Double-preconditioning for Fractional Linear Systems. Application to Stationary Fractional Partial Differential Equations

Xavier Antoine, Emmanuel Lorin

DOUBLE-PRECONDITIONING FOR FRACTIONAL LINEAR SYSTEMS. APPLICATION TO STATIONARY FRACTIONAL PARTIAL DIFFERENTIAL EQUATIONS

XAVIER ANTOINE* AND EMMANUEL LORIN†

Abstract. This paper is devoted to the numerical computation of *fractional linear systems*. The proposed approach is based on an efficient computation of Cauchy integrals allowing to estimate the real power of a (sparse) matrix A. A first preconditioner M is used to reduce the length of the Cauchy integral contour enclosing the spectrum of MA, hence allowing for a large reduction of the number of quadrature nodes along the integral contour. Next, ILU-factorizations are used to efficiently solve the linear systems involved in the computation of approximate Cauchy integrals. Numerical examples related to stationary (deterministic or stochastic) fractional Poisson-like equations are finally proposed to illustrate the methodology.

Key word. Real power of a matrix; Cauchy integral; preconditioning; deterministic and stochastic fractional stationary partial differential equations; unbounded domain.

1. Introduction. This paper is devoted to the efficient computation of the real power $\alpha \in \mathbb{R}_+^*$ of a large and sparse matrix $A \in \mathbb{C}^{n\times n}$ or $\mathbb{R}^{n\times n}$ which is supposed to be diagonalizable in $\mathbb{R}$ or $\mathbb{C}$, and to the solution to *fractional linear systems*

$$A^\alpha u = f\,, \tag{1.1}$$

where $f \in \mathbb{C}^n$ is given. The most natural method, also used in this paper, is based on the approximation of a Cauchy integral with a closed contour enclosing the spectrum of A. In this case, classical quadrature rules can be used for an accurate approximation of A^α [3]. Alternatively, A^α can be performed [6] by using Padé's approximants for z^α. Another approach, proposed in [6] and more specifically devoted to the computation of $A^\alpha b$ for a given vector b, is based on the solution to a differential system. A common point to all these approaches is that they require estimates of matrix inverses or solutions to linear systems. More generally, we refer to [6] for a discussion about the computation of $g(A)b$ for a holomorphic function g.

As said above, our strategy is based on the approximation of a Cauchy integral by a numerical quadrature rule [3, 6] involving J_A quadrature nodes/points, which is clearly expected to be embarrassingly parallel. Unless when specified, we assume that the spectrum of the matrix A is unknown so that a direct spectral decomposition in an orthonormal basis cannot be *a priori* used. Then, for $k \geqslant 0$, A^α is defined as (see e.g. Theorem 6.2.28 from [7])

$$A^\alpha = (2\pi \mathtt{i})^{-1}A^k \int_{\Gamma_A} z^{\alpha-k}(zI-A)^{-1}dz\,, \tag{1.2}$$

where Γ_A is a closed contour in the complex plane *enclosing the spectrum of the matrix* A, I is the identity matrix in $\mathbb{R}^{n\times n}$ and $\mathtt{i} = \sqrt{-1}$. In practice, when using the Cauchy integral to estimate A^α, it is clearly necessary to have some informations about the spectrum of the matrix A to define the contour path (see Section 3). Selecting k

*INSTITUT ELIE CARTAN DE LORRAINE, UNIVERSITÉ DE LORRAINE, UMR 7502, INRIA NANCY-GRAND EST, F-54506 VANDOEUVRE-LÈS-NANCY CEDEX, FRANCE, FRANCE (XAVIER.ANTOINE@UNIV-LORRAINE.FR).

†SCHOOL OF MATHEMATICS AND STATISTICS, CARLETON UNIVERSITY, OTTAWA, CANADA (ELORIN@MATH.CARLETON.CA); CENTRE DE RECHERCHES MATHÉMATIQUES, UNIVERSITÉ DE MONTRÉAL, MONTRÉAL, CANADA.

in (1.2) can be dependent on the location of the spectrum of A and the value of α. We refer to Subsection 4.6 for a discussion on the choice of the value of k. When using the Cauchy integral approach, two important issues related to the question of preconditioning can penalize the efficiency of the algorithm for solving a fractional linear system:

- first, the length $\ell(\Gamma_A)$ of the contour integral must be as small as possible to reduce the cost of the quadrature rule. Indeed, the number of linear systems to solve linearly grows according to the number J_A of quadrature points. To reduce this cost, we propose to use a preconditioned Cauchy integral formula based on a preconditioner M, leading to a contour length $\ell(\Gamma_{MA}) \ll \ell(\Gamma_A)$.
- Second, when the J_A (or J_{MA}) linear systems must be resolved, they also need to be preconditioned to be solved in conjunction with (for instance) a GMRES solver.

Proceeding this way, we then propose in Section 4 a double-preconditioning technique to efficiently estimate the real power of A. The first preconditioner allows for a reduction of the contour length, while the second preconditioner is used for efficiently solving the induced linear systems. Different Cauchy integral preconditioners are proposed and numerically tested. In Section 5, we present an efficient computational method for solving fractional linear systems, using the double-preconditioning method developed in Section 4.

This work is partially motivated by the computation of approximate solutions to deterministic or stochastic stationary fractional PDEs, and more specifically general fractional Poisson-like equations [9]. Such stationary equations can be solved approximately by using traditional finite difference methods which can require the solution to a so-called *fractional linear system*: find u such that $A^\alpha u = f$, for A, f, α given. A Cauchy integral preconditioning is then proposed in Section 6 to efficiently solve this problem for various cases of equations (deterministic or stochastic). Let us remark that this strategy, used here to solve Poisson-like equations, can also be naturally extended e.g. to fractional diffusion or Schrödinger equations (see again [9]). We propose several numerical experiments to illustrate the properties of the proposed approach for the stationary case.

Along the paper, some basic numerical experiments are presented to illustrate the main ideas and concepts. A discussion about the computational complexity of the derived method and a comparison with a direct finite difference approximation of the fractional Poisson equation is also proposed in Subsection 6.2. Some more elaborated experiments are reported in Subsection 6.3. We conclude in Section 7.

2. Fast computation of A^α when Sp(A) is given. An explicit knowledge of the spectrum $\mathrm{Sp}(A) := \{\lambda_k\}_{1\leqslant k\leqslant n}$ of the matrix A leads to an efficient computation of A^α. Such a situation occurs for instance when considering that the matrix A is a 3-, 5- or 7-points approximation of the Laplace operator with null Dirichlet boundary conditions on a finite interval, a square or a cube, respectively. In this case, the full spectrum (eigenvalues and eigenvectors) of the discrete laplacian A is indeed analytically known. Assuming that the transition matrix P_A and diagonal matrix Λ_A are explicitly known ($A = P_A\Lambda_A P_A^{-1}$), we then have: $A^\alpha = P_A\Lambda_A^\alpha P_A^{-1}$. Indeed, from (1.2) we can write that

$$\begin{aligned} A^\alpha &= (2\pi\mathtt{i})^{-1} A \int_\Gamma z^{\alpha-1}(zI - A)^{-1} dz = \left(P_A\Lambda_A P_A^{-1}\right)^\alpha \\ &= P_A (2\pi\mathtt{i})^{-1} \Lambda_A \int_\Gamma z^{\alpha-1}(zI - \Lambda_A)^{-1} dz P_A = P_A\Lambda_A^\alpha P_A^{-1}. \end{aligned}$$

Consequently, to solve $A^\alpha u = f$, with $f \in \mathbb{C}^n$ and A invertible, we can proceed as follows $u = A^{-\alpha} f = P_A^{-1}\Lambda_A^{-\alpha} P_A f$, which in practice leads to solving

$$\left\{\begin{array}{lcl} v & = & \Lambda_A^{-\alpha} P_A f, \\ P_A u & = & v\,. \end{array}\right.$$

Equivalently, for $A \in \mathbb{R}^{n\times n}$, by using the residue theorem one gets

$$\begin{aligned} A^\alpha = \sum_{k=1}^{n} \mathrm{Res}\big(z^\alpha (zI - A)^{-1}, \lambda_k\big) &= P_A^{-1} \sum_{k=1}^{n} \mathrm{Res}\big(z^\alpha (zI - \Lambda_A)^{-1}, \lambda_k\big) P_A \\ &= P_A^{-1} \sum_{k=1}^{n} D_A^{(k)} P_A, \end{aligned}$$

where $D_A^{(k)} = \{d_{A;ij}^{(k)}\}_{1\leqslant i,j\leqslant n}$, and

$$d_{A;jj}^{(k)} = \left\{\begin{array}{ll} \lambda_j^\alpha & \text{if } j = k \\ 0 & \text{if } j \neq k \end{array}\right. , \qquad d_{A;ij} = 0,\ \text{ if } i \neq j\,.$$

Obviously, we have $\Lambda_A = \sum_{k=1}^{n} D_A^{(k)}$. In this paper, we will exclude this situation, which makes trivial the computation of the solution to fractional linear systems.

3. Construction of the integral contour. In the general case, the direct strategy detailed in Section 2 cannot be used. We propose to develop an approach based on the discretization of the contour integral formula (1.2). Let us first consider the problem of building the contour Γ_A. When the spectrum location of the matrix A is known, Γ_A can be chosen such that its length is as small as possible. However, this is usually not the case, the crucial property of Γ_A being that it must enclose the whole spectrum of A. Various simple contours can be considered.

- A rectangular contour $\mathcal{G}(a, b, c, d)$ with left lower corner $a + \mathtt{i}b$ and right upper corner $c + \mathtt{i}d$.
- A circular contour $\mathcal{C}(z, R) := \{z + Re^{\mathtt{i}\theta}, \theta \in [0, 2\pi]\}$, centered at $z \in \mathbb{C}$ and with radius R.

In the following, Γ_A will refer to a rectangular contour and $\mathcal{C}_A$ to a circular one. The most natural and simple approach consists in evaluating the eigenvalue of A with largest modulus, i.e. $\lambda_\infty^{(A)} := \max_{1\leqslant i\leqslant n} |\lambda_i^{(A)}|$, where $\{\lambda_i^{(A)}\}_{1\leqslant i\leqslant n}$ denotes the (complex) eigenvalues of A (with possible multiplicity). As a consequence, we can define the contour as a circle $\mathcal{C}(\lambda_\infty^{(A)} + \varepsilon)$, where ε is a strictly positive number. When the contour is circular (with $k = 1$ in formula (1.2)), the Cauchy integral can be reformulated as follows

$$\begin{aligned} A^\alpha &= (2\pi\mathtt{i})^{-1} A \int_{\mathcal{C}_A} z^{\alpha-1}(zI - A)^{-1} dz \\ &= (2\pi)^{-1} A \int_0^{2\pi} \big((\lambda_\infty^{(A)} + \varepsilon)e^{\mathtt{i}\theta}\big)^{(\alpha-1)} \big((\lambda_\infty^{(A)} + \varepsilon)e^{\mathtt{i}\theta} I - A\big)^{-1} (\lambda_\infty^{(A)} + \varepsilon)e^{\mathtt{i}\theta} d\theta. \end{aligned}$$

Alternatively, we can construct Γ_A as $\mathcal{G}(\lambda_\infty^{(A)} - \varepsilon, \lambda_\infty^{(A)} - \varepsilon, \lambda_\infty^{(A)} + \varepsilon, \lambda_\infty^{(A)} + \varepsilon)$.

This general approach can unfortunately be inefficient from a practical point of view to numerically approximate the Cauchy integral by a quadrature formula, for instance with a clusterized spectrum. If the matrix A is hermitian, the contour can naturally be constructed more precisely. Typically, if $\lambda_{\min}^{(A)} = \min_{1\leqslant j\leqslant n} \lambda_j^{(A)}$ and $\lambda_{\max}^{(A)} = \max_{1\leqslant j\leqslant n} \lambda_j^{(A)}$ are computed by using a standard eigenvalue solver, then the simplest contour is a rectangle $\mathcal{G}_A(\lambda_{\min}^{(A)} - \varepsilon, -\varepsilon, \lambda_{\max}^{(A)} + \varepsilon, \varepsilon)$.

4. Cauchy integral preconditioner. In this section, we propose a *Cauchy integral preconditioning* strategy which potentially allows for a drastic reduction of the integral contour (1.2), then leading to a much faster algorithm than with a direct computation of A^α.

4.1. General consideration. A Cauchy integral preconditioner is a matrix M such that

$$(MA)^\alpha = (2\pi \mathtt{i})^{-1} MA \int_{\Gamma_{MA}} z^{\alpha-1}(zI - MA)^{-1} dz \,, \tag{4.1}$$

where we expect that $\ell(\Gamma_{MA}) \ll \ell(\Gamma_A)$, ℓ denoting the length of a curve in the complex plan. Typically, M will be chosen as a preconditioner for solving the linear system $Ax = b$, i.e. $M \approx A^{-1}$. However, additional constraints need to be added. The integral preconditioner of interest is two-fold

1. *clustering of the spectrum of the preconditioned matrix MA,*
2. *accurate estimate of the center of the spectrum of MA, more specifically* 1.

This idea is summarized in Fig. 1. Getting a shorter integration path for the Cauchy

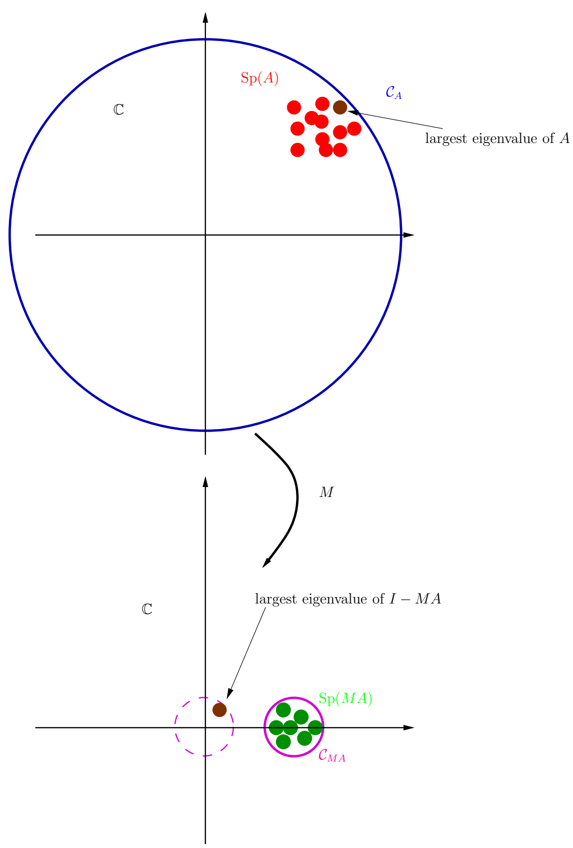

Fig. 1. *Clusterized spectra of the matrices A and MA, and their respective circular contours $\mathcal{C}_A$ and $\mathcal{C}_{MA}$ by using the above strategy.*

integral, i.e. leading to a small ratio $\lambda_\infty^{(I-MA)}/\lambda_\infty^{(A)}$, hence reduces the cost of the numerical quadrature used to approximate the Cauchy integral. Computing (1.2) from

(4.1) is expected to be more efficient than with a direct computation. To determine the contour for the preconditioned integral, we can proceed as for Γ_A but by computing the eigenvalue of $I-MA$ with largest amplitude, which is denoted by $\lambda_\infty^{(I-MA)}$. Next, we consider a circular contour $\mathcal{C}_{MA}=\mathcal{C}(1,\lambda_\infty^{(I-MA)}+\varepsilon)$ centered at 1 and with radius $\lambda_\infty^{(I-MA)}$. The reason for computing $\lambda_\infty^{(I-MA)}$ instead of $\lambda_\infty^{(MA)}$ is that $I-MA$ has a spectrum centered at 0, implying that $\mathrm{Sp}(MA)$ is centered at 1. An alternative to the circular contour is a square domain : $\mathcal{G}(-\lambda_\infty^{(I-MA)}-\varepsilon,-\lambda_\infty^{(I-MA)}-\varepsilon,\lambda_\infty^{(I-MA)}+\varepsilon,\lambda_\infty^{(I-MA)}+\varepsilon)$.

The following sections are devoted to the selection of the preconditioner M. Some constraints naturally arise, which makes its selection non-trivial.

4.2. Scaling Cauchy integral preconditioner. The simplest Cauchy integral preconditioner is a scaling matrix. Its interest may be limited, but in some cases it can be highly efficient. It simply consists in defining $M=c_AI$, where c_A is given by the 2-norm of the matrix A, i.e. $c_A=\|A\|_2:=\sup_{x\in\mathbb{R}^n-0}\|Ax\|_2/\|x\|_2$. Another possible choice, which is proved to be less efficient in practice, is $c_A=\lambda_\infty^{(A)}=\max_{i=1,\cdots,N}|\lambda_i^{(A)}|$. This simple scaling naturally implies that the following relation is satisfied

$$A^\alpha=M^{-\alpha}(MA)^\alpha\,, \tag{4.2}$$

and $\ell(\Gamma_{MA})<\ell(\Gamma_A)$. As a consequence, we expect a reduction of the length of the Cauchy integral contour and then an improvement of the overall efficiency of the algorithm for computing A^α. In general, the equality (4.2) is not valid, except for some very specific matrices and preconditioners.

4.3. Polynomial Cauchy integral preconditioner. The connection between $(MA)^\alpha$ and A^α is *a priori* not trivial if M and A do not commute. However, if M is for instance a polynomial preconditioner $p_K(A)$ [5], then obviously $p_K(A)A=Ap_K(A)$. The simplest approach to construct p_K consists in using a truncated Neumann series expansion. More precisely, for $\omega\in(0,2/\|A\|)$, $K\geqslant 1$ and $N:=I-\omega A$, we define

$$M=p_K(A)=\omega(I+N+\cdots+N^K). \tag{4.3}$$

Since $(\omega A)^{-1}=I+N+N^2+\cdots$, we can easily deduce the inequality: $\|I-MA\|\leqslant\|N^{K+1}\|\leqslant\|N\|^{K+1}$, where $||\cdot||$ is a matrix norm. Other polynomial preconditioners can be used (see Subsection 4.4) and more generally other types of Cauchy integral preconditioners may as well be implemented (see below) as long as they i) allow for a reduction of the length of the contour and ii) provide an efficient computation of A^α (resp. $A^{-\alpha}$) from $(MA)^\alpha$ (resp. $(MA)^{-\alpha}$). This leads to the following proposition which is important from a practical point of view.

Proposition 4.1. Assuming that M is a polynomial Cauchy integral preconditioner of the matrix A, then, for $\alpha\in\mathbb{R}^*$, we have $A^\alpha=M^{-\alpha}(MA)^\alpha$.

Proof. The proof is straightforward. For the matrix $A=\{A_{ij}\}_{1\leqslant i,j\leqslant n}$, we introduce $M=p_K(A)$, for $K\geqslant 1$. Then, one gets $AM=MA$ and (for $k=1$ in (1.2))

$$\begin{aligned}(MA)^\alpha &= (2\pi\mathtt{i})^{-1}MA\int_{\Gamma_{MA}}z^{\alpha-1}(zI-MA)^{-1}dz\\ &= (2\pi\mathtt{i})^{-1}AM\int_{\Gamma_{MA}}z^{\alpha-1}(zM^{-1}-A)^{-1}M^{-1}dz\,.\end{aligned}$$

Next, setting $z \leftarrow M^{-1}z$ and $\Gamma_A = M^{-1}\Gamma_{MA}$, we deduce that

$$\begin{aligned} (MA)^\alpha &= (2\pi \mathtt{i})^{-1} AM \int_{\Gamma_{MA}} M^{\alpha-1} z^{\alpha-1} (zI - A)^{-1} M^{-1} M dz \\ &= (2\pi \mathtt{i})^{-1} A \int_{\Gamma_A} M^{\alpha} z^{\alpha-1} (zI - A)^{-1} dz \\ &= M^\alpha (2\pi \mathtt{i})^{-1} A \int_{\Gamma_A} z^{\alpha-1} (zI - A)^{-1} dz = M^\alpha A^\alpha \,. \end{aligned} \tag{4.4}$$

□

Using a polynomial preconditioning leads to a reduction of the computational complexity of $p_K^\alpha(A)$ compared to A^α. In particular, we can easily prove that : $(p_K(A)A)^\alpha = p_K^\alpha(A)A^\alpha$, which means formally that $A^\alpha = p_K^{-\alpha}(A)(p_K(A)A)^\alpha$. However, evaluating A^α from $p_K^\alpha(A)$ is *a priori* not a simple task, although an iteration algorithm could be explored. At this stage, we propose an alternative preconditioning, particularly efficient for diagonally dominant matrices.

4.4. Differential-based preconditioner. We propose now a preconditioning method based on the solution to a differential system, used typically for computing $A^\alpha b$, for $b \in \mathbb{R}^n$. For $\alpha \in \mathbb{R}$, we recall [2, 3, 6] that the n-dimensional dynamical system

$$y'(\tau) = -\alpha(A - I)\big(I + \tau(A - I)\big)^{-1} y(\tau), \qquad y(0) = b, \tag{4.5}$$

is such that $y(\tau) = \big(I + \tau(A - I)\big)^{-\alpha} b$, $y(1) = A^{-\alpha} b$. Therefore, (4.5) can be used for computing $u = A^{-\alpha} f$. We can then approximate $A^{-\alpha} f$ as follows : $y(\tau) \approx \big(I - \alpha\tau(A - I)\big)f =: M_\tau f$. Thus, we have

$$(M_\tau A)^{-\alpha} = \frac{M_\tau A}{2\mathtt{i}\pi} \int_{\Gamma_{M_\tau A}} z^{-\alpha-1} (zI - M_\tau A)^{-1} dz = \frac{M_\tau}{2\mathtt{i}\pi} \int_{\Gamma_{M_\tau A}} z^{-\alpha-1} (zA - M_\tau)^{-1} dz \,.$$

Since M_τ is nothing but a parameterized polynomial preconditioner, we trivially have $AM_\tau = M_\tau A$ and then $(M_\tau A)^\alpha = M_\tau^\alpha A^\alpha$. This approach is partially relevant for non-diagonally dominant matrices when the approximations are accurate, i.e. for τ and α small enough. The preconditioning strategy is parallel to the one proposed with Cauchy integral, but this time applied to a differential system solver (Crank-Nicolson). This approach will be further investigated in a forthcoming paper.

4.5. Numerical approximations and experiments on contour integrals. From a practical point of view, the contour integral is numerically computed by using a quadrature rule leading to the approximate matrix computation (for $k = 1$ in (1.2))

$$A_{\mathbb{h}}^\alpha := (2\pi \mathtt{i})^{-1} A \sum_{1 \leqslant j \leqslant J_A} \mathbb{h}_j w_j z_j^{\alpha-1} (z_j I - A)^{-1} \,,$$

where $\{w_j\}_{1 \leqslant j \leqslant J_A}$ are the quadrature weights and $\{z_j\}_{1 \leqslant j \leqslant J_A}$ the integration nodes. The local discretization steps of the path are denoted by $\mathbb{h}_j$, and $\mathbb{h} = \max_{1 \leqslant j \leqslant J_A} \mathbb{h}_j$. In matrix norm, the order of convergence σ is such that

$$\big\| A_{\mathbb{h}}^\alpha - (2\pi \mathtt{i})^{-1} A \int_{\Gamma_A} z^{\alpha-1} (zI - A)^{-1} dz \big\| \leqslant Ch^\sigma \,.$$

In the following, we propose some numerical illustrations.

Experiment 1. Let us start by considering the one-dimensional operator $-\triangle + V$ defined on the computational domain $]-2;2[$ with homogeneous Dirichlet boundary conditions. The potential V is $V(x) = \mathtt{i}\exp(-20x^2)$. We use a 3-points finite difference discretization based on $n = 101$ interior points to approximate $-\triangle$. On Fig. 2, we represent two rectangular contours Γ_A and Γ_{MA}, where A is symmetric and M is the polynomial preconditioner $p_K(A)$ as defined in (4.3) for $K = 5$. We numerically get $\ell(\Gamma_A) \approx 5 \times 10^3$ and $\ell(\Gamma_{MA}) \approx 2$. Since $\ell(\Gamma_A)/\ell(\Gamma_{MA}) \approx 2.5 \times 10^3$, the numerical discretization based on Γ_{MA} is expected to be much faster than with Γ_A, for the same accuracy, since it needs far less discretization points.

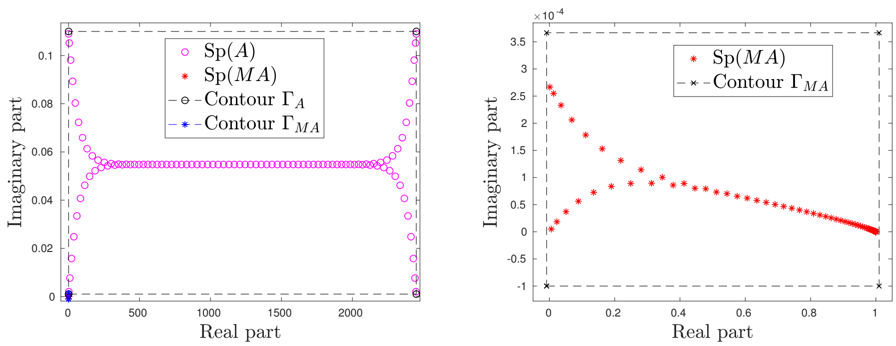

Fig. 2. ***Experiment 1.*** *(Left) Spectrum of the complex-valued matrices A and MA, and associated contours. (Right) Zoom on the spectrum of MA and contour Γ_{MA}.*

Experiment 2. In this second example, we consider a complex-valued random matrix $A \in \mathbb{C}^{n\times n}$ such that, for $1 \leqslant i,j \leqslant n$, $A_{ij} = \mathtt{rand}(0,1) + \mathtt{i}\,\mathtt{rand}(0,1)$, where $\mathtt{rand}(0,1)$ denotes a real number randomly chosen between 0 and 1 (that is taking its value in state space for a uniform distribution $\mathcal{U}(0,1)$). Moreover, we report the results for both $n = 101$ and $n = 1001$. We draw in Fig. 3 the corresponding spectra in the complex plane, including the contours Γ_A and Γ_{MA} for $n = 101$ (top) and $n = 1001$ (bottom). This shows the drastic clustering of the spectrum for the preconditioned matrix.

Experiment 3. Let us introduce the matrix $A = \{A_{ij}\}_{i,j} \in \mathbb{R}^{n\times n}$, defined by the two matrices B and C such that, for $1 \leqslant i,j \leqslant n$: $B_{ij} = n\mathtt{rand}(0,1)$, $C_{ij} = 20n + \mathtt{rand}(0,1)\delta_{ij}$, with $n = 100$, and $A = B + B^T + C$, which then has a real-valued spectrum. For $\alpha = 0.9$, we compare the relative error $\|A^\alpha_{\text{ref}} - A^\alpha_{\mathbb{h}}\|_2/\|A^\alpha_{\text{ref}}\|_2$ vs the number of quadrature points J_A and J_{MA}, with and without scaling preconditioner $M = I/\|A\|_2$ (see Subsection 4.2), for circular and rectangular contours in the preconditioned and non-preconditioned cases. The reference solution A^α_{ref} is computed by `matlab` through a spectral decomposition (see Subsection 2) and we use a composite midpoint quadrature rule. We first report on Fig. 4 (Top-Left) Sp(A), Sp(MA), the circular contours $\mathcal{C}_A$ and the preconditioned one $\mathcal{C}_{MA}$ with a scaling preconditioner, as well as the rectangular contours Γ_A and Γ_{MA} (with the same preconditioner). We then zoom in the neighborhood of Sp(A) in Fig. 4 (Top-Right), and in the neighborhood of Sp(MA) in Fig. 4 (Bottom-Left). We then compare in Fig. 4 (Bottom-Right) the convergence with respect to the contour choice (rectangle, circle). More specifically, we plot the relative error as a function of the number of quadrature points $J_{A,MA}$. As expected, the preconditioning improves the convergence rate for both the rectangular and circular contours. We also remark that the non-preconditioned rectangular

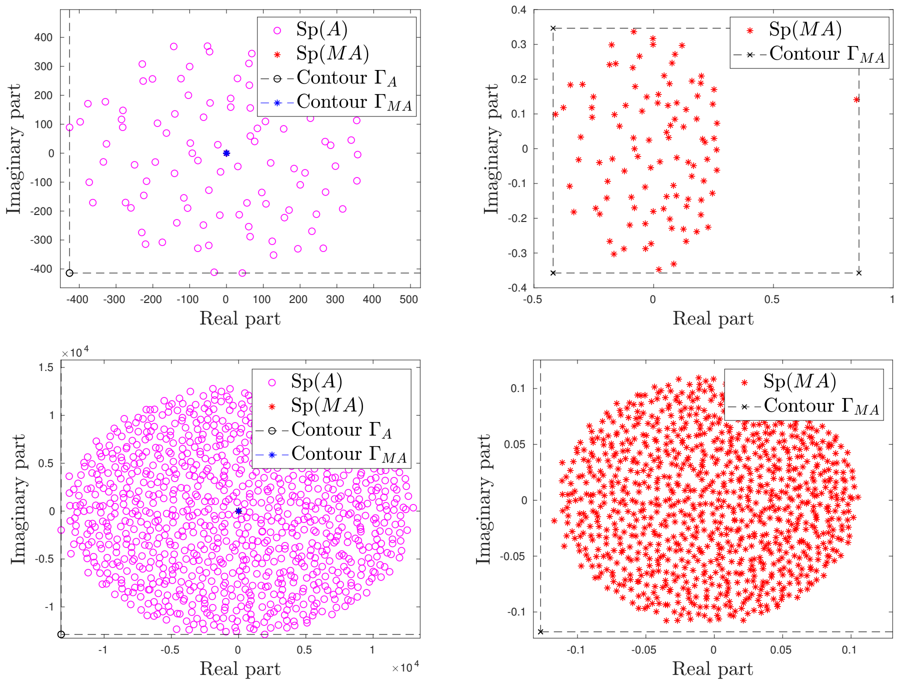

FIG. 3. ***Experiment 2.*** *Sp(A) and Sp(MA), and rectangular contour: (Top-Left) : $n = 101$; (Top-Right) : zoom for $n = 101$; (Bottom-Left) $n = 1001$; (Bottom-Right) : zoom for $n = 1001$.*

contour allows for a slightly more precise estimate of the Cauchy integral than for the non-preconditioned circular one. This is mainly due to the structure of the spectrum which is concentred around 0. As a consequence, the rectangle contour is very thin, then leading to a more accurate computation of the approximate operator $A_{\hbar}^{\alpha}$. The choice of the contour is naturally highly correlated to the structure of the spectrum.

Experiment 3bis. To complete the illustrations, let us consider the matrix $A = B + 0.75B^T + C$, where $B_{ij} = n\texttt{rand}(0,1)$ and $C_{ij} = 20n + \texttt{rand}(0,1)\delta_{ij}$, $1 \leqslant i, j \leqslant n$, for $n = 100$. The matrix A has a complex-valued spectrum. For $\alpha = 0.9$, Sp(A) is reported in Fig. 5 (Left) and a zoom on Sp(MA) is given in Fig. 5 (Right). We observe that the circular contour is more efficient here than the rectangular one (see Fig. 6).

4.6. Selection of the parameter k in the Cauchy integral formulation (1.2). We discuss now the selection of the Cauchy integral formulation, and more specifically the value of $k \in \mathbb{N}$ in formula (1.2). Since $z \in \Gamma_A$, we have $|z| > \rho(A)$, where $\rho(A)$ denotes the spectral radius of A. Denoting by $A_{\hbar}$ the approximate Cauchy integral using an order σ-composite-quadrature rule with $\hbar = \sup_j |z_{j+1} - z_j|$, there exists $c = c(A, \sigma) > 0$ such that

$$\frac{\|A_{\text{ref}}^{\alpha} - A_{\hbar}^{\alpha}\|}{\|A_{\text{ref}}^{\alpha}\|} \leqslant c\hbar^{\sigma} \sup_{z \in \Gamma_A} \frac{\left\| \dfrac{d^{\sigma}}{dz^{\sigma}} z^{\alpha-k}(zI - A)^{-1} \right\|}{\|A_{\text{ref}}^{\alpha}\|}. \qquad (4.6)$$

To minimize the error, this suggests that, if $\rho(A)$ is large, we should typically take $k \geqslant \lceil \alpha \rceil$, so that $k - \alpha \leqslant 0$. In practice, it is natural to simply select $k = \lceil \alpha \rceil$.

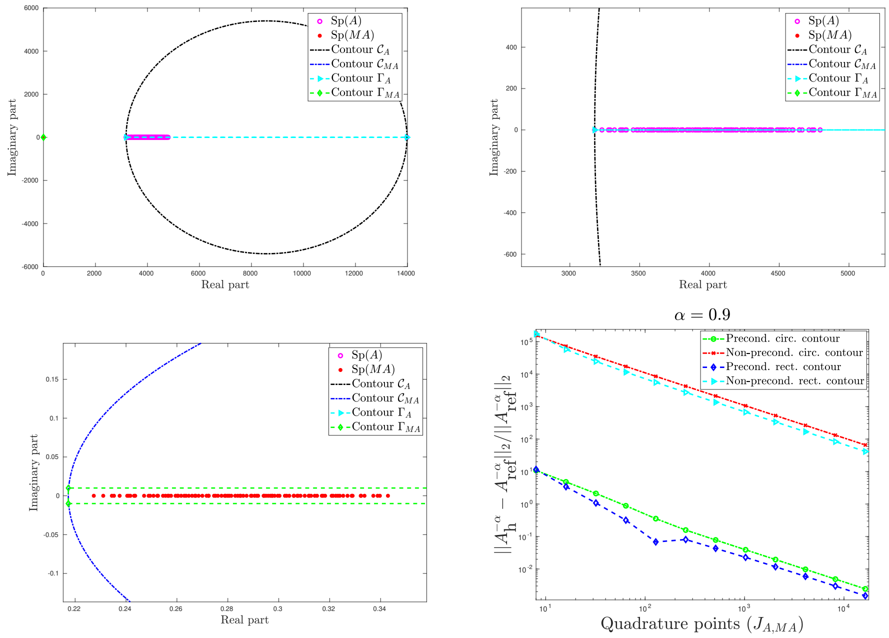

FIG. 4. ***Experiment 3.*** *(Top-Left) Sp(A) and Sp(MA), with $A \in \mathbb{R}^{n\times n}$, and $\mathcal{C}_A$, $\mathcal{C}_{MA}$, Γ_A, Γ_{MA}. (Top-Right) zoom on Sp(A). (Bottom-Left) zoom on Sp(MA). (Bottom-Right) Relative error vs the number of integration points for $\alpha = 0.9$.*

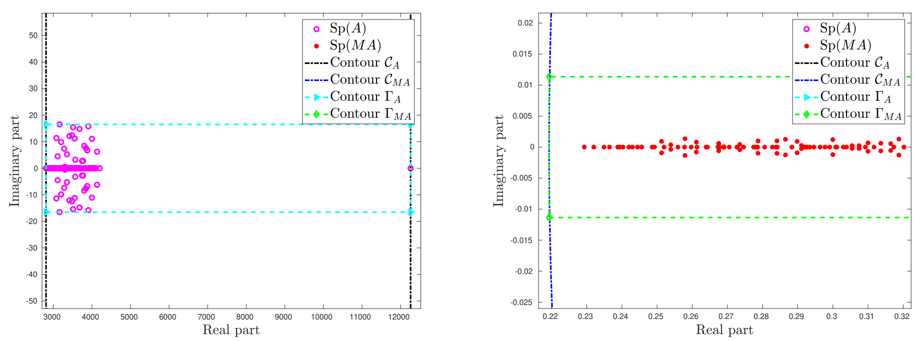

FIG. 5. ***Experiment 3bis.*** *(Left) Sp(A), Sp(MA) with $A \in \mathbb{R}^{n\times n}$, $\mathcal{C}_A$, $\mathcal{C}_{MA}$ and Γ_A, Γ_{MA}. (Right) Zoom on Sp(MA).*

However, whenever $\rho(A)$ is small, a natural choice in relation (1.2) is $k = 0$. Indeed, in this case, as $|z|$ is larger but close to $\rho(A)$, a small error (4.6) is expected and taking $k < \alpha$ could even deteriorate the approximation. For instance, it looks reasonable to use (1.2) with $k = \lceil\alpha\rceil$ for a direct evaluation of A^α and to use $k = 0$ for evaluating $(MA)^\alpha$ when M is an accurate (in the sense that $\rho(MA)$ is very small, typically < 1) Cauchy integral preconditioner. If $\rho(MA)$ is still larger than 1, it is preferable (theoretically) to take $k = \lceil\alpha\rceil$ to evaluate $(MA)^\alpha$. In the following, we arbitrary fix $k = 1$ (or $k = 0$), as most of the computations are done for $0 < \alpha < 1$ (or $1 < \alpha < 2$) and that $\rho(MA)$ will still be large enough to justify the fact that $k = \lceil\alpha\rceil$ provides a

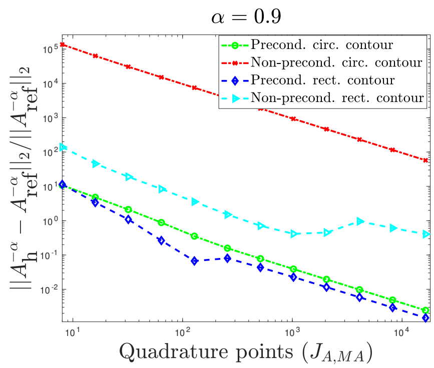

FIG. 6. ***Experiment 3bis.*** *Relative error ($\alpha = 0.9$) vs the number of integration points.*

better approximation than for $k = 0$. Notice that in the chosen benchmarks, we did not observe any noticeable effect of the selected formulation.

Experiment 4. To illustrate the discussion, we compare the relative error in 2-norm of A^α for $\alpha = 0.5$, where the matrix $A = \{A_{ij}\}_{1\leqslant i,j\leqslant n}$ is defined as: $A_{ij} = n\texttt{rand}(0,1) + \texttt{i}n\texttt{rand}(0,1)$, with $n = 400$. We compare the error (4.6) for $k = 0$,

$$A^\alpha = \frac{1}{2\texttt{i}\pi}\int_{\Gamma_A} z^\alpha (zI - A)^{-1} dz, \tag{4.7}$$

and $k = 1$

$$A^\alpha = \frac{A}{2\texttt{i}\pi}\int_{\Gamma_A} z^{\alpha-1} (zI - A)^{-1} dz. \tag{4.8}$$

We consider a circular contour where the number of quadrature nodes varies between 2 and 4096, and report in Fig. 7 the convergence of $\|A^\alpha - A^\alpha_{\mathbb{h}}\|_2/\|A^\alpha\|_2$ for $k = 0, 1$, in the non-preconditioned case, vs the number of quadrature points. We notice that taking $k = 0$ or $k = 1$ does not impact the behavior of the error.

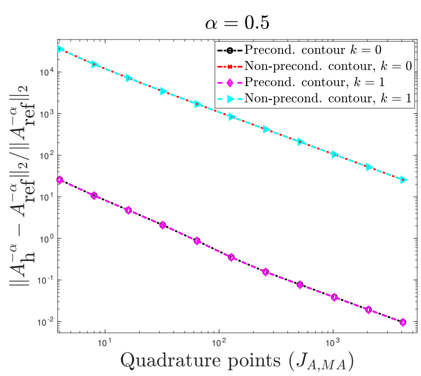

FIG. 7. **Experiment 4.** *Relative error ($\alpha = 0.5$) vs the number of integration points.*

5. Fractional linear systems $A^\alpha u = f$.

In the previous subsections, we developed an efficient methodology to estimate the real power of a matrix. In this paper, we are more specifically interested in the solution to *fractional linear systems* $A^\alpha u = f$, with $A \in \mathbb{C}^{n\times n}$, $f \in \mathbb{C}^n$, for some $\alpha \in \mathbb{R}_+^*$. For invertible matrices, we formally have $u = A^{-\alpha} f$.

5.1. Solution to fractional linear systems $A^\alpha u = f$, with $[M, A] = 0$. We assume here that M^α can efficiently be estimated numerically. If not, it is then more appropriate to proceed as in Subsection 5.2. We recall that for any matrix M such that $(MA)^\alpha = M^\alpha A^\alpha$, we can compute $A^{-\alpha} f$, from $(MA)^{-\alpha} f$, and

$$u \;=\; A^{-\alpha} f \;=\; M^\alpha (MA)^{-\alpha} f, \tag{5.1}$$

this approach being *a priori* valid for any invertible matrix $A \in \mathbb{C}^{n\times n}$. We can formally proceed as follows (e.g. for $k = 0$ in (1.2))

$$u = A^{-\alpha} f \;=\; (2\pi\mathtt{i})^{-1} \int_{\Gamma_A} z^{-\alpha} (zI - A)^{-1} f dz$$

where Γ_A encloses the spectrum of the matrix A. To estimate $(2\pi\mathtt{i})^{-1} \int_{\Gamma_A} z^{-\alpha}(zI - A)^{-1} f dz$, a Cauchy integral preconditioner is proposed. We denote by M a preconditioner for $A^{-\alpha}$, such that A and M commute: $[M, A] = 0$. Since $A^{-\alpha} = M^\alpha (MA)^{-\alpha}$, one gets

$$(MA)^{-\alpha} f \;=\; (2\pi\mathtt{i})^{-1} \int_{\Gamma_{MA}} z^{-\alpha} (zI - MA)^{-1} f dz\,.$$

Computed on a finite grid $\Gamma_{MA}^{(\mathbb{h})} \subsetneq \Gamma_{MA}$, with spatial resolution $\mathbb{h} = \max_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j$ and a quadrature of order σ, the approximate Cauchy integral to $(MA)^{-\alpha}$ is denoted by $S_{\mathbb{h}}^{(-\alpha)} \approx (MA)^{-\alpha}$ and is defined as

$$S_{\mathbb{h}}^{(-\alpha)} = (2\pi\mathtt{i})^{-1} \sum_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha} (z_j I - MA)^{-1}\,, \tag{5.2}$$

where $\{w_j\}_j$ are some interpolation weights. More precisely

- in the case of a rectangular contour, $z_j \in \Gamma_{MA}^{(\mathbb{h})}$ and $z_{j+1} = z_j + \mathbb{h}_{j+1}$, with $\mathbb{h}_j = \delta x_j + \mathtt{i}\delta y_j$. Denoting $(z_j I - MA)^{-1} f = u_j$, we have

$$\begin{aligned} u_{\mathbb{h}} \;&:=\; M^\alpha S_{\mathbb{h}}^{(-\alpha)} f = (2\pi\mathtt{i})^{-1} M^\alpha \sum_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha} u_j\,, \\ (z_j I - MA) u_j \;&=\; f, \text{ for all } 1 \leqslant j \leqslant J_{MA}\,, \end{aligned} \tag{5.3}$$

  i.e. $u_{\mathbb{h}} = M^\alpha S_{\mathbb{h}}^{(-\alpha)} f$.
- In the case of a circular contour of center z_c and radius $r_\varepsilon^{(A)}$, we have : $z_j = z_c + r_\varepsilon^{(A)} e^{\mathtt{i}\theta_j} \in \mathcal{C}_{MA}^{(\mathbb{h})}$ and $z_{j+1} = z_c + (z_j - z_c)e^{\mathtt{i}\delta\theta_{j+1}}$, with $\theta_{j+1} = \theta_j + \delta\theta_{j+1}$. We then consider the following quadrature

$$\begin{aligned} u_{\mathbb{h}} = (2\pi\mathtt{i})^{-1} M^\alpha \sum_{1\leqslant j\leqslant J_{MA}} \delta\theta_j w_j r_\varepsilon^{(MA)} e^{\mathtt{i}\theta_j} (r_\varepsilon^{(MA)})^{-\alpha} e^{-\mathtt{i}\alpha\theta_j} u_j, \\ (r_\varepsilon^{(MA)} e^{\mathtt{i}\theta_j} I - MA) u_j = f, \text{ for all } 1 \leqslant j \leqslant J_{MA}. \end{aligned}$$

A double-preconditioning is then implemented, the first one to reduce the contour length in the Cauchy integral, and then the second one to efficiently evaluate $(zI - MA)^{-1} f$, thus leading to

$$S_{\mathbb{h}}^{(-\alpha)} f = (2\pi\mathtt{i})^{-1} \sum_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha} (z_j I - MA)^{-1} f\,. \tag{5.4}$$

Since $\ell(\Gamma_{MA}) \ll \ell(\Gamma_A)$ (or $\ell(\mathcal{C}_{MA}) \ll \ell(\mathcal{C}_A)$), we get $J_{MA} \ll J_A$, which justifies the use of a Cauchy integral preconditioner M. Let us remark that when MA can be analytically diagonalized, the matrix power can be very efficiently computed, as stated in the following proposition.

Proposition 5.1. If MA is diagonalizable, then we have $MA = P_{MA}D_{MA}P_{MA}^{-1}$ and

$$S_{\mathbb{h}}^{(\alpha)}f = (2\pi\mathtt{i})^{-1}P_{MA}\Big[\sum_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha}(z_jI - D_{MA})^{-1}\Big]P_{MA}^{-1}f\,,$$

where D_{MA} is a diagonal matrix. As a consequence, in this case only one linear system (related to P_{MA}) has to be solved. However, except in some very simple cases (including low dimensional cases), P_{MA} and D_{MA} cannot be analytically calculated or computed.

Proof. Since MA is diagonalizable, we have

$$\begin{aligned}(MA)^\alpha &= (2\pi\mathtt{i})^{-1}\int_\Gamma z^\alpha(zI - MA)^{-1}dz = \big(P_{MA}D_{MA}P_{MA}^{-1}\big)^\alpha \\ &= P_{MA}(2\pi\mathtt{i})^{-1}\int_\Gamma z^\alpha(zI - D_{MA})^{-1}dzP_{MA} = P_{MA}D_{MA}^\alpha P_{MA}^{-1}\,.\end{aligned}$$

Next, we discretize the integral by using a classical quadrature formula:

$$S_{\mathbb{h}}^{-\alpha} = (2\pi\mathtt{i})^{-1}P_{MA}\Big[\sum_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha}(z_jI - D_{MA})^{-1}\Big]P_{MA}^{-1}\,,$$

which concludes the proof. □

In order to efficiently solve the linear systems (5.3), we simply compute in parallel the incomplete LU-factorizations [5]: for any $1 \leqslant j \leqslant J_{MA}$, $z_jI - MA \approx -L_jU_j$. We then define the preconditioners $N_j = -U_j^{-1}L_j^{-1}$ used to solve: $N_j(z_jI - MA)u_j = N_jf$. The J_{MA} linear systems are preconditioned and solved independently. On the other hand, if the systems are solved sequentially, $u_j^{(k)} \to_k u_{j+1}$ in $\mathbb{R}^n$ in at most n iterations and we can benefit from the previous computations

- From given $u_0^{(0)}$, solve $N_0(z_0I - MA)u_0 = N_0f$, for $z_0 \in \Gamma_{MA}^{(\mathbb{h})}$ (or $\in \mathcal{C}_{MA}^{(\mathbb{h})}$), by using the above algorithm, where $N_0 = -U_0^{-1}L_0^{-1}$.
- At index $j+1$: assuming u_j was previously computed, take as initial guess $u_{j+1}^{(0)} = u_j$ since for J_{MA} large enough, that is $|z_j - z_{j'}|$ small enough, we expect that u_{j+1} is close to u_j.
- It is not necessary to implement an ILU-factorization for any $1 \leqslant j \leqslant J_{MA}$. Basically, only a few ILU-factorizations are sufficient. By denoting $N_j = L_jU_j$, for j' close to j and by using continuity arguments, we expect that, in terms of conditioning, we have

$$\mathrm{cond}\big(N_j(z_jI - A)\big) \approx \mathrm{cond}\big(N_j(z_{j'}I - A)\big) \ll \mathrm{cond}(z_{j'}I - A)\,.$$

- Deduce $u = A^{-\alpha}f$, by estimating first $S_{\mathbb{h}}^{(-1-\alpha)}f$, then we have $u \approx u_{\mathbb{h}} := AM^{\alpha+1}S_{\mathbb{h}}^{(-1-\alpha)}f$.

We notice that performing a full LU-factorization on A provides a matrix M such that $[M, A] = 0$. However, computing M^α may be as almost complex as computing A^α. We therefore prefer to use ILU-factorizations.

5.2. Solution to fractional linear systems $A^\alpha u = f$, with $[M,A] \neq 0$. The most general and interesting case occurs when A and M do not commute. Then, we can no longer directly deduce the solution to $A^\alpha u = f$, from the solution to $M^\alpha (MA)^{-\alpha} f$. The natural procedure then consists in solving

$$M^\alpha (MA)^{-\alpha} A^\alpha u = M^\alpha (MA)^{-\alpha} f\,,$$

meaning that we precondition the linear system $A^\alpha u = f$ by $M^\alpha (MA)^{-\alpha}$ which is now only an (accurate) approximation to $A^{-\alpha}$. It is still necessary to be able to efficiently compute $M^\alpha (MA)^{-\alpha} x$ for any vector x. From a practical point of view, we have $M^\alpha (MA)^{-\alpha} x \approx M^\alpha S_{\mathbb{h}}^{(-\alpha)} x$, where $S_{\mathbb{h}}^{(-\alpha)} x$ is defined by (5.4) (setting $f = x$). The linear system is numerically solved by using an iterative scheme, but also requires intermediate solutions to sparse linear systems in order to estimate $M^\alpha (MA)^{-\alpha} x$. First, we approximate $(MA)^{-\alpha} x$ by $v_{\mathbb{h}}$ such that

$$\begin{array}{rcl} v_{\mathbb{h}} & = & (2\pi \mathtt{i})^{-1} M^\alpha \sum_{1 \leqslant j \leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha} u_j\,, \\ (z_j I - MA) u_j & = & x, \text{ for } 1 \leqslant j \leqslant J_{MA}\,. \end{array} \tag{5.5}$$

Next, we evaluate $M^\alpha v_{\mathbb{h}}$, which is more or less computationally complex. If M is a diagonal matrix (Jacobi) preconditioner, computing $M^\alpha v_{\mathbb{h}}$ is straightforward, while for ILU-preconditioning additional operations are needed, as described below.

5.3. Jacobi Cauchy integral preconditioner. Let us consider a Jacobi preconditioner, assuming that A is *diagonally dominant* and that $A_{ii} \neq 0$, for all $1 \leqslant i \leqslant n$. Setting $M = \mathrm{diag}(A_{11}^{-1}, \cdots, A_{nn}^{-1})$, we then have

$$(MA)^\alpha \;=\; (2\pi \mathtt{i})^{-1} \int_{\Gamma_{MA}} z^\alpha (zI - MA)^{-1} dz\,.$$

Similarly to the proof of Proposition 4.1 but noticing that *a priori* $AM \neq MA$ (in particular when the diagonal terms of A are not all equal), then $A^\alpha \neq M^{-\alpha}(MA)^\alpha$, with $\alpha \in \mathbb{R}^*$. Interestingly, M^α can however be very efficiently computed since M is diagonal.

5.4. ILU Cauchy integral preconditioner. Incomplete-LU factorizations appear as some natural candidates for solving fractional linear systems for two main reasons. First, they usually allow for a better preconditioning than Jacobi. Secondly, the triangular structure of the L and U matrices leads to an efficient computation of intermediate sparse linear systems. More specifically, we propose the following approach. We first implement an ILU-factorization $\widetilde{L}\widetilde{U}$ of the matrix A, with a threshold parameter $\zeta > 0$, and formally denote $M = (\widetilde{L}\widetilde{U})^{-1}$. In addition to (5.5), it is needed to approximate $M^\alpha v_{\mathbb{h}}$. In this goal, and unlike Jacobi preconditioning, it is necessary to solve additional triangular linear systems, i.e. we approximate $M^\alpha v_{\mathbb{h}}$, by $w_{\mathbb{h}}$ such that

$$\begin{array}{rcl} w_{\mathbb{h}} & = & (2\pi \mathtt{i})^{-1} \sum_{1 \leqslant j \leqslant J_M} \mathbb{h}_j w_j z_j^{-\alpha} v_j\,, \\ (z_j \widetilde{L}\widetilde{U} - I) v_j & = & \widetilde{L}\widetilde{U} v_{\mathbb{h}}, \text{ for } 1 \leqslant j \leqslant J_M\,. \end{array} \tag{5.6}$$

These new linear systems can be very efficiently solved since they are sparse and triangular. In addition, in order to improve the efficiency of the computation of $M^\alpha v_{\mathbb{h}}$, a Jacobi Cauchy integral preconditioner or scaling of M itself can be used as well, so that the quadrature is applied on a contour of reduced length which can be *a priori* as long as Γ_A, as proposed in Subsection 4.2.

5.5. Parallelization aspects. The computation of $(MA)^{-\alpha}$ can then be performed in parallel as follows. For p processors, we decompose Γ in p subcontours Γ_ℓ: $\Gamma = \cup_{\ell=1}^{p}\Gamma_\ell$ and $\ell(\Gamma_\ell) = \ell(\Gamma)/p$ and write

$$(MA)^{-\alpha} = \sum_{\ell=1}^{p}(MA)_\ell^{-\alpha} = \sum_{\ell=1}^{p}(2\pi\mathtt{i})^{-1}\int_{\Gamma_\ell} z^{-\alpha}(zI - MA)^{-1}dz\,.$$

We first implement an ILU-factorization and construct $\widetilde{L}$ and $\widetilde{U}$. For any fixed value of ℓ,

- we solve, for $\{z_j^{(\ell)}\}_j \in \Gamma_\ell$: $(z_j^{(\ell)}\widetilde{L}\widetilde{U} - A)u_j^\ell = f_j$,
- send&receive to the root processor the contribution of each Γ_ℓ, that is: $\sum_{z_j^{(\ell)}\in\Gamma_\ell}(2\pi\mathtt{i})^{-1}\mathbb{h}_j w_j z_j^{-\alpha}u_j$.

5.6. Numerical experiments on fractional linear systems. We provide now a few examples of numerical simulations to illustrate the methodology.

Experiment 5. In this example, we compare the efficiency of the different preconditioners implemented in GMRES for solving (1.1), where f is the unit vector. We report the convergence rate, represented as the residual history vs the GMRES iteration, where the solution is computed from

- a direct evaluation of the Cauchy integral without preconditioning (labelled No-precond.),
- by using an ILU preconditioner $M^{-\alpha}(MA)^{\alpha}$, with $M = \widetilde{L}\widetilde{U}$ for a drop tolerance at 10^{-4}, and a rectangular (ILU-precond. rect.) and circular contours (ILU-precond. circ.),
- with an ILU preconditioner M directly built on the sparse matrix A, and then the preconditioner M^α is used on A^α (and denoted M_α-precond.),
- and finally with an ILU preconditioner directly constructed from the full matrix A^α that we assume to be given (ILU-precond. on A^α).

The matrix A is defined as $A = (B+C) + (B+C)^T \in \mathbb{R}^{200\times 200}$, where

$$B_{ii} = 75\mathtt{rand}(0,1) + 15, \qquad B_{ii\pm1} = 5\mathtt{rand}(0,1) \mp 8, \qquad B_{ii\pm2} = \mathtt{rand}(0,1) \mp 1/2\,,$$

and $C_{ij} = \mathtt{rand}(0,1)$. We fix the tolerance to 10^{-15} in the GMRES, where the restart parameter is equal to 50. We report in Fig. 8 the results for the ILU-Cauchy integral preconditioner with (Left) $J_{A,MA} = 8$ and (Right) $J_{A,MA} = 128$. The number of GMRES iterations for the different preconditioners for a fixed number of quadrature nodes illustrates the efficiency of the proposed Cauchy integral preconditioning. For completeness, the same tests are performed by using a Jacobi Cauchy integral preconditioner (see Fig. 9).

Experiment 6. We now solve $A^\alpha u = f$, where A is a symmetric *diagonally dominant* full matrix which models a randomly perturbed Laplace operator, i.e. $-\triangle + d\mathcal{W}$, where $d\mathcal{W}$ is a small amplitude (2×10^{-2}) random and symmetric process, $n = 51$ and f is identically equal to 1. Moreover, we consider 3 values of the fractional order, i.e. $\alpha = 0.25$, $\alpha = 0.75$ and $\alpha = 1.5$. We then apply the Jacobi preconditioning for solving the linear systems related to $\big((z_c + r_\varepsilon^{(MA)}e^{\mathtt{i}\theta_j})I - MA\big)u_j$, in the following quadrature

$$\begin{gathered} u_{\mathbb{h}} = (2\pi)^{-1}M^\alpha(MA)\sum_{1\leqslant j\leqslant J_{MA}}\delta\theta w_j r_\varepsilon^{(A)}e^{\mathtt{i}\theta_j}(r_\varepsilon^{(A)})^{-\alpha-1}e^{-\mathtt{i}(\alpha+1)\theta_j}u_j, \\ \big((z_c + r_\varepsilon^{(MA)}e^{\mathtt{i}\theta_j})I - MA\big)u_j = f, \text{ for } 1\leqslant j\leqslant J_{MA}\,, \end{gathered}$$

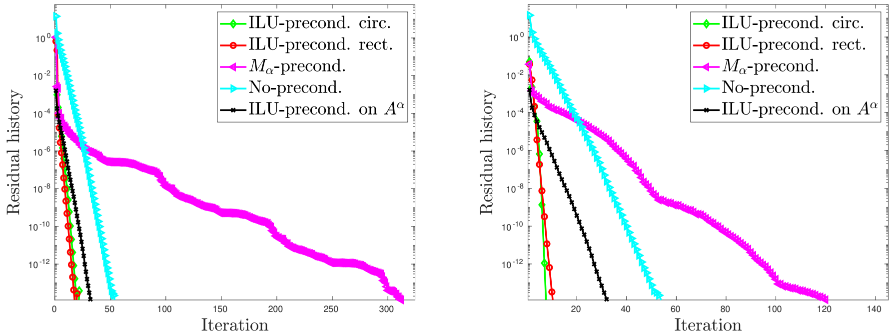

FIG. 8. **Experiment 5.** *Comparison of the residual history vs iterations of the GMRES algorithm (restarted at* 50 *iterations, and tolerance* 10^{-15}*) for various preconditioners: ILU Cauchy integral preconditioner (threshold at* 10^{-4}*), ILU-preconditioner on* A^α*. (Left):* $J_{A,MA} = 8$ *(Right):* $J_{A,MA} = 128$*.*

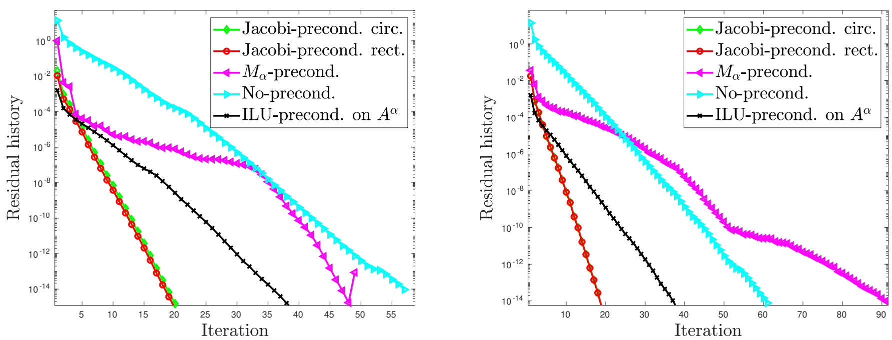

FIG. 9. **Experiment 5.** *Comparison of the residual history vs number of iterations of the GMRES (restarted after* 50 *iterations, and for a tolerance* 10^{-15}*) for different preconditioners: Jacobi preconditioner, ILU-preconditioner on* A^α*. (Left):* $J_{A,MA} = 8$ *(Right) :* $J_{A,MA} = 128$*.*

with $u_{\mathbb{h}} \approx u = A^{-\alpha} f$. Let us recall that $r_\varepsilon^{(A)} = r^{(A)} + \varepsilon$ and that the initial guess for computing u_{j+1} is taken as u_j. We report in Figs. 10 (Top/Bottom Left) the 2-norm error $\|u_{\mathbb{h}} - u_{\text{ref}}\|_2$ (in logscale) as a function of $J_{A,MA}$. We also provide the corresponding CPU-time with/without Jacobi preconditioning, as well as $\|A_{\mathbb{h}}^{-\alpha} - A^{-\alpha}\|_2$, where we have numerically estimated $A_{\mathbb{h}}^{-\alpha}$ from a direct (D) computation $(A_{\mathbb{h}}^{(\mathrm{D})})^{-\alpha}$ such that ($k = 1$ in relation (1.2))

$$(A_{\mathbb{h}}^{(\mathrm{D})})^{-\alpha} = (2\pi\mathtt{i})^{-1} A \sum_{1 \leqslant j \leqslant J_A} \mathbb{h}_j w_j z_j^{-\alpha-1} (z_j I - A)^{-1}\,, \tag{5.7}$$

or with a preconditioning $(A_{\mathbb{h}}^{(\mathrm{P})})^{-\alpha}$, from

$$(A_{\mathbb{h}}^{(\mathrm{P})})^{-\alpha} = (2\pi\mathtt{i})^{-1} MA \sum_{1 \leqslant j \leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha-1} (z_j I - MA)^{-1}\,. \tag{5.8}$$

The same test as above is also performed with $n = 501$ and $\alpha = 0.75$. The results are reported in Fig. 11, with $r^{(A)} = 5.15$ and $r_{MA} = 0.33$, i.e. with a ratio of about 15.5, illustrating the improved computational time.

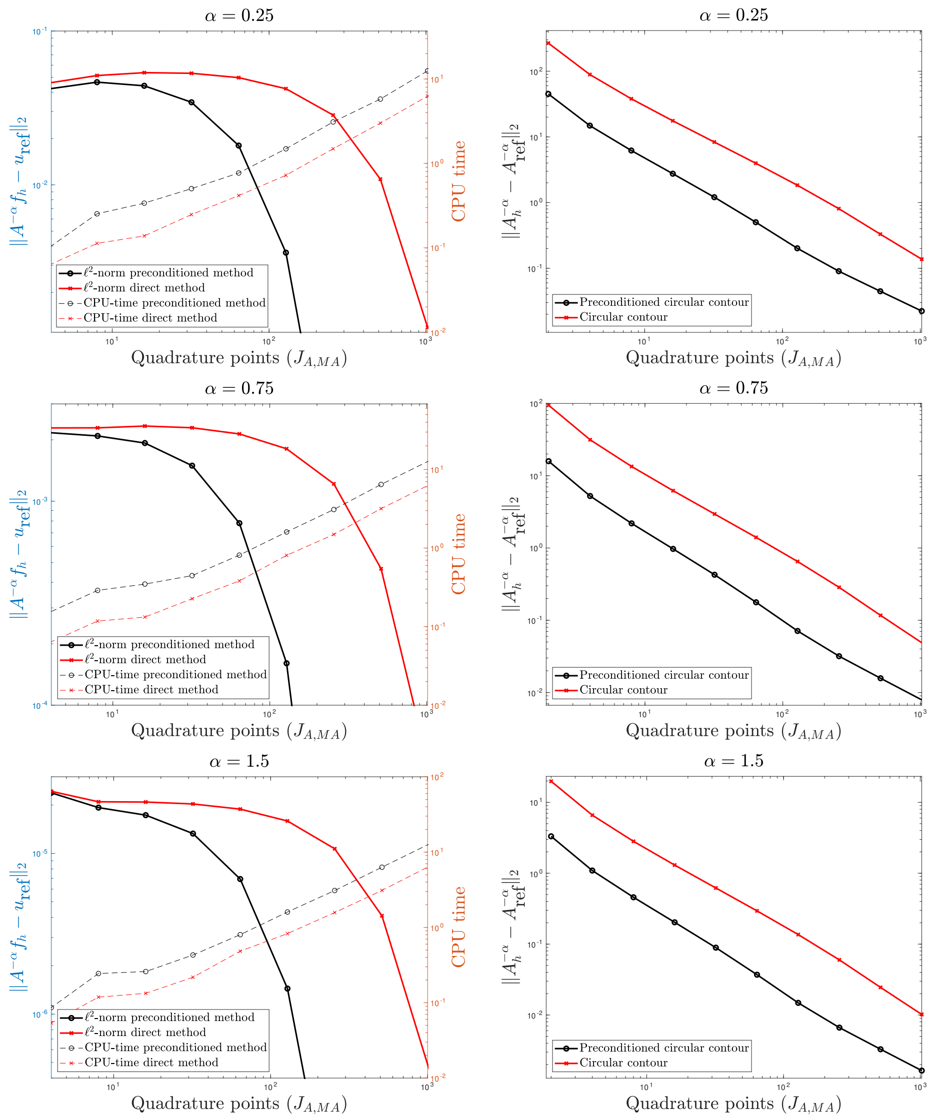

FIG. 10. **Experiment 6.** *(Top-Left) CPU-time (in seconds) in logscale, and* $\|A^{-\alpha} f_h - u_{ref}\|_2$*, where* $A^{-\alpha} f_h = u_{\mathbb{h}}$*, as a function of the number of quadrature points* $J_{A,MA}$*, with* $\alpha = 0.25$*, (Top-Right)* $\|A_{\mathbb{h}}^{-\alpha} - A_{ref}^{-\alpha}\|_2$ *in logscale as function of the number of quadrature points* $J_{A,MA}$*. (Middle-Left) and (Middle-Right) :* $\alpha = 0.75$*. (Bottom-Left) and (Bottom-Right) :* $\alpha = 1.5$*.*

Experiment 7. We propose the following numerical experiment to illustrate the differential-based preconditioner derived in this subsection for solving $A^{\alpha} u_h = f_h$, with $\alpha = 0.25$ in a case where $[M, A] = 0$. More precisely, we estimate $A^{-\alpha} f_h =$

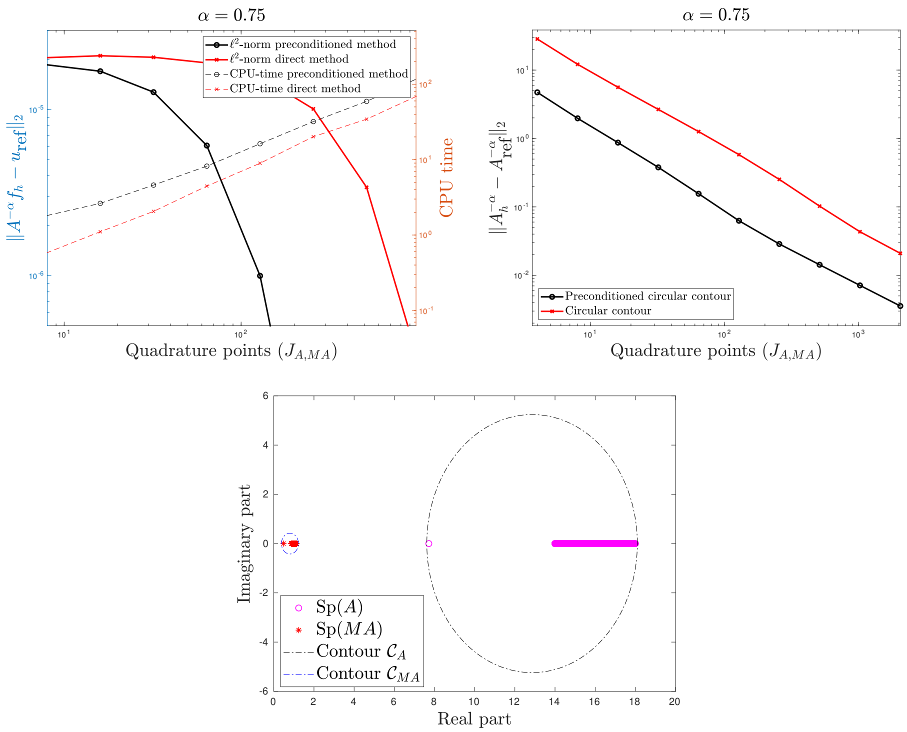

FIG. 11. **Experiment 6.** *(Top-Left) CPU-time (in seconds) in logscale, and* $\|A^{-\alpha} f_h - u_{ref}\|_2$ *as a function of the number of quadrature points* $J_{A,MA}$*, with* $\alpha = 0.75$*, (Top–Right)* $\|A_{\mathbb{h}}^{-\alpha} - A_{ref}^{-\alpha}\|_2$ *in logscale as a function of the number of quadrature points* $J_{A,MA}$*, with* $n = 501$*. (Bottom) Direct contour* $\mathcal{C}_A$ *and preconditioned contour* $\mathcal{C}_{MA}$*.*

$M_\tau^\alpha (M_\tau A)^{-\alpha} f_h$ and, for τ small enough, we have

$$
\begin{aligned}
M_\tau^\alpha (M_\tau A)^{-\alpha} f_h &= (2\pi \mathtt{i})^{-1} M_\tau^{\alpha+1} \int_{\Gamma_{M_\tau A}} z^{-\alpha-1} (zA - M_\tau)^{-1} f_h dz \\
&\approx u_{\mathbb{h}} = (2\pi \mathtt{i})^{-1} M_\tau^{\alpha+1} \sum_{1 \leqslant j \leqslant J_{M_\tau A}} \mathbb{h}_j w_j z_j^{-\alpha-1} (z_j A - M_\tau)^{-1} f_h \\
&\approx u_{\mathbb{h}} = (2\pi \mathtt{i})^{-1} \big(I - (\alpha+1)\tau (A - I) + \frac{\alpha(\alpha+1)\tau^2}{2} (I - A)^2 \big) \\
&\quad \times \sum_{1 \leqslant j \leqslant J_{M_\tau A}} \mathbb{h}_j w_j z_j^{-\alpha-1} (z_j A - M_\tau)^{-1} f_h \,.
\end{aligned}
$$

We consider A as a 3-point approximation of the Laplace operator on a segment $]-1;1[$, with $n = 101$ grid-points. We use some circular contours for both the non-preconditioned and preconditioned Cauchy integrals. In Fig. 12 (left), we report in logscale i) the CPU-time (in seconds) for the direct method (with $\mathcal{C}_A$) and double-preconditioned method (with $\mathcal{C}_{M_\tau A}$), and ii) $\|u_{\mathbb{h}} - u_{\text{ref}}\|_2$. We more precisely compare a Jacobi Cauchy integral preconditioner with a differential-based preconditioner M_τ with $\tau = 8 \times 10^{-1}$, $\tau = 9 \times 10^{-1}$, $\tau = 1$ and $\tau = 1.2$, and with a direct integral computation without preconditioner. We also use a Cauchy ILU-preconditioner ($\widetilde{L}\widetilde{U}$) with a drop-tolerance fixed to 10^{-1}, although in this case $[(\widetilde{L}\widetilde{U})^{-1}, A]$ is not necessarily close to zero. We also report $\|A_{\mathbb{h}}^{-\alpha} - A_{\text{ref}}^{-\alpha}\|_2$ in logscale in Fig. 12 (Right). The test illustrates that for a moderately dominant diagonal matrix, the differential-based preconditioning may be an alternative to Jacobi preconditioning, but an ILU-factorization can be used as well, if the drop tolerance is small enough.

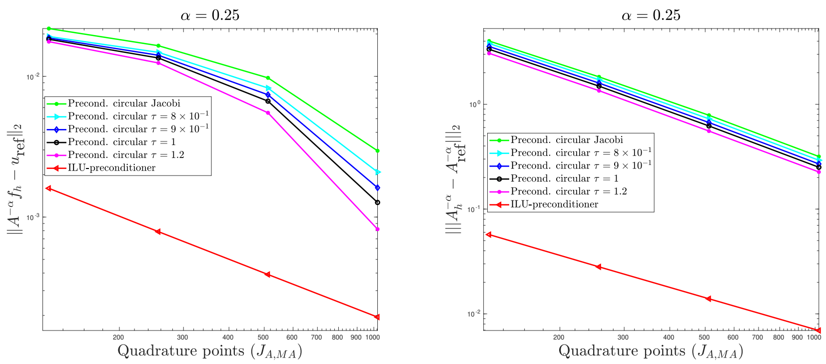

FIG. 12. **Experiment 7.** *Jacobi preconditioner, differential-based preconditioner* M_τ = *with* $\tau = 8 \times 10^{-1}$, $\tau = 9 \times 10^{-1}$, $\tau = 1$, *and* $\tau = 1.2$ *and ILU-preconditioner with a drop tolerance at* 10^{-1}. *(Left) In logscale* $\|A^{-\alpha} f_h - u_{ref}\|_2$ *where* $A^{-\alpha} f_h = u_{\mathbb{h}}$, *as a function of the number of quadrature points* $J_{A,MA}$, *with* $\alpha = 0.25$, *(Right)* $\|A_{\mathbb{h}}^{-\alpha} - A_{ref}^{-\alpha}\|_2$ *in logscale as a function of the number of quadrature points* $J_{A,MA}$, *with* $n = 101$. *(Right) Direct contour* $\mathcal{C}_A$ *and preconditioned contour* $\mathcal{C}_{MA}$.

6. Application to the approximation of stationary fractional PDEs.

The approximation of stationary and time-dependent fractional PDEs is currently a very active research area in particular due to the development of fractional models from physics (see e.g. [9]). We are here interested in the efficient computation of the solution to fractional Poisson-like equations thanks to the solutions to induced "fractional linear systems" $A^\alpha x = b$. The fractional Poisson equation on a bounded domain $\Omega \subset \mathbb{R}^d$ ($d = 1, 2, 3$) with null Dirichlet boundary condition on $\partial\Omega$ writes

$$\begin{aligned} -(-\triangle)^\alpha u &= f, \text{ in } \Omega, \\ u &= 0, \text{ on } \partial\Omega, \end{aligned} \tag{6.1}$$

where $\alpha \in (0, +\infty)$, $f \in L^p(\Omega)$, $1 < p < \infty$. The well-posedness of this problem is for instance studied in [1] for $\alpha \in (0, 1)$. In particular, it is proved that, for any function $f \in L^p(\Omega)$, with $1 < p < \infty$, the unique solution to the Dirichlet problem belongs to the functional space $\mathcal{L}^p_{2\alpha,\rm loc}(\Omega)$, where $\mathcal{L}^p_{2\alpha,\rm loc}(\Omega) := \big\{u \in L^p(\Omega) : u\varphi \in \mathcal{L}^p_{2\alpha}(\Omega) \text{ for any } \varphi \in C_0^\infty(\Omega)\big\}$, and $\mathcal{L}^p_{2\alpha}(\Omega) := \big\{u \in L^p(\Omega) : (-\triangle)^\alpha u \in L^p(\Omega)\big\}$. For any $u \in \mathcal{S}(\mathbb{R}^3)$ (i.e. the Schwartz's space of rapidly decaying C^∞-functions [11]) and $\alpha \in (0, 1)$, we have $(-\triangle)^\alpha u \in L^2(\mathbb{R}^3)$. An equivalent definition [4] in $\mathbb{R}^2$ can be stated for $\alpha \in (0, 1)$ and any $u \in \mathcal{S}(\mathbb{R}^2)$ [11] as

$$(-\triangle)^\alpha u(\boldsymbol{x}) = C(\alpha)\text{p.v.}\int_{\mathbb{R}^2} \frac{u(\boldsymbol{x}) - u(\boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|^{2+2\alpha}} d\boldsymbol{y} = C(\alpha) \lim_{\varepsilon \to 0^+} \int_{B_\varepsilon(\boldsymbol{x})} \frac{u(\boldsymbol{x}) - u(\boldsymbol{y})}{|\boldsymbol{x} - \boldsymbol{y}|^{2+2\alpha}} d\boldsymbol{y}, \tag{6.2}$$

where $B_\varepsilon(\boldsymbol{x})$ is the ball of radius ε and center $\boldsymbol{x}$, $C(\alpha)$ being the constant defined by

$$C(\alpha) := \Big(\int_{\mathbb{R}^2} \frac{1 - \cos(\xi_1)}{|\boldsymbol{\xi}|^{2+2\alpha}} d\boldsymbol{\xi}\Big)^{-1}. \tag{6.3}$$

The fractional laplacian can also be rewritten [4], for $\alpha \in (0,1)$ and any $u \in \mathcal{S}(\mathbb{R}^2)$, as

$$(6.4) \qquad (-\triangle)^{\alpha} u(\boldsymbol{x}) = -\frac{1}{2}C(\alpha)\text{p.v.}\int_{\mathbb{R}^2} \frac{u(\boldsymbol{x}+\boldsymbol{y}) - 2u(\boldsymbol{x}) + u(\boldsymbol{x}-\boldsymbol{y})}{|\boldsymbol{y}|^{2+2\alpha}} d\boldsymbol{y}.$$

Although nonlocal, this last equality is potentially interesting from a computational point of view (see formula (6.7)).

6.1. Fractional laplacian approximation. For the 2d computational domain $\Omega := \prod_{\ell=1}^{2}]-L_\ell;L_\ell[$, we introduce the inner uniform cartesian grid $\Omega_{\mathbf{h}}$, with $n := \Pi_{k=1}^{2} N_k$ total discretization points, defined by $\Omega_{\mathbf{h}} = \{\boldsymbol{x}_{i,j} = (x_{1,i}, x_{2,j})\}_{(i,j)\in\mathcal{I}}$, with $x_{1,i} := -L_1 + ih_1$, $x_{2,j} := -L_2 + jh_2$, $\mathcal{I} := \{(i,j) \in \mathbb{N}^2$ such that $1 \leqslant i \leqslant N_1$, $1 \leqslant j \leqslant N_2\}$, setting $h_\ell := 2L_\ell/(N_\ell+1)$, $\ell = 1,2$, and $\mathbf{h} := (h_1,h_2)$. When all the uniform discretization steps are equal along the directions, we define : $h := h_1 = h_2$, and then $n = N^2$, with $N := N_1 = N_2$.

To fix the ideas, let us now consider the following finite-difference approximation of the Laplacian operator $-\triangle$ based on a 5-point approximation scheme [10] along each direction for a function $\varphi := (\varphi_{i,j})$ set on the grid $\Omega_{\mathbf{h}}$

$$\begin{cases} -\triangle_{h_1}\varphi_{i,j} = \dfrac{\varphi_{i+2,j} - 16\varphi_{i+1,j} + 30\varphi_{i,j} - 16\varphi_{i-1,j} + \varphi_{i-2,j}}{12h_1^2}, \\ -\triangle_{h_2}\varphi_{i,j} = \dfrac{\varphi_{i,j+2} - 16\varphi_{i,j+1} + 30\varphi_{i,j} - 16\varphi_{i,j-1} + \varphi_{i,j-2}}{12h_2^2}. \end{cases}$$

A fourth-order approximation of the laplacian is then: $\Delta_h u_h := (\Delta_{h_1} + \Delta_{h_2})u_h$. Let $f_h = \{f_{i,j}\}_{(i,j)\in\mathcal{I}}$ be the projection of the function f on $\Omega_{\mathbf{h}}$, such that $f_{i,j} = f(\boldsymbol{x}_{i,j})$, $(i,j) \in \mathcal{I}$. Any other real space method (e.g. finite volume or finite element) could also be used within the method developed below. The approximate solution to system (6.1) is obtained by solving the fractional linear system $A_h^{\alpha} u_h = f_h$, corresponding to the discrete operator $-(-\Delta_h)^{\alpha}$. Let us assume that the approximation of $\triangle$ is at order q with discretization step h on the bounded domain $\Omega_{\mathbf{h}}$. The construction to the approximate solution u_h is performed by computing

$$(6.5) \qquad u_h = A^{-\alpha} f_h\,.$$

For the sake of conciseness, we use hereafter the notation "$A = A_h$". For a smooth function φ, one gets: $\triangle_h \varphi = \triangle\varphi + \mathcal{O}\big(h^q R_1(\phi)\big)$, so that as we use a *null Dirichlet boundary condition* [9] we obtain : $\triangle_h^{\alpha}\varphi = \triangle^{\alpha}\varphi + \mathcal{O}\big(h^{q\alpha} R_{\alpha}(\varphi)\big)$, with R_1 and R_α some smooth differential operators. To compute u_h, we propose to apply the strategy based on the efficient computation of Cauchy integrals. Inhomogeneous Dirichlet boundary conditions would complicate the approximation [9]. Let us also remark that usually real space approximations of the fractional Poisson equation are performed by directly approximating $(-\triangle)^{\alpha}$ by polynomials (see for instance [8]). The approach developed below is intended instead to illustrate that the efficient computation of matrix powers is an attractive alternative by numerically solving (6.5).

6.2. Computational complexity analysis in 2d. We recall that the fractional laplacian can also be rewritten [4] under the form (6.2), for $\alpha \in (0,1)$ and any $u \in \mathcal{S}(\mathbb{R}^2)$. A direct finite-difference approximation to (6.2) on a n-grid $\Omega_{\mathbf{h}} = \{\boldsymbol{x}_{i;j} = (x_{1,i}, x_{2,j})\,:\, 1 \leqslant i \leqslant N,\, 1 \leqslant j \leqslant N\}$ reads

$$(6.6) \qquad \mathcal{A}_{\alpha} u_h = f_h,$$

where $u_h := \{u_{i;j}\}_{1\leqslant i\leqslant N;1\leqslant j\leqslant N} \in \mathbb{C}^{N^2}$, with $u_{i;j} \approx u(\boldsymbol{x}_{i;j})$, and where the matrix $\mathcal{A}_\alpha$ is constructed by approximating (6.4) on the finite grid by

$$(6.7) \qquad -(-\triangle)^\alpha u(\boldsymbol{x}_{i;j}) \approx \frac{1}{2}C(\alpha)\sum_{k=1}^{N}\sum_{l=1}^{N}\frac{u_{i+k;j+l}-2u_{i;j}+u_{i-k;j-l}}{|\boldsymbol{y}_{k;l}|^{2+2\alpha}}h_1h_2.$$

The overall computational complexity to obtain the full matrix $\mathcal{A}_\alpha$ is at worse $\mathcal{O}(n^2)$, where the solution to (6.6) requires $\mathcal{O}(n^\beta)$ operations with $1 < \beta \leqslant 3$ related to the complexity for solving a *full* linear system (once) by a given brute force or specific algorithm. In contrast, for any (deterministic or stochastic) stationary operator, the methodology developed in Section 4 requires

- $\mathcal{O}(n)$ operations in order to construct a sparse approximate laplacian A.
- the computation of J_{MA} *sparse* linear systems, i.e. $\mathcal{O}(J_{MA}n^\gamma)$ operations, with $\gamma > 1$. This also contains the cost of the eigenvalue solver to estimate the largest and smallest eigenvalues to design the integral contour.
- The rest of the computation is a sparse matrix-vector product, thus requiring $\mathcal{O}(n)$ operations.

In fine, the overall computational complexity of the proposed method is $\mathcal{O}(J_{MA}n^\gamma)$, which must be compared to $\mathcal{O}(n^\beta + n^2)$. We conclude that a good preconditioned Cauchy integral approach allows for i) the use of sparse matrices, ii) efficient quadratures on *short length* contours, and thus is theoretically much more efficient than a direct approach.

We now state an important result of this paper. Consider the following system

$$(6.8) \qquad \begin{array}{rcl} -(-\triangle)^\alpha u & = & f, \text{ in } \Omega, \\ u & = & 0, \text{ on } \partial\Omega, \end{array}$$

where $\Omega \in \mathbb{R}^2$ is an open and bounded domain, and $f \in C^0(\Omega)$. Let us introduce the numerical solution $u_{h;\mathbb{h}} := M^\alpha S_{\mathbb{h}}^{(-\alpha)} f_h$, where

$$S_{\mathbb{h}}^{(-\alpha)} = (2\pi\mathtt{i})^{-1}MA\sum_{1\leqslant j\leqslant J_{MA}}\mathbb{h}_j w_j z_j^{-\alpha-1}(z_jI - MA)^{-1}\,.$$

Therefore, $u_{h;\mathbb{h}}$ is an approximation of the solution $u_h = A^{-\alpha}f_h$, the latter being itself an approximation to the solution u to system (6.8). In the sequel, we need the following discrete norms: for $v \in \ell^\infty(\Omega_{\mathbf{h}})$, $\|v\|_{\ell^\infty(\Omega_{\mathbf{h}})} := \max_{1\leqslant i\leqslant N_1;1\leqslant j\leqslant N_2}|v(x_{1,i},x_{2,j})|$, and for $v \in \ell^2(\Omega_{\mathbf{h}})$: $\|v\|_{\ell^2(\Omega_{\mathbf{h}})} := \big(h_1h_2\sum_{1\leqslant i\leqslant N_1;1\leqslant j\leqslant N_2}|v(x_{1,i},x_{2,j})|^2\big)^{1/2}$.

Theorem 6.1. We consider system (6.8). Let us denote by A an order $q \in 2\mathbb{N}^*$ finite-difference approximation to $-\triangle$ on the grid $\Omega_{\mathbf{h}}$, and by Π_h the projection operator from $C(\Omega)$ to $\ell^\infty(\Omega_{\mathbf{h}})$, such that $f_h := \Pi_h f = \{f(x_i,y_j)\}_{1\leqslant i\leqslant N_1;1\leqslant j\leqslant N_2}$. The approximate solution $u_{h;\mathbb{h}}$ on $\Omega_{\mathbf{h}}$ to the fractional linear system $A^\alpha u_h = f_h$ is constructed as follows:

$$\begin{array}{rcl} u_{h;\mathbb{h}} & := & (2\pi\mathtt{i})^{-1}M^\alpha(MA)\displaystyle\sum_{1\leqslant j\leqslant J_{MA}}\mathbb{h}_j w_j z_j^{-\alpha-1}u_j, \\ (z_jI - MA)u_j & = & f_h, \text{ for } 1\leqslant j\leqslant J_{MA}\,, \end{array}$$

where i) M is a Cauchy integral preconditioner such that $[M,A] = 0$, ii) J_{MA} is the total number of quadrature nodes on $\Gamma_{MA}^{(\mathbb{h})}$ (or $\mathcal{C}_A^{(\mathbb{h})}$), iii) $\{w_j\}_{1\leqslant j\leqslant J_{MA}}$ are the quadrature weights, and iv) $\{z_j\}_{1\leqslant j\leqslant J_{MA}} \in \Gamma_{MA}^{(\mathbb{h})}$ (or $\mathcal{C}_{MA}^{(\mathbb{h})}$) the quadrature nodes. Then, the following results hold

1. Let us assume that the Cauchy integral quadrature is of order $\sigma \in \mathbb{N}^*$, then there exists $C = C(\alpha, \Omega, A, M, \Gamma_{MA}) > 0$ and $D = D(f, \alpha, \Omega, A) > 0$, such that

$$\|u - u_{h;\mathbb{h}}\|_{\ell^2(\Omega_{\mathbf{h}})} \leqslant C \max_{1\leqslant j\leqslant J_{MA}} |\mathbb{h}_j|^\sigma \|f_h\|_{\ell^2(\Omega_{\mathbf{h}})} + D(h_1 h_2)^{q\alpha}\,. \tag{6.9}$$

2. Setting $n = N_1 N_2$ and for $A \in \mathbb{C}^{n\times n}$, a direct estimate of $A^{-\alpha}u_h$ requires $O(J_A n^{\beta_A})$ operations, with $1 < \beta_A < 3$. By using a *Cauchy integral preconditioner* M, only $J_{MA} \ll J_A$ linear systems have to be solved along Γ_{MA}. Performing p (parallel) ILU-factorizations N_j on $z_j I - A$ such that $\mathrm{cond}(N_j(z_j I - MA)) \ll \mathrm{cond}(z_j I - MA)$, the overall computational complexity of the double-preconditioning method is at most $\mathcal{O}(J_{MA} n^{\beta_{\mathrm{ILU}}})$, with $\beta_{\mathrm{ILU}} \gtrsim 1$ thanks to the cost for building the ILU-preconditioners.

Proof. We first prove (6.9). The approximate solution to (6.1) is defined by

$$u_h \quad = \quad A^{-\alpha} f_h \quad = \quad (2\pi\mathtt{i})^{-1} A \int_{\Gamma_A} z^{-\alpha-1}(zI - A)^{-1} f_h dz\,. \tag{6.10}$$

Assuming that an order $\sigma \in \mathbb{N}^*$ quadrature formula is used to approximate (6.10), we have

$$S_{\mathbb{h}}^{(-\alpha)} = (2\pi\mathtt{i})^{-1} MA \sum_{1\leqslant j\leqslant J_{MA}} \mathbb{h}_j w_j z_j^{-\alpha-1}(z_j I - MA)^{-1}.$$

In addition, one gets

$$(MA)^{-\alpha} f_h \quad = \quad (2\pi\mathtt{i})^{-1} MA \int_{\Gamma_{MA}} z^{-\alpha-1}(zI - MA)^{-1} f_h dz.$$

We therefore deduce that there exists $C_1 = C_1(\alpha, A, M, \Gamma_{MA}) > 0$ such that

$$\|S_{\mathbb{h}}^{(-\alpha)} - (MA)^{-\alpha}\|_2 \leqslant C_1 \max_{1\leqslant j\leqslant J_{MA}} |\mathbb{h}_j|^\sigma. \tag{6.11}$$

Next, we have: $u_{h;\mathbb{h}} - u_h = M^\alpha S_{\mathbb{h}}^{-\alpha} f_h - A^{-\alpha} f_h$. According to Proposition 5.1, the identity $A^{-\alpha} = M^\alpha (MA)^{-\alpha}$ yields

$$\begin{array}{rcl} \|u_{h;\mathbb{h}} - A^{-\alpha} f_h\|_{\ell^2(\Omega_{\mathbf{h}})} & = & \|M^\alpha S_{\mathbb{h}}^{-\alpha} f_h - M^\alpha (MA)^{-\alpha} f_h\|_{\ell^2(\Omega_{\mathbf{h}})} \\ & = & \|M^\alpha (S_{\mathbb{h}}^{-\alpha} - (MA)^{-\alpha}) f_h\|_{\ell^2(\Omega_{\mathbf{h}})} \\ & \leqslant & \|M^\alpha\|_2 \times \|S_{\mathbb{h}}^{(-\alpha)} - (MA)^{-\alpha}\|_2 \times \|f_h\|_{\ell^2(\Omega_{\mathbf{h}})}\,. \end{array}$$

From (6.11), we prove that there exists a positive constant $C = C(\alpha, \Omega, A, M, \Gamma_{MA}) > 0$ such that: $\|u_{h;\mathbb{h}} - A^{-\alpha} f_h\|_{\ell^2(\Omega_{\mathbf{h}})} \leqslant C \max_{1\leqslant j\leqslant J_{MA}} |\mathbb{h}_j|^p \|f_h\|_{\ell^2(\Omega_{\mathbf{h}})}$. Next, according to [9], one can find $D = D(f, \alpha, A, \Omega) > 0$ such that: $\|u - A^{-\alpha} f_h\|_{\ell^2(\Omega_{\mathbf{h}})} \leqslant D(h_1 h_2)^{q\alpha}$. We finally have

$$\begin{array}{rcl} \|u - u_h\|_{\ell^2(\Omega_{\mathbf{h}})} & \leqslant & \|u_{h;\mathbb{h}} - A^{-\alpha} f_h\|_{\ell^2(\Omega_{\mathbf{h}})} + \|u - A^{-\alpha} f_h\|_{\ell^2(\Omega_{\mathbf{h}})} \\ & \leqslant & C \max\limits_{1\leqslant j\leqslant J_{MA}} |\mathbb{h}_j|^\sigma \|f_h\|_{\ell^2(\Omega_{\mathbf{h}})} + D(h_1 h_1)^{q\alpha}\,. \end{array}$$

The second part of the theorem is straightforward. A direct estimate, i.e. without any preconditioner, requires the solution to J_A linear systems, each requiring $\mathcal{O}(n^{\beta_A})$

operations, for $1 < \beta_A < 1$. When a Cauchy integral preconditioner is used, only $J_{MA} \ll J_A$ linear systems have to be solved. For ILU-preconditioners, the overall complexity is simply $O(J_{MA} n^{\beta_{\rm ILU}})$, where $\beta_{\rm ILU} < \beta_A$. $\square$

The following remark is of interest for matrices with complex eigenvalues.

Remark 6.1. For matrices with a complex spectrum, the circular contour can also be used as follows: $\mathcal{C}_{MA} = \mathcal{C}(z_c, r_{MA})$, with center z_c and radius r_{MA}, and enclosing $\mathrm{Sp}(MA)$ corresponding to n poles to $(z_j I - MA)^{-1}$. In the following, we define $p_{MA} = J_{MA}/2$. In the case of a circular path, one also gets

$$(zI - MA)^{-1} = \frac{1}{2}\int_{-1}^{1} \big((re^{\mathtt{i}\pi\theta} + z_c + z)I - MA\big)^{-1} \frac{e^{\mathtt{i}\pi\theta}}{\big(e^{\mathtt{i}\pi\theta} + z_c e^{2\mathtt{i}\pi\theta}/r\big)} d\theta\,.$$

We set $z_j = \sigma_j^{-1} + z_c$ (see [12]), where

$$\sigma_j^{-1} = \begin{cases} r_{MA} e^{-\mathtt{i}\pi x_j}, & k = 1, \cdots, p_{MA}, \\ r_{MA} e^{-\mathtt{i}\pi x_{j-p}}, & k = p_{MA}+1, \cdots, 2p_{MA} = J_{MA}\,, \end{cases}$$

and

$$\widetilde{\sigma}_j = \begin{cases} \sigma_{j+p_{MA}}^{-1}, & k = 1, \cdots, p_{MA}, \\ \sigma_{j-p_{MA}}^{-1}, & k = p_{MA}+1, \cdots, 2p_{MA} = J_{MA}\,. \end{cases}$$

We first consider the construction of a preconditioner solving $(z_j I - MA)u_j = f_h$, for $n \in 2\mathbb{N}^*$,

$$(\widetilde{\sigma}_j I - A) \approx \begin{cases} L_{j+p_{MA}} U_{j+p_{MA}}, & j = 1 \cdots, p_{MA}\,, \\ L_{j-p_{MA}} U_{j-p_{MA}}, & j = p_{MA}+1 \cdots, 2p_{MA}\,. \end{cases}$$

These LU-factorizations can be used as preconditioners. Theorem 6.1 can easily be established for circular contours.

We can extend the methodology to equations of the form

$$\begin{array}{rcl} -(-\triangle)^{\alpha} u + Vu & = & f, \text{ in } \Omega, \\ u & = & 0, \text{ on } \partial\Omega, \end{array} \tag{6.12}$$

where $\alpha \in (0,1)$, $f \in L^p(\Omega)$ and $V := V(\boldsymbol{x}) \in L^{\infty}(\Omega)$, and with null Dirichlet boundary conditions on $\partial\Omega$. We propose the following finite difference approximation $(A^{\alpha} + V_h)u_h = f_h$, where i) the vector f_h and the matrix V_h are respectively the projection on $\Omega_{\mathbf{h}}$ of f and V, ii) $A = A_h$ is a finite difference approximation of $-\triangle$ on $\Omega_{\mathbf{h}}$ and iii) u_h is the approximate solution to u in (6.12). We formally have: $(I + A^{-\alpha}V_h)u_h = A^{-\alpha} f_h$. We then proceed as follows. We compute $A^{-\alpha} f_h$ and $A^{-\alpha} V_h$ by using the method developed above. Next,

1. we define $g_{\mathbb{h}}$ as an approximation to $A^{-\alpha} f_h$ following

$$\begin{array}{rcl} g_{\mathbb{h}} & := & (2\pi\mathtt{i})^{-1} A \displaystyle\sum_{1 \leqslant j \leqslant J_A} \mathbb{h}_j w_j z_j^{-\alpha-1} g_j, \\ (z_j I - A) g_j & = & f_h, \text{ for all } 1 \leqslant j \leqslant J_A\,, \end{array}$$

   where i) J_A is the total number of quadrature nodes on $\Gamma_A^{(\mathbb{h})}$, ii) $\{w_j\}_{1 \leqslant j \leqslant J_A}$ are some interpolation weights, and iii-a) $z_j \in \Gamma_A^{(\mathbb{h})}$ with $z_{j+1} = z_j + \mathbb{h}_{j+1}$ and $\mathbb{h}_j = \delta x_j + \mathtt{i}\delta y_j$ or iii-b) $z_j = z_c + r^{(A)} e^{\mathtt{i}\theta_j}$ and $z_{j+1} = z_c + (z_j - z_c)e^{\mathtt{i}\theta_{j+1}} = z_j e^{\mathtt{i}\delta\theta_{j+1}}$, with $\theta_{j+1} = \theta_j + \delta\theta_{j+1}$, where $\delta\theta_{j+1}$ is an angular increment.

2. Similarly, $B_{\mathbb{h}}$ is an approximation to $A^{-\alpha}V_h$

$$\begin{aligned} B_{\mathbb{h}}^{(i)} &:= (2\pi\mathtt{i})^{-1} A \sum_{1\leqslant j\leqslant J_A} \mathbb{h}_j w_j z_j^{-\alpha-1} v_j^{(i)}, \\ (z_j I - A) v_j^{(i)} &= V_h^{(i)}, \text{ for all } 1\leqslant j\leqslant J_A, \end{aligned}$$

where $V_h = [V_h^{(1)} \cdots V_h^{(n)}] \in \mathbb{R}^{n\times n}$ (resp. $B_{\mathbb{h}} = [B_{\mathbb{h}}^{(1)} \cdots B_{\mathbb{h}}^{(n)}] \in \mathbb{R}^{n\times n}$), setting $\{V_h^{(i)}\}_{1\leqslant i\leqslant n}$ (resp. $\{B_{\mathbb{h}}^{(i)}\}_{1\leqslant i\leqslant n}$) as the column vectors of V_h (resp. $B_{\mathbb{h}}$).

3. Finally, we solve : $(I - B_{\mathbb{h}})u_{h;\mathbb{h}} = g_{\mathbb{h}}$.

The computation of $B_{\mathbb{h}}^{(i)}$ is naturally embarrassingly parallel. Let us remark that Cauchy integral preconditioning can easily be combined with the above methodology for solving (6.12).

6.3. Numerical experiments on fractional Poisson equations. This section is devoted to some numerical experiments to illustrate the above approaches.

Experiment 8. 1d modified fractional Poisson equation. We consider : $-(-\triangle + V)^{\alpha}u = f$ on $\Omega =]-2,2[$, with $f(x) = \exp(-15x^2)$, $\alpha = 0.6$ and $V = 5$. We use a 5-point stencil approximate laplacian on $\Omega_{\mathbf{h}}$, where $n = 500$ and $A \in \mathbb{R}^{500\times 500}$. To analyze the performance of the proposed approach, we proceed as follows. We numerically compute $\lambda_{\min}^{(A)}$ and $\lambda_{\max}^{(A)}$ with a power and inverse-power methods, respectively, and define a circular contour $\mathcal{C}_A = \mathcal{C}(0, \lambda_{\infty}^{(I-A)} + \varepsilon_{\theta}^{(A)})$, with $\varepsilon_{\theta}^{(A)} = 5\times 10^{-2}$. The so-called direct method consists in computing

$$u_{h;\mathbb{h}} = (2\pi\mathtt{i})^{-1} \sum_{1\leqslant j\leqslant J_A-1} \mathbb{h}_j \big(\frac{z_j + z_{j+1}}{2}\big)^{-\alpha} (z_j I - A)^{-1} f_h\,, \tag{6.13}$$

with $z_j = z_c + r_{\varepsilon}^{(A)} e^{\mathtt{i}\theta_j}$. We define a Jacobi preconditioner $M = \mathrm{diag}(a_{11}^{-1}, \cdots, a_{nn}^{-1})$ and consider $\mathcal{C}_{MA} = \mathcal{C}(0, \lambda_{\infty}^{(I-MA)} + \varepsilon_{\theta}^{(MA)})$, where $\varepsilon_{\theta}^{(MA)} = 5\times 10^{-2}$. In the following, we compute only *one* CROUT (row) ILU factorization with tolerance 10^{-6}, setting the restart parameter to 20 iterations, $LU \approx \widetilde{z}I - A$ with $\widetilde{z} = \lambda_{\min}^{(MA)}$. We find $r_{\varepsilon}^{(A)} \approx 2.7$ and $r_{\varepsilon}^{(MA)} \approx 0.4$, corresponding to a gain factor equal to 6.7. In Fig. 13 (Right), we report in logscale i) the CPU-time (in seconds) for the direct method (with $\mathcal{C}_A$) and double-preconditioned method (with $\mathcal{C}_{MA}$), and ii) $\|u_{h;\mathbb{h}} - u_{\mathrm{ref}}\|_{\ell^2(\Omega_{\mathbf{h}})}$. The preconditioned approach converges much faster than the direct method which also requires more resources.

Experiment 9. 2d fractional Poisson equation. For $\Omega =]-5,5[\times]-1,1[$, we consider the fractional laplacian problem $-(-\triangle)^{\alpha}u = f$, with $f(\boldsymbol{x}) = \exp(-5x_1^2 - 10x_2^2)$ and $\alpha = 0.4$. We choose a simple 3-point stencil approximate laplacian on $\Omega_{\mathbf{h}} = \big\{(x_{1,i}, x_{2,j}) \in \Omega : 1\leqslant i\leqslant N_1,\, 1\leqslant j\leqslant N_2\big\}$, where $N_1 = 40$, $N_2 = 20$ and $A \in \mathbb{R}^{n\times n}$, for $n = 800$. The eigenvalues $\lambda_{\min}^{(A)}$ and $\lambda_{\max}^{(A)}$ are again computed by a power/inverse-power method. We define the rectangle contour $\Gamma_A = \mathcal{G}(\lambda_{\min}^{(A)} - \varepsilon, -\varepsilon, \lambda_{\max}^{(A)} + \varepsilon, \varepsilon)$, with $\varepsilon = 10^{-1}$. The direct method is based on (6.13), with $z_{j+1} = \mathbb{h}_{j+1} + z_j$ such that $\mathbb{h}_j := \delta x$ or $\mathbb{h}_j := \delta y$, leading to $\ell(\Gamma_A^{(\mathbb{h})}) = 2\big(\lambda_{\max}^{(A)} - \lambda_{\min}^{(A)} + 2\varepsilon\big)$, where J_A is the number of points to approximate Γ_A. For the Jacobi preconditioner M, we have $\ell(\Gamma_{MA}^{(\mathbb{h})}) = 2\big(\lambda_{\max}^{(MA)} - \lambda_{\min}^{(MA)} + 2\varepsilon\big)$. We calculate *one* CROUT (row) ILU-factorization, setting the tolerance to 10^{-6} and the value of the restart parameter to 20. Moreover, $LU \approx \widetilde{z}I - A$, with $\widetilde{z} = \lambda_{\min}^{(MA)}$. In Fig. 13 (Middle), we plot in logscale i) the CPU-time (in seconds) for both the direct method (with Γ_A) and double-preconditioned method (with Γ_{MA}), and ii) $\|A^{-\alpha}f_h - u_{\rm ref}\|_{\ell^2(\Omega)_h}$. It is clear that the preconditioned method is convergent much more rapidly than the direct one.

Experiment 9bis. 2d fractional Poisson equation. For $\Omega =]-2,2[^2$, we solve the fractional Poisson equation $-(-\triangle)^\alpha u = f$, for $f(\boldsymbol{x}) = 1$ and $\alpha = 0.4$. A 3-point stencil scheme is used for the laplacian on the square grid $\Omega_{\mathbf{h}}$, where $N = 50$, $A \in \mathbb{R}^{n\times n}$, and $n = 2500$. The power and inverse-power method provides $\lambda^{(A)}_{\rm min}$ and $\lambda^{(A)}_{\rm max}$. We use the circular contour $\mathcal{C}_A$, with $\varepsilon^{(A)}_\theta = 10^{-2}$. The direct method makes use of (6.13), with $z_j = z_c + r^{(A)}e^{\mathbf{i}\theta_j}$. We define the Jacobi preconditioner M and consider $\mathcal{C}^{(\mathbb{h})}_{MA}$, where $\varepsilon = 5\times 10^{-2}$. As in Experiment 9, one CROUT factorization is computed with the same parameters. We find $r^{(A)} \approx 8.34$ and $r^{(MA)} \approx 1.6$, corresponding to a gain factor equal to 5.2. In Fig. 13 (Middle), we provide in logscale i) the CPU-time (in seconds) for the direct method (with $\mathcal{C}_A$) and double-preconditioned method (with $\mathcal{C}_{MA}$), and ii) $\|A^{-\alpha}f_h - u_{\rm ref}\|_{\ell^2(\Omega_h)}$. The preconditioned method is definitively faster than the direct method, which is also more resources consuming.

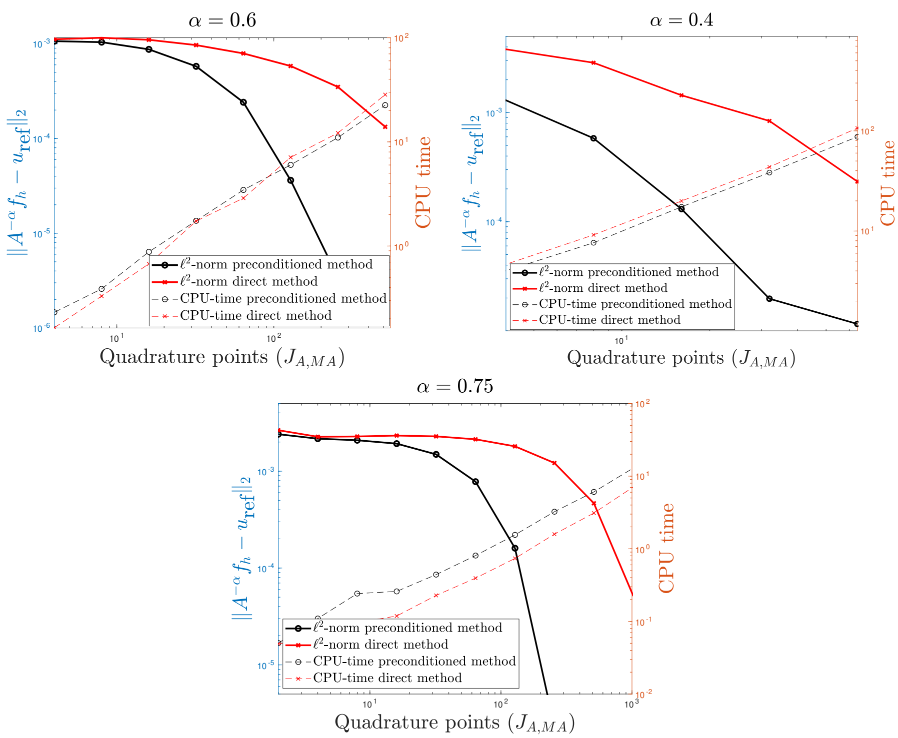

FIG. 13. *CPU-time (in seconds) in logscale, and* $\|A^{-\alpha}f_h - u_{\rm ref}\|_2$ *in logscale. (Left)* **Experiment 8.** *1d Poisson. (Middle)* **Experiment 9.** *2d Poisson. (Right)* **Experiment 9bis.** *2d Poisson.*

Experiment 10. We finally propose a series of experiments for $(-\triangle + V + d\mathcal{W})^\alpha u = f$ on a bounded domain $]-10,10[$ with null Dirichlet boundary conditions. The Cauchy integral is approximated by using $J_{A,MA} = 128$ quadrature nodes. For $-\triangle$, we use a 5-point scheme. In the following tests, we report the residual history vs the GMRES iteration number (the tolerance is 10^{-15} and the restart parameter is set to 50 iterations). More specifically using circular contours, we compare the convergence i) without Cauchy integral preconditioning (No precond.), ii) Jacobi Cauchy integral preconditioner (Jacobi precond.), iii) ILU Cauchy integral preconditioner (with scaling matrix for computing $M^{\alpha}x$, see Subsection 4.2) (ILU-precond.), iv) ILU factorization M on A and then M^{α} is used to precondition A^{α}, v) and finally no Cauchy integral preconditioning, but ILU preconditioning of A^{α}, assuming it is known (ILU-precond. on A^{α}). The convergence graphs (residual history vs GMRES iteration number) are given in Fig. 14 for

- **Experiment 10a.** $V = 0$ and the brownian motion $d\mathcal{W}$ is approximated by a symmetric random (uniform law) matrix of magnitude 0.12, and $\alpha = 0.75$.
- **Experiment 10b.** $V = 0$ and the brownian motion $d\mathcal{W}$ is computed by a unsymmetric random (uniform law) matrix with magnitude 0.06, and $\alpha = 0.75$.
- **Experiment 10c.** $V = 0$ and the brownian motion $d\mathcal{W}$ is approximated by a symmetric random (uniform law) matrix with magnitude 0.12, fixing $\alpha = 0.5$.
- **Experiment 10d.** $V = 100e^{-x^2}$ and the brownian motion $d\mathcal{W}$ is approximated by a symmetric random (uniform law) matrix of magnitude 0.12, and $\alpha = 0.75$.

These tests illustrate the fact that the convergence of the GMRES solver is highly dependent on the presence of a potential and the value of α. Overall, the ILU-Cauchy integral preconditioner allows for a faster (sometimes much faster) convergence than any other preconditioning approach.

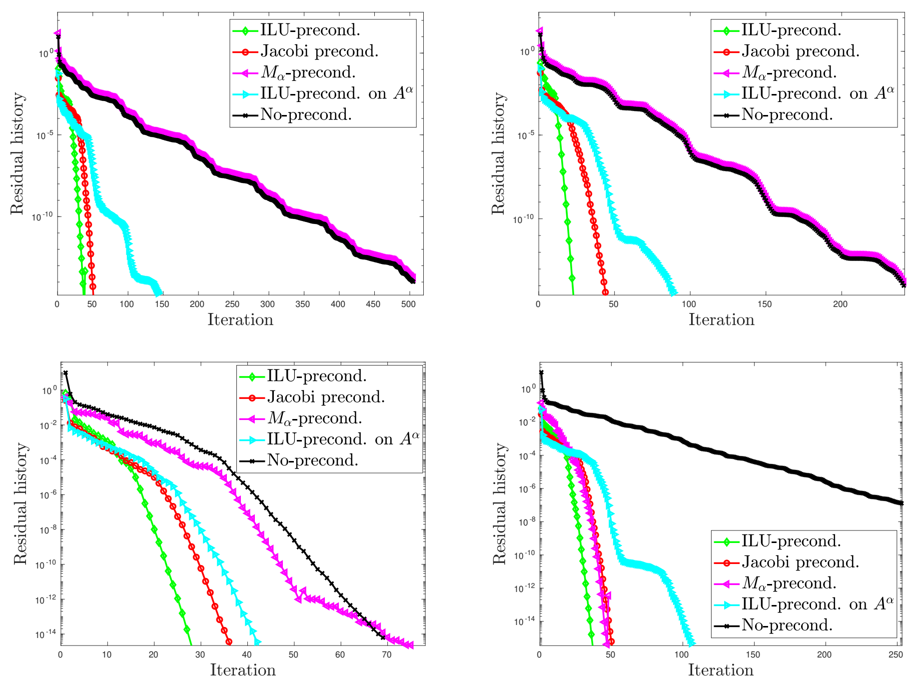

Fig. 14. *GMRES convergence. (Top-Left)* **Experiment 10a** *; (Top-Right)* **Experiment 10b***; (Botton-Left)* **Experiment 10c** *;(Bottom-Right)* **Experiment 10d**.

7. Conclusion. In this paper, we proposed an efficient method for computing the real power of a diagonalizable matrix A and algorithms for solving fractional linear systems, using quadrature rules for Cauchy integrals and contours enclosing the spectrum of A. Simple preconditioners are proposed for drastically reducing the computational complexity thanks to spectrum clustering. Some experiments are reported to illustrate the methodology. In particular, applications to (deterministic and stochastic) stationary fractional Poisson-like equations with Dirichlet boundary conditions are given. In a forthcoming paper, we will propose some realistic applications and comparisons with other methods such as the differential equation approach as defined in Subsection 4.4.

REFERENCES

[1] U. Biccari, M. Warma, and E. Zuazua. Local elliptic regularity for the Dirichlet fractional Laplacian. *Adv. Nonlinear Stud.*, 17(2):387–409, 2017.

[2] E. Carson and N.J. Higham. Accelerating the solution of linear systems by iterative refinement in three precisions. *SIAM J. Sci. Comput.*, 40(2):A817–A847, 2018.

[3] P. I. Davies and N. J. Higham. Computing $f(A)b$ for matrix functions f. In *QCD and numerical analysis III*, volume 47 of *Lect. Notes Comput. Sci. Eng.*, pages 15–24. Springer, Berlin, 2005.

[4] E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker's guide to the fractional Sobolev spaces. *Bulletin des Sciences Mathématiques*, 136(5):521–573, 2012.

[5] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2010.

[6] N.J. Higham. Evaluating Padé approximants of the matrix logarithm. *SIAM Journal on Matrix Analysis and Applications*, 22(4):1126–1135, 2001.

[7] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, Cambridge, 1991.

[8] Y. Huang and A. Oberman. Numerical methods for the fractional Laplacian: a finite difference–quadrature approach. *SIAM J. Numer. Anal.*, 52(6):3056–3084, 2014.

[9] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. Meerschaert, M. Ainsworth, and G. E. Karniadakis. What is the fractional laplacian? *arXiv:1801.09767v2*, 2018.

[10] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2004.

[11] M. E. Taylor. *Partial Differential Equations I. Basic Theory*, volume 115 of *Applied Mathematical Sciences*. Springer, New York, second edition, 2011.

[12] Y. Xi and Y. Saad. A rational function preconditioner for indefinite sparse linear systems. *SIAM J. Sci. Comput.*, 39(3):A1145–A1167, 2017.