Efficient Inversion of Matrix ϕ-Functions of Low Order

Luca Gemignani[a,1]

[a] *Dipartimento di Informatica, Largo B. Pontecorvo, 3, Pisa, 56127, Italy*

Abstract

The paper is concerned with efficient numerical methods for solving a linear system $\phi(A)\boldsymbol{x} = \boldsymbol{b}$, where $\phi(z)$ is a ϕ-function and $A \in \mathbb{R}^{N\times N}$. In particular in this work we are interested in the computation of $\phi(A)^{-1}\boldsymbol{b}$ for the case where $\phi(z) = \phi_1(z) = \dfrac{e^z - 1}{z}$ and $\phi(z) = \phi_2(z) = \dfrac{e^z - 1 - z}{z^2}$. Under suitable conditions on the spectrum of A we design fast algorithms for computing both $\phi_\ell(A)^{-1}$ and $\phi_\ell(A)^{-1}\boldsymbol{b}$ based on Newton's iteration and Krylov-type methods, respectively. Adaptations of these schemes for structured matrices are considered. In particular the cases of banded and more generally quasiseparable matrices are investigated. Numerical results are presented to show the effectiveness of our proposed algorithms.

Keywords: Matrix Inversion, Newton Iteration, Krylov Methods, Rank Structure, Matrix Function
2000 MSC: 65F05, 65F60

1. Introduction

Efficient numerical methods for computing the action of matrix ϕ-functions are of growing interest for the application of exponential integrators in the solution of stiff systems of differential equations (compare [24, 17, 18, 25, 14, 7] and the references given therein). The computation of the inverse of matrix ϕ-functions or, equivalently, the design of fast linear solvers for matrix ϕ-functions is useful in the solution of related inverse problems.

A fast efficient numerical method for computing $\psi_1(A)$ and $\psi_1(A)\boldsymbol{b}$ with $\psi_1(z) = 1/\phi_1(z)$, $\phi_1(z) = \dfrac{e^z - 1}{z}$, $A \in \mathbb{R}^{N\times N}$, has been presented in [5, 6]. The method exploits a partial fraction decomposition of the meromorphic function $\psi_1(z)$ and it is particularly suited for the application to structured matrices for which fast linear solvers exist. The same approach cannot be extended to other

Email address: luca.gemignani@unipi.it (Luca Gemignani)

[1] The author is partially supported by INDAM/GNCS and by the project PRA_2020_61 of the University of Pisa.

functions $\psi_\ell(z) = 1/\phi_\ell(z)$ with $\ell > 1$ due to the lack of explicit closed–form expressions of their poles.

The computation of $\psi_2(A)$ and $\psi_2(A)\boldsymbol{b}$ with $\psi_2(z) = 1/\phi_2(z)$, $\phi_2(z) = \dfrac{e^z - 1 - z}{z^2}$, $A \in \mathbb{R}^{N\times N}$, is also of relevant interest. We describe hereafter two applications.

1. `A nonlocal inverse problem.` Consider the nonlocal inverse problem defined as follows: We seek the vector $\boldsymbol{g} \in \mathbb{R}^N$ and the function $\boldsymbol{u} = \boldsymbol{u}(t)\colon [0,T] \to \mathbb{R}^N$ such that

$$\boldsymbol{u}'(t) = A\boldsymbol{u}(t) + \boldsymbol{g}, \quad \boldsymbol{u}(0) = \boldsymbol{u}_0, \quad \frac{1}{T}\int_0^T \boldsymbol{u}(t)dt = \boldsymbol{u}_1$$

with $A \in \mathbb{R}^{N\times N}$, $\boldsymbol{u}_0, \boldsymbol{u}_1 \in \mathbb{R}^N$. By imposing the integral condition we find that the vector $\boldsymbol{g}$ solves

$$\boldsymbol{u}_1 = \phi_1(TA)\boldsymbol{u}_0 + T\phi_2(TA)\boldsymbol{g}. \tag{1}$$

2. `Two-point inverse problems.` The computation of the unknown parameter $\boldsymbol{p}$ in the local boundary value problem [30, 35]

$$\boldsymbol{u}'(t) = A\boldsymbol{u}(t) + t\boldsymbol{p}, \quad \boldsymbol{u}(0) = \boldsymbol{q}, \quad \boldsymbol{u}(1) = \boldsymbol{g}, \tag{2}$$

$A \in \mathbb{R}^{N\times N}$, $t \in [0,1]$, amounts to solve a linear system with a matrix ϕ-function as coefficient matrix. Indeed, it is shown that

$$\boldsymbol{u}(1) = \boldsymbol{g} = e^A\boldsymbol{q} + \phi_2(A)\boldsymbol{p}. \tag{3}$$

Abstract formulations of 1 and 2 can also be considered [20, 36], where A is a differential operator which has to be discretized numerically. In this contribution we address the computation of $\psi_{\ell+1}(A)$ and $\psi_{\ell+1}(A)\boldsymbol{b}$, with $\psi_\ell(z) = 1/\phi_\ell(z)$ and $\ell > 0$, when A is a large sparse and/or structured matrix. Our extension relies on the Newton iteration for computing the inverse of a matrix. For review of this method see [27, 28]. This tool has already been successfully applied in [4, 29] for the inversion of matrices having a displacement rank structure.

More specifically, we first identify regions $\Omega \subset \mathbb{C}$ of the complex plane such that $\left|1 - \dfrac{\phi_{\ell+1}(z)}{\phi_\ell(z)}\right| < 1$, $\ell \geq 1$, for all $z \in \Omega$. Then, we show that if the eigenvalues of A lie in Ω, the Newton iteration applied for the inversion of $B = \phi_{\ell+1}(A)$, $\ell \geq 1$, with starting point $X_0 = \psi_\ell(A) = (\phi_\ell(A))^{-1}$ is quadratically converging to the inverse matrix of B. This means that given a method to compute $X_0 = \psi_\ell(A)$ we can apply the Newton iteration for approximating $B_0^{-1} = \psi_{\ell+1}(A)$. Moreover, since the iterative scheme only requires matrix multiplications it is amendable for fast implementations using structured representations of the matrices involved. In particular, fast adaptations for both displacement structured and quasiseparable matrices can be devised. Approximate compression techniques in the style of [4] can also be incorporated to take under control the growth of displacement or quasiseparable ranks.

The convergence results for the Newton iteration can also be exploited in a different perspective. It is easily seen that the intermediate approximations of $\psi_{\ell+1}(A)$ generated by Newton's iteration can be expressed as a polynomial of $X_0 B = \psi_\ell(A)\phi_{\ell+1}(A)$ thus providing the link to the development of Krylov-type methods for computing $\psi_{\ell+1}(A)\boldsymbol{b}$. In particular, the solution of the linear system $\phi_{\ell+1}(A)\boldsymbol{x} = \boldsymbol{b}$ can be approximated efficiently by means of a Krylov-type method like GMRES applied for solving the equivalent system $\psi_\ell(A)\phi_{\ell+1}(A)\boldsymbol{x} = \psi_\ell(A)\boldsymbol{b}$. When the eigenvalues of A lie in Ω then the convergence of GMRES applied to this system follows from the results in [31] (see Proposition 6.32 and its generalizations). The paramount advantage of such a Krylov-based approach is that matrix-by-vector multiplications are only required to find an approximation $X_k\boldsymbol{b}$ of the vector $\boldsymbol{x} = \psi_{\ell+1}(A)\boldsymbol{b}$. In particular, for $\ell = 1$ the projection method only involves products of the form $\psi_1(A)\boldsymbol{v}$ which can be computed using the methods introduced in [5, 6].

In principle, the proposed schemes can be applied recursively for evaluating $\psi_{\ell+1}(A)$ or $\psi_{\ell+1}(A)\boldsymbol{b}$, $\ell > 0$, provided that a method for computing $\psi_1(A)$ or $\psi_1(A)\boldsymbol{b}$ is available. Despite the generality of the approach, however, based on application and numerical issues in this work we focus on the case $\ell \in \{0, 1\}$, or at least ℓ small in value.

The paper is organized as follows. In Section 2 we analyze theoretical and computational properties of Newton's iteration for the inversion of matrix ϕ-functions. In Section 3 we devise a Krylov-type method for computing the action of these inverses on a vector. In Section 4 we present the results of numerical experiments illustrating the properties of this method whereas conclusions and future work are drawn in Section 5.

2. Newton Iteration for the Inversion of Matrix ϕ-Functions

In this section we design an iterative method based on Newton's iteration for the inversion of matrix ϕ-functions $\phi_\ell(A)$, $A \in \mathbb{C}^{N\times N}$, $\ell > 1$.

The ϕ-functions are entire functions defined for scalar arguments by the integral representation

$$\phi_0(z) = e^z, \quad \phi_\ell(z) = \frac{1}{(\ell-1)!}\int_0^1 e^{(1-\theta)z}\theta^{\ell-1}d\theta, \quad \ell \geq 1,\ z \in \mathbb{C}. \tag{4}$$

The ϕ-functions satisfy the recurrence relation

$$\phi_\ell(z) = z\phi_{\ell+1}(z) + \frac{1}{\ell!}, \quad \ell \geq 0, \tag{5}$$

and have the Taylor expansion

$$\phi_\ell(z) = \sum_{k=0}^{\infty} \frac{z^k}{(k+\ell)!}, \quad \ell \geq 0.$$

This latter can be extended to a matrix argument by setting for any $A \in \mathbb{C}^{N\times N}$

$$\phi_\ell(A) = \sum_{k=0}^{\infty} \frac{A^k}{(k+\ell)!}, \quad \ell \geq 0.$$

The function $\psi_\ell(z)$, $\ell \geq 0$, is a meromorphic function defined as the reciprocal of $\phi_\ell(z)$, that is,

$$\psi_\ell(z) = \phi_\ell(z)^{-1}, \quad \ell \geq 0.$$

Explicit series expansions are only known for $\psi_1(z)$. It holds [[1], formula 23.1.1]

$$\psi_1(z) = \phi_1(z)^{-1} = \sum_{k=0}^{+\infty} \frac{B_k}{k!} z^k, \quad |z| < 2\pi, \tag{6}$$

where B_k denotes the kth Bernoulli number. A different rational representation is derived in [6]. For any fixed $n > 0$ we have

$$\psi_1(z) = f_n(z) + 2(-1)^n \sum_{k=1}^{\infty} \left(\frac{z}{2\pi}\right)^{2(n+1)} \frac{1}{k^{2n}} (\left(\frac{z}{2\pi}\right)^2 + k^2)^{-1}, \tag{7}$$

where

$$f_n(z) = 1 - \frac{1}{2}z + \sum_{i=0}^{n-1} z^{2(i+1)} \frac{B_{2(i+1)}}{(2(i+1))!}.$$

The series on the rhs of (7) converges uniformly to $\psi_1(z)$ over any compact set $\mathcal{K} \subset \mathbb{C} \setminus \pm 2\pi i\mathbb{N}$. The polynomial contribution $f_n(z)$ is a partial sum of the power series expansion (6) aimed to improve the accuracy of the approximation around the removable singularity at the origin in the complex plane. Relation (7) provides a family of mixed polynomial/rational approximations of $\psi_1(A)$ of the form

$$\psi_1(A) \simeq r_{n,m}(A) = f_n(A) + 2(-1)^n \left(\frac{A}{2\pi}\right)^{2(n+1)} \sum_{k=1}^{m} \frac{1}{k^{2n}} (\left(\frac{A}{2\pi}\right)^2 + k^2)^{-1}. \tag{8}$$

Based on the computation of $\psi_1(A)$, under suitable assumption on the spectrum of A the Newton method for matrix inversion provides an effective tool for approximating $\psi_\ell(A)$ for $\ell > 1$. Newton's iteration [27, 28] for the inversion of a nonsingular matrix $B \in \mathbb{C}^{N\times N}$ is defined by :

$$X_0 \in \mathbb{C}^{N\times N}, \quad X_{k+1} = 2X_k - X_k B X_k, \ k \geq 0. \tag{9}$$

From

$$I - X_{k+1}B = (I - X_k B)^2 = (I - X_0 B)^{2^k},$$

we obtain that Newton's iteration (9) quadratically converges to B^{-1} provided that all eigenvalues of $R = I - X_0 B$ have modulus less than 1.

Let us first suppose that $A \in \mathbb{C}^{N\times N}$ has real eigenvalues only, that is, $\lambda \in \texttt{spec}(A) \Rightarrow \lambda \in \Omega = \mathbb{R}$. Observe that for real arguments ($z \in \mathbb{R}$) from

the integral representation (4) it follows that $\phi_\ell(z) > 0$ and, moreover, $\phi_\ell(z) > \phi_{\ell+1}(z)$ $\forall z \in \mathbb{R}$, $\ell \geq 1$. This means that

$$0 < 1 - \frac{\phi_{\ell+1}(\lambda_i)}{\phi_\ell(\lambda_i)} = 1 - \phi_{\ell+1}(\lambda_i)\psi_\ell(\lambda_i) < 1, \quad \forall \lambda_i \in \mathtt{spec}(A).$$

The next result immediately follows.

Proposition 1. *Let $A \in \mathbb{C}^{N\times N}$ be a matrix with all real eigenvalues. Then for any $\ell \geq 0$, $\phi_\ell(A)$ is invertible. Moreover, the Newton iteration (9) applied for the inversion of $B = \phi_{\ell+1}(A)$, $\ell \geq 1$, with starting point $X_0 = \psi_\ell(A) = (\phi_\ell(A))^{-1}$ is quadratically converging to the inverse matrix of B.*

The extension of this result for matrices with possibly complex eigenvalues requires some additional constraints. If $z = a + ib$, $a, b \in \mathbb{R}$, $i^2 = -1$, is a complex number then from the integral representation (4) it is found that for $\ell \geq 1$

$$\phi_\ell(z) = \frac{\displaystyle\int_0^1 e^{\tau a}\cos(\tau b)(1-\tau)^{\ell-1}d\tau + i\int_0^1 e^{\tau a}\sin(\tau b)(1-\tau)^{\ell-1}d\tau}{(\ell-1)!}.$$

Under the auxiliary assumption $b \in [-\pi/2, \pi/2]$ this implies that $\Re(\phi_\ell(z)) > 0$ and, hence, $\phi_\ell(z) \neq 0$. In addition, the residual $r(z) = \phi_\ell(z) - \phi_{\ell+1}(z)$ also satisfies

$$r(z) = \frac{\displaystyle\int_0^1 e^{\tau a}\cos(\tau b)\frac{\ell-1+\tau}{\ell}(1-\tau)^{\ell-1}d\tau + i\int_0^1 e^{\tau a}\sin(\tau b)\frac{\ell-1+\tau}{\ell}(1-\tau)^{\ell-1}d\tau}{(\ell-1)!}.$$

It follows that

$$|\Re(r(z))| < |\Re(\phi_\ell(z))|, \quad |\Im(r(z))| < |\Im(\phi_\ell(z))|$$

and therefore

$$\left|\frac{r(z)}{\phi_\ell(z)}\right| = \left|1 - \frac{\phi_{\ell+1}(z)}{\phi_\ell(z)}\right| < 1.$$

To sum up we arrive at the following extension of Proposition 1.

Proposition 2. *Let $A \in \mathbb{C}^{N\times N}$ be a matrix with all eigenvalues lying in the strip $\Omega = \mathbb{R} \times i[-\pi/2, \pi/2]$ in the complex plane. Then for any $\ell \geq 0$, $\phi_\ell(A)$ is invertible. Moreover, the Newton iteration (9) applied for the inversion of $B = \phi_{\ell+1}(A)$, $\ell \geq 1$, with starting point $X_0 = \psi_\ell(A) = (\phi_\ell(A))^{-1}$ is quadratically converging to the inverse matrix of B.*

Differently from the case of real spectrum some restrictions on the localization of the eigenvalues are needed for general matrices. Let us consider the tridiagonal Toeplitz matrix T of order N having subdiagonal, diagonal and superdiagonal entries given by 0.5, 0 and -0.5, respectively. The matrix has

h	1	N	N^2	N^4
$\rho(R)$	1.6852e+03	57.5590	0.5071	0.5000

Table 1: Values of the spectral radius $\rho(R)$ of $R = I - (\phi_1(A))^{-1}\phi_2(A)$ with $A = h^{-1}N^2T$, $N = 128$, and $T = \texttt{gallery}(\texttt{'tridiag'}, \texttt{N}, \texttt{0.5}, \texttt{0}, -\texttt{0.5})$ for different values of $h = 1, N, N^2, N^4$.

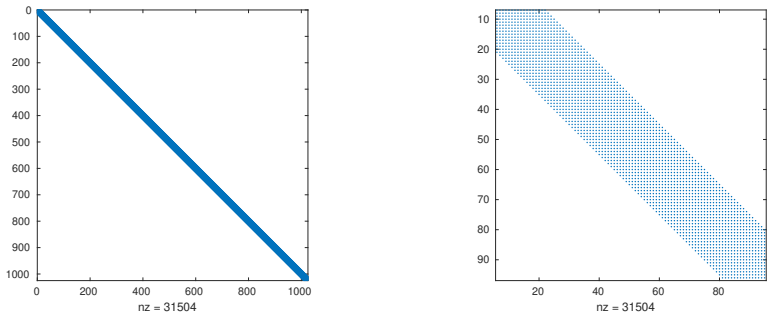
eigenvalues located on the imaginary axis in the interval $i[-1, 1]$. In Table 1, we report the computed spectral radius of $R = I - (\phi_1(A))^{-1}\phi_2(A)$, where $A = h^{-1}N^2T$ and $N = 128$, for different values of h.

Computational interest in Newton's method is especially due to the development of high-performance computing environments. The iterative scheme (9) basically require BLAS Level 3 routines which are easily implemented in parallel on a parallel computing system [26, 37]. Moreover, it is especially suited to take advantage of the sparsity and the structural properties of the matrices involved. The case of matrices having a displacement structure has been considered in [4, 28, 29]. In the next subsection, we focus on the application of Newton's method for inverting banded and more generally quasiseparable-type matrices arising from the discretization of partial differential equations. Recall that an $N \times N$ matrix A is quasiseparable of quasiseparability rank s if $\text{rank}(A(k+1:N, 1:k)) \leq s$ and $\text{rank}(A(1:k, k+1:N)) \leq s$ for $1 \leq k \leq N-1$. The class of quasiseparable matrices encompasses both banded matrices and their inverses.

2.1. Fast Adaptations for Structured Matrices

We begin by observing that for a given banded matrix A the matrix $B = \phi_\ell(A)$ or $B = \psi_\ell(A)$, $\ell \geq 1$, generally inherits the banded structure of A in some approximate way. For instance, in Figure 1 we illustrate the "spy" plots of $\psi_2(A)$ and its leading principal submatrix of order 95 when A is the 1D Laplacian matrix of order $N = 1024$. The threshold value is set to $1.0e-14$. The exact tridiagonal structure of A results in an approximate banded structure of B. Precise mathematical statements depend on quantities that are hard to compute and typically yield very pessimistic estimates (compare with [19] for the matrix exponential, the review [2] for more general analytic functions and [22] for some extensions to functions with singularities). In practice, suitable approximation/compression techniques are to be employed. Our preferred option is to look at the matrix B as a rank-structured matrix with the possibility to encode the structure by using numerical ranks.

Condensed representations for rank-structured matrices have been proposed in a variety of papers. Quasiseparable matrices are introduced in [10]. A complete review of their properties is presented in [11]. A quasiseparable representation of a matrix A is defined by two families of lower and upper generators that are computed by exploiting the low-rank properties of the submatrices of A located in its lower and upper triangular part, respectively. Given in input



(a) (b)

Figure 1: "Spy" plots of $\psi_2(A)$ and its leading principal submatrix of order 95 for A being the 1D Laplacian matrix of order $N = 1024$.

a quasiseparable representation of X_0 and B then the structured adaptation of (9) amounts to compute at each iteration one sum and two products of quasiseparable matrices possibly complemented with a compression/approximation technique used to take under control the growth of the quasiseparable generators. Generator-based algorithms to perform these operations are described in Chapter 4, 5 and 17 of [11].

A more flexible format for rank-structured matrices which is amenable for divide-and-conquer oriented techniques is called hierarchically semiseparable (HSS) representation [38]. This representation is found by combining recursive partitioning, compression of off-diagonal blocks and nestedeness for the generators of these blocks. In order to operate with HSS matrices efficiently, one exploits their representation with generators, demonstrated by the following 4×4 example:

$$A = \begin{bmatrix} D_1 & G_1H_2 & G_1R_1R_3Q_3 & G_1R_1R_4H_4 \\ P_2Q_1 & D_2 & P_2R_2R_3Q_3 & P_2R_2R_4H_4 \\ G_3L_1L_3Q_1 & G_3L_1L_4H_2 & D_3 & G_3H_4 \\ P_4L_2L_3Q_1 & P_4L_2L_4H_2 & P_4Q_3 & D_4 \end{bmatrix},$$

where D_i are square matrices of equal size. The representation is condensed if all the matrices L_i and R_i have sizes less than a small constant $k \ll N$. The generators G_i, P_i, H_i and Q_i are tall or skinny matrices. The value of k is related with the maximum rank of all off-diagonal blocks at all levels of the HSS recursive splitting of A [38]. Arithmetic operations between two matrices of order N expressed in a condensed HSS format can be performed in linear time w.r.t. N [38].

In view of the relation with the ranks of the off-diagonal blocks it is clear that any arithmetic operation (except inversion) performed on HSS matrices can increase their ranks. In particular, if C and B are $s-$quasiseparable matrices then $A = C \star B$, $\star \in \{\cdot, \pm\}$, is a quasiseparable matrix of quasiseparability rank at most $2s$. This makes possible to bound the ranks of the matrices generated

by the Newton iteration (9) in terms of the ranks of the input matrices X_0 and B. If these ranks are quite small the quadratic convergence right from the start of the Newton process is generally sufficient to control the growth of the generators. Otherwise, to use HSS structure efficiently under the iterative process we need some compression algorithm. A MatLab toolbox to carry out arithmetic operations among HSS matrices in exact or approximate compressed form is described in [23]. Using this package, for the sake of illustration we show in Figure 2 the rank properties of the matrices generated by Newton's iteration applied for the computation of $(\phi_2(A))^{-1}$ starting from $(\phi_1(A))^{-1}$ for a given rank-structured matrix A. Specifically, in our test we consider the matrix $A \in \mathbb{R}^{4096\times 4096}$ defined as follows:

$$A = \frac{1}{3}\begin{bmatrix} M & N & & \\ N & \ddots & \ddots & \\ & \ddots & \ddots & N \\ & & N & M \end{bmatrix} \tag{10}$$

with

$$M = \begin{bmatrix} -8 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -8 \end{bmatrix}, \quad N = \begin{bmatrix} 1 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 1 \end{bmatrix},$$

and M and N of size 64. The matrix is generated in the solution of 2D Laplace's equation with Dirichlet boundary conditions by Q1 finite elements [15]. In Figure 2 we show the numerical ranks of the off-diagonal blocks in the HSS representations of A, $(\phi_1(A))^{-1}$, $\phi_2(A)$ and the approximation X_7 of $(\phi_2(A))^{-1}$ generated by Newton's iteration applied for the inversion of $\phi_2(A)$ with starting point $X_0 = \psi_1(A) = (\phi_1(A))^{-1}$ stopped after 7 iterations with error $\| \phi_2(A)X_7 - I \|_2 \leq 2.3e-12$. The compression threshold value is set to $1.0e-12$.

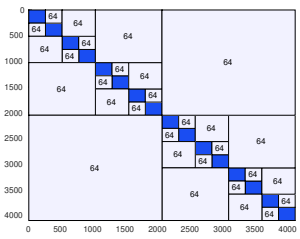
3. A Krylov-type Method for Computing the Action of ψ-Functions on a Vector

The above results indicate the possibility of approximating $B^{-1} = \psi_{\ell+1}(A)$, $\ell > 0$, using Newton's method with starting point $X_0 = \psi_\ell(A)$ provided that the eigenvalues of A are properly localized. Under the same assumption, we can apply some Krylov-type method like GMRES [31] for approximating $\psi_{\ell+1}(A)\boldsymbol{b}$ or, equivalently, for solving the linear system $\phi_{\ell+1}(A)\boldsymbol{x} = \boldsymbol{b}$. Indeed, from (9) we obtain that

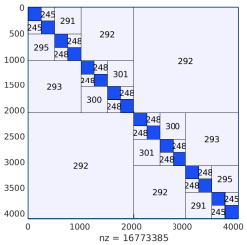
$$X_1 = 2X_0 - X_0BX_0 = (2I - X_0B)X_0 = p_1(X_0B)X_0,$$

with $p_1(z)$ a polynomial of degree 1. Inductively, we find that for $k > 0$

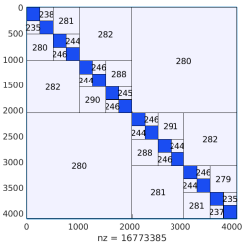
$$X_k = 2p_{k-1}(X_0B)X_0 - p_{k-1}(X_0B)X_0Bp_{k-1}(X_0B)X_0 = p_k(X_0B)X_0, \tag{11}$$



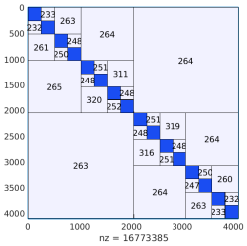
(a) Ranks of A



(b) Ranks of $X_0 = (\phi_1(A))^{-1}$



(c) Ranks of $\phi_2(A)$



(d) Ranks of $X_7 \simeq (\phi_2(A))^{-1}$

Figure 2: Illustration of the rank properties of HSS representations of the matrices involved in the computation of $(\phi_2(A))^{-1}$ by Newton's iteration for the matrix A given in (10). The compression threshold value is set to $1.0e-12$.

for a suitable polynomial $p_k(z)$ of degree $2^k - 1$. This means that the approximation $\boldsymbol{x}_k = X_k\boldsymbol{b}$ of the solution $\boldsymbol{x}$ of $B\boldsymbol{x} = \boldsymbol{b}$ satisfies

$$\boldsymbol{x}_k = p_k(X_0B)X_0\boldsymbol{b}, \quad k > 0.$$

Hence $\boldsymbol{x}_k$ belongs to the $m - th$ Krylov subspace, $m = 2^k$,

$$\mathcal{K}_m = \text{span}\{X_0\boldsymbol{b}, (X_0B)X_0\boldsymbol{b}, \ldots, (X_0B)^{m-1}X_0\boldsymbol{b}\}.$$

It follows that a Krylov-type method might be used for solving the equivalent system

$$X_0B\boldsymbol{x} = X_0\boldsymbol{b}. \tag{12}$$

For instance, GMRES [31] after m iterations returns an approximation $\boldsymbol{v}_k$ such that

$$\| X_0\boldsymbol{b} - X_0B\boldsymbol{v}_k \|_2 \leq \| X_0\boldsymbol{b} - X_0B\boldsymbol{x}_k \|_2 \leq \| X_0 \|_2 \| \boldsymbol{b} - B\boldsymbol{x}_k \|_2 .$$

In this case X_0 can play the role of a preconditioner suitably determined to ensure the convergence of the projection method. The next result immediately follows by combining Proposition 6.32 in [31] with the results of the previous section.

Proposition 3. *Let $A \in \mathbb{C}^{N\times N}$ be a diagonalizable matrix, i.e., $A = SDS^{-1}$, $D = \mathrm{diag}\,[\lambda_1, \ldots, \lambda_n]$, with all eigenvalues λ_i, $1 \le i \le N$, lying in the strip $\Omega = \mathbb{R} \times i[-\pi/2, \pi/2]$ in the complex plane. Let $\boldsymbol{x}_m$, $m \ge 0$, be the approximate solution of (12), with $B = \phi_{\ell+1}(A)$ and $X_0 = \psi_\ell(A)$, obtained from the m-th step of the GMRES algorithm, and let $\boldsymbol{r}_m = X_0\boldsymbol{b} - X_0B\boldsymbol{x}_m$. Then, we have*

$$\| \boldsymbol{r}_m \|_2 \le \kappa_2(S)\rho(R)^m \| \boldsymbol{r}_0 \|_2, \quad R = I - X_0B, \quad m \ge 0,$$

where $\rho(R) < 1$ denotes the spectral radius of R and $\kappa_2(S)$ is the 2-norm condition number of S.

The convergence estimates for GMRES can be extended to general matrices by replacing the spectral decomposition of A with its Jordan canonical form [32]. The bound in Proposition 3 depends on the 2-norm condition number of the eigenvector matrix S. This is satisfactory in the normal case but if A is far from normal, then $\kappa_2(S)$ may have large magnitude and this utterly invalidates the bound. Alternative GMRES convergence bounds based on the numerical range or the pseudospectra of the coefficient matrix have been proposed in the literature (see [12] and the references given therein). However, these bounds are not easy to compute and they present other drawbacks so that a common approach is to mitigate the impact of $\kappa_2(S)$ by assuming that the possible ill-conditioning is due only to a low-dimensional invariant subspace which contribution can be deflated in same way [12, 33].

The computational cost for the GMRES algorithm is dominated by the cost of matrix-vector multiplications with the matrix X_0B. It is worth noticing that from relation (5)

$$X_0B = \psi_\ell(A)\phi_{\ell+1}(A) = A^{-1}(I - \psi_\ell(A)/\ell!), \tag{13}$$

which implies that the multiplication of X_0B by a vector reduces to first multiply $\psi_\ell(A)$ by the same vector, and then solve a linear system with coefficient matrix A. This is particularly interesting for $\ell = 1$ since an efficient algorithm to evaluate $\psi_1(A)\boldsymbol{b}$ has been proposed in [5, 6]. The algorithm relies upon the family of polynomial/rational expansions of $\psi_1(z)$ given in (7). Based on (13), complementing the GMRES iterative solver with the approximation (7) provides an effective method for computing the action of $\psi_2(A)$ on a vector. A basic MatLab skeleton implementation is as follows:

Algorithm 1: Given in input the matrix $A \in \mathbb{C}^{N\times N}$ and the vector $\boldsymbol{b} \in \mathbb{C}^N$, this algorithm approximates the vector $\boldsymbol{w} = \psi_2(A)\boldsymbol{b}$

1: **Define** `funmv` $= @(\boldsymbol{z})A\backslash(I_N - r_{n,m}(A))\boldsymbol{z}$, where $r_{n,m}(A)$ is given in (8) ;
2: **Call** $\boldsymbol{w} =$ `gmres`$(@$`funmv`$, r_{n,m}(A)\boldsymbol{b}, tol, maxit)$;
3: **return** $\boldsymbol{w}$

In view of (13), Algorithm 1 can be the building block of a recursive procedure for the evaluation of $\psi_\ell(A)\boldsymbol{b}$, $\ell > 1$. Some explanations are, however,

tol	$1.0e-7$	$1.0e-9$	$1.0e-11$	$1.0e-13$
m	71	152	325	698
err	$1.8e-6$	$1.4e-8$	$8.8e-10$	$2.0e-11$

Table 2: Convergence and error estimates for the adaptive computation of $\boldsymbol{w} = \psi_1(A)\boldsymbol{z}$, with $A = \texttt{gallery}('hanowa', 128)$ and $\boldsymbol{z} = \texttt{ones}(128, 1)$, using the incremental approximation (14).

in order with respect to the execution of the first two steps. Specifically, the crucial point is the selection of the approximation $r_{n,m}(A)$ of $\psi_1(A)$.

For a general A, the choice of the approximation can be performed adaptively by means of an incremental scheme where we add one term at a time until a fixed tolerance is reached. Specifically, for a given value of n and tol we compute the approximation

$$\boldsymbol{w}_{m+1} = r_{n,m+1}(A)\boldsymbol{z} = \boldsymbol{w}_m + \frac{1}{(m+1)^{2n}} \left(\frac{A}{2\pi}\right)^{2(n+1)} \left(\left(\frac{A}{2\pi}\right)^2 + (m+1)^2 I_N\right)^{-1} \boldsymbol{z} \tag{14}$$

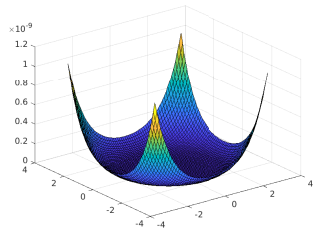
of $\boldsymbol{w} = \psi_1(A)\boldsymbol{z}$ until the stopping criterion $\| \boldsymbol{w}_{m+1} - \boldsymbol{w}_m \|_2 / \| \boldsymbol{w}_m \|_2 \leq tol$ is satisfied. The value of n should be chosen very small. In our experiments we use $n = 2$. In Table 2 for $A = \texttt{gallery}('hanowa', 128)$ we show the value of m and the corresponding error $err = \| \boldsymbol{w}_{m+1} - \boldsymbol{w} \|_2 / \| \boldsymbol{w} \|_2$ generated with different tolerances. Here $\boldsymbol{z} = \texttt{ones}(128, 1)$ and the vector $\boldsymbol{w} = \psi_1(A)\boldsymbol{z}$ is computed by using `expm` and the backslash operator.

Differently, for a normal matrix A the selection of n and m in (8) might be addressed a priori by looking at the corresponding approximation problem depending on the localization of the spectrum of A. In Figure 3 we show the plot of the absolute error $|\psi_1(z) - r_{n,m}(z)|$ for different values of n and m and for different domains.

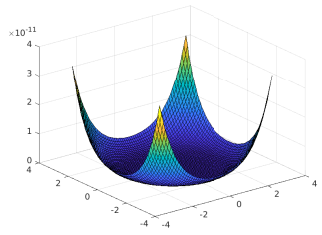
Recall that $\psi_1(z)$ is a meromorphic function with poles $\pm 2\pi i k$, $k \in \mathbb{N}$, and a removable singularity at the origin. It is remarkable that the approximation is quite accurate even close around the singular points, whereas the error increases appreciably with the size of the domain. A widespread approach to the computation of exponential and $\phi_\ell-$functions combines polynomial or Padé approximation with a few steps of scaling-and-squaring [16]. In principle, scaling-and-squaring may also be applied to our mixed polynomial-rational approximation, scaling the function argument by a suitable power of 2 and then making use of the squaring formulas

$$\psi_1(2z) = \frac{2\psi_1(z)}{e^z + 1} = \frac{2\psi_1(z)^2}{z + 2\psi_1(z)}. \tag{15}$$

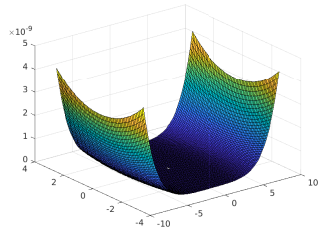
The efficient implementation of (15) is an ongoing research project and this scaling-and-squaring scheme is not considered here.



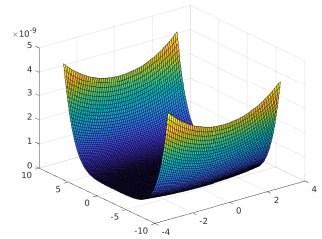
(a) $n = 2$ and $m = 32$ over $[-3, 3] \times i[-3, 3]$



(b) $n = 2$ and $m = 64$ over $[-3, 3] \times i[-3, 3]$



(c) $n = 2$ and $m = 64$ over $[-9, 9] \times i[-3, 3]$



(d) $n = 2$ and $m = 64$ over $[-3, 3] \times i[-9, 9]$

Figure 3: Surf plots of the absolute errors $|\psi_1(z) - r_{n,m}(z)|$ generated for different values of of n and m over different domains.

The evaluation of the function `funmv` at step 1 of Algorithm 1 basically amounts to compute $r_{n,m}(A)\boldsymbol{z}$. In the typical situation where $n \ll m$ this computations reduces to solve m shifted linear systems of the form

$$(A^2 + (2\pi k)^2 I_N)\boldsymbol{z}_i = (A + 2\pi i k I_N)(A - 2\pi i k I_N)\boldsymbol{z}_i = \boldsymbol{b}, \quad 1 \le i \le m. \quad (16)$$

There is an extensive literature on the solution of shifted linear systems. In the case of interest where A is quasiseparable we make use of the backward stable algorithm proposed in [5]. This algorithm saves about half of computations in the solution of the shifted linear systems by reusing pieces of the structured QR factorization of the matrix A. According to [5] for a quasiseparable matrix A of size N partitioned in blocks of size n that are represented via quasiseparable generators of length $r \ll m$ the arithmetic cost of solving the systems (16) is of the order $4n^2mN$.

Some numerical tests showing the effectiveness of Algorithm 1 are presented in Section 4.

4. Numerical Results

We have tested the application of Algorithm 1 for computing $\boldsymbol{w} = \psi_2(A)\boldsymbol{b}$ numerically by using MatLab. We do not exploit the use of restarting techniques

and specific selections of the initial guess in the `gmres` function by accepting the default values.

Numerical experiments have been carried out for comparison with the classical approach based on the Arnoldi method [9, 21, 13], where $\boldsymbol{w}$ is approximated by $\boldsymbol{w}_j = W_j \psi_2(H_j)\boldsymbol{e}_1 \parallel \boldsymbol{b} \parallel_2$, $j \geq 1$, and W_j and H_j are generated in the Arnoldi process. This scheme seems to be performing very efficiently whenever it works. The crux is that the performance is depending on a number of issues such as the choice of the starting vector and the stopping criterion in the Arnoldi process as well as the properties of the spectrum of the matrices H_j and the quality of the polynomial approximation of $\psi_2(z)$ on this spectrum. These issues can be difficult to tackle and resolve for a certain class of matrices. In order to illustrate these difficulties, let us consider the following examples.

1. In the first test we analyze some classical models of quasiseparable matrices generated by a rank-one perturbation of unitary and banded matrices. The matrices of size $N = 128$ are:

   (a) $A = Z + \epsilon \boldsymbol{e}\boldsymbol{e}^T$, $Z = \begin{bmatrix} & 1 & & \\ & & \ddots & \\ & & & 1 \\ 1 & & & \end{bmatrix}$, $\boldsymbol{e}^T = [1, \ldots, 1]$;

   (b) $A = B - \boldsymbol{e}\boldsymbol{e}_N^T$, $B = i \begin{bmatrix} 1 & & & \\ 1 & 2 & & \\ & \ddots & \ddots & \\ & & 1 & N \end{bmatrix}$;

   (c) $A = \mathrm{diag}(\xi, -\xi) \otimes I + \epsilon \boldsymbol{e}\boldsymbol{e}^T$, where $\xi =$ `lambertw`$(-2, -$`exp`$(-1))$, `lambertw` computes the Lambert W function [8] and ξ is the first pole of $\psi_2(z)$ [20].

   We have implemented the Arnoldi-based method in MatLab. As a stopping criterion we evaluate the relative error between two consecutive approximations $err_1^{(j)} = \parallel \boldsymbol{w}_{j+1} - \boldsymbol{w}_j \parallel_2 / \parallel \boldsymbol{w}_j \parallel_2$. As a measure of accuracy we also determine the relative error $err_2^{(j)} = \parallel \boldsymbol{w} - \boldsymbol{w}_j \parallel_2 / \parallel \boldsymbol{w} \parallel_2$, where $\boldsymbol{w}$ is computed in some way (varying with the considered example) to achieve a greater accuracy.

   The matrix in (1a) is well conditioned with eigenvalues far from the poles and the removable singularity of $\psi_2(z)$ and thus $\boldsymbol{w}$ is found using the direct approach based on `expm` and the backslash operator applied to $\phi_2(A)\boldsymbol{w} = \boldsymbol{b}$ with $\boldsymbol{b} = \boldsymbol{e}_1$. For $\epsilon = 0$ the matrices H_j, $1 \leq j \leq N-1$, generated by the Arnoldi scheme are lower bidiagonal matrices with zero diagonal entries. Hence, the direct approach do not work for computing the intermediate approximations $\boldsymbol{w}_j$. A method for dealing with singular matrices was proposed in [34]. This method determines $\phi_2(G)$ for a possibly singular G of size m at the cost of approximating the exponential of a matrix of size $3m$. The modified Arnoldi scheme complemented with this method and the backslash operator for computing $\boldsymbol{w}_j$ succeeds to find an approximation of $\boldsymbol{w}$ with relative error $err_2^{(j)}$ of order $1.0e-12$ in 15 iterations. For

(n,m)	$(3,8)$	$(3,16)$	$(3,32)$
$it_{\texttt{gmres}}$	17	17	17
$r_{\texttt{gmres}}$	$4.8e-14$	$4.8e-14$	$4.8e-14$
err_2	$8.7e-14$	$5.3e-14$	$5.3e-14$

Table 3: Performance of Algorithm 1 applied to the matrix in (1a) of size $N = 128$ with $\epsilon = 1.0e-14$ and the tolerance of `gmres` set at $tol = 1.0e-12$. .

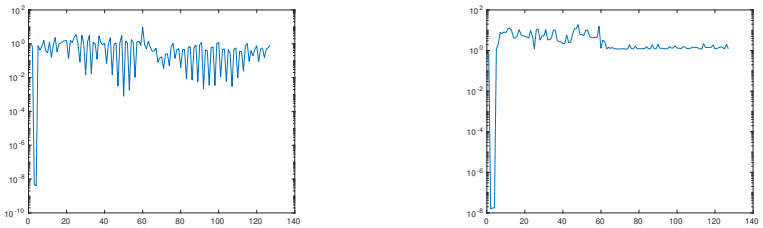
k	1	2	3	4	5	6
res	0.3	0.1	0.01	$1.7e-4$	$2.9e-8$	$8.9e-16$

Table 4: Convergence history of Newton's iteration applied for the inversion of $\phi_2(A)$, where A is the matrix in (1a) of size $N = 128$ with $\epsilon = 1.0e-14$.

a small nonzero ϵ the construction is numerically stable but it generates submatrices H_j that are severely ill-conditioned due to the occurrence of eigenvalues clustered around the origin. However, the modified Arnoldi scheme is still effective by reaching an approximation of $\boldsymbol{w}$ with relative error $err_2^{(j)}$ of order $1.0e-12$ in 18 iterations for $\epsilon = 1.0e-14$. For comparison in Table 3 we describe the results obtained by Algorithm 1 applied to the matrix in (1a). We show the value of n and m in the rational approximant, the number $it_{\texttt{gmres}}$ of iterations of `gmres`, the relative residual $r_{\texttt{gmres}}$ of the approximation returned by `gmres` together with the relative error err_2. Since the mixed polynomial-rational approximation performs quite well on this example we have also tested the performance of Newton's iteration described in Section 2. The direct and the modified algorithm can equivalently be used for computing $\phi_2(A)$. An initial guess is found by setting $X_0 = r_{3,8}(A)$. In Table 4 we illustrate the convergence history of the Newton scheme by showing the residual $res = \| \phi_2(A)X_k - I_N \|_2$.

The rapid convergence of the modified Arnoldi scheme on example (1a) can be explained in terms of its approximation properties. The eigenvalues of the matrices H_j are zero or clustered around zero and, therefore, the polynomial approximation induced by the Arnoldi scheme behaves as the series expansion (6). The dependence on the spectral properties of the matrices H_j is the mixed blessing for the Arnoldi scheme. To see this, let us consider the matrix in example (1b). The eigenvalues of A are located close to the imaginary axis in the interval between 1 and $N = 128$. The modified Arnoldi scheme applied for solving the problem $\phi_2(A)\boldsymbol{w} = \boldsymbol{b}$ with $\boldsymbol{b} = \boldsymbol{e}_1$ returns an approximation with relative error $err_2^{(j)}$ of order $1.0e-12$ in 20 iterations. Differently, the same algorithm applied for solving the problem $\phi_2(A)\boldsymbol{w} = \boldsymbol{b}$ with $\boldsymbol{b} = \texttt{rand}(N, 1)$ is unable to provide an approximation with relative error less than or equal to $1.0e-7$ in N

iterations. In the first case the eigenvalues of the matrices H_j are the corresponding diagonal entries of A and the approximation is quite effective. On the contrary, in the second case the eigenvalues of the matrices H_j are scattered in the complex plane and the approximations improve slowly. This effect combined with the simultaneous loss of orthogonality in the vectors of W_j leads to a poor accuracy. Concerning Algorithm 1, we observe that the eigenvalues λ_j of A are not in a region Ω of guaranteed convergence and we find that $\left|1 - \frac{\phi_2(\lambda_j)}{\phi_1(\lambda_j)}\right| > 1$ for about half of the eigenvalues. Notwithstanding that, for $\boldsymbol{b} = \boldsymbol{e}_1$ Algorithm 1 complemented with the adaptive computation of $r_{2,m}(A)$ converges with error $1.1e-9$ after 15 iterations of `gmres` whereas for $\boldsymbol{b} = \texttt{rand}(N, 1)$ the convergence requires 65 iterations of `gmres` with error $1.2e-8$.
To further evidence the role of the spectral properties of A and its approximations H_j, we consider the matrix in (1c) whose eigenvalues are clustered around the two points ξ and $-\xi$ with ξ being a pole of $\psi_2(z)$. Here ϵ is set to be equal $1.0e-8$, $\boldsymbol{w}$ is fixed equal to $\boldsymbol{e}$ and then $\boldsymbol{b}$ is determined by $\phi_2(A)\boldsymbol{w} = \boldsymbol{b}$. The spectrum of the matrices H_j accumulates around the two points ξ and ξ and the origin. In two steps the errors reach a minimum value of order $1.0e-8$ and after that rapidly grow and stabilize around the unit. This loss of accuracy displayed in Figure 4 is caused by catastrophic cancellation which determines the loss of orthogonality of the computed basis. The selection of the threshold value is critical to obtaining a feasible approximation.



(a) Semilog plot of $err_1^{(j)}$ for the matrix in (1c) (b) Semilog plot of $err_2^{(j)}$ for the matrix in (1c)

Figure 4: Semilog plots of $err_1^{(j)}$ and $err_2^{(j)}$ for the matrices defined in (1a) and (1c) of size $N = 128$ with $\epsilon = 1.0e-14, 1.0e-8$, respectively.

For comparison in Table 5 we show the results obtained by Algorithm 1 applied to the matrix in (1c).

2. In general, our approach can outperform the Arnoldi method when this latter exhibits a poor convergence. An example is given by the matrix $A = \texttt{gallery}('hanowa', 256)$. In Figure 5 we show the plot of the error $err_2^{(j)}$ generated in the computation of $\psi_2(A)\boldsymbol{b}$ with $\boldsymbol{b} = \texttt{ones}(256, 1)$. The process is stable and at the very end we compute an accurate approximation using the Hessenberg reduction of A. The matrix has eigenvalues

(n,m)	$(3,16)$	$(3,32)$	$(3,64)$	$(3,128)$
$it_{\texttt{gmres}}$	3	3	3	3
$r_{\texttt{gmres}}$	$1.8e-15$	$1.8e-15$	$1.7e-15$	$1.8e-15$
err_2	$5.0e-3$	$4.4e-5$	$3.6e-7$	$1.8e-8$

Table 5: Performance of Algorithm 1 applied to the matrix in (1c) of size $N = 128$ with $\epsilon = 1.0e-8$ and the tolerance of `gmres` set at $tol = 1.0e-12$. .

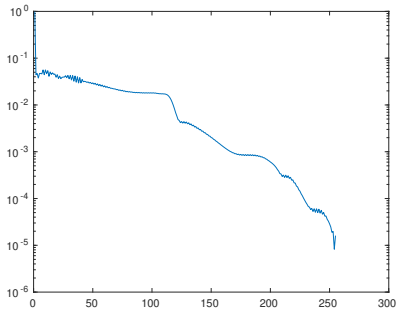


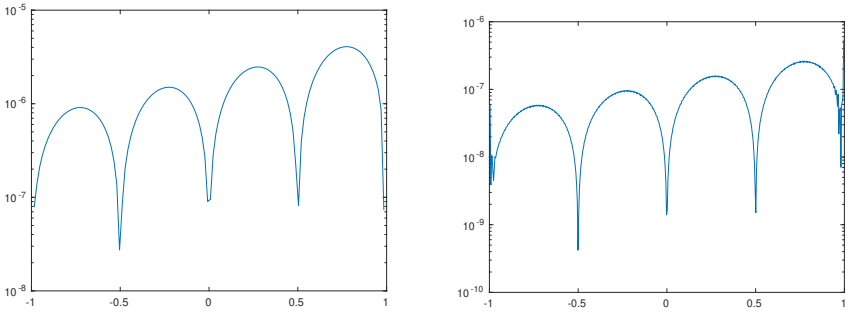
Figure 5: Semilog plot of $err_2^{(j)}$ generated by the Arnoldi method applied to $A = \texttt{gallery}('hanowa', 256)$.

located on a straight line parallel to the imaginary axis. The poor convergence is probably related with the periodicity of the complex exponential function which is not easily captured by the polynomial approximation induced by the Arnoldi construction. Algorithm 1 complemented with the adaptive computation of $r_{2,m}(A)$ returns an approximation with error $1.0e-5$ after 14 iterations of `gmres`. The value of m ranges in the interval $[165, 238]$. Observe that the eigenvalues of A are not lying in the strip $\Omega = \mathbb{R} \times i[-\pi/2, \pi/2]$ in the complex plane. Indeed, the preconditioned matrix has eigenvalues of modulus in the range $[0.59, 1, 59]$. The preconditioning is still effective and this opens an interesting perspective for future researches.

Numerical tests have been also performed to investigate the application of Algorithm 1 in the solution of the inverse problems described in the introduction. For the sake of illustration let us consider the following differential problem:

$$\frac{\partial u(z,t)}{\partial t} = \frac{e^{z-4}}{\sigma^2}\frac{\partial^2 u(z,t)}{\partial z^2} + tf(z), \quad f(z) = \sin(2\pi z), \ (z,t) \in [-1,1] \times [0,1], \tag{17}$$

with boundary conditions $u(-1,t) = u(1,t) = 0$, $u(z,0) = 0$ and $\sigma = 10$. The differential problem has been solved in Mathematica using the internal function NDSolve with extended precision. The computed solution $u(z,t)$ evaluated at $t = 1$ is used to define $h(z) = u(z,1)$. Then the inverse problem concerns



(a) (b)

Figure 6: Plots of the error vectors generated by Algorithm 1 for $N = 128$ (a) and $N = 512$ (b).

the reconstruction of $f(z)$ from the boundary conditions and the additional constraint $u(z, 1) = h(z)$. Using a discretization in space by finite differences over $N+2$ equispaced points in the interval $[-1, 1]$ leads to the first order system

$$\frac{\mathrm{d}\boldsymbol{u}}{\mathrm{d}t} = A\boldsymbol{u}(t) + t\boldsymbol{f},$$

$$A = \left(\frac{N+1}{2\sigma}\right)^2 \mathrm{diag}(e^{z_1-4}, \ldots, e^{z_N-4}) \begin{bmatrix} -2 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -2 \end{bmatrix}$$

with conditions

$$\boldsymbol{u}(0) = \boldsymbol{0}, \quad \boldsymbol{u}(1) = \boldsymbol{h},$$

and $\boldsymbol{f} = [f(z_1), \ldots, f(z_N)]^T$, $\boldsymbol{h} = [h(z_1), \ldots, h(z_N)]^T$. The unknown vector $\boldsymbol{f}$ can thus be determined by means of formula (3). The matrix A is similar to a negative definite matrix and therefore our methods can be applied. In Figure 6 we plot the absolute error vector with entries $|\hat{f}_i - f(z_i)|$, $1 \le i \le N$, $N \in \{128, 512\}$, where $\hat{f}_i$ are generated by Algorithm 1 with $n = 2$, $m = 32$, $tol = 1.0e-10$ and $maxit = 40$. The `gmres` command detects convergence at iteration 7 and 8 for $N = 128$ and $N = 512$, respectively. The finer discretization produces a small error. Similar plots are observed for the vectors generated by using `expm` and the backslash operator. The condition number of the matrices involved is of order $1.0e+8$.

5. Conclusions and Future Work

In this paper we have presented two approaches based on Newton's iteration and Krylov-type methods for the efficient computation of the inverse of a matrix

ϕ-function as well as the action of this inverse matrix on a vector. In particular, an appealing iterative procedure for computing $\psi_2(A)\boldsymbol{v}$ has been devised. Numerical experiments show that the proposed methods exhibit good robustness and convergence properties. The iterative scheme for the approximation of $\psi_2(A)\boldsymbol{v}$ requires at each step to compute an approximation of products of the form $\psi_1(A)\boldsymbol{w}$ by solving several linear systems whose matrices differ from A by a complex multiple of the identity matrix. Future work is concerned with the efficient solution of these shifted systems using the techniques introduced in [3]. Another interesting research topic would be the design of an adaptive modification of the algorithm in [5, 6] for computing $\psi_1(A)$ capable to determine automatically the "best" polynomial/rational approximation formula for $\psi_1(z)$.

Acknowledgment

The author would like to thank Prof. Paola Boito and Prof. Yuli Eidelman for useful discussions and feedback. The author is also indebted to Yuli Eidelman for the English translation of reference [20].

References

[1] M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*. National Bureau of Standards Applied Mathematics Series, No. 55. U. S. Government Printing Office, Washington, D.C., 1964. For sale by the Superintendent of Documents.

[2] M. Benzi. Localization in matrix computations: theory and applications. In *Exploiting hidden structure in matrix computations: algorithms and applications*, volume 2173 of *Lecture Notes in Math.*, pages 211–317. Springer, Cham, 2016.

[3] D. Bertaccini, M. Popolizio, and F. Durastante. Efficient approximation of functions of some large matrices by partial fraction expansions. *Int. J. Comput. Math.*, 96(9):1799–1817, 2019.

[4] D. A. Bini and B. Meini. Approximate displacement rank and applications. In *Structured matrices in mathematics, computer science, and engineering, II (Boulder, CO, 1999)*, volume 281 of *Contemp. Math.*, pages 215–232. Amer. Math. Soc., Providence, RI, 2001.

[5] P. Boito, Y. Eidelman, and L. Gemignani. Efficient solution of parameter-dependent quasiseparable systems and computation of meromorphic matrix functions. *Numer. Linear Algebra Appl.*, 25(6):e2141, 13, 2018.

[6] P. Boito, Y. Eidelman, and L. Gemignani. Computing the reciprocal of a ϕ-function by rational approximation. *Adv. Comput. Math.*, 48(1), 2022.

[7] M. Caliari, F. Cassini, and F. Zivcovich. BAMPHI: Matrix-free and transpose-free action of linear combinations of ϕ-functions from exponential integrators. *J. Comput. Appl. Math.*, 423:Paper No. 114973, 2023.

[8] R. M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Adv. Comput. Math.*, 5(4):329–359, 1996.

[9] V. L. Druskin and L. A. Knizhnerman. Two polynomial methods for calculating functions of symmetric matrices. *Zh. Vychisl. Mat. i Mat. Fiz.*, 29(12):1763–1775, 1989.

[10] Y. Eidelman and I. Gohberg. On a new class of structured matrices. *Integral Equations Operator Theory*, 34(3):293–324, 1999.

[11] Y. Eidelman, I. Gohberg, and I. Haimovici. *Separable type representations of matrices and fast algorithms. Vol. 1*, volume 234 of *Operator Theory: Advances and Applications*. Birkhäuser/Springer, Basel, 2014. Basics. Completion problems. Multiplication and inversion algorithms.

[12] M. Embree. How descriptive are GMRES convergence bounds?, 2022.

[13] E. Gallopoulos and Y. Saad. Efficient solution of parabolic equations by Krylov approximation methods. *SIAM J. Sci. Statist. Comput.*, 13(5):1236–1264, 1992.

[14] S. Gaudreault, G. Rainwater, and M. Tokman. KIOPS: a fast adaptive Krylov subspace solver for exponential integrators. *J. Comput. Phys.*, 372:236–255, 2018.

[15] J. Gondzio and P. Zhlobich. Multilevel quasiseparable matrices in pde-constrained optimization, 2011.

[16] N. J Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM review*, 51(4):747–764, 2009.

[17] N. J. Higham and L. Lin. Matrix functions: a short course. In *Matrix functions and matrix equations*, volume 19 of *Ser. Contemp. Appl. Math. CAM*, pages 1–27. Higher Ed. Press, Beijing, 2015.

[18] M. Hochbruck and A. Ostermann. Exponential integrators. *Acta Numer.*, 19:209–286, 2010.

[19] A. Iserles. How large is the exponential of a banded matrix? *New Zealand J. Math.*, 29(2):177–192, 2000. Dedicated to John Butcher.

[20] A. V. Karev and I. V. Tikhonov. The distribution of zeros of a Mittag-Leffler type entire function with applications in the theory of inverse problems. *Chelyab. Fiz.-Mat. Zh.*, 2(4):430–446, 2017.

[21] L. A. Knizhnerman. Calculation of functions of nonsymmetric matrices by means of Arnoldi's method. *Zh. Vychisl. Mat. i Mat. Fiz.*, 31(1):5–16, 1991.

[22] S. Massei and L. Robol. Decay bounds for the numerical quasiseparable preservation in matrix functions. *Linear Algebra Appl.*, 516:212–242, 2017.

[23] S. Massei, L. Robol, and D. Kressner. hm-toolbox: MATLAB software for HODLR and HSS matrices. *SIAM J. Sci. Comput.*, 42(2):C43–C68, 2020.

[24] B. V. Minchev and W. M. Wright. A review of exponential integrators for first order semi-linear problems. Technical report, Norwegian University of Science and Technology, 2005.

[25] J. Niesen and W. M. Wright. Algorithm 919: a Krylov subspace algorithm for evaluating the ϕ-functions appearing in exponential integrators. *ACM Trans. Math. Software*, 38(3):Art. 22, 19, 2012.

[26] V. Pan. Parametrization of Newton's iteration for computations with structured matrices and applications. *Comput. Math. Appl.*, 24(3):61–75, 1992.

[27] V. Pan and J. Reif. Fast and efficient parallel solution of dense linear systems. *Comput. Math. Appl.*, 17(11):1481–1491, 1989.

[28] V. Y. Pan. Newton's iteration for matrix inversion, advances and extensions. In *Matrix methods: theory, algorithms and applications*, pages 364–381. World Sci. Publ., Hackensack, NJ, 2010.

[29] V. Y. Pan, Y. Rami, and X. Wang. Structured matrices and Newton's iteration: unified approach. *Linear Algebra Appl.*, 343/344:233–265, 2002. Special issue on structured and infinite systems of linear equations.

[30] A. I. Prilepko, D. G. Orlovsky, and I. A. Vasin. *Methods for solving inverse problems in mathematical physics*, volume 231 of *Monographs and Textbooks in Pure and Applied Mathematics*. Marcel Dekker, Inc., New York, 2000.

[31] Y. Saad. *Iterative methods for sparse linear systems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[32] G. Sacchi. A new convergence model for the GMRES method. Master's thesis, Master's thesis, AlmaMater Studiorum, Università di Bologna, 2017. https://amslaurea.unibo.it/13501/1/sacchi_giulia_tesi_mag.pdf.

[33] G. Sacchi and V. Simoncini. A GMRES convergence analysis for localized invariant subspace ill-conditioning. *SIAM J. Matrix Anal. Appl.*, 40(2):542–563, 2019.

[34] R. B. Sidje. Expokit: A software package for computing matrix exponentials. *ACM Trans. Math. Softw.*, 24(1):130–156, mar 1998.

[35] A. Y. Suhov. A spectral method for the time evolution in parabolic problems. *J. Sci. Comput.*, 29(2):201–217, 2006.

[36] I. V. Tikhonov and Y. Eidelman. Questions of the well-posedness of direct and inverse problems for an evolution equation of special type. *Mat. Zametki*, 56(2):99–113, 160, 1994.

[37] M. Veneva and A. Ayriyan. Performance analysis of effective methods for solving band matrix slaes after parabolic nonlinear pdes. In K. Georgiev, M. Todorov, and I. Georgiev, editors, *Advanced Computing in Industrial Mathematics: 12th Annual Meeting of the Bulgarian Section of SIAM December 20-22, 2017, Sofia, Bulgaria Revised Selected Papers*, pages 407–419. Springer International Publishing, Cham, 2019.

[38] J. Xia, S. Chandrasekaran, M. Gu, and X. S. Li. Fast algorithms for hierarchically semiseparable matrices. *Numer. Linear Algebra Appl.*, 17(6):953–976, 2010.