

Efficient implementation of rational approximations to fractional differential operators

Lidia Aceto · Paolo Novati

Received: date / Accepted: date

Abstract This paper deals with some numerical issues about the rational approximation to fractional differential operators provided by the Padé approximants. In particular, the attention is focused on the fractional Laplacian and on the **Caputo's** derivative which, in this context, occur into the definition of anomalous diffusion problems and of time fractional differential equations (FDEs), respectively. The paper provides the algorithms for an efficient implementation of the IMEX schemes for semi-discrete anomalous diffusion problems and of the short-memory-FBDF methods for Caputo's FDEs.

Keywords fractional Laplacian operator · Caputo fractional derivative · matrix functions · Gauss-Jacobi rule · Padé approximants

Mathematics Subject Classification (2000) 65F60 · 65D32 · 35R11 · 34A08

1 Introduction

Let $A \in \mathbb{R}^{n \times n}$ be a sparse and banded matrix that represents the finite difference approximation of a differential operator with given initial and/or boundary conditions. In some situations of practical interest the matrix fractional power A^α , $\alpha > 0$, if well-defined, can be used to construct numerical methods for equations

This work was partially supported by GNCS-INdAM, University of Pisa (grant PRA_2017.05) and FRA-University of Trieste.

L. Aceto
Dipartimento di Matematica, Università di Pisa
via F. Buonarroti, 1/C, 56127 Pisa (Italy)
Tel.: +39-050-2213836
E-mail: lidia.aceto@unipi.it

P. Novati
Dipartimento di Matematica e Geoscienze, Università di Trieste
via Valerio 12/1, 34127 Trieste (Italy)
Tel.: +39-040-5582643
E-mail: novati@units.it

involving fractional derivatives. For instance, denoting by h a suitable stepsize and taking

$$A = h^{-1} \cdot \text{bidiag}(1, -1),$$

which represents the uniform backward difference approximation of the first derivative, it is known that A^α represents the Grünwald-Letnikov fractional derivative of order $\alpha \in (0, 1)$ (see, for instance, [16, 19, 20]). Another simple example is given by

$$A = h^{-2} \cdot \text{tridiag}(-1, 2, -1),$$

in which A^α , $\alpha \in (1/2, 1]$, is assumed to be a uniform finite difference approximation of the one-dimensional fractional Laplacian operator of order 2α with homogeneous Dirichlet boundary conditions (see [14, 15]). In this view, any numerical scheme able to compute the matrix fractional powers can be potentially used to define a method for fractional equations. Nevertheless, when working with fractional powers, it must be kept in mind that raising to a fractional number destroys the sparsity structure of the underlying integer order approximation. As consequence the corresponding solver may be extremely expensive for large values of n . In order to face this problem, in [3, 4, 18] the authors have studied rational approximations to A^α , that is,

$$A^\alpha \approx [q_k(A)]^{-1} p_k(A), \quad (1)$$

where $p_k, q_k \in \Pi_k$, the set of polynomials of degree less or equal than k . Assuming that $k \ll n$, the action of A^α is then approximated through the action of sparse matrices. We remark that each ‘‘short memory’’ approach, in which A^α is simply replaced by its m -banded version, for a given m fixed in some way, represents a special case of (1) in which $k = m$ and $q_k(z) = 1$.

Obviously the choice of k in (1) plays a crucial role and depends on many factors, such as the operator, the differential problem, the properties of the rational form p_k/q_k . On the other side, its value is responsible for the overall computational cost. Beside this classical matter between accuracy and cost, there is another computational problem that may appear when working with rational approximations, that is, the conditioning of $p_k(A)$ and $q_k(A)$. Since these two matrices will be involved in the linear algebra tasks required by the differential solver, it is fundamental that their condition numbers are not responsible for inaccuracy, otherwise there will be an a-priori barrier for the choice of k .

In this paper we deal with a particular rational approximation (1), used in [3, 4], that arises from a standard integral representation of A^α for $\alpha \in (0, 1)$. This rational approximation, that is essentially a scaled Padé form, is quite accurate but the conditioning of $p_k(A)$ and $q_k(A)$ becomes quickly (with respect to k) very large if A represents a differential operator. We remark that the error analysis given in [3, Theorem 3.6] (and also the one in [18, Theorem 6] based on a related approach) for the approximation of the Caputo’s fractional derivative reveals that the choice of k is strictly related to the number of discretization points (that is, the size of A). The same holds for the approximation of the fractional Laplacian. In this case, from [4, Theorem 3 and Corollary 4] we have that an error of type

$$O\left(\exp\left(-4k\sqrt[4]{\lambda_{\min}/\lambda_{\max}}\right)\right)$$

can be predicted, where λ_{\min} and λ_{\max} are respectively the smallest and the largest eigenvalue of a given discretization of the Laplacian. Of course, $\sqrt[4]{\lambda_{\min}/\lambda_{\max}} \rightarrow 0$ improving the quality of the discretization.

The above arguments substantially states that using these kind of approximation a suitable value of k may be rather large in principle so that any integrator that involves $p_k(A)$ and $q_k(A)$ may result highly inaccurate because of the conditioning. In order to **approach** this problem, and hence to fix the computational drawback of the integrators used in [3,4], here we use a simple but reliable strategy that allows to keep the conditioning under control. The basic idea is to reorganize the algorithm of any integrator based on the approximation (1) in order that all linear algebra tasks only (iteratively) involve matrices of the type $(A + \cdot I)$. Just to provide an elementary example, if one needs to solve $A^m \mathbf{x} = \mathbf{b}$, where A is moderately ill-conditioned and $m > 1$, it is rather clear that the computation of A^m should be avoided in favor of the sequential solution $A\mathbf{y}_1 = \mathbf{b}$, $A\mathbf{y}_2 = \mathbf{y}_1, \dots, A\mathbf{x} = \mathbf{y}_{m-1}$ (see e.g. [22]). While the idea is simple, its application may be nontrivial and **quite** technical, depending on the method employed to the differential problem to solve. In this paper we focus our attention on methods for the anomalous diffusion and Caputo's type fractional differential equations (FDEs). In particular we provide the computational details for an efficient implementation of some IVP solvers for semi-discrete anomalous diffusion problems and for the well-known Fractional BDFs (FBDFs).

The paper is organized as follows. In Section 2 we recall the basic features about the rational approximation of fractional operators through the Gauss-Jacobi approximation of the corresponding integral representation. In Section 3 we present two algorithms that can be fruitfully employed to solve linear systems involving commutative products of matrices. In Section 4 we consider the numerical solution of anomalous diffusion problems by means of implicit-explicit linear multistep methods implemented with the algorithms of Section 3. Finally, Section 5 is devoted to the solution of time fractional differential **equations** by using a predictor-corrector implementation of short-memory-FBDF methods.

2 Background on the Padé type rational approximations

From the theory of matrix functions (see [13] for a survey), we know that the fractional power of matrix can be written as a contour integral

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

where Γ is a suitable closed contour of the complex plane not containing the origin and enclosing the spectrum of A , $\sigma(A)$, in its interior. A numerical method for the computation of A^α can be defined by any quadrature approximation of (2), that leads to a weighted sum of terms of the type $(z_j I - A)^{-1}$, $z_j \in \Gamma$. If $\sigma(A)$ is close to the origin then so is Γ , and because of the nature of the integrand a good approximation requires that many z_j are taken close to the origin. As consequence, the product of the terms $(z_j I - A)^{-1}$, that defines the polynomial $q_k(A)$ of (1), is expected to be very ill-conditioned. This is a general argument, but in what follows we explain in detail what happens to the rational approximation based on the Gauss-Jacobi rule. First of all we recall the following result [7].

Proposition 1 *Let $A \in \mathbb{R}^{n \times n}$ be such that $\sigma(A) \subset \mathbb{C} \setminus (-\infty, 0]$. For $\alpha \in (0, 1)$ the following representation holds*

$$A^\alpha = \frac{A \sin(\alpha\pi)}{\alpha\pi} \int_0^\infty (\rho^{1/\alpha} I + A)^{-1} d\rho. \quad (3)$$

By considering the change of variable

$$\rho^{1/\alpha} = \tau \frac{1-t}{1+t}, \quad \tau > 0, \quad (4)$$

we can rewrite (3) as

$$A^\alpha = \frac{2 \sin(\alpha\pi) \tau^\alpha}{\pi} A \int_{-1}^1 (1-t)^{\alpha-1} (1+t)^{-\alpha} (\tau(1-t)I + (1+t)A)^{-1} dt,$$

which leads to the use of a k -point Gauss-Jacobi rule and hence to a rational approximation (1). In particular, we have

$$\left(\frac{A}{\tau}\right)^\alpha \approx \frac{A}{\tau} \sum_{j=1}^k \frac{2 \sin(\alpha\pi)}{\pi} \frac{w_j}{1+\vartheta_j} \left(\frac{1-\vartheta_j}{1+\vartheta_j} I + \frac{A}{\tau}\right)^{-1}, \quad (5)$$

where $w_j, \vartheta_j, j = 1, 2, \dots, k$, are respectively the weights and nodes of the Gauss-Jacobi quadrature rule with weight function $(1-t)^{\alpha-1}(1+t)^{-\alpha}$.

As remembered in the Introduction, for an error analysis of this formula for the approximation of Caputo's fractional derivative and the fractional Laplacian we refer to [3, Theorem 3.6] and [4, Theorem 3 and Corollary 4], respectively. Nevertheless, an optimal value of k will finally depend on the accuracy required to the integrator and hence it will be closely related to the problem, as shown in [4, Section 5, Subsection 5.1] for the case of IVPs involving the fractional Laplacian and where a 'conservative' hint about the choice of k is given. The situation is even more complicated when dealing with the **Caputo's** derivative. In [3, Section 4.1] the consistency analysis is given with respect to the quality of the discretization and it is shown that k can be properly selected by working scalarly. **In the present work, based on the computational experience of our previous, we just want to explain how to bypass the computational drawback (bottleneck) caused by accuracy requirements that force k to be large. As for the dependence on α , we remark that the change of variable (4) forces the singularity to stay in the weight function, so that the integrand function is analytic in an open subset of the complex plane containing $[-1, 1]$.** In this sense, the quality of the approximation is substantially independent of α (cf. the experiments of [3, Section 3.1] and [4, Section 4]).

In [9, Lemma 4.4] it has been proved that the k -point Gauss-Jacobi quadrature corresponds to the $(k-1, k)$ -Padé approximant of $(z/\tau)^{\alpha-1}$ centered at 1, hereafter denoted by $\mathcal{R}_{k-1,k}(z/\tau; 1)$. In this sense, the approximation (5) can be written as

$$\left(\frac{A}{\tau}\right)^\alpha \approx \frac{A}{\tau} \mathcal{R}_{k-1,k}\left(\frac{A}{\tau}; 1\right). \quad (6)$$

We also remember that $\mathcal{R}_{k-1,k}(z/\tau; 1)$ can be expressed in terms of hypergeometric functions using the result given in [10, Theorem 4.1]

$$\mathcal{R}_{k-1,k}\left(\frac{z}{\tau}; 1\right) = \frac{{}_2F_1(1-k, 1-\alpha-k; 1-2k; 1-z/\tau)}{{}_2F_1(-k, \alpha-k; 1-2k; 1-z/\tau)}.$$

Denoting with $\mathcal{P}_\ell^{(\beta, \gamma)}$ the Jacobi polynomial of degree ℓ and using [21, eq. 142, p. 464] and the symmetry of such polynomials, we can equivalently write

$$\mathcal{R}_{k-1, k} \left(\frac{z}{\tau}; 1 \right) = \frac{(1 - z/\tau)^{k-1} \mathcal{P}_{k-1}^{(1-\alpha, \alpha)} \left(\frac{1+z/\tau}{1-z/\tau} \right)}{(1 - z/\tau)^k \mathcal{P}_k^{(\alpha-1, -\alpha)} \left(\frac{1+z/\tau}{1-z/\tau} \right)}.$$

We remark that the nodes of the Gauss-Jacobi rule ϑ_j , $j = 1, 2, \dots, k$, are in fact the roots of $\mathcal{P}_k^{(\alpha-1, -\alpha)}(\xi)$.

Proposition 2 *Let ϑ_j be the j -th root of $\mathcal{P}_k^{(\alpha-1, -\alpha)}(\xi)$ and ζ_r be the r -th root of $\mathcal{P}_{k-1}^{(1-\alpha, \alpha)}(\xi)$. Setting*

$$\eta_j = \tau \frac{1 - \vartheta_j}{1 + \vartheta_j} \equiv \tau \bar{\vartheta}_j, \quad j = 1, 2, \dots, k, \quad (7)$$

$$\varepsilon_r = \tau \frac{1 - \zeta_r}{1 + \zeta_r} \equiv \tau \bar{\zeta}_r, \quad r = 1, 2, \dots, k-1, \quad (8)$$

we have

$$\mathcal{R}_{k-1, k} \left(\frac{z}{\tau}; 1 \right) = c \tau \prod_{r=1}^{k-1} (z + \varepsilon_r) \prod_{j=1}^k (z + \eta_j)^{-1}, \quad (9)$$

with

$$c = \frac{\eta_k}{\tau} \frac{\binom{k-\alpha}{k-1}}{\binom{k+\alpha-1}{k}} \prod_{j=1}^{k-1} \frac{\eta_j}{\varepsilon_j}. \quad (10)$$

Proof According to the change of variable

$$\xi = \frac{1 + z/\tau}{1 - z/\tau},$$

and by (7)-(8), we can write

$$(1 - z/\tau)^k \mathcal{P}_k^{(\alpha-1, -\alpha)} \left(\frac{1 + z/\tau}{1 - z/\tau} \right) = \bar{c} \prod_{j=1}^k \left(\frac{z}{\tau} + \bar{\vartheta}_j \right),$$

$$(1 - z/\tau)^{k-1} \mathcal{P}_{k-1}^{(1-\alpha, \alpha)} \left(\frac{1 + z/\tau}{1 - z/\tau} \right) = \hat{c} \prod_{r=1}^{k-1} \left(\frac{z}{\tau} + \bar{\zeta}_r \right),$$

with \bar{c} and \hat{c} suitable constants. These can be immediately determined by taking into account that for a Jacobi polynomial $\mathcal{P}_\ell^{(\beta, \gamma)}$ it holds

$$\mathcal{P}_\ell^{(\beta, \gamma)}(1) = \binom{\ell + \beta}{\ell}.$$

In particular,

$$\bar{c} = \binom{k + \alpha - 1}{k} \left(\prod_{j=1}^k \bar{\vartheta}_j \right)^{-1}, \quad \hat{c} = \binom{k - \alpha}{k - 1} \left(\prod_{r=1}^{k-1} \bar{\zeta}_r \right)^{-1},$$

from which the statement follows. \square

Therefore, by using (9)-(10) the $(k-1, k)$ -Padé approximant of $(A/\tau)^{\alpha-1}$ can be expressed by

$$\left(\frac{A}{\tau}\right)^{\alpha-1} \approx c\tau \prod_{r=1}^{k-1} (A + \varepsilon_r I) \prod_{j=1}^k (A + \eta_j I)^{-1}.$$

Denoting by

$$M = \prod_{j=1}^k M_j, \quad M_j = A + \eta_j I, \quad (11)$$

$$N = c\tau^\alpha \prod_{j=1}^k N_j, \quad N_j = A + \varepsilon_j I, \quad \varepsilon_k = 0, \quad (12)$$

the rational approximation of A^α takes the following form (see (1))

$$A^\alpha \approx M^{-1}N. \quad (13)$$

It is worth noting that since $\eta_j > 0$ for each j , if $\sigma(A) \subset \mathbb{C}^+$ then $\sigma(M_j) \subset \mathbb{C}^+$. Similar remarks can be done for the matrix N_j . Nevertheless, if $\sigma(A)$ is closed to the origin both M and N are expected to be severely ill-conditioned for growing values of k . Indeed, increasing k we have an increasing number of Gauss-Jacobi nodes close to 1 and therefore, by (7) and (8), an increasing number of η_j and ε_r close to zero. Just to provide a numerical example, in Figure 1 we compare the Euclidean condition number of A^α , $\kappa(A^\alpha)$, with the one of the two matrices M and N with respect to k , that can be exactly computed knowing λ_{\min} and λ_{\max} (the smallest and the largest eigenvalue of A , respectively) by

$$\kappa(M) = \prod_{j=1}^k \frac{\lambda_{\max} + \eta_j}{\lambda_{\min} + \eta_j}, \quad \kappa(N) = \frac{\lambda_{\max}}{\lambda_{\min}} \prod_{j=1}^{k-1} \frac{\lambda_{\max} + \varepsilon_j}{\lambda_{\min} + \varepsilon_j}. \quad (14)$$

In this case we have considered $A = \text{tridiag}(-1, 2, -1)$, $\alpha = 1.2, 1.8$ and $\tau = \sqrt{\lambda_{\min}\lambda_{\max}}$ (see (4)). This value of τ has been showed to be the one that optimizes the asymptotic convergence factor of the approximation (5) when A is a symmetric positive definite matrix, cf. [4]. Moreover, using the **Matlab function jacpts** implemented in Chebfun by Hale and Townsend [12], we have computed the values ϑ_j and ζ_r which allow to obtain η_j and ε_r by (7) and (8).

3 The basic algorithms

We start this section with a simple strategy to approximate the matrix-vector multiplication $A^\alpha \mathbf{b}$. By (13), we need to solve efficiently

$$M\mathbf{x} = N\mathbf{b}. \quad (15)$$

The following Algorithm 1 solves linear systems of the type

$$l_k(A)\mathbf{y} = r_k(A)\mathbf{c},$$

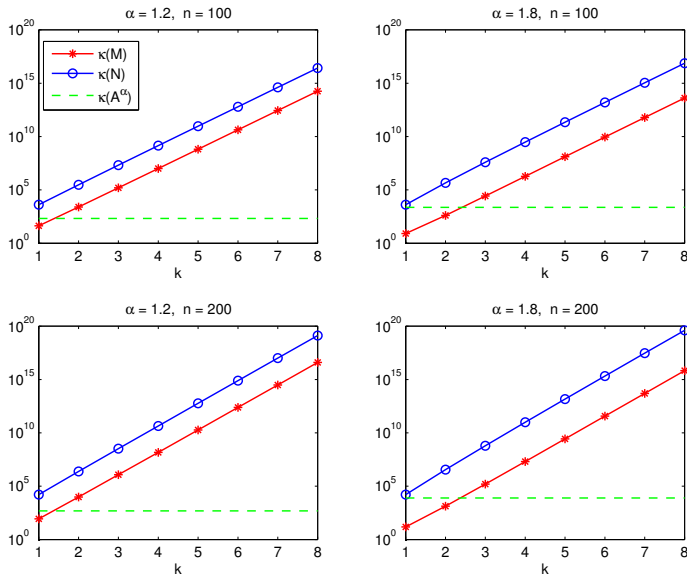


Fig. 1 Comparison between the Euclidean condition numbers of M and N computed by (14), and the one of A^α , for $\alpha = 1.2, 1.8$.

Algorithm 1

- 1: set $\mathbf{y}^{(0)} = \mathbf{c}$
 - 2: for $j = 1$ to k
 - 3: $d_j = \rho_j - \lambda_j$
 - 4: solve $(A + \lambda_j I)\mathbf{v}^{(j)} = \mathbf{y}^{(j-1)}$
 - 5: $\mathbf{y}^{(j)} = \mathbf{y}^{(j-1)} + d_j \mathbf{v}^{(j)}$
 - 6: end for
-

where

$$l_k(A) = \prod_{j=1}^k (A + \lambda_j I), \quad r_k(A) = \prod_{j=1}^k (A + \rho_j I).$$

Therefore, for (15) one can use Algorithm 1. The algorithm is really simple but allows to work always with a shift of the matrix A without forming explicitly M and N . In this sense it is also possible to consider a preconditioned version in which the preconditioner is computed only once (see e.g. [5]).

In order to show the efficiency of the algorithm, in Figure 2 we consider the comparison with both the Cholesky and the Conjugate Gradient method applied to (15), that is, with M as coefficient matrix, again for $A = \text{tridiag}(-1, 2, -1)$. In Algorithm 1, the subsystems $M_j \mathbf{v}^{(j)} = \mathbf{y}^{(j-1)}$ are solved with the Cholesky method. The bottleneck determined by the ill-conditioning of M , and overtaken by the algorithm, is quite clear. We have taken again $\tau = \sqrt{\lambda_{\min} \lambda_{\max}}$.

We remark that, in general terms, the computation of $A^\alpha \mathbf{b}$ is not an easy task if A is large. Excluding the use of a direct approach (see [13]) one typically resort to a Krylov projection. Since z^α has a branch point, the convergence of the standard Krylov method (based on the construction of the Krylov subspaces generated by

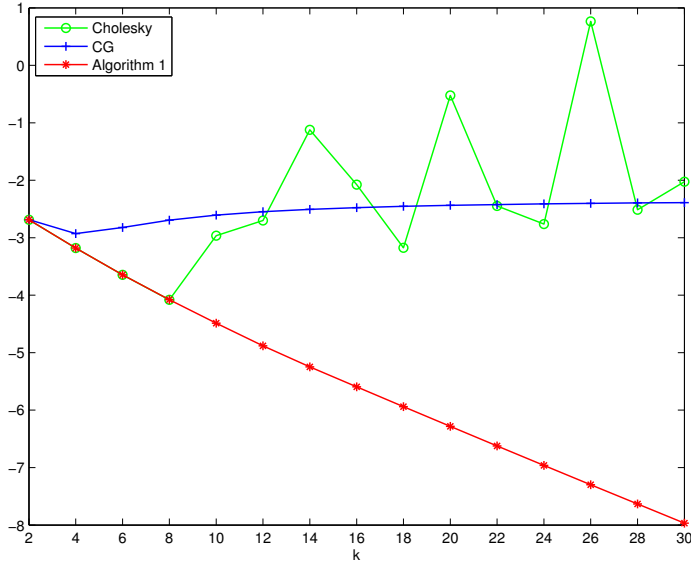


Fig. 2 Computed errors $\|\mathbf{y}^{(k)} - \mathbf{z}\|_\infty$, $\mathbf{z} = A^\alpha \mathbf{b}$, for different values of k , where $\mathbf{y}^{(k)}$ is the numerical solution of (15). We compare the results of Algorithm 1 with the Cholesky and the Conjugate Gradient method (CG) applied directly to (15). In the experiment $\alpha = 1.8$, $n = 200$, and \mathbf{b} represents the uniform discretization of $x(1-x)$ in $[0, 1]$.

A and \mathbf{b}) may be extremely slow even if no inversion is required. On the other side the convergence can be accelerated using a rational Krylov method [17], that, at each iteration, requires the solution of a linear system with A shifted, as well as Algorithm 1.

As we shall see later, when constructing integrators for fractional differential equations using the factorization (13) it is possible to encounter linear equations of the type

$$(M + \omega N)\mathbf{x} = N\mathbf{b} \quad \text{or} \quad (M + \omega N)\mathbf{x} = M\mathbf{b}, \quad \omega > 0. \quad (16)$$

We have (see (11) and (12))

$$M + \omega N = \prod_{j=1}^k (A + \eta_j I) + \omega c \tau^\alpha \prod_{j=1}^k (A + \varepsilon_j I). \quad (17)$$

Setting $\{a_0, a_1, \dots, a_{k-1}\}$ and $\{b_0, b_1, \dots, b_{k-1}\}$ such that

$$M = a_0 I + \dots + a_{k-1} A^{k-1} + A^k, \quad N = c \tau^\alpha (b_0 I + \dots + b_{k-1} A^{k-1} + A^k), \quad (18)$$

and $\hat{\omega} = \omega c \tau^\alpha$, we obtain

$$\begin{aligned} \frac{1}{1 + \hat{\omega}} (M + \omega N) &= \frac{1}{1 + \hat{\omega}} \left(\sum_{j=0}^{k-1} (a_j + \hat{\omega} b_j) A^j + (1 + \hat{\omega}) A^k \right) \\ &= \sum_{j=0}^{k-1} \frac{a_j + \hat{\omega} b_j}{1 + \hat{\omega}} A^j + A^k = \sum_{j=0}^{k-1} c_j A^j + A^k. \end{aligned} \quad (19)$$

Finally, whenever the roots of such polynomial, denoted by $(-\sigma_j)$, have been computed, we have

$$M + \omega N = (1 + \hat{\omega}) \prod_{j=1}^k (A + \sigma_j I). \quad (20)$$

Obviously, the above expression allows to solve (16) through Algorithm 1. The basic steps can be summarized by Algorithm 2.

Algorithm 2

- 1: compute $a_j, b_j, j = 0, \dots, k-1$, the coefficients in (18)
 - 2: set $\hat{\omega} = \omega c \tau^\alpha$
 - 3: evaluate $c_j = \frac{a_j + \hat{\omega} b_j}{1 + \hat{\omega}}, j = 0, \dots, k-1$
 - 4: compute $(-\sigma_j)$, the roots of the polynomial in (19)
 - 5: solve (16) using (20) and Algorithm 1
-

4 Anomalous diffusion

Consider the following fractional in space reaction-diffusion differential equation

$$\frac{\partial u(x, t)}{\partial t} = -\kappa_\alpha (-\Delta)^\alpha u(x, t) + f(x, t, u), \quad x \in \Omega \subset \mathbb{R}^d, t \in (0, T), \quad (21)$$

subject to homogeneous Dirichlet boundary conditions

$$u(x, t)|_{\hat{\Omega}} = 0, \quad \hat{\Omega} = \mathbb{R}^d \setminus \Omega,$$

and the initial condition

$$u(x, 0) = u_0(x).$$

The symmetric space fractional derivative $(-\Delta)^\alpha$ of order $2\alpha, \alpha \in (1/2, 1]$, can be defined through the spectral decomposition of the homogeneous Dirichlet Laplace operator $(-\Delta)$, [15, Definition 2].

We consider the spatial discretization of (21) by taking a uniform mesh of stepsize h and using the matrix transfer technique proposed in [14,15] by Ilić et al.

$$-(-\Delta)^\alpha u \approx -\frac{1}{h^{2\alpha}} A^\alpha u, \quad (22)$$

where $h^{-2}A$ is the approximate matrix representation of the standard Laplacian obtained by using any finite difference method.

Hence, by (22), the matrix transfer technique transforms the fractional reaction-diffusion problem into the system of first-order differential equations

$$\begin{aligned} \frac{dy}{dt} &= -\frac{\kappa_\alpha}{h^{2\alpha}} A^\alpha y + f(t, y), & t \in (0, T], \\ y(0) &= y_0, \end{aligned} \quad (23)$$

with the usual meaning for $y = y(t)$ and $f(t, y)$.

Working as in Section 2, we approximate the solution of (23) by solving

$$\begin{aligned} \frac{dw}{dt} &= -\frac{\kappa_\alpha}{h^{2\alpha}} M^{-1} N w + f(t, w), & t \in (0, T], \\ w(0) &= y_0, \end{aligned} \quad (24)$$

where M and N are defined accordingly to (11)-(12).

Below we show an implementation of some multistep IVP solvers for (24) discretizing in time with a uniform stepsize δ , i.e., $t_j = j\delta$, $j \geq 0$. Because of the stiff nature of the problem we consider implicit schemes but at the same time we always assume to handle explicitly the nonlinear forcing term $f(t, w)$, that is, we consider IMEX (implicit-explicit) integrators.

4.1 Implicit-explicit linear multistep methods

A generic m -step implicit-explicit linear multistep method for (24) can be written as

$$\sum_{i=0}^m \alpha_i w_{j-i+1} = \delta \sum_{i=0}^m \beta_i \left(-\frac{\kappa_\alpha}{h^{2\alpha}} M^{-1} N w_{j-i+1} \right) + \delta \sum_{i=1}^m \gamma_i f(t_{j-i+1}, w_{j-i+1}), \quad (25)$$

where $w_j \approx w(t_j)$, **and where** the set of coefficients $\{(\alpha_i, \beta_i, \gamma_i)\}_{i=0, \dots, m}$, $\gamma_0 = 0$, needs to be suitably defined to ensure proper linear stability properties and to achieve a certain order of convergence (see [1, 2] for a complete analysis). Setting

$$s = \kappa_\alpha \frac{\delta}{h^{2\alpha}},$$

and

$$\Gamma_1 = \sum_{i=1}^m \alpha_i w_{j-i+1}, \quad \Gamma_2 = \sum_{i=1}^m \beta_i w_{j-i+1}, \quad \Gamma_3 = \delta \sum_{i=1}^m \gamma_i f(t_{j-i+1}, w_{j-i+1}),$$

after some manipulations of (25) one obtains

$$(\alpha_0 M + \beta_0 s N) \left(w_{j+1} + \frac{\Gamma_1 - \Gamma_3}{\alpha_0} \right) = s N \left(\frac{\beta_0}{\alpha_0} (\Gamma_1 - \Gamma_3) - \Gamma_2 \right). \quad (26)$$

In this way w_{j+1} can be computed using Algorithm 2. We remark that for classical implicit multistep methods $\alpha_0 \beta_0 > 0$ so that the matrix pencil $\alpha_0 M + \beta_0 s N$ remains positive definite.

Example 1 In order to work with an IMEX Adams type method, we just need to couple an Adams-Moulton (α_i, β_i) and an Adams-Bashforth (α_i, γ_i) of equal order. For an IMEX BDF method of order $p = m \leq 6$ the coefficients $\{(\alpha_i, \beta_i)\}_{i=0, \dots, p}$ are the ones of the classical p -step BDF that are coupled with the coefficients $\{\gamma_i\}_{i=1, \dots, p}$ satisfying (see [2, 8])

$$\gamma_1 \zeta^{p-1} + \dots + \gamma_{p-1} \zeta + \gamma_p = \zeta^p - (\zeta - 1)^p.$$

4.2 Numerical examples

We start with the one-dimensional case. We consider the spatial domain $\Omega = (-1, 1)$, $u_0(x) = (1 - x^2)^{-1+\alpha/2}$ and

$$f(x, t, u) = \gamma (t + 1)^{\gamma-1} (1 - x^2)^{-1+\alpha/2}.$$

The exact solution is given by

$$u(x, t) = (t + 1)^\gamma (1 - x^2)^{-1+\alpha/2}.$$

In our experiments, we select the model parameters $\kappa_\alpha = 0.1$ and $\alpha = 1.2$. We discretize the spatial domain with a uniform mesh having stepsize $h = 2/501$ and we consider the standard 3-point central difference discretization of the Laplacian $h^{-2}A = h^{-2}\text{tridiag}(-1, 2, -1)$. We solve (23) and (24) with the IMEX Adams methods of order 2 (Figures 3 and 4) and of order 4 (Figures 5 and 6) in the time interval $[0, 1]$, using the time-step $\delta = 10^{-3}$. Below we recall the coefficients of the methods:

- second-order IMEX Adams:

$$(\alpha_0, \alpha_1, \alpha_2) = (1, -1, 0), \quad (\beta_0, \beta_1, \beta_2) = \frac{1}{2}(1, 1, 0), \quad (\gamma_0, \gamma_1, \gamma_2) = \frac{1}{2}(0, 3, 1);$$

- forth-order IMEX Adams:

$$(\alpha_0, \alpha_1, \alpha_2, \alpha_3, \alpha_4) = (1, -1, 0, 0, 0), \quad (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4) = \frac{1}{24}(9, 19, -5, 1, 0),$$

$$(\gamma_0, \gamma_1, \gamma_2, \gamma_3, \gamma_4) = \frac{1}{24}(0, 55, -59, 37, -9).$$

We have a double aim. The first one is to show that the use of Algorithm 2 for solving (26) can improve the results attainable with the Cholesky decomposition naively applied to $\alpha_0 M + \beta_0 s N$. We refer to the first approach by “iterated Cholesky” while we denote the second one by “standard Cholesky”. To this aim we take as benchmark the numerical solution of (23) in which A^α is computed to the machine precision, that is, the classical matrix transfer technique. Figures 3 and 5 refer to this comparison. The second aim is to compare the results of these three methods with the exact solution, and this is done in Figures 4 and 6. In all pictures we consider the numerical solution at $t = 1$.

In each figure we plot three pictures in order to consider an increasingly accurate rational approximation to A^α given by (6). In this sense we take $k = 7, 14, 21$. Figures 3 and 5 show that the improvement of this approximation is only possible when working with the iterated Cholesky because of the extremely large conditioning of $\alpha_0 M + \beta_0 s N$ for growing k . On the other side Figures 4 and 6 reveal that by choosing k large enough the rational approximation implemented with the iterated Cholesky yields essentially the same results of the matrix transfer technique.

Concerning some hints about a suitable a-priori choice of k , the analysis given in [4, §5.1] applies to the present case. This analysis however requires some numerical estimates about the error of the rational approximation (13), given in that paper, that we do not report in the present one. Of course a good choice of k can substantially reduce the overall computational cost but our aim here is just to

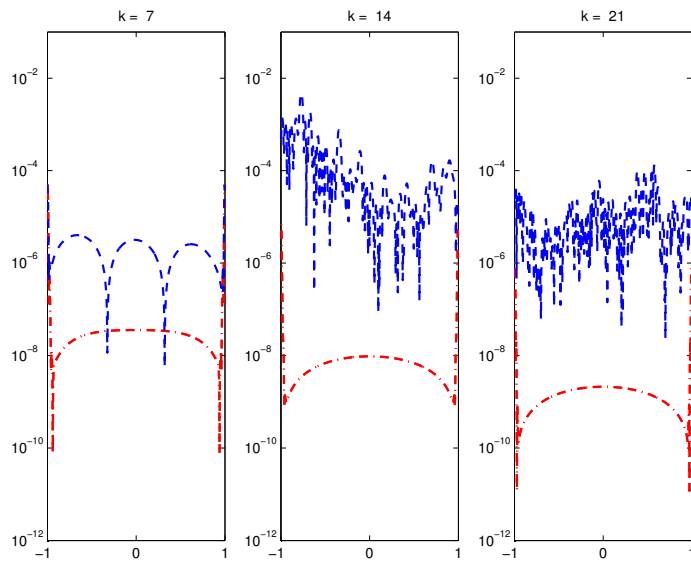


Fig. 3 Pointwise error at $t = 1$ of the iterated (dash-dotted line) and the standard Cholesky (dashed line) method with respect to the reference solution given by the matrix transfer technique. The time-stepping is performed by the IMEX Adams method of order 2.

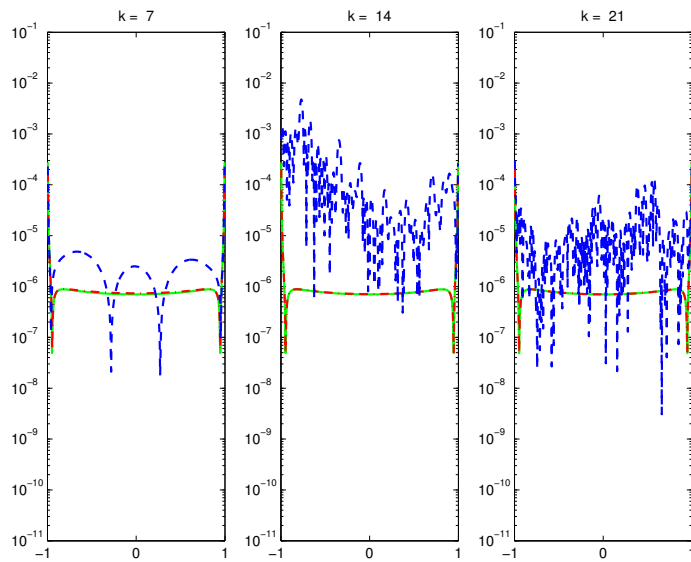


Fig. 4 Pointwise error at $t = 1$ of the iterated (dash-dotted line) and the standard Cholesky (dashed line) method, and the matrix transfer approach (solid line), with respect to the exact solution. The time-stepping is performed by the IMEX Adams method of order 2.

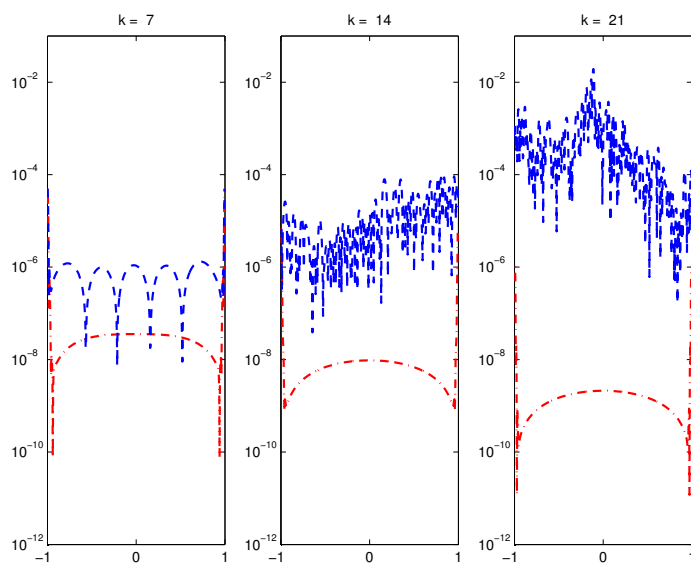


Fig. 5 Pointwise error at $t = 1$ of the iterated (dash-dotted line) and the standard Cholesky (dashed line) method with respect to the reference solution given by the matrix transfer technique. The time-stepping is performed by the IMEX Adams method of order 4.

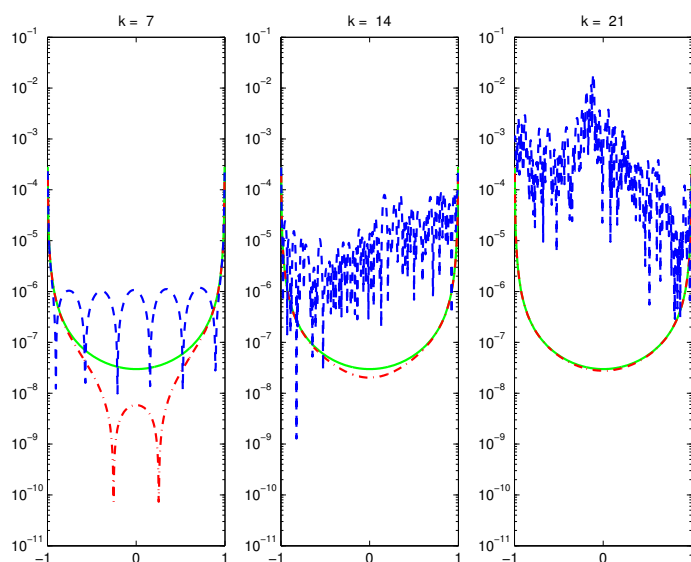


Fig. 6 Pointwise error at $t = 1$ of the iterated (dash-dotted line) and the standard Cholesky (dashed line) method, and the matrix transfer approach (solid line), with respect to the exact solution. The time-stepping is performed by the IMEX Adams method of order 4.

show that for the accuracy we are not forced to keep k small in order to preserve the conditioning.

Now we consider (21) in two space dimensions. We set $\Omega = (0, 1) \times (0, 1)$, $u_0(x_1, x_2) = 0$, and

$$f(x_1, x_2, t, u) = t^{2\alpha} \frac{\kappa_\alpha}{16} \sum_{j=1}^4 (1 + \mu_j^\alpha) v_j + 2\alpha t^{2\alpha-1} \sin^3(\pi x_1) \sin^3(\pi x_2) - \kappa_\alpha u,$$

where

$$\begin{aligned} v_1 &= 9 \sin(\pi x_1) \sin(\pi x_2), & \mu_1 &= 2\pi^2, \\ v_2 &= -3 \sin(\pi x_1) \sin(3\pi x_2), & \mu_2 &= 10\pi^2, \\ v_3 &= -3 \sin(3\pi x_1) \sin(\pi x_2), & \mu_3 &= 10\pi^2, \\ v_4 &= \sin(3\pi x_1) \sin(3\pi x_2), & \mu_4 &= 18\pi^2. \end{aligned}$$

The exact solution to this problem is

$$u(x_1, x_2, t) = t^{2\alpha} \sin^3(\pi x_1) \sin^3(\pi x_2).$$

In our experiments we select $\kappa_\alpha = 10$ and $\alpha = 1.2$. We use $n = 70$ points in each spatial domain direction and we approximate the two-dimensional Laplacian using the 5-point stencil finite-difference method. We solve (23) and (24) with the IMEX Adams methods of order 4 in the time interval $[0, 0.1]$, using the time-step $\delta = 10^{-2}$. The numerical solution provided by the rational approach based on the Gauss-Jacobi rule with $k = 14, 21, 28$, the matrix transfer technique and the exact solution are drawn at $t = 0.1$. The results are reported in Figures 7 and 8. **As one can see from these two pictures, comments similar to those made in the one-dimensional case can also be done in this case.**

5 Caputo's FDEs

Let $\alpha_0, \alpha_1, \dots, \alpha_\ell$ be the $\ell + 1$ coefficients of a BDF of order ℓ , with $1 \leq \ell \leq 6$, which discretizes the derivative operator (see [11] Chapter III.1 for a background). Let moreover

$$A_\ell = \begin{pmatrix} \alpha_0 & 0 & & 0 \\ \vdots & \alpha_0 & 0 & \\ \alpha_\ell & & \ddots & 0 \\ 0 & \ddots & & \ddots & 0 \\ & 0 & \alpha_\ell & \cdots & \alpha_0 \end{pmatrix} \in \mathbb{R}^{(n+1) \times (n+1)}, \quad (27)$$

whose spectrum, $\sigma(A_\ell)$, consists of the point $\alpha_0 > 0$.

In this situation, $A_\ell^\alpha e_1$, $e_1 = (1, 0, \dots, 0)^T$, $\alpha \in (0, 1)$, contains the whole set of coefficients of the corresponding Fractional BDF (FBDF) for solving FDEs (see e.g. [19] for an exhaustive overview) of the type

$${}_0D_t^\alpha y(t) = g(t, y(t)), \quad t \in (0, T], \quad (28)$$

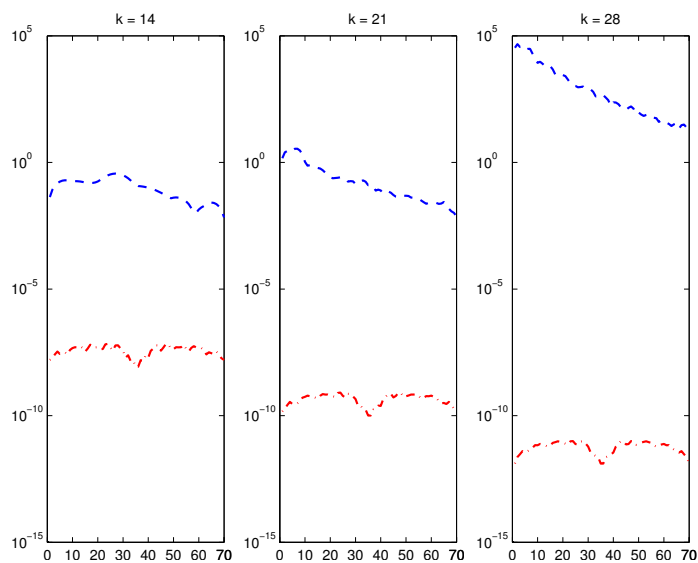


Fig. 7 Step-by-step maximum error at $t = 0.1$ of the iterated (dash-dotted line) and the standard Cholesky (dashed line) method with respect to the reference solution given by the matrix transfer technique. The time-stepping is performed by the IMEX Adams method of order 4.

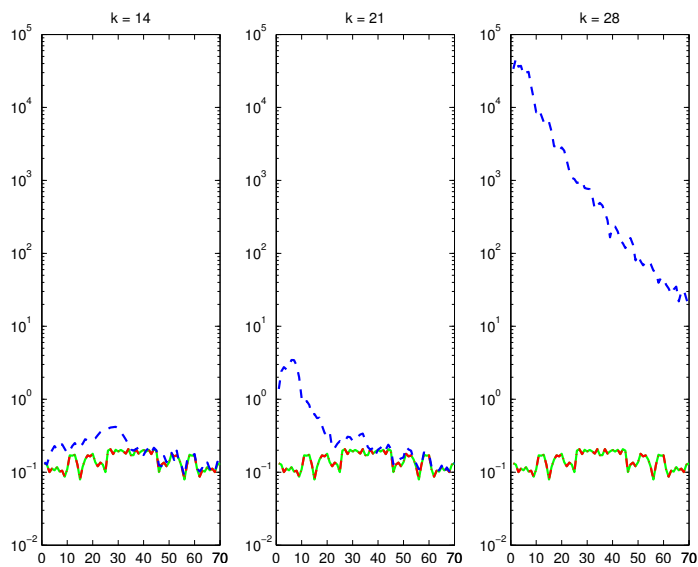


Fig. 8 Step-by-step maximum error at $t = 0.1$ of the iterated (dash-dotted line) and the standard Cholesky (dashed line) method, and the matrix transfer approach (solid line), with respect to the exact solution. The time-stepping is performed by the IMEX Adams method of order 4.

where ${}_0D_t^\alpha$ denotes the Caputo's fractional derivative operator, and where we assume to consider a uniform discretization of the time domain $t_j = j\delta, j = 0, 1, \dots, n, \delta = T/n$. FBDFs of order $\ell \geq 2$ have been introduced in [16], and extend the Grünwald-Letnikov discretization of the fractional derivative (see again [19]). We remark that the j -th entry of $A_\ell^\alpha e_1$ is just the j -th coefficient of the Taylor expansion of the generating function of the method

$$\omega_\ell^{(\alpha)}(\zeta) = \left(\alpha_0 + \alpha_1 \zeta + \dots + \alpha_\ell \zeta^\ell \right)^\alpha$$

around $\zeta = 0$.

The discrete problem provided by the FBDFs of order ℓ applied for solving (28) can be written in matrix form as follows

$$(A_\ell^\alpha \otimes I_s) Y + \mathcal{S} = \delta^\alpha G(Y), \quad (29)$$

where s is the dimension of the FDE, I_s is the identity matrix of order s , $Y = (y_1, \dots, y_n)^T \approx (y(t_1), \dots, y(t_n))^T$, $G(Y) = (g(t_1, y_1), \dots, g(t_n, y_n))^T$ and \mathcal{S} denotes a suitable starting phase. When $\ell = 1$,

$$\mathcal{S} = -(A_1^\alpha \otimes I_s) (\mathbf{1} \otimes y_0) \quad (30)$$

where $y_0 \in \mathbb{R}^s$ represents the initial value and $\mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^n$.

Using any rational approximation of the form

$$A_\ell^\alpha \approx A_\ell \sum_{j=1}^k \gamma_j (A_\ell + \eta_j I)^{-1}, \quad (31)$$

the explicit knowledge of the coefficients $\gamma_j, \eta_j, j = 1, \dots, k$, allows to construct the polynomials $p_k, q_k \in \Pi_k$ such that

$$A_\ell^\alpha \approx [q_k(A_\ell)]^{-1} p_k(A_\ell). \quad (32)$$

Note that $p_k(A_\ell)$ and $q_k(A_\ell)$ are still lower triangular matrices of bandwidth equal to ℓk , so that the approximation (32) can be used to define an implicit ℓk -step formula for FDEs (see [3, 18]). To this purpose, the use of the k -point Gauss-Jacobi rule as described in Section 2 leads to set in (31)

$$\gamma_j = \frac{2 \sin(\alpha\pi) \tau^\alpha}{\pi} \frac{w_j}{1 + \vartheta_j}, \quad \eta_j = \tau \frac{1 - \vartheta_j}{1 + \vartheta_j}.$$

Here w_j and ϑ_j denote, respectively, the weights and nodes of the Gauss-Jacobi quadrature rule with weight function $(1-t)^{\alpha-1}(1+t)^{-\alpha}$; concerning the parameter τ , a good choice is

$$\tau = \frac{(7 + \ell)}{2n} k$$

according to the analysis carried out in [3, Section 3]. Consequently, the polynomials q_k occurring in (32) are given by

$$q_k(A_\ell) = \prod_{j=1}^k (A_\ell + \eta_j I),$$

while the form of the polynomials p_k can be deduced from (12).

As already explained in [3] any numerical scheme based on the rational approximation (32) of the fractional derivative leads to a pure short-memory method so that the history part is not explicitly approximated. In this sense the approach is completely different from the techniques based on the accurate but cheap computation of this part through an efficient approximation of the underlying kernel function. For a background on this kind of approaches, we quote the recent work [6] and the references therein contained. **Every theoretical issue about the short-memory approach here considered (including computational cost, accuracy, consistency, linear stability, length of the formula, dependence on α) has been studied in [3].**

If we replace A_ℓ^α by $M^{-1}N$ in (29) accordingly to (32), and we multiply both side of the resulting equation from the left by $M \otimes I_s$, we obtain

$$(N \otimes I_s)Y + (M \otimes I_s)\mathcal{S} = \delta^\alpha(M \otimes I_s)G(Y), \quad (33)$$

where now Y represents the numerical solution provided by the ℓk -step method. The above equation can be equivalently rewritten as

$$(N \otimes I_s)Y = (M \otimes I_s)(\delta^\alpha G(Y) - \mathcal{S}), \quad (34)$$

and hence, by (11)-(12), and using the Kronecker product rules, we obtain

$$\prod_{j=1}^k (N_j \otimes I_s)Y = \prod_{j=1}^k (M_j \otimes I_s) \left(\frac{\delta^\alpha}{c\tau^\alpha} G(Y) - \frac{1}{c\tau^\alpha} \mathcal{S} \right). \quad (35)$$

The predictor-corrector approach for (35) can then be implemented. This is done in the Algorithm 3. We remark that the two linear systems occurring in such algorithm are in fact solved by k forward substitutions, in which the coefficient matrices are ℓ -banded. Indeed, for each $j = 1, \dots, n$, each system represent a $(\ell k + 1)$ -term difference equation and using Algorithm 1 we express it through $(\ell + 1)$ -term difference equations of the type

$$(\alpha_0 + \varepsilon_j)z_n + \alpha_1 z_{n-1} + \dots + \alpha_\ell z_{n-\ell} = \Phi,$$

that needs to be solved k times, and where Φ is explicitly known, α_j is from (27) and ε_j from (12). For this reason, the cost of one time-step of Algorithm 3 is comparable with the one of ℓk -step formula (33).

5.1 A numerical example

Consider the following nonlinear problem

$$\begin{aligned} {}_0D_t^\alpha u(x, t) &= \frac{\partial(p(x)u(x, t))}{\partial x} + K_\alpha \frac{\partial^2 u(x, t)}{\partial x^2} + ru(x, t) \left(1 - \frac{u(x, t)}{K} \right), \\ u(0, t) &= u(5, t) = 0, \quad t \in [0, 1], \\ u(x, 0) &= x^2(5 - x)^2, \quad x \in [0, 5]. \end{aligned}$$

This is a particular instance of the time fractional Fokker-Planck equation with a nonlinear source term [23]. In population biology it is used to model the spread of

Algorithm 3

-
- 1: compute $-\frac{1}{c\tau^\alpha}\mathcal{S}$
 - 2: set $Y^{(E)} = \mathbf{1} \otimes y_0$
 - 3: for $i = 1$ to n
 - 4: (predictor) compute $Y^{(1)}$ by solving

$$\prod_{j=1}^k (N_j \otimes I_s) Y^{(1)} = \prod_{j=1}^k (M_j \otimes I_s) \left(\frac{\delta^\alpha}{c\tau^\alpha} G(Y^{(E)}) - \frac{1}{c\tau^\alpha} \mathcal{S} \right)$$

using Algorithm 1

- 5: update $Y^{(E)} = (y_1^{(E)}, \dots, y_{i-1}^{(E)}, y_i^{(1)}, \dots, y_n^{(1)})^T$,
- 6: (corrector) compute $Y^{(2)}$ by solving

$$\prod_{j=1}^k (N_j \otimes I_s) Y^{(2)} = \prod_{j=1}^k (M_j \otimes I_s) \left(\frac{\delta^\alpha}{c\tau^\alpha} G(Y^{(E)}) - \frac{1}{c\tau^\alpha} \mathcal{S} \right)$$

using Algorithm 1

- 7: define $y_i := e_i^T Y^{(2)} \approx y(t_i)$
 - 8: end for
-

invasive species: its solution $u(x, t)$ represents the population density at location x and time t and the nonlinear source term in the equation models the population growth (in this form it is known as Kolmogorov-Fisher's growth term). In addition, r is the intrinsic growth rate of a species and K is the environmental carrying capacity, representing the maximum sustainable population density.

The application of the classical second-order semidiscretization in space with stepsize $h = 5/(s + 1)$ leads to the following initial value problem

$${}_0D_t^\alpha y(t) = Jy(t) + g(y(t)), \quad t \in (0, 1], \quad y(0) = y_0, \quad (36)$$

where, for each $i = 1, 2, \dots, s$, $(y(t))_i \equiv y_i(t) \approx u(ih, t)$, $y_i(0) = u(ih, 0)$, $(g(y))_i = ry_i(1 - y_i/K)$, and J is a tridiagonal matrix whose significant entries are

$$\begin{aligned} J_{ii} &= p'(x_i) - \frac{2K_\alpha}{h^2}, & i &= 1, \dots, s, \\ J_{i,i-1} &= -\frac{p(x_i)}{2h} + \frac{K_\alpha}{h^2}, & J_{i-1,i} &= \frac{p(x_{i-1})}{2h} + \frac{K_\alpha}{h^2}, & i &= 1, \dots, s-1. \end{aligned}$$

In our experiments, we set $\alpha = 0.8$, $p(x) = -1$, $r = 0.2$, $K_\alpha = K = 1$ (see [23, Example 5.4]); for $s = 24$, we solve (36) over a uniform meshgrid with stepsize $\delta = 1/1024$ using the following methods, based on the choice of $\ell = 1$:

- FBDF of order 1 (FBDF1), that is, by solving the full recursion (29), in which the nonlinear problem is solved by the Newton method;
- the k -step method (here $\ell = 1$) that solves (34) as in [3], and again based on the Newton method, denoted by SM, Short Memory;
- the method based on Algorithm 3, denoted by ISM, Iterated Short Memory.

We remark that the rational approximation used by the short-memory approaches is attained by defining in (4)

$$\tau = 4k/n,$$

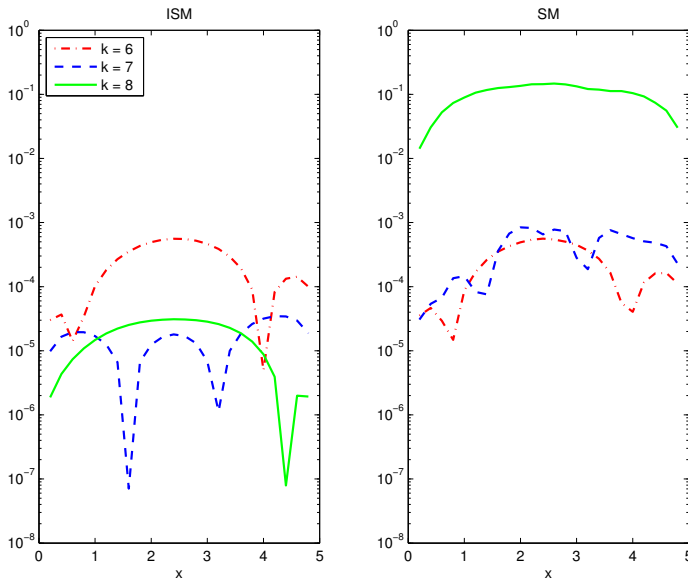


Fig. 9 Pointwise error of the k -step methods SM and ISM, for $k = 6, 7, 8$, with respect to the reference solution of the FBDF1 at $t = 1$.

as indicated in [3].

In Figure 9 we take as reference solution the one computed using the FBDF1. At $t = 1$, we compare the error of the SM and the ISM on the discrete spatial domain of the equation, changing the value of k . The right picture clearly shows the difficulties of the SM method to keep close to the FBDF1 for $k = 8$ (and also for $k > 8$ even if not reported), while the iterated approach does not suffer from this situation.

In Figure 10 we compare the step-by-step maximum norm of the error of the three methods with respect to a reference solution computed using the FBDF1 with stepsize $\delta/10$. Keeping in mind that the aim of a short-memory technique is to simulate as better as possible the behavior of the underlying full recursion, Figure 10 perfectly shows the potential of the ISM, which is able to improve the approximation for growing k regardless of the conditioning of the matrix N in (34) that, on the other side, heavily affects the behavior of the SM.

6 Conclusions

The use of rational approximations to matrix fractional powers represents a promising strategy for the solution of equations involving fractional operators. Indeed, good accuracy is attainable with a significant computational savings. In this paper we have tried to solve a linear algebra problem that were still open, that is, the severe ill-conditioning of the matrix polynomials involved in the rational approximation. In principle, this matter can be **approached** by means of quite simple algorithms, but their use inside the integrators is not straightforward and **requires** suitable adaptation of the **schemes**.

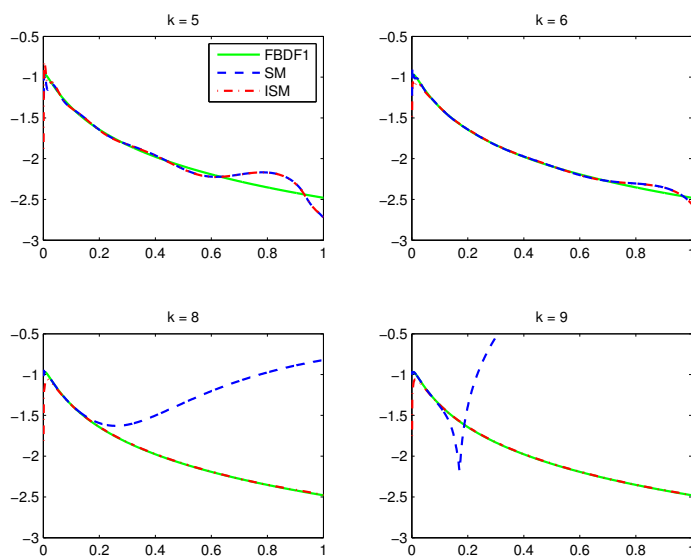


Fig. 10 Step-by-step error (in logarithmic scale) for the solutions of (36) for the FBDF1 (solid line), the SM (dashed line) and the ISM (dash-dotted line). For the short memory approaches we consider $k = 5, 6, 8, 9$.

References

1. G. Akrivis, M. Crouzeix, C. Makridakis, Implicit-explicit multistep finite element methods for nonlinear parabolic problems, *Math. Comp.*, 67, 457–477 (1998).
2. G. Akrivis, M. Crouzeix, C. Makridakis, Implicit-explicit multistep methods for quasilinear parabolic equations, *Numer. Math.*, 82, 521–541 (1999).
3. L. Aceto, C. Magherini, P. Novati, On the construction and properties of m -step methods for FDEs, *SIAM J. Sci. Comput.*, 37, A653–A675 (2015).
4. L. Aceto, P. Novati, Rational approximation to the fractional Laplacian operator in reaction-diffusion problems, *SIAM J. Sci. Comput.*, 39, A214–A228 (2017).
5. M. Benzi, D. Bertaccini, Approximate inverse preconditioning for shifted linear systems, *BIT*, 43, 231–244 (2003).
6. D. Baffet, J. S. Hesthaven, A kernel compression scheme for fractional differential equations, *SIAM J. Numer. Anal.*, 55, 496–520 (2017).
7. D. A. Bini, N. J. Higham, B. Meini, Algorithms for the matrix p th root, *Numer. Algorithms*, 39, 349–378 (2005).
8. M. Crouzeix, Une méthode multipas implicite-explicite pour l’approximation des équations d’évolution paraboliques, *Numer. Math.*, 35, 257–276 (1980).
9. A. Frommer, S. Güttel, M. Schweitzer, Efficient and stable Arnoldi restarts for matrix functions based on quadrature, *SIAM J. Matrix Anal. Appl.*, 35, 661–683 (2014).
10. O. Gómilko, F. Greco, K. Ziętak, A Padé family of iterations for the matrix sign function and related problems, *Numer. Linear Algebra Appl.*, 19, 585–605 (2012).
11. E. Hairer, S. P. Norsett, G. Wanner, Solving ordinary differential equations I. Nonstiff problems. Second edition. Springer Series in Computational Mathematics, 8. Springer-Verlag, Berlin (1993).
12. N. Hale, A. Townsend, Fast and accurate computation of Gauss-Legendre and Gauss-Jacobi quadrature nodes and weights, *SIAM J. Sci. Comput.*, 35, A652–A672 (2013).
13. N. J. Higham, Functions of matrices. Theory and computation. SIAM, Philadelphia, PA (2008).
14. M. Ilić, F. Liu, I. Turner, V. Anh, Numerical approximation of a fractional-in-space diffusion equation I, *Fract. Calc. Appl. Anal.*, 8, 323–341 (2005).

15. M. Ilić, F. Liu, I. Turner, V. Anh, Numerical approximation of a fractional-in-space diffusion equation II - with nonhomogeneous boundary conditions, *Fract. Calc. Appl. Anal.*, 9, 333–349 (2006).
16. C. Lubich, Discretized fractional calculus, *SIAM J. Math. Anal.*, 17, 704–719 (1986).
17. I. Moret, Rational Lanczos approximations to the matrix square root and related functions, *Numer. Linear Alg. Appl.*, 16, 431–445 (2009).
18. P. Novati, Numerical approximation to the fractional derivative operator, *Numer. Math.*, 127, 539–566 (2014).
19. I. Podlubny, Fractional differential equations. Mathematics in Science and Engineering, 198. Academic Press, Inc., San Diego, CA (1999).
20. I. Podlubny, Matrix approach to discrete fractional calculus, *Fract. Calc. Appl. Anal.*, 3, 359–386 (2000).
21. A. P. Prudnikov, Yu. A. Brychkov, O. I. Marichev, Integrals and Series, vol. 3: More Special Functions. Gordon and Breach Science Publishers, New York (1990).
22. H. A. Van der Vorst, An iterative solution method for solving $f(A)x = b$, using Krylov subspace information obtained for the symmetric positive definite matrix A , *J. Comput. Appl. Math.*, 18, 249–263 (1987).
23. Q. Yang, F. Liu, I. Turner, Stability and convergence of an effective numerical method for the Time-Space Fractional Fokker-Planck Equation with a nonlinear source term, *Int. J. Differ. Equ.*, Art. ID 464321, 22 pp. (2010).