



**HAL**  
open science

# Double-Preconditioning Techniques for Fractional Partial Differential Equation Solvers

Xavier Antoine, Emmanuel Lorin

► **To cite this version:**

Xavier Antoine, Emmanuel Lorin. Double-Preconditioning Techniques for Fractional Partial Differential Equation Solvers. *Multiscale Science and Engineering*, 2022, 4, pp.137-160. 10.1007/s42493-022-00084-x . hal-02340820

**HAL Id: hal-02340820**

**<https://hal.science/hal-02340820v1>**

Submitted on 31 Oct 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



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

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

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

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

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

75 **2. Fast computation of  $A^\alpha$  when  $\text{Sp}(A)$  is given.** An explicit knowledge of  
 76 the spectrum  $\text{Sp}(A) := \{\lambda_k\}_{1 \leq k \leq n}$  of the matrix  $A$  leads to an efficient computation  
 77 of  $A^\alpha$ . Such a situation occurs for instance when considering that the matrix  $A$  is a  
 78 3-, 5- or 7-points approximation of the Laplace operator with null Dirichlet boundary  
 79 conditions on a finite interval, a square or a cube, respectively. In this case, the  
 80 full spectrum (eigenvalues and eigenvectors) of the discrete laplacian  $A$  is indeed  
 81 analytically known. Assuming that the transition matrix  $P_A$  and diagonal matrix  $\Lambda_A$   
 82 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  
 83 (1.2) we can write that

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

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

$$87 \quad \begin{cases} v &= \Lambda_A^{-\alpha} P_A f, \\ P_A u &= v. \end{cases}$$

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

$$89 \quad \begin{aligned} A^\alpha &= \sum_{k=1}^n \operatorname{Res}(z^\alpha (zI - A)^{-1}, \lambda_k) = P_A^{-1} \sum_{k=1}^n \operatorname{Res}(z^\alpha (zI - \Lambda_A)^{-1}, \lambda_k) P_A \\ &= P_A^{-1} \sum_{k=1}^n D_A^{(k)} P_A, \end{aligned}$$

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

$$91 \quad d_{A;ij}^{(k)} = \begin{cases} \lambda_j^\alpha & \text{if } j = k \\ 0 & \text{if } j \neq k \end{cases}, \quad d_{A;ij} = 0, \text{ if } i \neq j.$$

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

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

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

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

$$112 \quad \begin{aligned} A^\alpha &= (2\pi i)^{-1} A \int_{\mathcal{C}_A} z^{\alpha-1} (zI - A)^{-1} dz \\ &= (2\pi)^{-1} A \int_0^{2\pi} ((\lambda_\infty^{(A)} + \varepsilon)e^{i\theta})^{(\alpha-1)} ((\lambda_\infty^{(A)} + \varepsilon)e^{i\theta} I - A)^{-1} (\lambda_\infty^{(A)} + \varepsilon)e^{i\theta} d\theta. \end{aligned}$$

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

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

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

124 **4.1. General consideration.** A Cauchy integral preconditioner is a matrix  $M$   
 125 such that

$$126 \quad (4.1) \quad (MA)^\alpha = (2\pi i)^{-1} MA \int_{\Gamma_{MA}} z^{\alpha-1} (zI - MA)^{-1} dz,$$

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

- 131 1. *clustering of the spectrum of the preconditioned matrix  $MA$ ,*
- 132 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

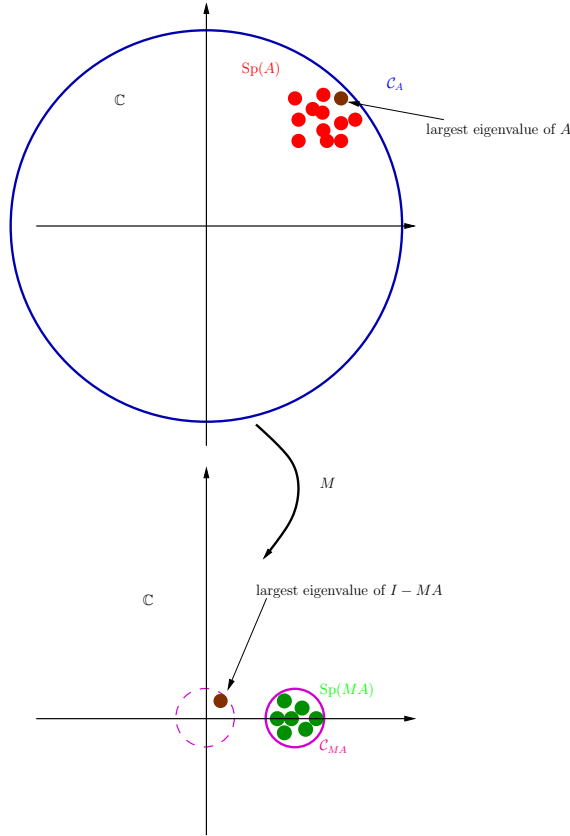


FIG. 1. Clusterized spectra of the matrices  $A$  and  $MA$ , and their respective circular contours  $C_A$  and  $C_{MA}$  by using the above strategy.

133 integral, i.e. leading to a small ratio  $\lambda_\infty^{(I-MA)} / \lambda_\infty^{(A)}$ , hence reduces the cost of the  
 134 numerical quadrature used to approximate the Cauchy integral. Computing (1.2) from  
 135

136 (4.1) is expected to be more efficient than with a direct computation. To determine the  
 137 contour for the preconditioned integral, we can proceed as for  $\Gamma_A$  but by computing  
 138 the eigenvalue of  $I - MA$  with largest amplitude, which is denoted by  $\lambda_\infty^{(I-MA)}$ . Next,  
 139 we consider a circular contour  $\mathcal{C}_{MA} = \mathcal{C}(1, \lambda_\infty^{(I-MA)} + \varepsilon)$  centered at 1 and with radius  
 140  $\lambda_\infty^{(I-MA)}$ . The reason for computing  $\lambda_\infty^{(I-MA)}$  instead of  $\lambda_\infty^{(MA)}$  is that  $I - MA$  has  
 141 a spectrum centered at 0, implying that  $\text{Sp}(MA)$  is centered at 1. An alternative to  
 142 the circular contour is a square domain :  $\mathcal{G}(-\lambda_\infty^{(I-MA)} - \varepsilon, -\lambda_\infty^{(I-MA)} - \varepsilon, \lambda_\infty^{(I-MA)} +$   
 143  $\varepsilon, \lambda_\infty^{(I-MA)} + \varepsilon)$ .

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

146 **4.2. Scaling Cauchy integral preconditioner.** The simplest Cauchy integral  
 147 preconditioner is a scaling matrix. Its interest may be limited, but in some cases it  
 148 can be highly efficient. It simply consists in defining  $M = c_A I$ , where  $c_A$  is given  
 149 by the 2-norm of the matrix  $A$ , i.e.  $c_A = \|A\|_2 := \sup_{x \in \mathbb{R}^n - 0} \|Ax\|_2 / \|x\|_2$ . An-  
 150 other possible choice, which is proved to be less efficient in practice, is  $c_A = \lambda_\infty^{(A)} =$   
 151  $\max_{i=1, \dots, N} |\lambda_i^{(A)}|$ . This simple scaling naturally implies that the following relation is  
 152 satisfied

$$153 \quad (4.2) \quad A^\alpha = M^{-\alpha} (MA)^\alpha,$$

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

158 **4.3. Polynomial Cauchy integral preconditioner.** The connection between  
 159  $(MA)^\alpha$  and  $A^\alpha$  is *a priori* not trivial if  $M$  and  $A$  do not commute. However, if  $M$  is for  
 160 instance a polynomial preconditioner  $p_K(A)$  [5], then obviously  $p_K(A)A = Ap_K(A)$ .  
 161 The simplest approach to construct  $p_K$  consists in using a truncated Neumann series  
 162 expansion. More precisely, for  $\omega \in (0, 2/\|A\|)$ ,  $K \geq 1$  and  $N := I - \omega A$ , we define

$$163 \quad (4.3) \quad M = p_K(A) = \omega(I + N + \dots + N^K).$$

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

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

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

$$175 \quad \begin{aligned} (MA)^\alpha &= (2\pi i)^{-1} MA \int_{\Gamma_{MA}} z^{\alpha-1} (zI - MA)^{-1} dz \\ &= (2\pi i)^{-1} AM \int_{\Gamma_{MA}} z^{\alpha-1} (zM^{-1} - A)^{-1} M^{-1} dz. \end{aligned}$$

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

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

178  $\square$

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

185 **4.4. Differential-based preconditioner.** We propose now a preconditioning  
186 method based on the solution to a differential system, used typically for computing  
187  $A^\alpha b$ , for  $b \in \mathbb{R}^n$ . For  $\alpha \in \mathbb{R}$ , we recall [2, 3, 6] that the  $n$ -dimensional dynamical  
188 system

$$189 \quad (4.5) \quad y'(\tau) = -\alpha(A - I)(I + \tau(A - I))^{-1}y(\tau), \quad y(0) = b,$$

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

$$193 \quad (M_\tau A)^{-\alpha} = \frac{M_\tau A}{2\mathbf{i}\pi} \int_{\Gamma_{M_\tau A}} z^{-\alpha-1}(zI - M_\tau A)^{-1}dz = \frac{M_\tau}{2\mathbf{i}\pi} \int_{\Gamma_{M_\tau A}} z^{-\alpha-1}(zA - M_\tau)^{-1}dz.$$

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

200 **4.5. Numerical approximations and experiments on contour integrals.**  
201 From a practical point of view, the contour integral is numerically computed by using  
202 a quadrature rule leading to the approximate matrix computation (for  $k = 1$  in (1.2))

$$203 \quad A_h^\alpha := (2\pi\mathbf{i})^{-1}A \sum_{1 \leq j \leq J_A} \mathfrak{h}_j w_j z_j^{\alpha-1} (z_j I - A)^{-1},$$

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

$$207 \quad \|A_h^\alpha - (2\pi\mathbf{i})^{-1}A \int_{\Gamma_A} z^{\alpha-1}(zI - A)^{-1}dz\| \leq Ch^\sigma.$$

208 In the following, we propose some numerical illustrations.

209 **Experiment 1.** Let us start by considering the one-dimensional operator  $-\Delta + V$   
210 defined on the computational domain  $] - 2; 2[$  with homogeneous Dirichlet boundary  
211 conditions. The potential  $V$  is  $V(x) = i \exp(-20x^2)$ . We use a 3-points finite differ-  
212 ence discretization based on  $n = 101$  interior points to approximate  $-\Delta$ . On Fig. 2,  
213 we represent two rectangular contours  $\Gamma_A$  and  $\Gamma_{MA}$ , where  $A$  is symmetric and  $M$  is  
214 the polynomial preconditioner  $p_K(A)$  as defined in (4.3) for  $K = 5$ . We numerically  
215 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  
216 discretization based on  $\Gamma_{MA}$  is expected to be much faster than with  $\Gamma_A$ , for the same  
217 accuracy, since it needs far less discretization points.

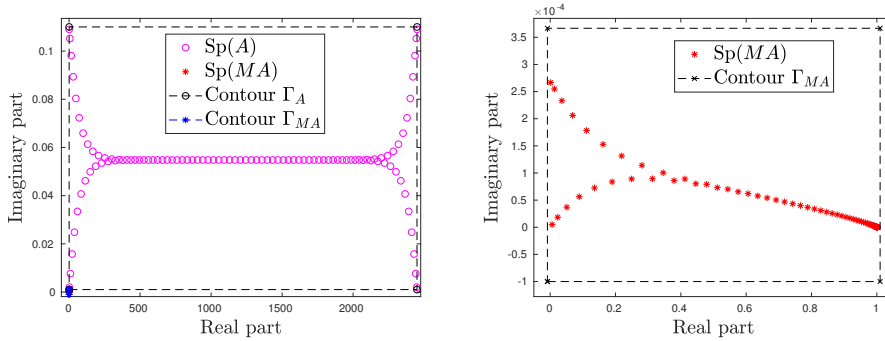


FIG. 2. **Experiment 1.** (Left) Spectrum of the complex-valued matrices  $A$  and  $MA$ , and associated contours. (Right) Zoom on the spectrum of  $MA$  and contour  $\Gamma_{MA}$ .

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

226 **Experiment 3.** Let us introduce the matrix  $A = \{A_{ij}\}_{i,j} \in \mathbb{R}^{n \times n}$ , defined by  
227 the two matrices  $B$  and  $C$  such that, for  $1 \leq i, j \leq n$ :  $B_{ij} = n \mathbf{rand}(0, 1)$ ,  $C_{ij} =$   
228  $20n + \mathbf{rand}(0, 1) \delta_{ij}$ , with  $n = 100$ , and  $A = B + B^T + C$ , which then has a real-valued  
229 spectrum. For  $\alpha = 0.9$ , we compare the relative error  $\|A_{\text{ref}}^\alpha - A_{\text{h}}^\alpha\|_2 / \|A_{\text{ref}}^\alpha\|_2$  vs the  
230 number of quadrature points  $J_A$  and  $J_{MA}$ , with and without scaling preconditioner  
231  $M = I / \|A\|_2$  (see Subsection 4.2), for circular and rectangular contours in the pre-  
232 conditioned and non-preconditioned cases. The reference solution  $A_{\text{ref}}^\alpha$  is computed by  
233 `matlab` through a spectral decomposition (see Subsection 2) and we use a composite  
234 midpoint quadrature rule. We first report on Fig. 4 (Top-Left)  $\text{Sp}(A)$ ,  $\text{Sp}(MA)$ , the  
235 circular contours  $\mathcal{C}_A$  and the preconditioned one  $\mathcal{C}_{MA}$  with a scaling preconditioner, as  
236 well as the rectangular contours  $\Gamma_A$  and  $\Gamma_{MA}$  (with the same preconditioner). We then  
237 zoom in the neighborhood of  $\text{Sp}(A)$  in Fig. 4 (Top-Right), and in the neighborhood  
238 of  $\text{Sp}(MA)$  in Fig. 4 (Bottom-Left). We then compare in Fig. 4 (Bottom-Right) the  
239 convergence with respect to the contour choice (rectangle, circle). More specifically,  
240 we plot the relative error as a function of the number of quadrature points  $J_{A,MA}$ .  
241 As expected, the preconditioning improves the convergence rate for both the rectan-  
242 gular and circular contours. We also remark that the non-preconditioned rectangular



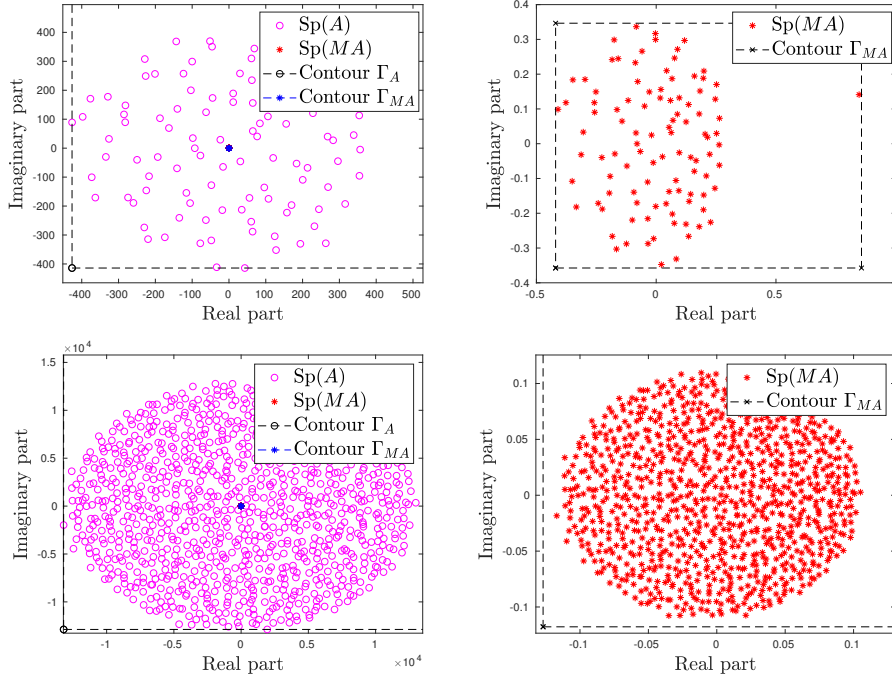


FIG. 3. **Experiment 2.**  $Sp(A)$  and  $Sp(MA)$ , and rectangular contour: (Top-Left) :  $n = 101$ ; (Top-Right) : zoom for  $n = 101$ ; (Bottom-Left)  $n = 1001$ ; (Bottom-Right) : zoom for  $n = 1001$ .

243 contour allows for a slightly more precise estimate of the Cauchy integral than for the  
 244 non-preconditioned circular one. This is mainly due to the structure of the spectrum  
 245 which is concentrated around 0. As a consequence, the rectangle contour is very thin,  
 246 then leading to a more accurate computation of the approximate operator  $A_{\mathfrak{h}}^{\alpha}$ . The  
 247 choice of the contour is naturally highly correlated to the structure of the spectrum.  
 248 **Experiment 3bis.** To complete the illustrations, let us consider the matrix  $A =$   
 249  $B + 0.75B^T + C$ , where  $B_{ij} = n\text{rand}(0, 1)$  and  $C_{ij} = 20n + \text{rand}(0, 1)\delta_{ij}$ ,  $1 \leq i, j \leq n$ ,  
 250 for  $n = 100$ . The matrix  $A$  has a complex-valued spectrum. For  $\alpha = 0.9$ ,  $Sp(A)$  is  
 251 reported in Fig. 5 (Left) and a zoom on  $Sp(MA)$  is given in Fig. 5 (Right). We  
 252 observe that the circular contour is more efficient here than the rectangular one (see  
 253 Fig. 6).

254 **4.6. Selection of the parameter  $k$  in the Cauchy integral formulation**  
 255 (1.2). We discuss now the selection of the Cauchy integral formulation, and more  
 256 specifically the value of  $k \in \mathbb{N}$  in formula (1.2). Since  $z \in \Gamma_A$ , we have  $|z| > \rho(A)$ ,  
 257 where  $\rho(A)$  denotes the spectral radius of  $A$ . Denoting by  $A_{\mathfrak{h}}$  the approximate Cauchy  
 258 integral using an order  $\sigma$ -composite-quadrature rule with  $\mathfrak{h} = \sup_j |z_{j+1} - z_j|$ , there  
 259 exists  $c = c(A, \sigma) > 0$  such that

$$260 \quad (4.6) \quad \frac{\|A_{\text{ref}}^{\alpha} - A_{\mathfrak{h}}^{\alpha}\|}{\|A_{\text{ref}}^{\alpha}\|} \leq c\mathfrak{h}^{\sigma} \sup_{z \in \Gamma_A} \frac{\left\| \frac{d^{\sigma}}{dz^{\sigma}} z^{\alpha-k} (zI - A)^{-1} \right\|}{\|A_{\text{ref}}^{\alpha}\|}.$$

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

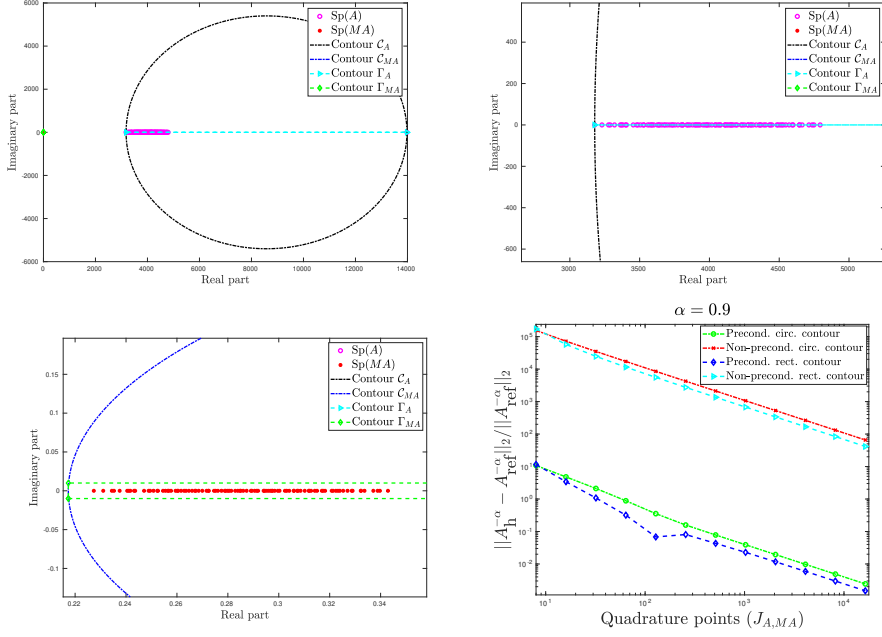


FIG. 4. **Experiment 3.** (Top-Left)  $Sp(A)$  and  $Sp(MA)$ , with  $A \in \mathbb{R}^{n \times n}$ , and  $C_A$ ,  $C_{MA}$ ,  $\Gamma_A$ ,  $\Gamma_{MA}$ . (Top-Right) zoom on  $Sp(A)$ . (Bottom-Left) zoom on  $Sp(MA)$ . (Bottom-Right) Relative error vs the number of integration points for  $\alpha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

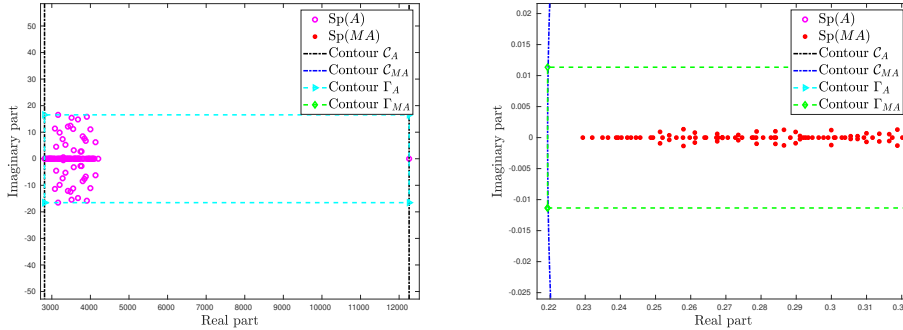


FIG. 5. **Experiment 3bis.** (Left)  $Sp(A)$ ,  $Sp(MA)$  with  $A \in \mathbb{R}^{n \times n}$ ,  $C_A$ ,  $C_{MA}$  and  $\Gamma_A$ ,  $\Gamma_{MA}$ . (Right) Zoom on  $Sp(MA)$ .

263 However, whenever  $\rho(A)$  is small, a natural choice in relation (1.2) is  $k = 0$ . Indeed,  
 264 in this case, as  $|z|$  is larger but close to  $\rho(A)$ , a small error (4.6) is expected and taking  
 265  $k < \alpha$  could even deteriorate the approximation. For instance, it looks reasonable to  
 266 use (1.2) with  $k = \lceil \alpha \rceil$  for a direct evaluation of  $A^\alpha$  and to use  $k = 0$  for evaluating  
 267  $(MA)^\alpha$  when  $M$  is an accurate (in the sense that  $\rho(MA)$  is very small, typically  
 268  $< 1$ ) Cauchy integral preconditioner. If  $\rho(MA)$  is still larger than 1, it is preferable  
 269 (theoretically) to take  $k = \lceil \alpha \rceil$  to evaluate  $(MA)^\alpha$ . In the following, we arbitrary fix  
 270  $k = 1$  (or  $k = 0$ ), as most of the computations are done for  $0 < \alpha < 1$  (or  $1 < \alpha < 2$ )  
 271 and that  $\rho(MA)$  will still be large enough to justify the fact that  $k = \lceil \alpha \rceil$  provides a

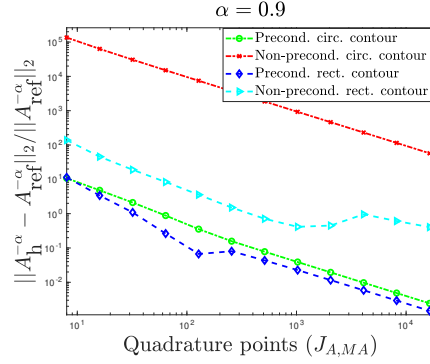


FIG. 6. *Experiment 3bis*. Relative error ( $\alpha = 0.9$ ) vs the number of integration points.

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

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

$$277 \quad (4.7) \quad A^\alpha = \frac{1}{2i\pi} \int_{\Gamma_A} z^\alpha (zI - A)^{-1} dz,$$

278 and  $k = 1$

$$279 \quad (4.8) \quad A^\alpha = \frac{A}{2i\pi} \int_{\Gamma_A} z^{\alpha-1} (zI - A)^{-1} dz.$$

280 We consider a circular contour where the number of quadrature nodes varies between  
 281 2 and 4096, and report in Fig. 7 the convergence of  $\|A_h^\alpha - A_{\text{ref}}^\alpha\|_2 / \|A_{\text{ref}}^\alpha\|_2$  for  $k = 0, 1$ ,  
 282 in the non-preconditioned case, vs the number of quadrature points. We notice that  
 283 taking  $k = 0$  or  $k = 1$  does not impact the behavior of the error.

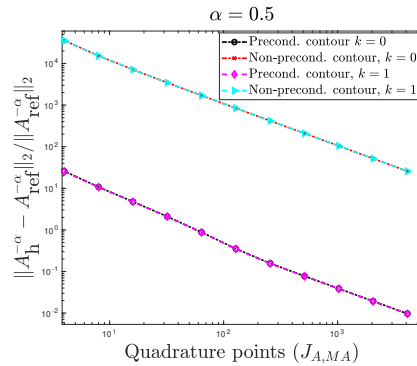


FIG. 7. *Experiment 4*. Relative error ( $\alpha = 0.5$ ) vs the number of integration points.

284 **5. Fractional linear systems  $A^\alpha u = f$ .** In the previous subsections, we devel-  
 285 oped an efficient methodology to estimate the real power of a matrix. In this paper, we

286 are more specifically interested in the solution to *fractional linear systems*  $A^\alpha u = f$ ,  
 287 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  
 288  $u = A^{-\alpha} f$ .

289 **5.1. Solution to fractional linear systems  $A^\alpha u = f$ , with  $[M, A] = 0$ .** We  
 290 assume here that  $M^\alpha$  can efficiently be estimated numerically. If not, it is then more  
 291 appropriate to proceed as in Subsection 5.2. We recall that for any matrix  $M$  such  
 292 that  $(MA)^\alpha = M^\alpha A^\alpha$ , we can compute  $A^{-\alpha} f$ , from  $(MA)^{-\alpha} f$ , and

$$293 \quad (5.1) \quad u = A^{-\alpha} f = M^\alpha (MA)^{-\alpha} f,$$

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

$$296 \quad u = A^{-\alpha} f = (2\pi i)^{-1} \int_{\Gamma_A} z^{-\alpha} (zI - A)^{-1} f dz$$

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

$$301 \quad (MA)^{-\alpha} f = (2\pi i)^{-1} \int_{\Gamma_{MA}} z^{-\alpha} (zI - MA)^{-1} f dz.$$

302 Computed on a finite grid  $\Gamma_{MA}^{(h)} \subseteq \Gamma_{MA}$ , with spatial resolution  $h = \max_{1 \leq j \leq J_{MA}} h_j$   
 303 and a quadrature of order  $\sigma$ , the approximate Cauchy integral to  $(MA)^{-\alpha}$  is denoted  
 304 by  $S_h^{(-\alpha)} \approx (MA)^{-\alpha}$  and is defined as

$$305 \quad (5.2) \quad S_h^{(-\alpha)} = (2\pi i)^{-1} \sum_{1 \leq j \leq J_{MA}} h_j w_j z_j^{-\alpha} (z_j I - MA)^{-1},$$

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

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

$$309 \quad (5.3) \quad \begin{aligned} u_h &:= M^\alpha S_h^{(-\alpha)} f = (2\pi i)^{-1} M^\alpha \sum_{1 \leq j \leq J_{MA}} h_j w_j z_j^{-\alpha} u_j, \\ (z_j I - MA) u_j &= f, \text{ for all } 1 \leq j \leq J_{MA}, \end{aligned}$$

310 i.e.  $u_h = M^\alpha S_h^{(-\alpha)} f$ .

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

$$314 \quad \begin{aligned} u_h &= (2\pi i)^{-1} M^\alpha \sum_{1 \leq j \leq J_{MA}} \delta\theta_j w_j r_\varepsilon^{(MA)} e^{i\theta_j} (r_\varepsilon^{(MA)})^{-\alpha} e^{-i\alpha\theta_j} u_j, \\ (r_\varepsilon^{(MA)} e^{i\theta_j} I - MA) u_j &= f, \text{ for all } 1 \leq j \leq J_{MA}. \end{aligned}$$

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

$$318 \quad (5.4) \quad S_h^{(-\alpha)} f = (2\pi i)^{-1} \sum_{1 \leq j \leq J_{MA}} h_j w_j z_j^{-\alpha} (z_j I - MA)^{-1} f.$$

319 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  
 320 the use of a Cauchy integral preconditioner  $M$ . Let us remark that when  $MA$  can  
 321 be analytically diagonalized, the matrix power can be very efficiently computed, as  
 322 stated in the following proposition.

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

$$325 \quad S_{\mathfrak{h}}^{(\alpha)} f = (2\pi\mathfrak{i})^{-1} P_{MA} \left[ \sum_{1 \leq j \leq J_{MA}} \mathfrak{h}_j w_j z_j^{-\alpha} (z_j I - D_{MA})^{-1} \right] P_{MA}^{-1} f,$$

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

330 **Proof.** Since  $MA$  is diagonalizable, we have

$$331 \quad \begin{aligned} (MA)^\alpha &= (2\pi\mathfrak{i})^{-1} \int_{\Gamma} z^\alpha (zI - MA)^{-1} dz = (P_{MA}D_{MA}P_{MA}^{-1})^\alpha \\ &= P_{MA}(2\pi\mathfrak{i})^{-1} \int_{\Gamma} z^\alpha (zI - D_{MA})^{-1} dz P_{MA} = P_{MA}D_{MA}^\alpha P_{MA}^{-1}. \end{aligned}$$

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

$$333 \quad S_{\mathfrak{h}}^{-\alpha} = (2\pi\mathfrak{i})^{-1} P_{MA} \left[ \sum_{1 \leq j \leq J_{MA}} \mathfrak{h}_j w_j z_j^{-\alpha} (z_j I - D_{MA})^{-1} \right] P_{MA}^{-1},$$

334 which concludes the proof.  $\square$

335

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

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

$$351 \quad \text{cond}(N_j(z_j I - A)) \approx \text{cond}(N_j(z_{j'} I - A)) \ll \text{cond}(z_{j'} I - A).$$

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

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

357 **5.2. Solution to fractional linear systems**  $A^\alpha u = f$ , with  $[M, A] \neq 0$ . The  
358 most general and interesting case occurs when  $A$  and  $M$  do not commute. Then,  
359 we can no longer directly deduce the solution to  $A^\alpha u = f$ , from the solution to  
360  $M^\alpha(MA)^{-\alpha}f$ . The natural procedure then consists in solving

$$361 \quad M^\alpha(MA)^{-\alpha}A^\alpha u = M^\alpha(MA)^{-\alpha}f,$$

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

$$369 \quad (5.5) \quad \begin{aligned} v_{\mathfrak{h}} &= (2\pi\mathbf{i})^{-1}M^\alpha \sum_{1 \leq j \leq J_{MA}} \mathfrak{h}_j w_j z_j^{-\alpha} u_j, \\ (z_j I - MA)u_j &= x, \text{ for } 1 \leq j \leq J_{MA}. \end{aligned}$$

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

373 **5.3. Jacobi Cauchy integral preconditioner.** Let us consider a Jacobi pre-  
374 conditioner, assuming that  $A$  is *diagonally dominant* and that  $A_{ii} \neq 0$ , for all  $1 \leq i \leq$   
375  $n$ . Setting  $M = \text{diag}(A_{11}^{-1}, \dots, A_{nn}^{-1})$ , we then have

$$376 \quad (MA)^\alpha = (2\pi\mathbf{i})^{-1} \int_{\Gamma_{MA}} z^\alpha (zI - MA)^{-1} dz.$$

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

381 **5.4. ILU Cauchy integral preconditioner.** Incomplete-LU factorizations ap-  
382 pear as some natural candidates for solving fractional linear systems for two main  
383 reasons. First, they usually allow for a better preconditioning than Jacobi. Secondly,  
384 the triangular structure of the  $L$  and  $U$  matrices leads to an efficient computation of  
385 intermediate sparse linear systems. More specifically, we propose the following ap-  
386 proach. We first implement an ILU-factorization  $\tilde{L}\tilde{U}$  of the matrix  $A$ , with a threshold  
387 parameter  $\zeta > 0$ , and formally denote  $M = (\tilde{L}\tilde{U})^{-1}$ . In addition to (5.5), it is needed  
388 to approximate  $M^\alpha v_{\mathfrak{h}}$ . In this goal, and unlike Jacobi preconditioning, it is necessary  
389 to solve additional triangular linear systems, i.e. we approximate  $M^\alpha v_{\mathfrak{h}}$ , by  $w_{\mathfrak{h}}$  such  
390 that

$$391 \quad (5.6) \quad \begin{aligned} w_{\mathfrak{h}} &= (2\pi\mathbf{i})^{-1} \sum_{1 \leq j \leq J_M} \mathfrak{h}_j w_j z_j^{-\alpha} v_j, \\ (z_j \tilde{L}\tilde{U} - I)v_j &= \tilde{L}\tilde{U}v_{\mathfrak{h}}, \text{ for } 1 \leq j \leq J_M. \end{aligned}$$

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

397 **5.5. Parallelization aspects.** The computation of  $(MA)^{-\alpha}$  can then be per-  
 398 formed in parallel as follows. For  $p$  processors, we decompose  $\Gamma$  in  $p$  subcontours  $\Gamma_\ell$ :  
 399  $\Gamma = \cup_{\ell=1}^p \Gamma_\ell$  and  $\ell(\Gamma_\ell) = \ell(\Gamma)/p$  and write

$$400 \quad (MA)^{-\alpha} = \sum_{\ell=1}^p (MA)_\ell^{-\alpha} = \sum_{\ell=1}^p (2\pi i)^{-1} \int_{\Gamma_\ell} z^{-\alpha} (zI - MA)^{-1} dz.$$

401 We first implement an ILU-factorization and construct  $\tilde{L}$  and  $\tilde{U}$ . For any fixed value  
 402 of  $\ell$ ,

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

406 **5.6. Numerical experiments on fractional linear systems.** We provide  
 407 now a few examples of numerical simulations to illustrate the methodology.

408 **Experiment 5.** In this example, we compare the efficiency of the different pre-  
 409 conditioners implemented in GMRES for solving (1.1), where  $f$  is the unit vector.  
 410 We report the convergence rate, represented as the residual history vs the GMRES  
 411 iteration, where the solution is computed from

- 412 • a direct evaluation of the Cauchy integral without preconditioning (labelled  
 413 No-precond.),
- 414 • by using an ILU preconditioner  $M^{-\alpha}(MA)^\alpha$ , with  $M = \tilde{L}\tilde{U}$  for a drop toler-  
 415 ance at  $10^{-4}$ , and a rectangular (ILU-precond. rect.) and circular contours  
 416 (ILU-precond. circ.),
- 417 • with an ILU preconditioner  $M$  directly built on the sparse matrix  $A$ , and  
 418 then the preconditioner  $M^\alpha$  is used on  $A^\alpha$  (and denoted  $M_\alpha$ -precond.),
- 419 • and finally with an ILU preconditioner directly constructed from the full  
 420 matrix  $A^\alpha$  that we assume to be given (ILU-precond. on  $A^\alpha$ ).

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

$$422 \quad B_{ii} = 75\mathbf{rand}(0, 1) + 15, \quad B_{ii\pm 1} = 5\mathbf{rand}(0, 1) \mp 8, \quad B_{ii\pm 2} = \mathbf{rand}(0, 1) \mp 1/2,$$

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

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

$$437 \quad u_{\mathfrak{h}} = (2\pi)^{-1} M^\alpha(MA) \sum_{1 \leq j \leq J_{MA}} \delta\theta w_j r_\varepsilon^{(A)} e^{i\theta_j} (r_\varepsilon^{(A)})^{-\alpha-1} e^{-i(\alpha+1)\theta_j} u_j,$$

$$((z_c + r_\varepsilon^{(MA)} e^{i\theta_j})I - MA)u_j = f, \text{ for } 1 \leq j \leq J_{MA},$$

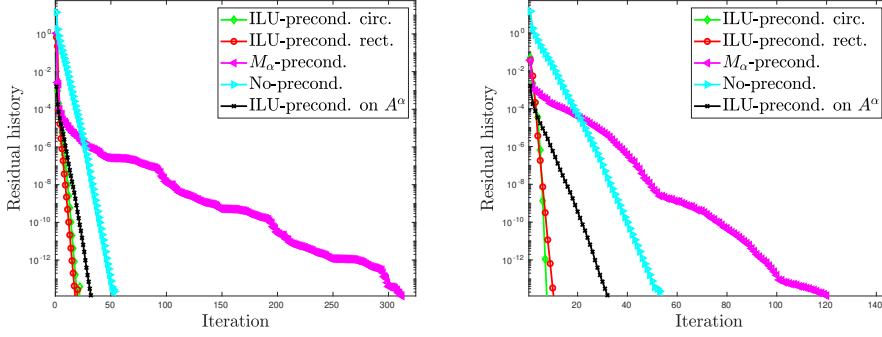


FIG. 8. **Experiment 5.** Comparison of the residual history vs iterations of the GMRES algorithm (restarted at 50 iterations, and tolerance  $10^{-15}$ ) for various preconditioners: ILU Cauchy integral preconditioner (threshold at  $10^{-4}$ ), ILU-preconditioner on  $A^\alpha$ . (Left):  $J_{A,MA}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

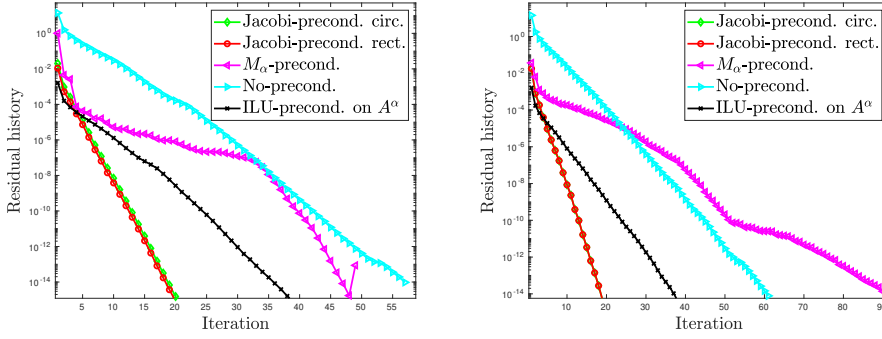


FIG. 9. **Experiment 5.** Comparison of the residual history vs number of iterations of the GMRES (restarted after 50 iterations, and for a tolerance  $10^{-15}$ ) for different preconditioners: Jacobi preconditioner, ILU-preconditioner on  $A^\alpha$ . (Left):  $J_{A,MA} = 8$  (Right):  $J_{A,MA} =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

438 with  $u_h \approx u = A^{-\alpha} f$ . Let us recall that  $r_\varepsilon^{(A)} = r^{(A)} + \varepsilon$  and that the initial guess  
 439 for computing  $u_{j+1}$  is taken as  $u_j$ . We report in Figs. 10 (Top/Bottom Left) the  
 440 2-norm error  $\|u_h - u_{\text{ref}}\|_2$  (in logscale) as a function of  $J_{A,MA}$ . We also provide the  
 441 corresponding CPU-time with/without Jacobi preconditioning, as well as  $\|A_h^{-\alpha} -$   
 442  $A^{-\alpha}\|_2$ , where we have numerically estimated  $A_h^{-\alpha}$  from a direct (D) computation  
 443  $(A_h^{(D)})^{-\alpha}$  such that ( $k = 1$  in relation (1.2))

$$444 \quad (5.7) \quad (A_h^{(D)})^{-\alpha} = (2\pi i)^{-1} A \sum_{1 \leq j \leq J_A} h_j w_j z_j^{-\alpha-1} (z_j I - A)^{-1},$$

445 or with a preconditioning  $(A_h^{(P)})^{-\alpha}$ , from

$$446 \quad (5.8) \quad (A_h^{(P)})^{-\alpha} = (2\pi i)^{-1} M A \sum_{1 \leq j \leq J_{MA}} h_j w_j z_j^{-\alpha-1} (z_j I - M A)^{-1}.$$

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



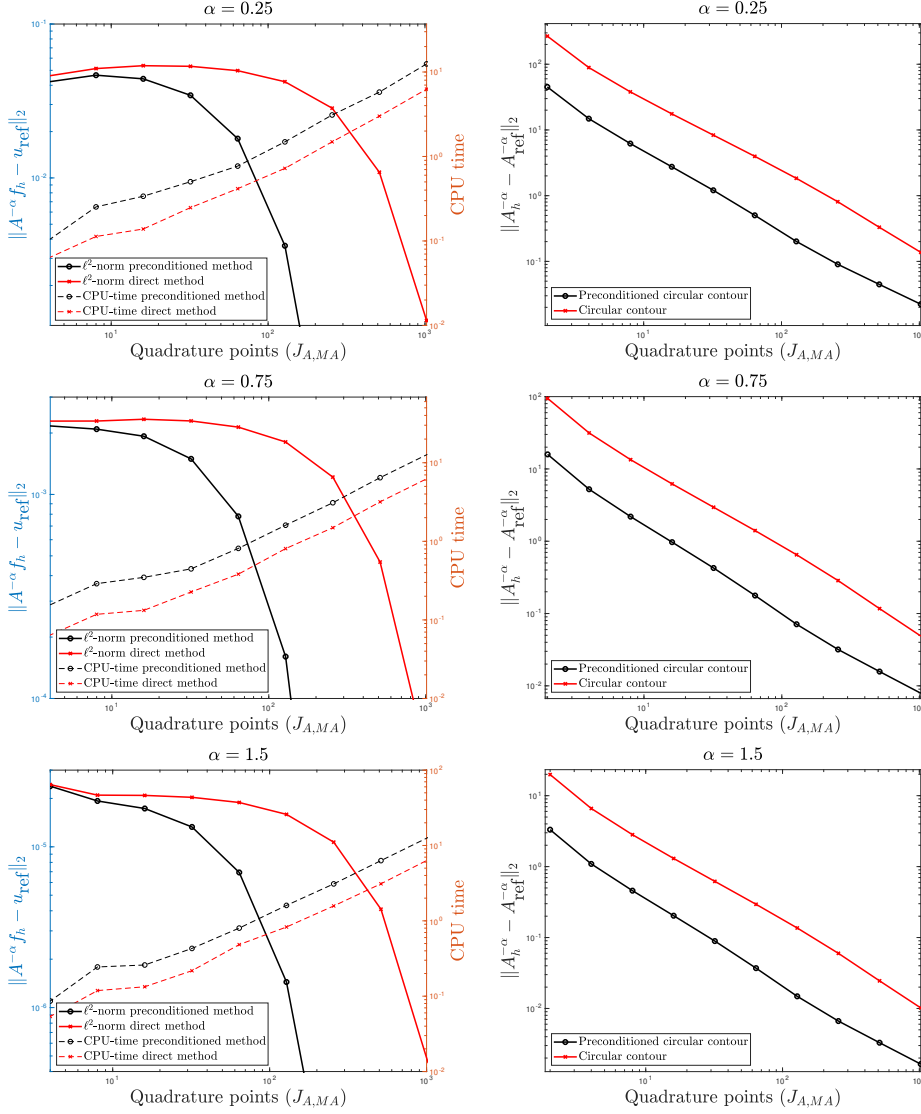


FIG. 10. **Experiment 6.** (Top-Left) CPU-time (in seconds) in logscale, and  $\|A^{-\alpha} f_h - u_{ref}\|_2$ , where  $A^{-\alpha} f_h = u_h$ , as a function of the number of quadrature points  $J_{A,MA}$ , with  $\alpha = 0.25$ , (Top-Right)  $\|A_h^{-\alpha} - A_{ref}^{-\alpha}\|_2$  in logscale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

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

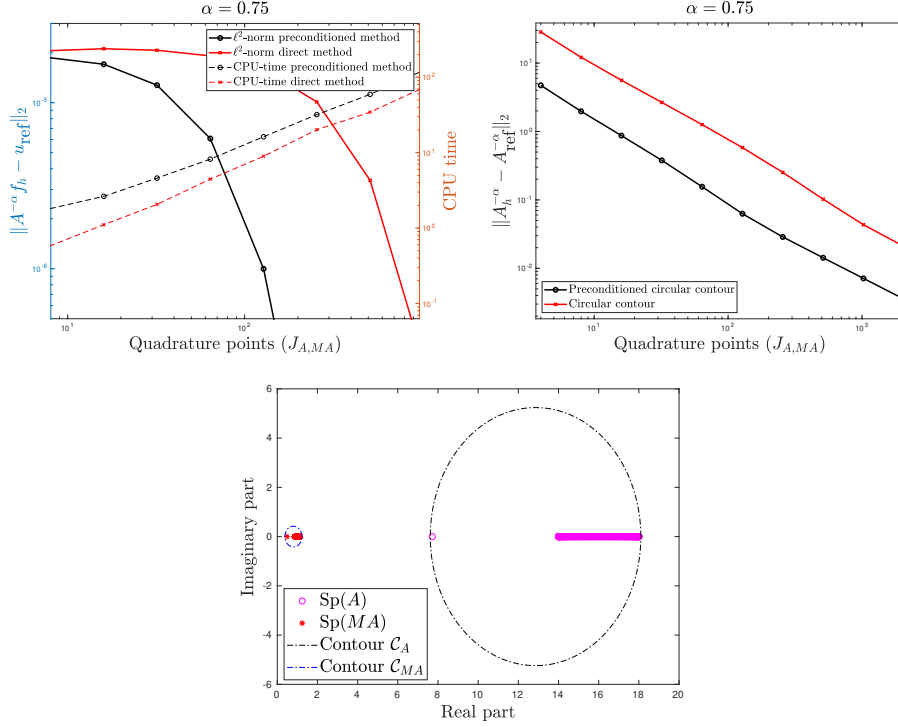


FIG. 11. **Experiment 6.** (Top-Left) CPU-time (in seconds) in logscale, and  $\|A^{-\alpha} f_h - u_{\text{ref}}\|_2$  as a function of the number of quadrature points  $J_{A,MA}$ , with  $\alpha = 0.75$ , (Top-Right)  $\|A_h^{-\alpha} - A_{\text{ref}}^{-\alpha}\|_2$  in logscale as a function of the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

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

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

454

455 We consider  $A$  as a 3-point approximation of the Laplace operator on a segment  
456  $] - 1; 1[$ , with  $n = 101$  grid-points. We use some circular contours for both the non-  
457 preconditioned and preconditioned Cauchy integrals. In Fig. 12 (left), we report in  
458 logscale i) the CPU-time (in seconds) for the direct method (with  $\mathcal{C}_A$ ) and double-  
459 preconditioned method (with  $\mathcal{C}_{M_\tau A}$ ), and ii)  $\|u_h - u_{\text{ref}}\|_2$ . We more precisely compare  
460 a Jacobi Cauchy integral preconditioner with a differential-based preconditioner  $M_\tau$   
461 with  $\tau = 8 \times 10^{-1}$ ,  $\tau = 9 \times 10^{-1}$ ,  $\tau = 1$  and  $\tau = 1.2$ , and with a direct integral compu-  
462 tation without preconditioner. We also use a Cauchy ILU-preconditioner ( $\tilde{L}\tilde{U}$ ) with a  
463 drop-tolerance fixed to  $10^{-1}$ , although in this case  $[(\tilde{L}\tilde{U})^{-1}, A]$  is not necessarily close  
464 to zero. We also report  $\|A_h^{-\alpha} - A_{\text{ref}}^{-\alpha}\|_2$  in logscale in Fig. 12 (Right). The test illus-

465 trates that for a moderately dominant diagonal matrix, the differential-based preconditioning may be an alternative to Jacobi preconditioning, but an ILU-factorization  
 466 can be used as well, if the drop tolerance is small enough.

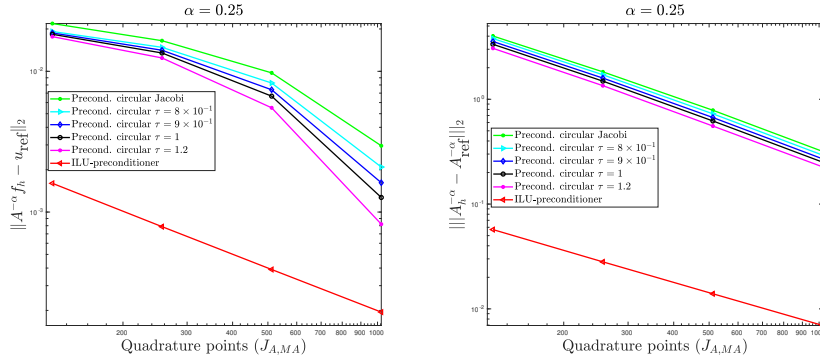


FIG. 12. **Experiment 7.** Jacobi preconditioner, differential-based preconditioner  $M_\tau$  with  $\tau = 8 \times 10^{-1}$ ,  $\tau = 9 \times 10^{-1}$ ,  $\tau = 1$ , and  $\tau = 1.2$  and ILU-preconditioner with a drop tolerance at  $10^{-1}$ . (Left) In logscale  $\|A^{-\alpha} f_h - u_{ref}\|_2$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

467

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

$$475 \quad (6.1) \quad \begin{aligned} -(-\Delta)^\alpha u &= f, \text{ in } \Omega, \\ u &= 0, \text{ on } \partial\Omega, \end{aligned}$$

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

$$484 \quad (6.2) \quad (-\Delta)^\alpha u(\mathbf{x}) = C(\alpha) \text{p.v.} \int_{\mathbb{R}^2} \frac{u(\mathbf{x}) - u(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^{2+2\alpha}} d\mathbf{y} = C(\alpha) \lim_{\varepsilon \rightarrow 0^+} \int_{B_\varepsilon(\mathbf{x})} \frac{u(\mathbf{x}) - u(\mathbf{y})}{|\mathbf{x} - \mathbf{y}|^{2+2\alpha}} d\mathbf{y},$$

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

$$486 \quad (6.3) \quad C(\alpha) := \left( \int_{\mathbb{R}^2} \frac{1 - \cos(\xi_1)}{|\xi|^{2+2\alpha}} d\xi \right)^{-1}.$$

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

$$489 \quad (6.4) \quad (-\Delta)^\alpha u(\mathbf{x}) = -\frac{1}{2}C(\alpha)\text{p.v.} \int_{\mathbb{R}^2} \frac{u(\mathbf{x} + \mathbf{y}) - 2u(\mathbf{x}) + u(\mathbf{x} - \mathbf{y})}{|\mathbf{y}|^{2+2\alpha}} d\mathbf{y}.$$

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

492 **6.1. Fractional laplacian approximation.** For the 2d computational domain  
 493  $\Omega := \prod_{\ell=1}^2 [-L_\ell; L_\ell]$ , we introduce the inner uniform cartesian grid  $\Omega_{\mathbf{h}}$ , with  $n :=$   
 494  $\prod_{k=1}^2 N_k$  total discretization points, defined by  $\Omega_{\mathbf{h}} = \{\mathbf{x}_{i,j} = (x_{1,i}, x_{2,j})\}_{(i,j) \in \mathcal{I}}$ , with  
 495  $x_{1,i} := -L_1 + ih_1$ ,  $x_{2,j} := -L_2 + jh_2$ ,  $\mathcal{I} := \{(i, j) \in \mathbb{N}^2 \text{ such that } 1 \leq i \leq N_1, 1 \leq j \leq$   
 496  $N_2\}$ , setting  $h_\ell := 2L_\ell/(N_\ell + 1)$ ,  $\ell = 1, 2$ , and  $\mathbf{h} := (h_1, h_2)$ . When all the uniform  
 497 discretization steps are equal along the directions, we define :  $h := h_1 = h_2$ , and then  
 498  $n = N^2$ , with  $N := N_1 = N_2$ .

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

$$502 \quad \begin{cases} -\Delta_{h_1} \varphi_{i,j} = \frac{\varphi_{i+2,j} - 16\varphi_{i+1,j} + 30\varphi_{i,j} - 16\varphi_{i-1,j} + \varphi_{i-2,j}}{12h_1^2}, \\ -\Delta_{h_2} \varphi_{i,j} = \frac{\varphi_{i,j+2} - 16\varphi_{i,j+1} + 30\varphi_{i,j} - 16\varphi_{i,j-1} + \varphi_{i,j-2}}{12h_2^2}. \end{cases}$$

503 A fourth-order approximation of the laplacian is then:  $\Delta_h u_h := (\Delta_{h_1} + \Delta_{h_2})u_h$ . Let  
 504  $f_h = \{f_{i,j}\}_{(i,j) \in \mathcal{I}}$  be the projection of the function  $f$  on  $\Omega_{\mathbf{h}}$ , such that  $f_{i,j} = f(\mathbf{x}_{i,j})$ ,  
 505  $(i, j) \in \mathcal{I}$ . Any other real space method (e.g. finite volume or finite element) could  
 506 also be used within the method developed below. The approximate solution to system  
 507 (6.1) is obtained by solving the fractional linear system  $A_h^\alpha u_h = f_h$ , corresponding  
 508 to the discrete operator  $-(\Delta_h)^\alpha$ . Let us assume that the approximation of  $\Delta$  is at  
 509 order  $q$  with discretization step  $h$  on the bounded domain  $\Omega_{\mathbf{h}}$ . The construction to  
 510 the approximate solution  $u_h$  is performed by computing

$$511 \quad (6.5) \quad u_h = A^{-\alpha} f_h.$$

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

522 **6.2. Computational complexity analysis in 2d.** We recall that the frac-  
 523 tional laplacian can also be rewritten [4] under the form (6.2), for  $\alpha \in (0, 1)$  and any  
 524  $u \in \mathcal{S}(\mathbb{R}^2)$ . A direct finite-difference approximation to (6.2) on a  $n$ -grid  $\Omega_{\mathbf{h}} = \{\mathbf{x}_{i,j} =$   
 525  $(x_{1,i}, x_{2,j}) : 1 \leq i \leq N, 1 \leq j \leq N\}$  reads

$$526 \quad (6.6) \quad \mathcal{A}_\alpha u_h = f_h,$$

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

$$529 \quad (6.7) \quad -(-\Delta)^\alpha u(\mathbf{x}_{i;j}) \approx \frac{1}{2} C(\alpha) \sum_{k=1}^N \sum_{l=1}^N \frac{u_{i+k;j+l} - 2u_{i;j} + u_{i-k;j-l}}{|\mathbf{y}_{k;l}|^{2+2\alpha}} h_1 h_2.$$

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

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

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

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

$$547 \quad (6.8) \quad \begin{aligned} -(-\Delta)^\alpha u &= f, \text{ in } \Omega, \\ u &= 0, \text{ on } \partial\Omega, \end{aligned}$$

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

$$550 \quad S_{\mathbf{h}}^{(-\alpha)} = (2\pi\mathbf{i})^{-1} M A \sum_{1 \leq j \leq J_{MA}} \mathfrak{h}_j w_j z_j^{-\alpha-1} (z_j I - M A)^{-1}.$$

551 Therefore,  $u_{h;\mathbf{h}}$  is an approximation of the solution  $u_h = A^{-\alpha} f_h$ , the latter being itself  
 552 an approximation to the solution  $u$  to system (6.8). In the sequel, we need the follow-  
 553 ing discrete norms: for  $v \in \ell^\infty(\Omega_{\mathbf{h}})$ ,  $\|v\|_{\ell^\infty(\Omega_{\mathbf{h}})} := \max_{1 \leq i \leq N_1; 1 \leq j \leq N_2} |v(x_{1,i}, x_{2,j})|$ ,  
 554 and for  $v \in \ell^2(\Omega_{\mathbf{h}})$ :  $\|v\|_{\ell^2(\Omega_{\mathbf{h}})} := (h_1 h_2 \sum_{1 \leq i \leq N_1; 1 \leq j \leq N_2} |v(x_{1,i}, x_{2,j})|^2)^{1/2}$ .

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

$$560 \quad \begin{aligned} u_{h;\mathbf{h}} &:= (2\pi\mathbf{i})^{-1} M^\alpha (M A) \sum_{1 \leq j \leq J_{MA}} \mathfrak{h}_j w_j z_j^{-\alpha-1} u_j, \\ (z_j I - M A) u_j &= f_h, \text{ for } 1 \leq j \leq J_{MA}, \end{aligned}$$

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

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

$$568 \quad (6.9) \quad \|u - u_{h;h}\|_{\ell^2(\Omega_h)} \leq C \max_{1 \leq j \leq J_{MA}} |\mathfrak{h}_j|^\sigma \|f_h\|_{\ell^2(\Omega_h)} + D(h_1 h_2)^{q\alpha}.$$

569 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  
 570  $O(J_A n^{\beta_A})$  operations, with  $1 < \beta_A < 3$ . By using a *Cauchy integral pre-*  
 571 *conditioner*  $M$ , only  $J_{MA} \ll J_A$  linear systems have to be solved along  
 572  $\Gamma_{MA}$ . Performing  $p$  (parallel) ILU-factorizations  $N_j$  on  $z_j I - A$  such that  
 573  $\text{cond}(N_j(z_j I - MA)) \ll \text{cond}(z_j I - MA)$ , the overall computational com-  
 574 plexity of the double-preconditioning method is at most  $\mathcal{O}(J_{MA} n^{\beta_{\text{ILU}}})$ , with  
 575  $\beta_{\text{ILU}} \gtrsim 1$  thanks to the cost for building the ILU-preconditioners.  
 576

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

$$578 \quad (6.10) \quad u_h = A^{-\alpha} f_h = (2\pi i)^{-1} A \int_{\Gamma_A} z^{-\alpha-1} (zI - A)^{-1} f_h dz.$$

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

$$581 \quad S_h^{(-\alpha)} = (2\pi i)^{-1} MA \sum_{1 \leq j \leq J_{MA}} \mathfrak{h}_j w_j z_j^{-\alpha-1} (z_j I - MA)^{-1}.$$

582 In addition, one gets

$$583 \quad (MA)^{-\alpha} f_h = (2\pi i)^{-1} MA \int_{\Gamma_{MA}} z^{-\alpha-1} (zI - MA)^{-1} f_h dz.$$

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

$$585 \quad (6.11) \quad \|S_h^{(-\alpha)} - (MA)^{-\alpha}\|_2 \leq C_1 \max_{1 \leq j \leq J_{MA}} |\mathfrak{h}_j|^\sigma.$$

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

$$588 \quad \begin{aligned} \|u_{h;h} - A^{-\alpha} f_h\|_{\ell^2(\Omega_h)} &= \|M^\alpha S_h^{-\alpha} f_h - M^\alpha (MA)^{-\alpha} f_h\|_{\ell^2(\Omega_h)} \\ &= \|M^\alpha (S_h^{-\alpha} - (MA)^{-\alpha}) f_h\|_{\ell^2(\Omega_h)} \\ &\leq \|M^\alpha\|_2 \times \|S_h^{(-\alpha)} - (MA)^{-\alpha}\|_2 \times \|f_h\|_{\ell^2(\Omega_h)}. \end{aligned}$$

589 From (6.11), we prove that there exists a positive constant  $C = C(\alpha, \Omega, A, M, \Gamma_{MA}) >$   
 590  $0$  such that:  $\|u_{h;h} - A^{-\alpha} f_h\|_{\ell^2(\Omega_h)} \leq C \max_{1 \leq j \leq J_{MA}} |\mathfrak{h}_j|^p \|f_h\|_{\ell^2(\Omega_h)}$ . Next, according  
 591 to [9], one can find  $D = D(f, \alpha, A, \Omega) > 0$  such that:  $\|u - A^{-\alpha} f_h\|_{\ell^2(\Omega_h)} \leq D(h_1 h_2)^{q\alpha}$ .  
 592 We finally have

$$593 \quad \begin{aligned} \|u - u_h\|_{\ell^2(\Omega_h)} &\leq \|u_{h;h} - A^{-\alpha} f_h\|_{\ell^2(\Omega_h)} + \|u - A^{-\alpha} f_h\|_{\ell^2(\Omega_h)} \\ &\leq C \max_{1 \leq j \leq J_{MA}} |\mathfrak{h}_j|^\sigma \|f_h\|_{\ell^2(\Omega_h)} + D(h_1 h_2)^{q\alpha}. \end{aligned}$$

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

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

599

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

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

$$605 \quad (zI - MA)^{-1} = \frac{1}{2} \int_{-1}^1 ((re^{i\pi\theta} + z_c + z)I - MA)^{-1} \frac{e^{i\pi\theta}}{(e^{i\pi\theta} + z_c e^{2i\pi\theta}/r)} d\theta.$$

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

$$607 \quad \sigma_j^{-1} = \begin{cases} r_{MA} e^{-i\pi x_j}, & k = 1, \dots, p_{MA}, \\ r_{MA} e^{-i\pi x_{j-p}}, & k = p_{MA} + 1, \dots, 2p_{MA} = J_{MA}, \end{cases}$$

608 and

$$609 \quad \tilde{\sigma}_j = \begin{cases} \sigma_{j+p_{MA}}^{-1}, & k = 1, \dots, p_{MA}, \\ \sigma_{j-p_{MA}}^{-1}, & k = p_{MA} + 1, \dots, 2p_{MA} = J_{MA}. \end{cases}$$

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

$$612 \quad (\tilde{\sigma}_j I - A) \approx \begin{cases} L_{j+p_{MA}} U_{j+p_{MA}}, & j = 1 \dots, p_{MA}, \\ L_{j-p_{MA}} U_{j-p_{MA}}, & j = p_{MA} + 1 \dots, 2p_{MA}. \end{cases}$$

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

615 We can extend the methodology to equations of the form

$$616 \quad (6.12) \quad \begin{aligned} -(-\Delta)^\alpha u + Vu &= f, \text{ in } \Omega, \\ u &= 0, \text{ on } \partial\Omega, \end{aligned}$$

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

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

$$625 \quad \begin{aligned} g_{\mathbf{h}} &:= (2\pi i)^{-1} A \sum_{1 \leq j \leq J_A} \mathfrak{h}_j w_j z_j^{-\alpha-1} g_j, \\ (z_j I - A)g_j &= f_h, \text{ for all } 1 \leq j \leq J_A, \end{aligned}$$

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

630 2. Similarly,  $B_{\mathbf{h}}$  is an approximation to  $A^{-\alpha}V_{\mathbf{h}}$

$$631 \quad B_{\mathbf{h}}^{(i)} := (2\pi\mathbf{i})^{-1}A \sum_{1 \leq j \leq J_A} \mathfrak{h}_j w_j z_j^{-\alpha-1} v_j^{(i)},$$

$$(z_j I - A)v_j^{(i)} = V_{\mathbf{h}}^{(i)}, \text{ for all } 1 \leq j \leq J_A,$$

632 where  $V_{\mathbf{h}} = [V_{\mathbf{h}}^{(1)} \cdots V_{\mathbf{h}}^{(n)}] \in \mathbb{R}^{n \times n}$  (resp.  $B_{\mathbf{h}} = [B_{\mathbf{h}}^{(1)} \cdots B_{\mathbf{h}}^{(n)}] \in \mathbb{R}^{n \times n}$ ),  
 633 setting  $\{V_{\mathbf{h}}^{(i)}\}_{1 \leq i \leq n}$  (resp.  $\{B_{\mathbf{h}}^{(i)}\}_{1 \leq i \leq n}$ ) as the column vectors of  $V_{\mathbf{h}}$  (resp.  
 634  $B_{\mathbf{h}}$ ).

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

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

639 **6.3. Numerical experiments on fractional Poisson equations.** This section  
 640 is devoted to some numerical experiments to illustrate the above approaches.

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

$$648 \quad (6.13) \quad u_{\mathbf{h};\mathbf{h}} = (2\pi\mathbf{i})^{-1} \sum_{1 \leq j \leq J_A-1} \mathfrak{h}_j \left( \frac{z_j + z_{j+1}}{2} \right)^{-\alpha} (z_j I - A)^{-1} f_{\mathbf{h}},$$

649 with  $z_j = z_c + r_{\varepsilon}^{(A)} e^{i\theta_j}$ . We define a Jacobi preconditioner  $M = \text{diag}(a_{11}^{-1}, \dots, a_{nn}^{-1})$   
 650 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,  
 651 we compute only *one* CROUT (row) ILU factorization with tolerance  $10^{-6}$ , setting  
 652 the restart parameter to 20 iterations,  $LU \approx \tilde{z}I - A$  with  $\tilde{z} = \lambda_{\min}^{(MA)}$ . We find  
 653  $r_{\varepsilon}^{(A)} \approx 2.7$  and  $r_{\varepsilon}^{(MA)} \approx 0.4$ , corresponding to a gain factor equal to 6.7. In Fig.  
 654 13 (Right), we report in logscale i) the CPU-time (in seconds) for the direct method  
 655 (with  $\mathcal{C}_A$ ) and double-preconditioned method (with  $\mathcal{C}_{MA}$ ), and ii)  $\|u_{\mathbf{h};\mathbf{h}} - u_{\text{ref}}\|_{\ell^2(\Omega_{\mathbf{h}})}$ .  
 656 The preconditioned approach converges much faster than the direct method which  
 657 also requires more resources.

658 **Experiment 9. 2d fractional Poisson equation.** For  $\Omega = ]-5, 5[ \times ]-1, 1[$ , we  
 659 consider the fractional laplacian problem  $-(\Delta)^{\alpha}u = f$ , with  $f(\mathbf{x}) = \exp(-5x_1^2 -$   
 660  $10x_2^2)$  and  $\alpha = 0.4$ . We choose a simple 3-point stencil approximate laplacian on  $\Omega_{\mathbf{h}} =$   
 661  $\{(x_{1,i}, x_{2,j}) \in \Omega : 1 \leq i \leq N_1, 1 \leq j \leq N_2\}$ , where  $N_1 = 40$ ,  $N_2 = 20$  and  $A \in \mathbb{R}^{n \times n}$ ,  
 662 for  $n = 800$ . The eigenvalues  $\lambda_{\min}^{(A)}$  and  $\lambda_{\max}^{(A)}$  are again computed by a power/inverse-  
 663 power method. We define the rectangle contour  $\Gamma_A = \mathcal{G}(\lambda_{\min}^{(A)} - \varepsilon, -\varepsilon, \lambda_{\max}^{(A)} + \varepsilon, \varepsilon)$ ,  
 664 with  $\varepsilon = 10^{-1}$ . The direct method is based on (6.13), with  $z_{j+1} = \mathfrak{h}_{j+1} + z_j$  such  
 665 that  $\mathfrak{h}_j := \delta x$  or  $\mathfrak{h}_j := \delta y$ , leading to  $\ell(\Gamma_A^{(\mathbf{h})}) = 2(\lambda_{\max}^{(A)} - \lambda_{\min}^{(A)} + 2\varepsilon)$ , where  $J_A$  is  
 666 the number of points to approximate  $\Gamma_A$ . For the Jacobi preconditioner  $M$ , we have  
 667  $\ell(\Gamma_{MA}^{(\mathbf{h})}) = 2(\lambda_{\max}^{(MA)} - \lambda_{\min}^{(MA)} + 2\varepsilon)$ . We calculate *one* CROUT (row) ILU-factorization,  
 668 setting the tolerance to  $10^{-6}$  and the value of the restart parameter to 20. Moreover,  
 669  $LU \approx \tilde{z}I - A$ , with  $\tilde{z} = \lambda_{\min}^{(MA)}$ . In Fig. 13 (Middle), we plot in logscale i) the CPU-  
 670 time (in seconds) for both the direct method (with  $\Gamma_A$ ) and double-preconditioned



671 method (with  $\Gamma_{MA}$ ), and ii)  $\|A^{-\alpha}f_h - u_{\text{ref}}\|_{\ell^2(\Omega)_h}$ . It is clear that the preconditioned  
 672 method is convergent much more rapidly than the direct one.

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

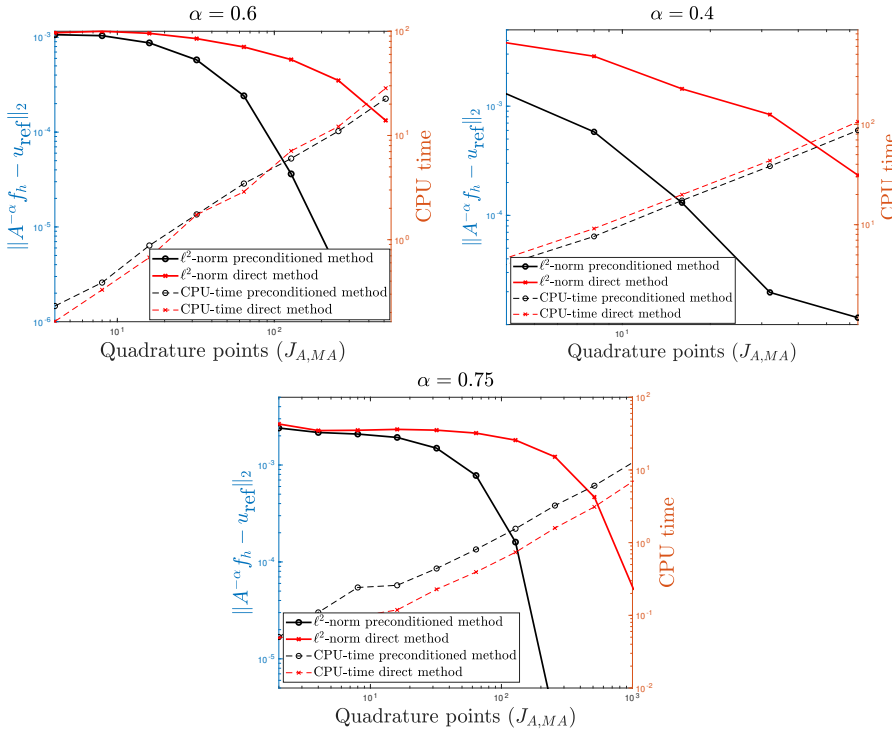


FIG. 13. CPU-time (in seconds) in logscale, and  $\|A^{-\alpha}f_h - u_{\text{ref}}\|_2$  in logscale. (Left) **Experiment 8. 1d Poisson.** (Middle) **Experiment 9. 2d Poisson.** (Right) **Experiment 9bis. 2d Poisson.**

684

685 **Experiment 10.** We finally propose a series of experiments for  $(-\Delta + V + dW)^\alpha u = f$   
 686 on a bounded domain  $]-10, 10[$  with null Dirichlet boundary conditions. The Cauchy  
 687 integral is approximated by using  $J_{A,MA} = 128$  quadrature nodes. For  $-\Delta$ , we  
 688 use a 5-point scheme. In the following tests, we report the residual history vs the  
 689 GMRES iteration number (the tolerance is  $10^{-15}$  and the restart parameter is set to  
 690 50 iterations). More specifically using circular contours, we compare the convergence  
 691 i) without Cauchy integral preconditioning (No preconditioning), ii) Jacobi Cauchy integral

692 preconditioner (Jacobi precond.), iii) ILU Cauchy integral preconditioner (with scaling  
 693 matrix for computing  $M^\alpha x$ , see Subsection 4.2) (ILU-precond.), iv) ILU factorization  
 694  $M$  on  $A$  and then  $M^\alpha$  is used to precondition  $A^\alpha$ , v) and finally no Cauchy integral  
 695 preconditioning, but ILU preconditioning of  $A^\alpha$ , assuming it is known (ILU-precond.  
 696 on  $A^\alpha$ ). The convergence graphs (residual history vs GMRES iteration number) are  
 697 given in Fig. 14 for

- 698 • **Experiment 10a.**  $V = 0$  and the brownian motion  $d\mathcal{W}$  is approximated by  
 699 a symmetric random (uniform law) matrix of magnitude 0.12, and  $\alpha = 0.75$ .
- 700 • **Experiment 10b.**  $V = 0$  and the brownian motion  $d\mathcal{W}$  is computed by a  
 701 unsymmetric random (uniform law) matrix with magnitude 0.06, and  $\alpha =$   
 702 0.75.
- 703 • **Experiment 10c.**  $V = 0$  and the brownian motion  $d\mathcal{W}$  is approximated  
 704 by a symmetric random (uniform law) matrix with magnitude 0.12, fixing  
 705  $\alpha = 0.5$ .
- 706 • **Experiment 10d.**  $V = 100e^{-x^2}$  and the brownian motion  $d\mathcal{W}$  is approxi-  
 707 mated by a symmetric random (uniform law) matrix of magnitude 0.12, and  
 708  $\alpha = 0.75$ .

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

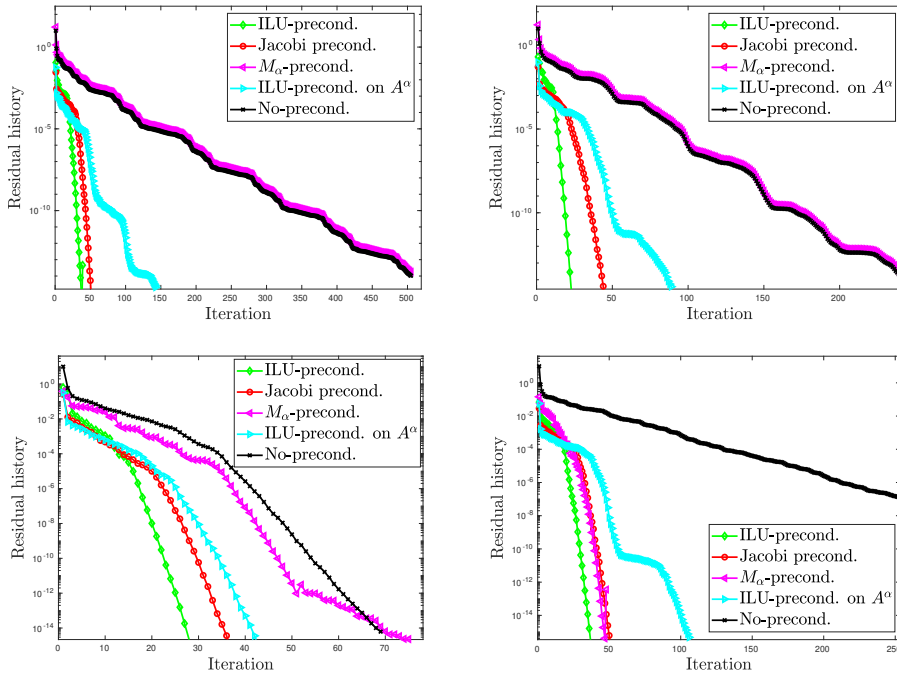


FIG. 14. GMRES convergence. (Top-Left) Experiment 10a ; (Top-Right) Experiment 10b; (Bottom-Left) Experiment 10c ;(Bottom-Right) Experiment 10d.

713 **7. Conclusion.** In this paper, we proposed an efficient method for computing  
 714 the real power of a diagonalizable matrix  $A$  and algorithms for solving fractional

715 linear systems, using quadrature rules for Cauchy integrals and contours enclosing  
716 the spectrum of  $A$ . Simple preconditioners are proposed for drastically reducing the  
717 computational complexity thanks to spectrum clustering. Some experiments are re-  
718 ported to illustrate the methodology. In particular, applications to (deterministic and  
719 stochastic) stationary fractional Poisson-like equations with Dirichlet boundary con-  
720 ditions are given. In a forthcoming paper, we will propose some realistic applications  
721 and comparisons with other methods such as the differential equation approach as  
722 defined in Subsection 4.4.

723

#### REFERENCES

- 724 [1] U. Biccari, M. Warma, and E. Zuazua. Local elliptic regularity for the Dirichlet fractional  
725 Laplacian. *Adv. Nonlinear Stud.*, 17(2):387–409, 2017.
- 726 [2] E. Carson and N.J. Higham. Accelerating the solution of linear systems by iterative refinement  
727 in three precisions. *SIAM J. Sci. Comput.*, 40(2):A817–A847, 2018.
- 728 [3] P. I. Davies and N. J. Higham. Computing  $f(A)b$  for matrix functions  $f$ . In *QCD and numerical*  
729 *analysis III*, volume 47 of *Lect. Notes Comput. Sci. Eng.*, pages 15–24. Springer, Berlin,  
730 2005.
- 731 [4] E. Di Nezza, G. Palatucci, and E. Valdinoci. Hitchhiker’s guide to the fractional Sobolev spaces.  
732 *Bulletin des Sciences Mathématiques*, 136(5):521–573, 2012.
- 733 [5] G. H. Golub and G. Meurant. *Matrices, Moments and Quadrature with Applications*. Princeton  
734 Series in Applied Mathematics. Princeton University Press, Princeton, NJ, 2010.
- 735 [6] N.J. Higham. Evaluating Padé approximants of the matrix logarithm. *SIAM Journal on Matrix*  
736 *Analysis and Applications*, 22(4):1126–1135, 2001.
- 737 [7] R. A. Horn and C. R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press,  
738 Cambridge, 1991.
- 739 [8] Y. Huang and A. Oberman. Numerical methods for the fractional Laplacian: a finite difference–  
740 quadrature approach. *SIAM J. Numer. Anal.*, 52(6):3056–3084, 2014.
- 741 [9] A. Lischke, G. Pang, M. Gulian, F. Song, C. Glusa, X. Zheng, Z. Mao, W. Cai, M. Meer-  
742 schaert, M. Ainsworth, and G. E. Karniadakis. What is the fractional laplacian?  
743 *arXiv:1801.09767v2*, 2018.
- 744 [10] J. C. Strikwerda. *Finite Difference Schemes and Partial Differential Equations*. Society for  
745 Industrial and Applied Mathematics (SIAM), Philadelphia, PA, second edition, 2004.
- 746 [11] M. E. Taylor. *Partial Differential Equations I. Basic Theory*, volume 115 of *Applied Mathe-*  
747 *matical Sciences*. Springer, New York, second edition, 2011.
- 748 [12] Y. Xi and Y. Saad. A rational function preconditioner for indefinite sparse linear systems.  
749 *SIAM J. Sci. Comput.*, 39(3):A1145–A1167, 2017.