach, in efficiently solving fractional diffusion problems, especially when α is close to 1. However, we emphasize that our study primarily focused on α close to 1, as the Laplacian preconditioner remains the most effective choice for $\alpha \approx 2$.

6 Conclusions and future perspectives

In this work, we have extended the rational preconditioning approach for the Riesz operator $-(-\Delta)^{\frac{\alpha}{2}}$, $\alpha \in (1, 2]$, introduced in [8] by defining new preconditioning strategies based on sinc and Gauss-Laguerre quadratures.

A crucial aspect that emerged from our analysis is the discrepancy between approximation accuracy and preconditioning effectiveness. While Gauss-Laguerre quadrature provides a more precise approximation of the fractional power operator, this does not necessarily translate into better preconditioning performance. In fact, an effective preconditioner must not only approximate the operator well but also minimize the number of inversions required to maintain a stable iteration count as

the problem size grows. From this perspective, sinc quadrature emerges as the best choice, as it strikes an optimal balance between approximation quality and computational efficiency, allowing for an a priori selection of parameters based on error analysis. In contrast, the Gauss-Jacobi preconditioner requires an increase in the parameter to sustain acceptable performance as the problem size increases. These results suggest that using preconditioners based on exponentially convergent quadratures represents a significant step forward in efficiently solving fractional diffusion problems, particularly for values of α close to 1, with a special quote to SE that can be used choosing its parameter in advance, while the GL preconditioner can only be used by trial-and-error.

Looking ahead, a promising research direction is extending the proposed methods to more complex fractional models, such as an anisotropic Riesz operator with different variable coefficients in the different directions or problems defined on irregular domains where the introduction of non-uniform grids is often mandatory. Another potential avenue could be the integration of these preconditioners with rational Krylov or reduced conjugate gradient methods to explore whether a combined approach can further enhance computational efficiency.

Acknowledgements The authors are members of the research group *Gruppo Nazionale per il Calcolo Scientifico*, *Istituto Nazionale di Alta Matematica* (GNCS-INdAM).

Author Contributions All authors contributed equally to this work

Funding Open access funding provided by Università degli Studi di Roma Tor Vergata within the CRUI-CARE Agreement. This work was partially supported by the "INdAM - GNCS Project", code CUP_E53C23001670001. The second author acknowledges the MUR Excellence Department Project MatMod@TOV awarded to the Department of Mathematics, University of Rome Tor Vergata, CUP E83C23000330006. Her work was partially granted by the Italian Ministry of University and Research (MUR) through the PRIN 2022-PNRR "MATHPROCULT: MATHematical tools for predictive maintenance and PROtection of CULTtural heritage" (CUP J53D23015940001).

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of Interest The authors declare that they have no conflict of interest.

Competing Interests The authors declare no competing interests.

References

1. Moghaderi, H., Dehghan, M., Donatelli, M., Mazza, M.: Spectral analysis and multigrid preconditioners for two-dimensional space-fractional diffusion equations. J. Comput. Phys. **350**, 992–1011 (2017). https://doi.org/10.1016/j.jcp.2017.08.064
2. Pang, H.-K., Sun, H.-W.: Multigrid method for fractional diffusion equations. J. Comput. Phys. **231**(2), 693–703 (2012). https://doi.org/10.1016/j.jcp.2011.10.005
3. Lei, S.-L., Sun, H.-W.: A circulant preconditioner for fractional diffusion equations. J. Comput. Phys. **242**, 715–725 (2013). https://doi.org/10.1016/j.jcp.2013.02.025
4. Donatelli, M., Mazza, M., Serra-Capizzano, S.: Spectral analysis and structure preserving preconditioners for fractional diffusion equations. J. Comput. Phys. **307**, 262–279 (2016). https://doi.org/10.1016/j.jcp.2015.11.061
5. Huang, X., Lin, X.-L., Ng, M.K., Sun, H.-W.: Spectral analysis for preconditioning of multi-dimensional Riesz fractional diffusion equations. Numer. Math. Theory Methods Appl. **15**(3), 565–591 (2022). https://doi.org/10.4208/nmtma.oa-2022-0032
6. Lin, F.-R., Yang, S.-W., Jin, X.-Q.: Preconditioned iterative methods for fractional diffusion equation. J. Comput. Phys. **256**, 109–117 (2014). https://doi.org/10.1016/j.jcp.2013.07.040
7. Donatelli, M., Krause, R., Mazza, M., Trotti, K.: Multigrid preconditioners for anisotropic space-fractional diffusion equations. Adv. Comput. Math. **46**(3), 49–31 (2020). https://doi.org/10.1007/s10444-020-09790-2
8. Aceto, L., Mazza, M.: A rational preconditioner for multi-dimensional Riesz fractional diffusion equations. Comput. Math. Appl. **143**, 372–382 (2023). https://doi.org/10.1016/j.camwa.2023.05.016
9. Ilic, M., Liu, F., Turner, I., Anh, V.: Numerical approximation of a fractional-in-space diffusion equation. I. Fract. Calc. Appl. Anal. **8**(3), 323–341 (2005)
10. Komatsu, H.: Fractional powers of operators. Pacific J. Math. **19**, 285–346 (1966)
11. Harizanov, S., Lazarov, R., Margenov, S., Marinov, P., Pasciak, J.: Comparison analysis of two numerical methods for fractional diffusion problems based on the best rational approximations of t^γ on $[0, 1]$. In: Advanced Finite Element Methods with Applications. Lecture Notes in Computational Science and Engineering, vol. 128, pp. 165–185. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-14244-5_9
12. Harizanov, S., Lazarov, R., Margenov, S., Marinov, P., Pasciak, J.: Analysis of numerical methods for spectral fractional elliptic equations based on the best uniform rational approximation. J. Comput. Phys. **408**, 109285–21 (2020). https://doi.org/10.1016/j.jcp.2020.109285
13. Harizanov, S., Lazarov, R., Margenov, S., Marinov, P., Vutov, Y.: Optimal solvers for linear systems with fractional powers of sparse SPD matrices. Numer. Linear Algebra Appl. **25**(5), 2167–24 (2018). https://doi.org/10.1002/nla.2167
14. Harizanov, S., Margenov, S.: Positive approximations of the inverse of fractional powers of SPD M-matrices. In: Control Systems and Mathematical Methods in Economics. Lecture Notes in Econom. and Math. Systems, vol. 687, pp. 147–163. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-75169-6_8
15. Vabishchevich, P.N.: Numerically solving an equation for fractional powers of elliptic operators. J. Comput. Phys. **282**, 289–302 (2015). https://doi.org/10.1016/j.jcp.2014.11.022
16. Hofreither, C.: A unified view of some numerical methods for fractional diffusion. Comput. Math. Appl. **80**(2), 332–350 (2020). https://doi.org/10.1016/j.camwa.2019.07.025
17. Aceto, L., Bertaccini, D., Durastante, F., Novati, P.: Rational Krylov methods for functions of matrices with applications to fractional partial differential equations. J. Comput. Phys. **396**, 470–482 (2019). https://doi.org/10.1016/j.jcp.2019.07.009
18. Massei, S., Robol, L.: Rational Krylov for Stieltjes matrix functions: convergence and pole selection. BIT Numer. Math. **61**(1), 237–273 (2021). https://doi.org/10.1007/s10543-020-00826-z
19. Li, Y., Zikatanov, L., Zuo, C.: A reduced conjugate gradient basis method for fractional diffusion. SIAM J. Sci. Comput. **46**(5), 68–87 (2024). https://doi.org/10.1137/23M1575913
20. Aceto, L., Novati, P.: Rational approximation to the fractional Laplacian operator in reaction-diffusion problems. SIAM J. Sci. Comput. **39**(1), 214–228 (2017). https://doi.org/10.1137/16M1064714
21. Aceto, L., Novati, P.: Efficient implementation of rational approximations to fractional differential operators. J. Sci. Comput. **76**(1), 651–671 (2018). https://doi.org/10.1007/s10915-017-0633-2
22. Aceto, L., Novati, P.: Rational approximations to fractional powers of self-adjoint positive operators. Numer. Math. **143**(1), 1–16 (2019). https://doi.org/10.1007/s00211-019-01048-4

23. Aceto, L., Novati, P.: Exponentially convergent trapezoidal rules to approximate fractional powers of operators. J. Sci. Comput. **91**(2), 55–18 (2022). https://doi.org/10.1007/s10915-022-01837-4
24. Aceto, L., Novati, P.: Fast and accurate approximations to fractional powers of operators. IMA J. Numer. Anal. **42**(2), 1598–1622 (2022). https://doi.org/10.1093/imanum/drab002
25. Bonito, A., Lei, W., Pasciak, J.E.: On sinc quadrature approximations of fractional powers of regularly accretive operators. J. Numer. Math. **27**(2), 57–68 (2019). https://doi.org/10.1515/jnma-2017-0116
26. Bonito, A., Pasciak, J.E.: Numerical approximation of fractional powers of elliptic operators. Math. Comp. **84**(295), 2083–2110 (2015). https://doi.org/10.1090/S0025-5718-2015-02937-8
27. Casulli, A., Robol, L.: Low-rank tensor structure preservation in fractional operators by means of exponential sums. BIT **63**(2), 30–26 (2023). https://doi.org/10.1007/s10543-023-00974-y
28. Vabishchevich, P.N.: Numerical solution of time-dependent problems with fractional power elliptic operator. Comput. Methods Appl. Math. **18**(1), 111–128 (2018). https://doi.org/10.1515/cmam-2017-0028
29. Vabishchevich, P.N.: Approximation of a fractional power of an elliptic operator. Numer. Linear Algebra Appl. **27**(3), 2287–14 (2020). https://doi.org/10.1002/nla.2287
30. Crouzeix, M., Palencia, C.: The numerical range is a $(1+\sqrt{2})$ -spectral set. SIAM J. Matrix Anal. Appl. **38**(2), 649–655 (2017). https://doi.org/10.1137/17M1116672
31. Lund, J., Bowers, K.L.: Sinc Methods for Quadrature and Differential Equations, p. 304. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA (1992). https://doi.org/10.1137/1.9781611971637
32. Aceto, L., Mazza, M., Serra-Capizzano, S.: Fractional Laplace operator in two dimensions, approximating matrices, and related spectral analysis. Calcolo **57**(3), 1–25 (2020). https://doi.org/10.1007/s10092-020-00369-3
33. Garoni, C., Serra-Capizzano, S.: Generalized Locally Toeplitz Sequences: Theory and Applications, vol. 1. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53679-8
34. Mazza, M., Donatelli, M., Manni, C., Speleers, H.: On the matrices in B-spline collocation methods for Riesz fractional equations and their spectral properties. Numer. Linear Algebra Appl. **30**(1), 2462–23 (2023). https://doi.org/10.1002/nla.2462
35. Hale, N., Trefethen, L.N.: Chebfun and numerical quadrature. Sci China Math **55**(9), 1749–1760 (2012). https://doi.org/10.1007/s11425-012-4474-z
36. Güttel, S.: Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. GAMM-Mitteilungen **36**(1), 8–31 (2013). https://doi.org/10.1002/gamm.201310002