Rational Krylov methods for functions of matrices with applications to fractional partial differential equations☆

L. Aceto[a], D. Bertaccini[b,c,*], F. Durastante[d], P. Novati[e]

[a] *Università di Pisa, Dipartimento di Matematica, Via F. Buonarroti, 1/C, Pisa, Italy*
[b] *Università di Roma "Tor Vergata", Dipartimento di Matematica, viale della Ricerca Scientifica 1, Roma, Italy. Email: bertaccini@mat.uniroma2.it*
[c] *Istituto per le Applicazioni del Calcolo (IAC) "M. Picone", National Research Council (CNR), Roma, Italy.*
[d] *Università di Pisa, Dipartimento di Informatica, largo Bruno Pontecorvo 3, Pisa, Italy.*
[e] *Università di Trieste, Department of Mathematics and Geosciences, via Valerio 12/1, 34127 Trieste, Italy*

Abstract

In this paper we propose a new choice of poles to define reliable rational Krylov methods. These methods are used for approximating function of positive definite matrices. In particular, the fractional power and the fractional resolvent are considered because of their importance in the numerical solution of fractional partial differential equations. The numerical experiments on some fractional partial differential equation models confirm that the proposed approach is promising.

Keywords: Fractional Laplacian, Matrix functions, Krylov methods, Gauss-Jacobi rule
2010 MSC: 65f60, 35r11, 65d32

1. Introduction

This paper deals with some computational issues concerning the use of Krylov-type methods for computing certain functions of matrices occurring in the solution of fractional partial differential equations. In particular, we focus on the following model problems:

$$\begin{cases} (-\Delta)^{\frac{\alpha}{2}} u = s(\mathbf{x}, u), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega, \end{cases} \tag{1}$$

and

$$\begin{cases} \frac{\partial u}{\partial t} = -\mu(-\Delta)^{\frac{\alpha}{2}} u + s(\mathbf{x}, u, t), & (\mathbf{x}, t) \in \Omega \times (t_0, T], \\ u(\mathbf{x}, t_0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}, t) = 0 \quad (\text{or} \quad \partial_n u(\mathbf{x}, t) = 0), & (\mathbf{x}, t) \in \partial\Omega \times (t_0, T], \end{cases} \tag{2}$$

where $\alpha \in (1, 2]$, $\mu > 0$, and $-(-\Delta)^{\alpha/2}$ is the fractional Laplacian operator with homogeneous Dirichlet (or Neumann) boundary conditions. As usual, this operator can be defined using the spectral decomposition of the Laplacian, that is,

$$(-\Delta)^{\alpha/2} u = \sum_{p=1}^{\infty} \mu_p^{\alpha/2} c_p \varphi_p, \quad u = \sum_{p=1}^{\infty} c_p \varphi_p,$$

where $\{\varphi_p\}_{p=1}^{\infty}$ are the eigenfunctions of $(-\Delta)$ and $\{\mu_p\}_{p=1}^{\infty}$ are the corresponding positive real eigenvalues. In this view, any positive definite matrix representing an appropriate discretization $A_n \in \mathbb{R}^{n \times n}$ of the Laplacian operator (that is, $\mathbf{x}^T A_n \mathbf{x} > 0$ for any nonzero $\mathbf{x} \in \mathbb{C}^n$) can be used to approximate $(-\Delta)^{\alpha/2}$ by means of $A_n^{\alpha/2}$. This approach is known as *Matrix Transfer*

☆The authors are members of the INdAM Research group GNCS, and this work was supported in part by the INDAM-GNCS 2018 projects "Tecniche innovative per problemi di algebra lineare" and "Risoluzione numerica di equazioni di evoluzione integrali e differenziali con memoria", by the University of Pisa (Grant PRA_2017_05), and FRA-University of Trieste. D. Bertaccini gratefully acknowledges the MIUR Excellence Department Project awarded to the Department of Mathematics, University of Rome Tor Vergata, CUP E83C18000100006 and the Tor Vergata University "MISSION: SUSTAINABILITY"project "NUMnoSIDS", CUP E86C18000530005.

*Corresponding author

URL: lidia.aceto@unipi.it (L. Aceto), bertaccini@mat.uniroma2.it (D. Bertaccini), fabio.durastante@di.unipi.it (F. Durastante), novati@units.it (P. Novati)

Technique (MTT) [1–6]. Here we use finite differences, finite volumes and finite elements methods to generate the underlying approximation A_n.

For what concerns the steady state problem (1), the solution can be approximated by any method able to compute the action of $A_n^{-\alpha/2}$. On the other hand, the solution of (2) requires also the discretization in time. Since A_n inherits the stiffness of the Laplacian operator, the time integration of (2) needs to be implicit. Therefore, at each time step, we need to compute the action of matrix functions of the type $\left(I + \nu A_n^{\alpha/2}\right)^{-1}$, where I denotes the identity matrix of size n and $\nu > 0$ is a parameter that depends on the stepsize and the integrator. Assuming that the stiffness is only due to the Laplacian operator, here we consider IMEX-type methods, which allow an explicit treatment of the forcing term in (2).

In this work, we analyze some Krylov-type methods approximating the action of the product of the above matrix functions by a given vector. We compare the behavior of the polynomial, shift-and-invert, extended, and rational Krylov methods in terms of accuracy and computational cost on various test problems and different discretizations. Our aim is to show that our rational Krylov method can outperform the other approaches provided that the poles of the underlying rational forms are suitably defined. To this purpose, we define the poles using the Gauss-Jacobi approach for the computation of $A_n^{-\alpha/2}$ as in [7–9], and then extend this idea for the computation of $\left(I + \nu A_n^{\alpha/2}\right)^{-1} \mathbf{v}$ giving also some details for a reliable implementation. We prove that our new poles are real and simple, so that they can be computed with a root-finder less prone to error amplification. We remark that the computation of $\left(I + \nu A_n^{\alpha/2}\right)^{-1} \mathbf{v}$ with Krylov methods has been considered in [10], but by using only the shift-and-invert approach.

The paper is organized as follows. In Section 2 we recall the basic features of Krylov methods of polynomial and rational types tailored for approximating function of matrices, while in Section 3 we give a theoretical and computational analysis of the poles of the proposed rational methods. In Section 4 we discuss some issues of the MTT based on finite differences, finite elements, and finite volumes approximation applied to the Laplacian operator. Finally, Section 5 is devoted to some numerical experiments in which we compare the performances of our approach with the other Krylov methods presented in Section 2.

2. Rational Krylov Methods

We recall that for a given matrix $A_n \in \mathbb{R}^{n\times n}$ and a function f that is analytic on and inside a closed contour Γ that encloses the spectrum of A_n, the matrix function $f(A_n)$ is defined as

$$f(A_n) = \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A_n)^{-1}\, d\, z.$$

Let V_k be an orthogonal matrix whose columns $\mathbf{v}_1, \ldots, \mathbf{v}_k$ span an arbitrary Krylov subspace $\mathcal{W}_k(A_n, \mathbf{v})$ of dimension k. We obtain an approximation of $f(A_n)\mathbf{v}$ by

$$\begin{aligned} f(A_n)\mathbf{v} =& \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A_n)^{-1}\mathbf{v}\, d\, z \\ \approx& \frac{1}{2\pi i}\int_\Gamma f(z)V_k(zI - V_k^T A_n V_k)^{-1}V_k^T\mathbf{v}\, d\, z \\ =& V_k f(V_k^T A_n V_k)V_k^T\mathbf{v}. \end{aligned} \tag{3}$$

Different methods for the approximation of matrix functions are obtained for different choices of the projection spaces $\mathcal{W}_k(A_n, \mathbf{v})$. Given a set of scalars $\{\sigma_1, \ldots, \sigma_{k-1}\} \subset \overline{\mathbb{C}}$ (the extended complex plane), that are not eigenvalues of A_n, let

$$q_{k-1}(z) = \prod_{j=1}^{k-1}(\sigma_j - z).$$

The rational Krylov subspace of order k associated with A_n, $\mathbf{v}$ and q_{k-1} is defined by

$$\mathcal{Q}_k(A_n, \mathbf{v}) = [q_{k-1}(A_n)]^{-1}\,\mathcal{K}_k(A_n, \mathbf{v}),$$

where

$$\mathcal{K}_k(A_n, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A_n\mathbf{v}, \ldots, A_n^{k-1}\mathbf{v}\}$$

is the standard polynomial Krylov space. By defining the matrices

$$C_j = (\mu_j\sigma_j A_n - I)\,(\sigma_j I - A_n)^{-1},$$

where $\{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}}$ are such that $\sigma_j \neq \mu_j^{-2}$, it is known that the rational Krylov space can also be written as follows (see [11])

$$\mathcal{Q}_k(A_n, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1}\cdots C_2C_1\mathbf{v}\}.$$

This general formulation allows to recast most of the classical Krylov methods in terms of a rational Krylov method with a specific choice of σ_j

and μ_j. In particular, the standard (polynomial) Krylov method in which $\mathcal{W}_k(A_n, \mathbf{v}) = \mathcal{K}_k(A_n, \mathbf{v})$ can be recovered by defining $\mu_j = 1$ and $\sigma_j = \infty$ for each j. The extended Krylov method (see [12, 13]), in which

$$\mathcal{W}_{2k-1}(A_n, \mathbf{v}) = \text{Span}\{\mathbf{v}, A_n^{-1}\mathbf{v}, A_n\mathbf{v}, \ldots, A_n^{-(k-1)}\mathbf{v}, A_n^{k-1}\mathbf{v}\},$$

is obtained by setting

$$(\mu_j, \sigma_j) = \begin{cases} (1, \infty), & \text{for } j \text{ even}, \\ (0, 0), & \text{for } j \text{ odd}. \end{cases}$$

The shift-and-invert (single pole) rational Krylov, see [14, 15], where

$$\mathcal{W}_k(A_n, \mathbf{v}) = \text{Span}\{\mathbf{v}, (\sigma I - A_n)^{-1}\mathbf{v}, \ldots, (\sigma I - A_n)^{-(k-1)}\mathbf{v}\},$$

is defined by taking $\mu_j = 0$ and $\sigma_j = \sigma$ for each j.

As for the use of Krylov methods for the computation of functions of operators involving $(-\Delta)^{\alpha/2}$, that is for our cases of interest, we quote here [5] and [10] in which the standard Krylov method and the shift-and-invert approach are investigated, respectively. On the same line, the computation of the matrix square root ($\alpha = 1$) is done by using the extended Krylov method in [16], see also [17], and the rational Lanczos approximation in [18].

In this work we consider a rational Krylov approach in which $\mu_j = 0$ for each j, and where $-\sigma_j = \xi_j > 0$ are suitably defined (see Section 3). In this view, the method here presented is a rational Krylov method in which

$$\mathcal{W}_k(A_n, \mathbf{v}) = \text{Span}\{\mathbf{v}, (\xi_1 I + A_n)^{-1}\mathbf{v}, \ldots, (\xi_{k-1} I + A_n)^{-1} \cdots (\xi_1 I + A_n)^{-1}\mathbf{v}\}. \tag{4}$$

Starting from $\mathbf{v}_1 = \mathbf{v}/\beta$, where $\beta = \|\mathbf{v}\|_2$, we determine $\mathbf{v}_{j+1}$ by orthogonalizing the vector

$$\mathbf{w}_j = (\xi_j I + A_n)^{-1}\mathbf{v}_j \tag{5}$$

against $\mathbf{v}_1, \ldots, \mathbf{v}_j$, followed by normalization. In this way, a sequence of vectors $\{\mathbf{v}_j\}_{j=1}^k$ is generated such that

$$\mathbf{v}_j = (\xi_j I + A_n) \sum_{i=1}^{j+1} h_{i,j}\mathbf{v}_i, \qquad \text{for } j \le k-1,$$

$$\mathbf{v}_k = (\xi_k I + A_n) \sum_{i=1}^{k} h_{i,k}\mathbf{v}_i + (\xi_k I + A_n) h_{k+1,k}\mathbf{v}_{k+1}$$

obtaining the following Arnoldi–like decomposition:

$$V_k^T A_n V_k = (I - H_k D_k) H_k^{-1} - h_{k+1,k} V_k^T A_n \mathbf{v}_{k+1} \mathbf{e}_k^T H_k^{-1},$$

where $D_k = \mathrm{diag}(\{\xi_j^{-1}\}_{j=1}^k)$ and H_k is the Hessemberg matrix $H_k = [h_{i,j}]$. Since

$$h_{k+1,k} V_k^T A_n \mathbf{v}_{k+1} \mathbf{e}_k^T H_k^{-1} = h_{k+1,k} V_k^T V_{k+1} H_{k+1} \mathbf{e}_{k+1} \mathbf{e}_k^T H_k^{-1} = O_k,$$

we find the following expression for the projected matrix

$$V_k^T A_n V_k = (I - H_k D_k) H_k^{-1}.$$

Finally, we approximate $f(A_n)\mathbf{v}$ as

$$f(A_n)\mathbf{v} \approx \beta V_k f(V_k^T A_n V_k) \mathbf{e}_1,$$

where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^k$.

As is well known, rational Krylov methods are generally quite fast (in terms of iterations with respect to the polynomial counterpart) whenever A_n represents an unbounded self-adjoint operator. In this view, the most computational demanding part of these methods is typically due to the solution of the linear systems (5). The computational cost of an iterative solver for (5), whenever A_n is large, is highly problem-dependent and often influences very much the performance of these algorithms and therefore their competitiveness with respect to polynomial methods. Anyway, in the examples considered here, we always deal with sparse and, in particular, banded matrices A_n. In this sense, we compute the approximate solution of (5) by means of sparse direct solvers. Alternatively, one may consider iterative solvers with preconditioners able to handle multiple shifts (see e.g. [19–22]).

In Section 3 we propose to select the poles ξ_j by exploiting some insights on the functions to be approximated here, i.e. $f(z) = z^{-\alpha/2}$ and $f(z) = (1+\nu z^{\alpha/2})^{-1}$, by working with suitable rational approximations. Clearly, one could look for an automatic selection of the poles that avoids the use of such information. An example of application of this strategy is represented by the `RKFIT` algorithm [23]. It can be used for computing a rational approximation for a matrix function $f(A_n)$ by solving a rational least square problem of the form

$$\text{given } f(A_n), A_n, \mathbf{v}, \text{ find } R_{k+\ell,k}(\cdot) \text{ to minimize } \|f(A_n)\mathbf{v} - R_{k+\ell,k}(A_n)\mathbf{v}\|_2^2,$$

where f is a matrix function (or an approximation), and $R_{k+\ell,k}$ is a rational function of type $(k+\ell, k)$ and $\ell \geq -k$. Thus, the minimum is taken on the roots of the polynomials defining $R_{k+\ell,k}$. To effectively use this procedure, one needs to find some starting poles for $R_{k+\ell,k}(z)$ and the values of k and ℓ.

3. Poles Selection

Our proposal for selecting the poles for the construction of the rational Krylov subspace (4) relies on the rational approximation of $z^{-\alpha/2}$ proposed in [7–9]. In particular, following [8, eq. (5)], we get

$$z^{-\alpha/2} \approx \sum_{j=1}^{k} \frac{2\sin(\frac{\alpha}{2}\pi)\tau^{1-\alpha/2}}{\pi} \frac{\omega_j}{1+\theta_j} \left(\frac{\tau(1-\theta_j)}{1+\theta_j} + z \right)^{-1} \triangleq R_{k-1,k}(z),$$

where ω_j and θ_j are, respectively, the weights and nodes of the Gauss–Jacobi quadrature formula with weight function $(1-x)^{-\frac{\alpha}{2}}(1+x)^{\frac{\alpha}{2}-1}$ and τ is a positive real parameter that should be defined suitably. In addition, denoting by ζ_r the rth zero of the Jacobi polynomial $\mathcal{P}_{k-1}^{(\alpha/2,1-\alpha/2)}(z)$ and setting

$$\epsilon_r = \tau \frac{1-\zeta_r}{1+\zeta_r}, \quad r = 1, 2, \ldots, k-1, \tag{6}$$

$$\eta_j = \frac{\tau(1-\theta_j)}{1+\theta_j}, \quad j = 1, 2, \ldots, k, \tag{7}$$

from [8, Proposition 1] we can express $R_{k-1,k}(z)$ as the rational function

$$R_{k-1,k}(z) = \frac{p_{k-1}(z)}{q_k(z)} = \frac{\chi \prod_{r=1}^{k-1}(z+\epsilon_r)}{\prod_{j=1}^{k}(z+\eta_j)},$$

where

$$\chi = \frac{\eta_k}{\tau^{\alpha/2}} \frac{\binom{k+\alpha/2-1}{k-1}}{\binom{k-\alpha/2}{k}} \prod_{j=1}^{k-1} \frac{\eta_j}{\epsilon_j}.$$

It is worth noting that $\{\theta_j\}_{j=1}^{k}$ are the zeros of the Jacobi polynomial $\mathcal{P}_k^{(-\alpha/2,\alpha/2-1)}(z)$. In this context, this implies that the values η_j are all real and simple and therefore the roots of $q_k(z)$ as well.

Given a positive definite matrix A_n with spectrum $\sigma(A_n) \subseteq [\lambda_{\min}, \lambda_{\max}]$, a specific rational approximation for the matrix function $f(A_n) = A_n^{-\alpha/2}$

can be deduced from the above, provided that the parameter τ is selected. To this purpose, setting

$$\tilde{\tau}_k \triangleq \lambda_{\min}\left(\frac{\alpha/2}{2ke}\right)^2 \exp\left(2W\left(\frac{4k^2e}{(\alpha/2)^2}\right)\right)$$

with $W(\cdot)$ denoting the Lambert-W function, in [7, Propositions 3.4, 4.1] it has been shown that for the matrix function $A_n^{-\alpha/2}$ a reliable value of τ is given by

$$\tau = \tau_k := \begin{cases} \tilde{\tau}_k, & \text{if } k \le \overline{k}, \\ \left(\tilde{\sigma}_k + \sqrt{\tilde{\sigma}_k^2 + (\lambda_{\min}\lambda_{\max})^{1/2}}\right)^2, & \text{if } k > \overline{k}, \end{cases}$$

where

$$\tilde{\sigma}_k \triangleq -\frac{\alpha/2}{8k}\ln\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)\lambda_{\max}^{1/2},$$

and

$$\overline{k} = \frac{(\alpha/2)^2}{8}\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)^{1/2}\left[\ln\left(\frac{\lambda_{\max}}{\lambda_{\min}}\right)+2\right].$$

This choice is the result of an analysis on the Padé-type approximations of $z^{-\alpha/2}$. Now, by using the above arguments, we can get an approximation also for $f(z) = (1+\nu z^{\alpha/2})^{-1}$. Indeed, we can write

$$\begin{aligned}(1+\nu z^{\alpha/2})^{-1} =& \frac{1}{1+\nu(z^{-\alpha/2})^{-1}} \approx \frac{1}{1+\nu(R_{k-1,k}(z))^{-1}} = \frac{1}{1+\nu\frac{q_k(z)}{p_{k-1}(z)}} \\ =& \frac{p_{k-1}(z)}{p_{k-1}(z)+\nu q_k(z)} := \frac{p_{k-1}(z)}{\tilde{q}_k(z)}.\end{aligned}$$

For the roots of $\tilde{q}_k(z)$ we observe the following useful result.

Proposition 1. *All the roots of the polynomial $\tilde{q}_k(z)$ are real and simple.*

To prove this we need two auxiliary results. First, there is an interlacing of the zeros of Jacobi polynomials of different orders and weights.

Theorem 1. *[24, Theorem 2.3] Let $\beta, \gamma > -1$. For any $t, s \in [0,2]$, let*

- *$-1 < \theta_1 < \theta_2 < \ldots < \theta_n < 1$ be the zeros of $\mathcal{P}_n^{(\beta,\gamma)}$, and*
- *$-1 < \zeta_1 < \zeta_2 < \ldots < \zeta_{n-1} < 1$ be the zeros of $\mathcal{P}_{n-1}^{(\beta+t,\gamma+s)}$.*

Then

$$-1 < \theta_1 < \zeta_1 < \theta_2 < \zeta_2 < \ldots < \theta_{n-1} < \zeta_{n-1} < \theta_n < 1.$$

Theorem 2. *[25, Theorem 8] Let p and q be real polynomials. Then p and q have strictly alternating roots if and only if all polynomials in the space*

$$\{c_1\, p + c_2\, q \,:\, c_i \in \mathbb{R}, i = 1, 2\}$$

have real and simple roots.

Proof of Proposition 1. To apply Theorem 2, we need to show that the roots of the polynomials $p_{k-1}(z)$ and $q_k(z)$ are strictly interlaced. By construction we know that the roots of $p_{k-1}(z)$ are the real values $\{-\epsilon_r\}_{r=1}^{k-1}$ in (6), while the roots of $q_k(z)$ are the real values $\{-\eta_j\}_{j=1}^{k}$ in (7). By Theorem 1 applied to the Jacobi polynomials $\mathcal{P}_k^{(-\alpha/2,\alpha/2-1)}(z)$ and $\mathcal{P}_{k-1}^{(\alpha/2,1-\alpha/2)}(z)$ we have

$$-1 < \theta_1 < \zeta_1 < \theta_2 < \zeta_2 < \ldots < \theta_{k-1} < \zeta_{k-1} < \theta_k < 1. \tag{8}$$

Then, for the interlacing between the $\{-\epsilon_r\}_{r=1}^{k-1}$ and the $\{-\eta_j\}_{j=1}^{k}$, we need to prove that $-\eta_1 < -\epsilon_1 < -\eta_2 < -\epsilon_2 < \ldots < -\eta_{k-1} < -\epsilon_{k-1} < -\eta_k$ or, equivalently,

$$\eta_1 > \epsilon_1 > \eta_2 > \epsilon_2 > \ldots > \eta_{k-1} > \epsilon_{k-1} > \eta_k.$$

Now, using (6) and (7) for $i = 1, \ldots, k-1$, we have

$$\epsilon_i > \eta_{i+1} \quad \Leftrightarrow \quad \frac{\tau\,(1-\zeta_i)}{1+\zeta_i} > \frac{\tau\,(1-\theta_{i+1})}{1+\theta_{i+1}}.$$

Since $\tau > 0$, $(1+\zeta_i) > 0$ and $(1+\theta_{i+1}) > 0$, the previous inequalities are satisfied if and only if $\zeta_i < \theta_{i+1}$, which is true by (8). Similar arguments lead to verify that $\eta_i > \varepsilon_i$. By using Theorem 2, the proof is complete. □

In addition, since all the coefficients of $\tilde{q}_k(z)$ are strictly positive by construction, according to the Descartes' rule of signs, we are also sure that all its roots are negative.

In consideration of all the above arguments, the poles that we use for the computation of the proposed rational Krylov methods are positive, real and simple. In fact, when $f(A_n) = A_n^{-\alpha/2}$, in (4) we take as poles the opposite of the roots of $q_k(z)$ (i.e., $\xi_j = \eta_j$), while, when $f(A_n) = (I + \nu A_n^{\alpha/2})^{-1}$, we take as poles the opposite of the roots of $\tilde{q}_k(z)$. Thus, one of the main

features of our proposal is represented by the possibility to work only with real arithmetic. From now on, we refer to the rational Krylov methods based on these poles as *Krylov Jacobi*.

Just to provide an example of computed poles, in Figure 1 we focus on the matrix A_n given by the centered differences discretization of the 1D Laplacian with $n = 2^{12}$ and $\alpha = 1.2$. In this case, for $k = 10, 20, 30$, we computed the roots $\{\theta_j\}_{j=1}^{k}$ and $\{\zeta_r\}_{r=1}^{k-1}$ by the `JACOBI_POLYNOMIAL` Matlab code; see [26]. On the other hand, we used the package MPSolve to compute the roots of $\tilde{q}_k(z)$; see [27, 28].

4. Solving the model problems by MTT using Krylov methods

Computing efficiently functions of matrix-vector products by Krylov methods is beneficial for many applications. However, in order to provide reasonable comparisons, here we consider one class of mathematical tools which is quite wide and important because it forms the ground of many numerical models: (evolutionary) fractional partial differential equations.

As a case study, let us consider the d–dimensional Laplacian operator

$$\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}, \tag{9}$$

on the domain Ω subject to Dirichlet (or Neumann) boundary conditions. The *Matrix Transfer Technique* (MTT) introduced in [3, 4] considers approximations to the fractional Laplacian of the form

$$(-\Delta)^{\frac{\alpha}{2}} \approx A_n^{\alpha/2},$$

where A_n is a suitable matrix approximation of the standard Laplacian. In this work, we consider the following discretizations:

- finite differences (FD);
- finite elements methods (FEM) with first order Lagrangian elements on a triangular tessellation;
- cell–centered finite volume (FV) methods with a piecewise linear basis, [29].

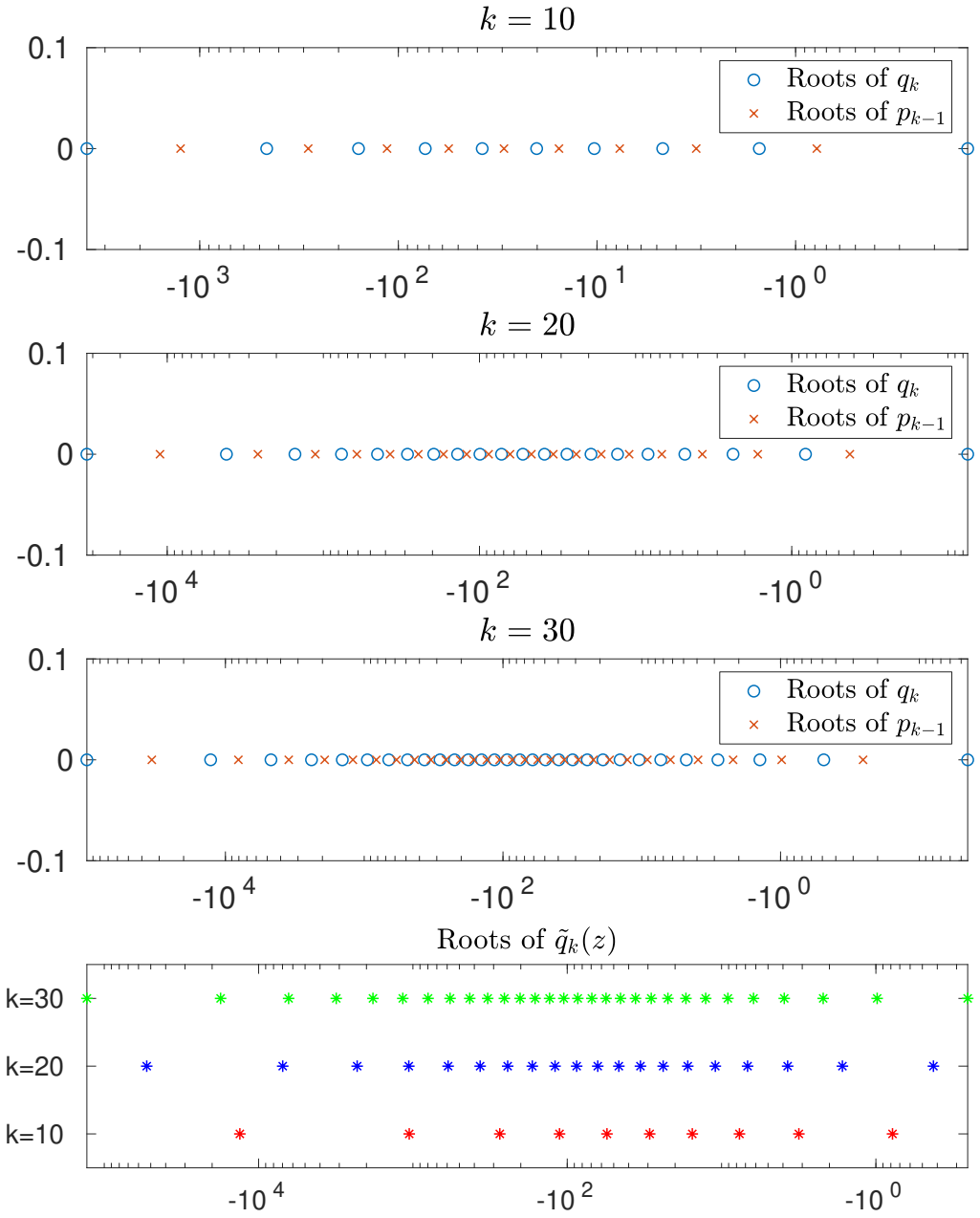


Figure 1: Poles for the computation of the function $f(z) = (1 + \nu z^{\alpha/2})^{-1}$ for the matrix A_n given by the centered differences discretization of the 1D Laplacian with $n = 2^{12}$, $\alpha = 1.2$, and $\nu = 1/(n + 1)$.

For FEM and FV we get $A_n = M_n^{-1}K_n$, where K_n denotes the stiffness matrix, while M_n is the mass matrix for the FEM or the diagonal matrix of the cell volumes for FV. In these cases, even if A_n is generally not symmetric, it is similar to a positive definite matrix. In fact, $M_n^{\frac{1}{2}} A_n M_n^{-\frac{1}{2}} = M_n^{-\frac{1}{2}} K_n M_n^{-\frac{1}{2}}$. Whenever the mass matrix M_n is not trivial, i.e. A_n is not a multiple of K_n, we should solve linear systems with coefficient matrices $(\xi_j I + M_n^{-1}K_n)$, by means of a direct method. This can be inappropriate because it would require assembling the matrix $A_n = M_n^{-1}K_n$ that can be dense. Therefore, instead of computing the basis of the rational Krylov space with A_n, we generate it working with $\hat{A}_n = \xi_j M_n + K_n$. This is completely analogous to what is usually done for generalized eigenvalues for the pencil (K_n, M_n) with rational Krylov methods, see, e.g., [30, 31]. Then, with the attained basis, we approximate the matrix–vector product $f(A_n)\mathbf{v}$ as in (3).

Finally, we stress that direct solvers for sparse linear systems can benefit by the use of permutations to reduce fill–in. This can be mostly useful when unstructured matrices are used for either FEM or FV discretizations.

4.1. Integration in time

If we consider the solution of the problem (2) semidiscretized with respect to the space variables, we get the initial value problem

$$\begin{cases} \mathbf{y}'(t) = -A_n^{\frac{\alpha}{2}}\mathbf{y}(t) + \mathbf{s}(t), & t \in (t_0, T], \\ \mathbf{y}(t_0) = \mathbf{y}_0, \end{cases} \tag{10}$$

where $\mathbf{y}(t) : \mathbb{R} \to \mathbb{R}^n$ and $\mathbf{y_0} \in \mathbb{R}^n$ (with respect to problem (2), to simplify notation, here we set $\mu = 1$ and $s(\mathbf{x}, u, t) = s(\mathbf{x}, t)$). To integrate (10) in time, we apply a *Linear Multistep Method* (LMM) over an equispaced grid $t_j = t_0 + j\delta_t, j = 0, 1, \ldots, n_t$, with $\delta_t = (T - t_0)/n_t$ the selected stepsize. Setting $\mathbf{g}(\mathbf{y}, t) = -A_n^{\frac{\alpha}{2}}\mathbf{y}(t) + \mathbf{s}(t)$, we get

$$\sum_{j=0}^{\ell} \alpha_j \mathbf{y}^{(m+j)} = \delta_t \sum_{j=0}^{\ell} \beta_j \mathbf{g}^{(m+j)}, \qquad m = 0, 1, \ldots, n_t - \ell, \tag{11}$$

with

$$\mathbf{y}^{(m+j)} \approx \mathbf{y}(t_{m+j}), \quad \mathbf{g}^{(m+j)} = \mathbf{g}(\mathbf{y}^{(m+j)}, t_{m+j}).$$

Since the fractional Laplacian operator is unbounded and give stiffness , implicit schemes are preferable to avoid possible severe stepsize restrictions

in order to satisfy stability requirements; see, e.g., [32]. Using implicit schemes, we need to solve linear systems of the form

$$\left(\alpha_\ell I + \delta_t \beta_\ell A_n^{\frac{\alpha}{2}}\right) \mathbf{y}^{(m+\ell)} = \delta_t \sum_{j=0}^{\ell-1} \beta_j \mathbf{g}^{(m+j)} - \sum_{j=0}^{\ell-1} \alpha_j \mathbf{y}^{(m+j)},\ m = 0, 1, \ldots, n_t - \ell,$$

that can be recast in the computation of a function of matrix times vector, that is,

$$\mathbf{y}^{(m+\ell)} = f(A_n)\tilde{\mathbf{y}}^{(m)}, \qquad m = 0, 1, \ldots, n_t - \ell,$$

where

$$f(z) = \left(1 + \delta_t \frac{\beta_\ell}{\alpha_\ell} z^{\frac{\alpha}{2}}\right)^{-1} \tag{12}$$

and

$$\tilde{\mathbf{y}}^{(m)} = \frac{1}{\alpha_\ell} \left(\delta_t \sum_{j=0}^{\ell-1} \beta_j \mathbf{g}^{(m+j)} - \sum_{j=0}^{\ell-1} \alpha_j \mathbf{y}^{(m+j)} \right).$$

By using similar arguments, we can also deal with semi–linear problems having non–linear forcing term

$$\begin{cases} \mathbf{y}'(t) = -\mu A_n^{\frac{\alpha}{2}} \mathbf{y}(t) + \mathbf{s}(t, \mathbf{y}(t)), \quad t \in (t_0, T], \\ \mathbf{y}(t_0) = \mathbf{y}_0. \end{cases} \tag{13}$$

Assuming that the function $\mathbf{s}$ is not responsible for further stiffness, we can use *implicit–explicit methods* (IMEX), similarly to [8]. A generic ℓ-step IMEX method for (13) can be written as

$$\sum_{j=0}^{\ell} \alpha_j \mathbf{y}^{(m+j)} = -\delta_t \mu \sum_{j=0}^{\ell} \beta_j A_n^{\frac{\alpha}{2}} \mathbf{y}^{(m+j)} + \delta_t \sum_{j=0}^{\ell-1} \gamma_j \mathbf{s}(t_{m+j}, \mathbf{y}^{(m+j)}). \tag{14}$$

For further details on IMEX methods see, e.g., [33, 34]. Therefore, we need again to compute matrix functions of the form (12).

5. Numerical Experiments

The examples in this section are collected in two groups. In Section 5.1 we deal with the discrete version of the steady state problem (1), while in Section 5.2 the time dependent problem (2).

All the numerical experiments are performed on a laptop running Linux with 8 Gb memory and CPU Intel® Core™ i7-4710HQ CPU with clock 2.50 GHz. The codes are written and executed in MATLAB R2018a. The following external codes are used in our routines:

- MPSolve package for the computation of the polynomial zeros; see [27],
- `rat_krylov` for the construction of the basis of the rational Kyrlov methods; see [11],
- `EKS` class for the construction of the basis of the extended Krylov method; see [13],
- FENICS library [35] v.2018.1 to assemble the finite elements matrices.
- FiPy library [36] v.3.1.3-dev2-g11937196 to assemble the finite volume matrices.

The auxiliary linear systems are solved here by Matlab's standard backslash. Whenever is feasible, the reference solution $\mathbf{u}^*$ for the various problem is computed directly by the Schur–Parlett algorithm. We denote by $\varepsilon = \|\mathbf{u}^* - \mathbf{u}\|_2/\|\mathbf{u}^*\|_2$ the relative error. All the timings are measured in seconds averaged on one hundred runs. For the definition of the poles, the interval $[\lambda_{\min}, \lambda_{\max}]$ is always assumed explicitly known.

5.1. Stationary problems

We report the results for the steady state problem (1) discretized using second order centered differences for the 1D and 2D Laplacian on the domains $[0, 1]$ and $[0, 1]^2$, respectively. The right–hand side terms for the two problems are $f(x) = \sin(\pi x)$, and $f(x, y) = \sin(\pi x)\sin(\pi y)$. We observe the expected behavior: rational Krylov Jacobi algorithms outperform (in term of iterations) the polynomial Krylov; see Figure 2 for the behavior for the one dimensional problem, and Figure 3 for the two dimensional. In both cases the shift parameter σ for the shift–and–invert method is computed as $\sigma = \sqrt{\lambda_{\min}(A_n)\lambda_{\max}(A_n)}$, cf. [18].

5.2. Time–dependent problems

In this section, we focus on the problem (2). First, we set $s = 0$, $\mu = 1$, $T = 1$,

$$u(x, y, 0) = x^2y^2(1 - x)(1 - y), \tag{15}$$

on the unit square $\Omega = [0, 1]^2$ and consider Dirichlet boundary conditions. We refer to this test problem as P1. By using finite differences (FD), finite volumes (FV) and finite elements methods (FEM) we discretize this problem and test on it the underlying rational Krylov methods. Then,

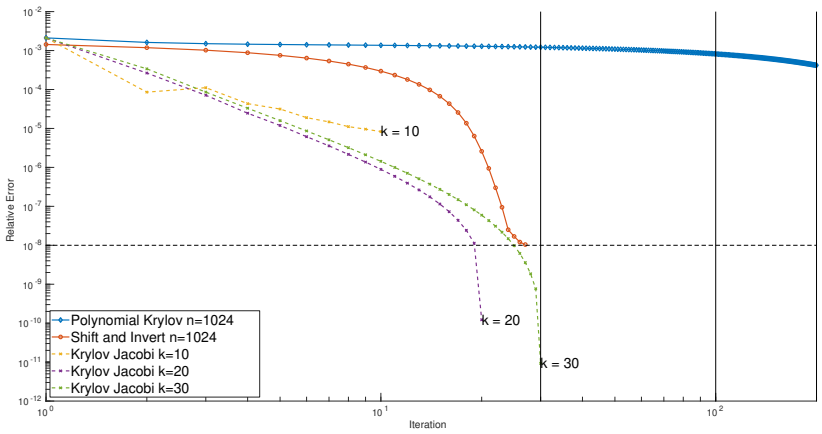


Figure 2: Comparisons of the convergence curve in term of the relative error ε with the reference solution for the 1D stationary problem discretized by finite differences for $\alpha = 1.2, k = 10, 20, 30$. The behavior for other values of α is similar.

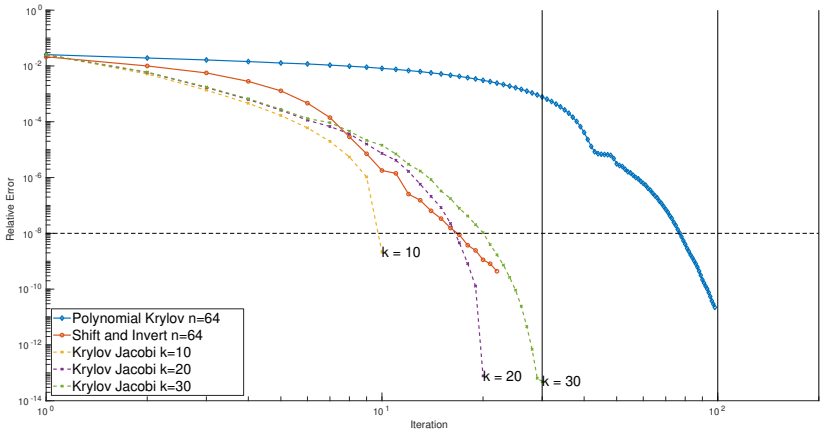


Figure 3: Comparisons of the convergence curve in term of the relative error ε with the reference solution for the 2D stationary problems discretized by finite differences for $\alpha = 1.5, k = 10, 20, 30$. The behavior for other values of α is similar.

we consider the problem (2) with s a nonzero polynomial function of the solution, producing the *fractional Allen–Cahn* equation, that is

$$\begin{cases} \dfrac{\partial u}{\partial t} + \mu(-\Delta)^{\frac{\alpha}{2}} u = -(u^3 - u), & (\mathbf{x}, t) \in \Omega \times [t_0, T], \\ u(\mathbf{x}, t_0) = u_0(\mathbf{x}), & \mathbf{x} \in \Omega, \\ \partial_n u(\mathbf{x}, t) = 0, & (\mathbf{x}, t) \in \partial\Omega \times (t_0, T]. \end{cases} \tag{16}$$

Here $\mu > 0$ is a small parameter defining the thickness of the interface separating the different phases and $u_0 \in \mathbb{L}^2(\Omega)$. In our numerical experiments we choose $\Omega = [0, 1]^2$, $[t_0, T] = [0, 4]$, and

$$u_0(\mathbf{x}) \equiv u_0(x, y) = 0.25 \sin(2\pi x) \sin(2\pi y). \tag{17}$$

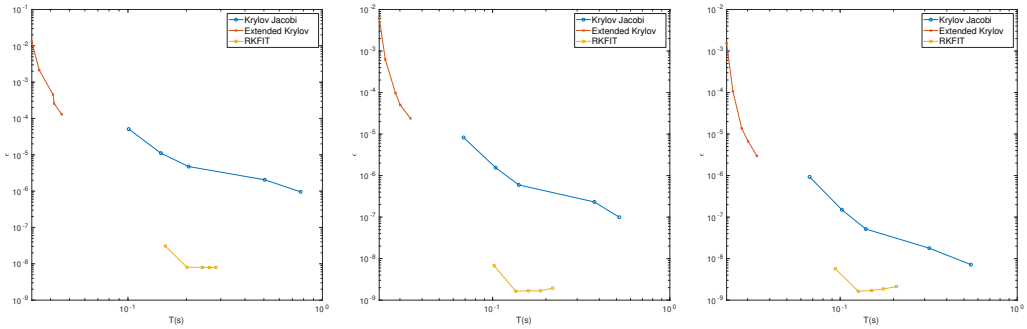
Refer to this test problem as P2. We apply finite volumes and finite elements methods to discretize this problem. In both cases the obtained matrices A_n are positive definite, and thus the analysis in Section 3 applies straightforwardly.

P1 by using FD. We consider the five–point stencil of the FD discretization for the Laplacian operator on the grid with $n_x = n_y = n_t = 2^6$ points. The method for marching in time is the *implicit Euler*. Therefore, as discussed in Section 4, in order to advance in time, we need to compute the matrix function

$$f(A_n) = (I + c\,\delta_t A_n^{\alpha/2})^{-1},$$

where c is a constant. In Figure 4 the relative error and execution time comparison for the first iterate of the method is reported. The poles for the RKFIT algorithm are computed on the complete matrix function on a reduced size grid ($n_x = n_y = 2^4$ and $n_t = 2^6$) from the initial guess $\{\sigma_j = +\infty\}_{j=1}^{k}$. Note that the RKFIT can compute only $k = 10$ different nodes. We recall also that for the extended Krylov algorithm the number k represent the size of the Krylov subspaces, thus the number of linear auxiliary linear systems solved is $\lfloor k/2 \rfloor$. This explain the behavior with respect to time: the Krylov space of the same dimension for the extended Krylov costs roughly a half of the rational subspace of the same size.

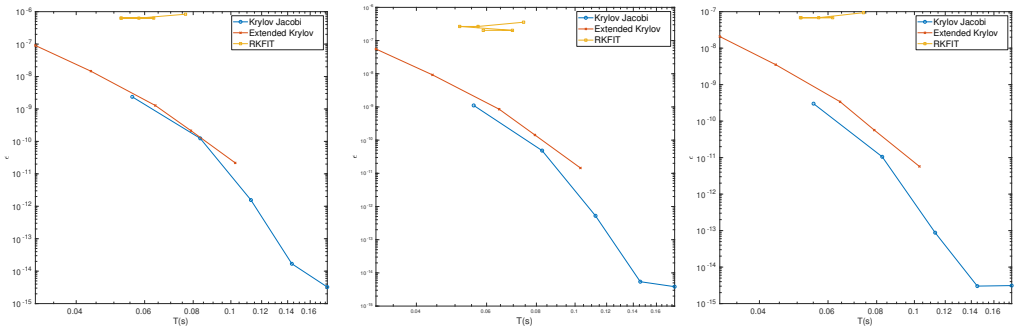
P1 by using FV. We consider the cell–centered FV discretization of the problem (2) on the same grid used for finite differences. In particular, $n_x = n_y = n_t = 2^6$. Results in Figure 5 are comparable with those of the FD discretization. The RKFIT algorithm produces less than the requested



(a) $\alpha = 1.2$ (b) $\alpha = 1.5$ (c) $\alpha = 1.8$

Figure 4: P1– FD: The nodes in the time–error graph corresponds to $k = 10, 15, 20, 25, 30$ poles.

number of nodes and at a greater computational effort. By fixing the size of the Krylov space, we observe that the Krylov Jacobi method is more accurate than the extended Krylov method which, on the other hand, keeps the same ratio with respect to the achieved timings.

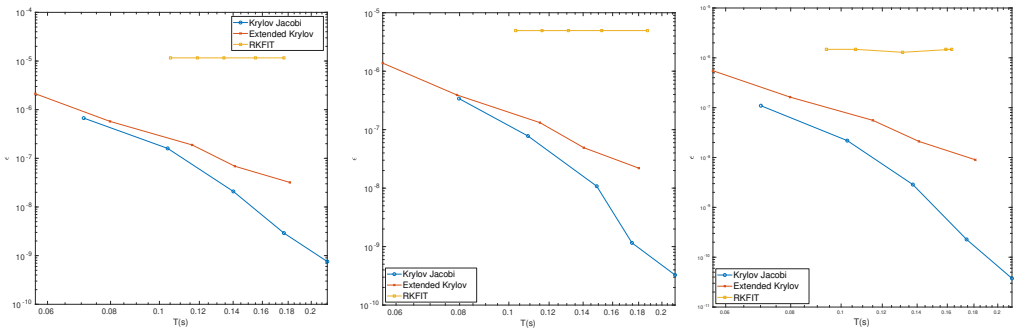


(a) $\alpha = 1.2$ (b) $\alpha = 1.5$ (c) $\alpha = 1.8$

Figure 5: P1– FV: The nodes in the time–error graph corresponds to $k = 10, 15, 20, 25, 30$ poles.

P1 by using FEM. Consider here the FEM discretization of problem (2) on the structured mesh made by triangular cells with straight sides, i.e., each side of the squared domain Ω is divided into n_x and n_y rectangles, and then divided into a pair of triangles, with Lagrangian elements of order 1. From Figure 6 we observe that the RKFIT algorithm produces again less than

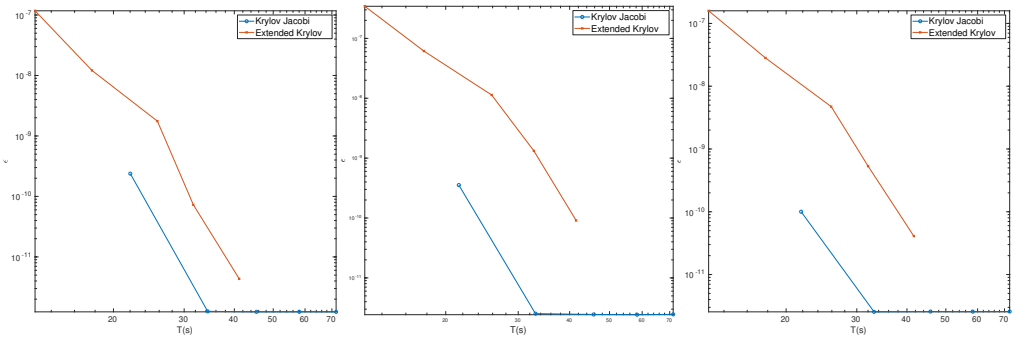
the requested number of nodes and at a greater computational effort. Considering Krylov spaces of the same size, the Krylov Jacobi method is more accurate than the extended Krylov. Moreover, the latter in this example is only marginally better with respect to the execution time.



(a) $\alpha = 1.2$ (b) $\alpha = 1.5$ (c) $\alpha = 1.8$

Figure 6: P1– FEM: The nodes in the time–error graph corresponds to $k = 10, 15, 20, 25, 30$ poles.

P2 by using FV. Consider the cell–centered FV for (16) over a uniform quadrangular mesh with $n_x = n_y = 64$ intervals on the unit square $[0, 1]^2$. To solve this problem we consider the application of the IMEX backward Euler method with $\delta_t = 10^{-2}$. The results are collected in Figure 7, from which



(a) $\alpha = 1.2$ (b) $\alpha = 1.5$ (c) $\alpha = 1.8$

Figure 7: P2– FV: Allen–Cahn equation (16) with $\mu = 10^{-4}$. The nodes in the time–error graph corresponds to $k = 10, 15, 20, 25, 30$ poles.

we observe that the Krylov Jacobi method achieves the better accuracy

already with $k = 5$ nodes. The RKFIT Algorithm, tuned on the problem with $n_x = n_y = 32$, fails to compute acceptable nodes in this case, see the example of the errors for $\alpha = 1.8$ given in Figure 8.

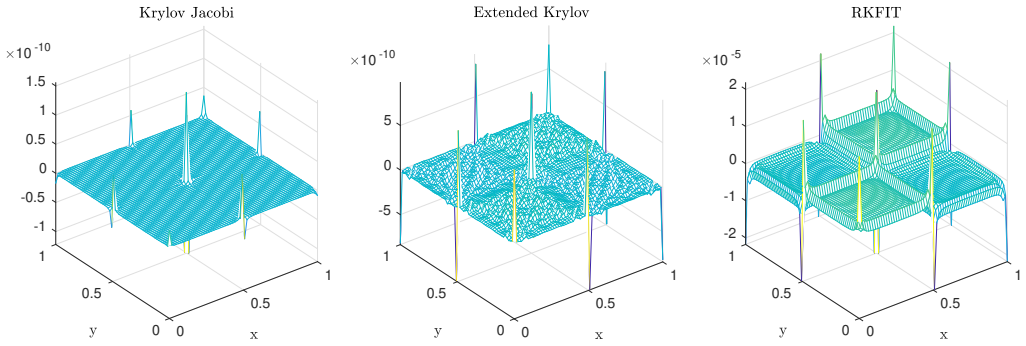


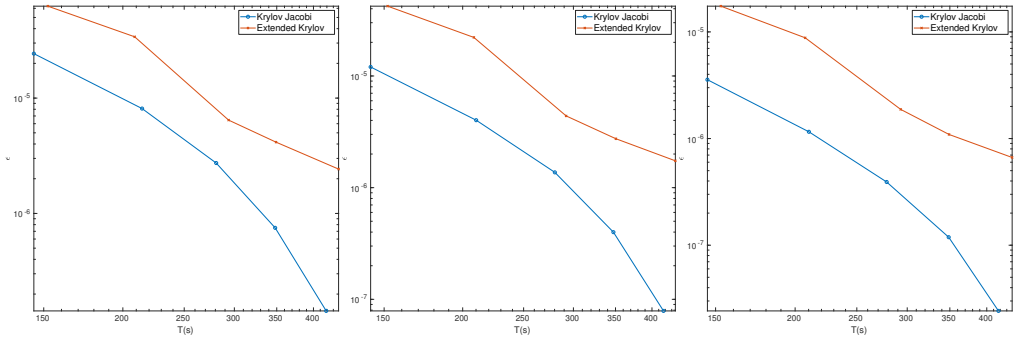
Figure 8: P2– FV: Comparison of the relative error for the Allen–Cahn equation (16) with $\mu = 10^{-2}$.

P2 by using FEM. Consider the discretization of (16) by means of Lagrange linear elements over a uniform finite element mesh on the unit square $[0, 1]^2$, i.e., a mesh consisting of triangular cells with straight sides dividing each side of the square into n_x and n_y rectangles, each divided into a pair of triangles, for $n_x = n_y = 80$. The integration method in time is the IMEX backward Euler with $\delta_t = 10^{-2}$. The performance of the considered rational Krylov method are shown in Figure 9. Note that the behavior is analogous to the one observed for the time–dependent problem (2) (compare the results with those in Figure 6).

6. Conclusion and perspectives

We considered certain matrix functions involving the fractional power of real positive definite matrices computed by rational Krylov methods with application to the numerical approximation of fractional-in-space partial differential equations. We proposed a new choice of the poles, real and simple by construction, that appears to be effective and faster with respect to polynomial, shift-and-invert, extended Krylov methods, and RKFIT.

Within the same framework, one can apply the proposed rational Krylov Jacobi method also to *fractional linear multistep methods* [37] for solving (2) in which the first order derivative in time is replaced by a fractional derivative of order β, with $\beta \in (0, 1)$.



(a) $\alpha = 1.2$ (b) $\alpha = 1.5$ (c) $\alpha = 1.8$

Figure 9: P2– FEM: Allen–Cahn equation (16) with $\mu = 10^{-3}$. The nodes in the time–error graph corresponds to $k = 10, 15, 20, 25, 30$ poles.

In principle, the proposed method can work on irregular domains, Robin or Dirichlet BC, see, e.g., [38–40], and can be used also in contexts of adaptivity, provided that these generate a symmetric positive definite matrix. In the latter case, studying how the selected poles vary as the mesh is changed would be of interest, and could also open new alternative approaches.

[1] K. Burrage, N. Hale, D. Kay, An efficient implicit FEM scheme for fractional-in-space reaction-diffusion equations, SIAM J. Sci. Comput. 34 (4) (2012) A2145–A2172. doi:10.1137/110847007.
URL https://doi.org/10.1137/110847007

[2] P. N. Vabishchevich, Numerical solution of time-dependent problems with fractional power elliptic operator, Comput. Methods Appl. Math. 18 (1) (2018) 111–128. doi:10.1515/cmam-2017-0028.
URL https://doi.org/10.1515/cmam-2017-0028

[3] M. Ilic, F. Liu, I. Turner, V. Anh, Numerical approximation of a fractional-in-space diffusion equation. I, Fract. Calc. Appl. Anal. 8 (3) (2005) 323–341.

[4] M. Ilic, 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 (4) (2006) 333–349.

[5] Q. Yang, I. Turner, F. Liu, M. Ilić, Novel numerical methods for solving the time-space fractional diffusion equation in two dimensions, SIAM J. Sci. Comput. 33 (3) (2011) 1159–1180. doi:10.1137/100800634.
URL https://doi.org/10.1137/100800634

[6] Q. Yang, I. Turner, T. Moroney, F. Liu, A finite volume scheme with preconditioned Lanczos method for two-dimensional space-fractional reaction-diffusion equations, Appl. Math. Model. 38 (15-16) (2014) 3755–3762. doi:10.1016/j.apm.2014.02.005.
URL https://doi.org/10.1016/j.apm.2014.02.005

[7] L. Aceto, P. Novati, Rational approximations to fractional powers of self-adjoint

positive operators, Numer. Math.doi:10.1007/s00211-019-01048-4.
URL https://doi.org/10.1007/s00211-019-01048-4

[8] L. Aceto, P. Novati, Efficient implementation of rational approximations to fractional differential operators, J. Sci. Comput. 76 (1) (2018) 651–671. doi:10.1007/s10915-017-0633-2.
URL https://doi.org/10.1007/s10915-017-0633-2

[9] L. Aceto, P. Novati, Rational approximation to the fractional Laplacian operator in reaction-diffusion problems, SIAM J. Sci. Comput. 39 (1) (2017) A214–A228. doi:10.1137/16M1064714.
URL https://doi.org/10.1137/16M1064714

[10] I. Moret, P. Novati, Krylov subspace methods for functions of fractional differential operators, Math. Comp. 88 (315) (2019) 293–312. doi:10.1090/mcom/3332.
URL https://doi.org/10.1090/mcom/3332

[11] S. Güttel, Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection, GAMM-Mitt. 36 (1) (2013) 8–31. doi:10.1002/gamm.201310002.
URL https://doi.org/10.1002/gamm.201310002

[12] V. Druskin, L. Knizhnerman, Extended Krylov subspaces: approximation of the matrix square root and related functions, SIAM J. Matrix Anal. Appl. 19 (3) (1998) 755–771. doi:10.1137/S0895479895292400.
URL https://doi.org/10.1137/S0895479895292400

[13] L. Knizhnerman, V. Simoncini, A new investigation of the extended Krylov subspace method for matrix function evaluations, Numer. Linear Algebra Appl. 17 (4) (2010) 615–638. doi:10.1002/nla.652.
URL https://doi.org/10.1002/nla.652

[14] I. Moret, P. Novati, RD-rational approximations of the matrix exponential, BIT 44 (3) (2004) 595–615. doi:10.1023/B:BITN.0000046805.27551.3b.
URL https://doi.org/10.1023/B:BITN.0000046805.27551.3b

[15] J. van den Eshof, M. Hochbruck, Preconditioning Lanczos approximations to the matrix exponential, SIAM J. Sci. Comput. 27 (4) (2006) 1438–1457. doi:10.1137/040605461.
URL https://doi.org/10.1137/040605461

[16] V. Druskin, L. Knizhnerman, Extended Krylov subspaces: approximation of the matrix square root and related functions, SIAM J. Matrix Anal. Appl. 19 (3) (1998) 755–771. doi:10.1137/S0895479895292400.
URL https://doi.org/10.1137/S0895479895292400

[17] B. Iannazzo, C. Manasse, A Schur logarithmic algorithm for fractional powers of matrices, SIAM J. Matrix Anal. Appl. 34 (2) (2013) 794–813. doi:10.1137/120877398.
URL https://doi.org/10.1137/120877398

[18] I. Moret, Rational Lanczos approximations to the matrix square root and related functions, Numer. Linear Algebra Appl. 16 (6) (2009) 431–445. doi:10.1002/nla.625.
URL https://doi.org/10.1002/nla.625

[19] M. Benzi, D. Bertaccini, Approximate inverse preconditioning for shifted linear systems, BIT 43 (2) (2003) 231–244. doi:10.1023/A:1026089811044.
URL https://doi.org/10.1023/A:1026089811044

[20] D. Bertaccini, Efficient preconditioning for sequences of parametric complex symmetric linear systems, Electron. Trans. Numer. Anal. 18 (2004) 49–64.
URL http://www.emis.de/journals/ETNA/vol.18.2004/pp49-64.dir/pp49-64.pdf

[21] S. Bellavia, D. Bertaccini, B. Morini, Nonsymmetric preconditioner updates in Newton-Krylov methods for nonlinear systems, SIAM J. Sci. Comput. 33 (5) (2011) 2595–2619. doi:10.1137/100789786.
URL https://doi.org/10.1137/100789786

[22] D. Bertaccini, F. Durastante, Interpolating preconditioners for the solution of sequence of linear systems, Comput. Math. Appl. 72 (4) (2016) 1118–1130. doi:10.1016/j.camwa.2016.06.023.
URL https://doi.org/10.1016/j.camwa.2016.06.023

[23] M. Berljafa, S. Güttel, The RKFIT algorithm for nonlinear rational approximation, SIAM J. Sci. Comput. 39 (5) (2017) A2049–A2071. doi:10.1137/15M1025426.
URL https://doi.org/10.1137/15M1025426

[24] K. Driver, K. Jordaan, N. Mbuyi, Interlacing of the zeros of Jacobi polynomials with different parameters, Numer. Algorithms 49 (1-4) (2008) 143–152. doi:10.1007/s11075-008-9162-2.
URL https://doi.org/10.1007/s11075-008-9162-2

[25] P. Brändén, On operators on polynomials preserving real-rootedness and the Neggers-Stanley conjecture, J. Algebraic Combin. 20 (2) (2004) 119–130. doi:10.1023/B:JACO.0000047295.93525.df.
URL https://doi.org/10.1023/B:JACO.0000047295.93525.df

[26] Jacobi Polynomials.
URL https://people.sc.fsu.edu/~jburkardt/m_src/jacobi_polynomial/jacobi_polynomial.html

[27] D. A. Bini, L. Robol, Solving secular and polynomial equations: a multiprecision algorithm, J. Comput. Appl. Math. 272 (2014) 276–292. doi:10.1016/j.cam.2013.04.037.
URL https://doi.org/10.1016/j.cam.2013.04.037

[28] D. A. Bini, G. Fiorentino, Design, analysis, and implementation of a multiprecision polynomial rootfinder, Numer. Algorithms 23 (2-3) (2000) 127–173. doi:10.1023/A:1019199917103.
URL https://doi.org/10.1023/A:1019199917103

[29] R. E. Ewing, T. Lin, Y. Lin, On the accuracy of the finite volume element method based on piecewise linear polynomials, SIAM J. Numer. Anal. 39 (6) (2002) 1865–1888. doi:10.1137/S0036142900368873.
URL https://doi.org/10.1137/S0036142900368873

[30] A. Ruhe, Rational Krylov: a practical algorithm for large sparse nonsymmetric matrix pencils, SIAM J. Sci. Comput. 19 (5) (1998) 1535–1551. doi:10.1137/S1064827595285597.
URL https://doi.org/10.1137/S1064827595285597

[31] A. Ruhe, The rational Krylov algorithm for nonsymmetric eigenvalue problems. III. Complex shifts for real matrices, BIT 34 (1) (1994) 165–176. doi:10.1007/BF01935024.
URL https://doi.org/10.1007/BF01935024

[32] J. Lambert, Numerical Methods for Ordinary Differential Systems: The Initial Value Problem, Wiley, New York, 1991.

[33] U. M. Ascher, S. J. Ruuth, B. T. R. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM J. Numer. Anal. 32 (3) (1995) 797–823. doi:10.1137/0732037.
URL https://doi.org/10.1137/0732037

[34] M. Crouzeix, Une méthode multipas implicite-explicite pour l'approximation des équations d'évolution paraboliques, Numer. Math. 35 (3) (1980) 257–276. doi:10.1007/BF01396412.
URL https://doi.org/10.1007/BF01396412

[35] M. S. Alnæs, J. Blechta, J. Hake, A. Johansson, B. Kehlet, A. Logg, C. Richardson, J. Ring, M. E. Rognes, G. N. Wells, The FEniCS Project Version 1.5, Archive of Numerical Software 3 (100). doi:10.11588/ans.2015.100.20553.

[36] J. E. Guyer, D. Wheeler, J. A. Warren, FiPy: Partial Differential Equations with Python, Computing in Science Engineering 11 (3) (2009) 6–15. doi:10.1109/MCSE.2009.52.

[37] C. Lubich, Discretized fractional calculus, SIAM J. Math. Anal. 17 (3) (1986) 704–719. doi:10.1137/0517050.
URL https://doi.org/10.1137/0517050

[38] F. Gibou, R. P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (1) (2002) 205–227. doi:10.1006/jcph.2001.6977.
URL https://doi.org/10.1006/jcph.2001.6977

[39] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains—the Voronoi interface method, J. Comput. Phys. 298 (2015) 747–765. doi:10.1016/j.jcp.2015.06.026.
URL https://doi.org/10.1016/j.jcp.2015.06.026

[40] J. Papac, A. Helgadottir, C. Ratsch, F. Gibou, A level set approach for diffusion and Stefan-type problems with Robin boundary conditions on quadtree/octree adaptive Cartesian grids, J. Comput. Phys. 233 (2013) 241–261. doi:10.1016/j.jcp.2012.08.038.
URL https://doi.org/10.1016/j.jcp.2012.08.038