Solving matrix polynomial equations arising in queueing problems

Dario A. Bini a, Guy Latouche b, Beatrice Meini a,∗

aDipartimento di Matematica “L. Tonelli”, Università di Pisa, via Buonarroti 2, 56127 Pisa, Italy
bDépartement d’Informatique, U.L.B., C.P. 212, Boulevard du Triomphe 1050, Bruxelles, Belgium

Received 3 October 2000; accepted 29 June 2001

Abstract

The matrix equation \( \sum_{i=0}^{n} A_i X^i = 0 \), where the \( A_i \)’s are \( m \times m \) matrices, is encountered in the numerical solution of Markov chains which model queueing problems. We provide here a unifying framework in terms of Möbius’ mapping to relate different resolution algorithms having a quadratic convergence. This allows us to compare algorithms like logarithmic reduction (LR) and cyclic reduction (CR), which extend Graeffe’s iteration to matrix polynomials, and the invariant subspace (IS) approach, which extends Cardinal’s algorithm. We devise new iterative techniques having quadratic convergence and present numerical experiments. © 2002 Elsevier Science Inc. All rights reserved.

Keywords: Matrix equations; Markov chains; Möbius map; Cyclic reduction; Logarithmic reduction; Invariant subspace; Matrix sign function

1. Introduction

Let \( A_i, i = 0, \ldots, n \), be \( m \times m \) matrices and consider the matrix equation

\[
\sum_{i=0}^{n} A_i X^i = 0.
\] (1)

The solution of this matrix equation is fundamental in the analysis of queueing problems modeled by Markov chains of M/G/1 type [23,27,28,33]. In certain cases,
\( n = +\infty \) and the left-hand side of (1) is a matrix power series; when \( n = 2 \), we have the so-called Quasi-Birth-and-Death (QBD) problems which are of particular interest [23]. Eq. (1) is encountered in many other applications like ladder networks, system theory, polynomial computations. We refer the reader to [3,17] for more references to applications.

We place ourselves squarely in the context of Markov chains analysis. In that case, the matrices \( A_i, i = 0, \ldots, n \), are such that \( I + A_1 \geq 0, A_i \geq 0 \) for \( i \neq 1 \), and \( \sum_{i=0}^{n} A_i + I \) is an irreducible stochastic matrix. Moreover, one is interested in the minimal nonnegative solution of (1), usually denoted by \( G \); this means that \( G \) is the nonnegative solution which is entrywise less than any other possible nonnegative solution of (1). We refer the reader to [28] for details on the existence and uniqueness of such solution \( G \).

Different techniques have been introduced in the literature to calculate that minimal solution. Algorithms based on functional iterations, having linear convergence, are analyzed in [11,12,20,25,32]. A few methods with quadratic convergence have also been proposed; the logarithmic reduction (LR) technique is defined in [22] for the special case \( n = 2 \) and a similar iteration, the cyclic reduction (CR) method, is extended in [4–6] to arbitrary values for \( n \leq +\infty \); the invariant subspace (IS) algorithm is introduced in [1] for general finite values of \( n \). Finally, we also mention the doubling technique proposed in [24] and Newton’s scheme analyzed in [21]—these, however, are less efficient because of their high computational cost per iteration, and will not be considered further here. A recent survey on methods to determine the solutions of (1) in a general context is presented in [17].

We pursue here the analysis and critical comparison between CR and IS performed in [26]. We show that Möbius’ mapping \( z(w) = (1 + w)/(1 - w) \) and its inverse \( w(z) = (z - 1)/(z + 1) \) are the fundamental keys for expressing the relation between the CR and LR algorithms on the one hand, and the IS algorithm on the other hand. More precisely, we show that the first two methods are particular extensions of Graeffe’s iteration [18,29] to matrix polynomials [15] and that their quadratic convergence is due to the implicit use of the square function \( S(z) = z^2 \). The IS algorithm, for its part, coincides with Cardinal’s algorithm [9,10], applied to a suitable Frobenius matrix and its convergence is due to the quadratic convergence of the sequence \( x_{i+1} = J(x_i) \), where \( J(x) \) is Joukowski’s function \( J(x) = \frac{1}{2}(x + x^{-1}) \). This is significant because the two functions are directly related by means of Möbius’ mapping since \( w(J(z)) = S(w(z)) \), as one easily verifies.

The paper is organized as follows. We recall in Section 2 the main properties of Möbius’ mapping, we analyze the correspondence between Joukowski’s and the square functions and we explain why these are important in the context of the computation of the matrix \( G \). In Section 3 we recall various iterative algorithms for factoring scalar polynomials (the iterations of Cardinal, Chebyshev, Graeffe and Sebastião e Silva) and relate them by means of Möbius’ mapping.

In Section 4 we show how the LR, CR and IS iterative procedures are based on generalizations to matrix polynomials of Graeffe’s and Cardinal’s procedures ana-
lyzed in Section 3, and we develop in Section 5 a new algorithm based on Chebyshev’s iterations. We show that this new procedure is better than the IS procedure. We give some numerical illustration in Section 6, and argue in Section 7 that the CR procedure (and its LR variant) is the method of choice in the context of Markov chains, both in terms of its speed of convergence and of its numerical stability.

2. Möbius’ map

Here we recall the main properties of Möbius’ map and prepare the tools for designing and relating algorithms for solving (1). Denote by $\mathbb{C}^+$ and $\mathbb{C}^-$ the half-planes made up by the complex numbers with strictly positive and strictly negative real part, respectively, and denote by $\mathcal{D}$ the open unit disk.

Möbius’ map $z(x)$ and its inverse $w(t)$ are defined by the functions

\begin{equation}
\begin{align*}
z(x) &= \frac{1 + x}{1 - x}, \quad w(t) = \frac{t - 1}{t + 1}
\end{align*}
\end{equation}

of a complex variable, where $x \neq 1$ and $t \neq -1$, respectively. The functions $z(x)$ and $w(t)$, respectively, map the unit circle without the point 1 and without the point $-1$ into the imaginary axis, and they map the imaginary axis into the unit circle without the point $-1$ and the point 1, respectively. Moreover, $z(x)$ and $w(t)$, respectively, map $\mathcal{D}$ into $\mathbb{C}^+$ and into $\mathbb{C}^-$ and they map the complement of the closed unit disk into $\mathbb{C}^-$ and into $\mathbb{C}^+$, respectively.

These are important features of the transformations, and we shall write that the mapping $z(x)$ transforms the coordinates of the unit circle into the coordinates of the imaginary axis.

2.1. Mappings of roots of polynomials

The two mappings (2) are very useful when we have to recast algorithms by interchanging the roles of the unit circle and of the imaginary axis, as in the analysis of the stability of polynomials and of the inertia of matrices. Given a polynomial $p(x)$, we may either apply this transformation to the polynomial itself or to its variable $x$. In the latter case we obtain in a natural way the operator

$\mathcal{P} : \Pi_n \to \Pi_n$

defined on the set

$$\Pi_n = \left\{ p(x) = \sum_{i=0}^{n} p_i x^i, \ p_i \in \mathbb{C} \right\}$$

of polynomials of degree at most $n$ having complex coefficients, by the following equation:

$$\mathcal{P}(p(x)) \equiv q(t) = (1 + t)^n p(w(t)).$$

(3)
It can be easily verified that, if \( p(x) \) has \( k \) roots in \( D \) and \( n - k \) roots in the complement of the closed unit disk, then \( q(t) \) has \( k \) roots in \( C^+ \) and \( n - k \) roots in \( C^- \).

Even though in principle there is no difference in the nature of the polynomials \( p(x) \) and \( q(t) \) in \( II_n \), we refer to \( II_n \) as being in the domain of the unit circle when we want to point out its role as the domain of definition of \( \mathcal{P} \), and in the domain of the imaginary axis when we refer to the co-domain, where \( \mathcal{P} \) takes its values. One motivation of this notational choice is that the operator \( \mathcal{P}(\cdot) \) allows us to transform algorithms for polynomial and matrix computations defined in the domain of the unit circle into algorithms defined in the domain of the imaginary axis.

### 2.2. Mapping of operators

Möbius’ mapping may also be applied to operators. When one considers successive iterations of the square function

\[
S(x) = x^2,
\]

the complex numbers are naturally partitioned into three subsets: the unit circle, its interior \( D \) and its exterior. Sequences obtained from the recursion \( x_{i+1} = S(x_i) \) quadratically converge to zero if \( |x_0| < 1 \) and to \( \infty \) if \( |x_0| > 1 \); there is in general no convergence for starting points with \( |x_0| = 1 \).

For Joukowski’s function

\[
J(t) = \frac{1}{2}(t + t^{-1}),
\]

the complex numbers are partitioned into \( C^+ \), \( C^- \) and the imaginary axis: sequences obtained from the recursions \( t_{i+1} = J(t_i) \) converge to 1 if \( t_0 \in C^+ \), to \(-1\) if \( t_0 \in C^- \) and fail to converge if \( t_0 \) is on the imaginary axis.

The square function and Joukowski’s function are related through Möbius’ map by the following properties which can be proved by direct inspection (see also [19]):

\[
w(-t) = \frac{1}{w(t)}, \quad z(-x) = \frac{1}{z(x)}, \quad (4)
\]

\[
w(t^{-1}) = -w(t), \quad z(x^{-1}) = -z(x), \quad (5)
\]

\[
w(J(t)) = S(w(t)), \quad z(S(x)) = J(z(x)), \quad (6)
\]

\[
w(-S(t)) = J(w(t)), \quad z(J(x)) = -S(z(x)). \quad (7)
\]

Now, let us consider a function \( f(x) \) of the variable \( x \) in the coordinates of the unit circle. We may apply the change of coordinates \( t = z(x) \) both to the independent variable \( x \) and to \( f \), thereby transforming \( f(x) \) into the function \( g(t) = z(f(w(t))) \) defined for \( t \) in the set of coordinates of the imaginary axis. In this way, any function defined in the domain of the unit circle has its counterpart in the domain of the imaginary axis.
This is the key observation that allows us to recast algorithms defined in the domain of the unit circle into algorithms defined in the domain of the imaginary axis.

2.3. Spectrum of $G$

The minimal nonnegative solution $G$ of (1) has its natural environment in the domain of the unit circle because of the following property [13,14]. Define $p(x) = \sum_{i=0}^{n} A_i x^i$ and denote by $\xi_1, \xi_2, \ldots$ the roots of $\det(p(x))$, ordered in increasing value of their modulus. The eigenvalues of $G$ are the $m$ smallest modulus roots $\xi_1, \ldots, \xi_m$, the eigenvalue $\xi_m$ with largest modulus is real, simple, and $|\xi_m| \leq 1$. Furthermore, $|\xi_{m+1}|$ is strictly greater than $\xi_m$, and is at least equal to 1. Thus, it comes as no surprise that methods which use repeated application of the square function will be useful in directly determining $G$. Due to these spectral properties, the matrix $G$ is also the minimal solvent in the notation of [17].

The approach followed in [1], i.e., the IS algorithm, consists in applying Möbius’ map (2), thus replacing (1) with the equation

$$\sum_{i=0}^{n} A_i (I - W)^{n-i}(I + W)^i = \sum_{i=0}^{n} U_i W^i = 0. \quad (8)$$

The solution $W$ having eigenvalues in $\mathbb{C}^- \cup \{0\}$ provides the solution $G$ by means of the formula $G = (I - W)^{-1}(I + W)$; moreover, the uniqueness of the minimal nonnegative solution $G$ of (1) implies the uniqueness of $W$. The natural environment of $W$ is in the domain of the imaginary axis, and it is clear why one should use Joukowski’s function as the basis for iterative algorithms to compute $W$, as is done in [1].

3. Polynomial factorization

The functions $S(x)$ and $J(t)$ are used to solve polynomial computational problems like approximating polynomial factors or polynomial roots.

3.1. Sebastião e Silva and Cardinal

The algorithms of Sebastião e Silva [35] and of Cardinal [10] are two such applications. The first one uses the square function, and the second uses Joukowski’s function.

Given a scalar polynomial $p(x)$, Sebastião e Silva’s procedure is based on the sequence $\phi_{i+1}(x) = \phi_i^2 \mod p(x)$, starting with an initial polynomial $\phi_0(x)$ (say, $\phi_0(x) = x$). Approximations to the factors of $p(x)$ containing equimodular zeros are determined by applying the Euclidean scheme to $p(x)$ and $\phi_i(x)$ for a large enough $i$. In Cardinal’s algorithm, one generates the sequence of polynomials $\psi_{i+1}(x) = \ldots$
\( \frac{1}{2}(\psi_i(x) + \psi_i(x)^{-1}) \mod p(x) \), starting from an initial polynomial \( \psi_0(x) \) and one determines two factors of \( p(x) \) containing the zeros with positive (respectively, negative) real parts from \( \gcd(\psi_i(x) + 1, \psi_i(x) - 1) \), for \( i \) large enough.

In matrix form, Sebastião e Silva’s algorithm corresponds to applying the power method to the Frobenius matrix

\[
F = \begin{pmatrix}
0 & 1 & & \\
& 0 & 1 & \\
& & \ddots & \ddots \\
& -p_0 & -p_1 & \ldots & -p_{n-1}
\end{pmatrix}
\]

associated with the polynomial \( p(x) \), and Cardinal’s method corresponds to applying the matrix sign iteration to the same matrix. One easily verifies from (7) that Cardinal’s method may be interpreted as resulting from the application of M"{o}bius’ map to the method of Sebastião e Silva. In fact, \( \phi_i \) is associated to \( \psi_i \) by the mapping defined in Section 2, i.e.,

\[
\psi_i(x) = z(\phi_i(x)) \mod p(x)
\]

for all \( i \) if we choose the initial polynomials \( \phi_0(x) \) and \( \psi_0(x) \) such that \( \psi_0(x) = z(\phi_0(x)) \mod p(x) \). To see this, assume that (10) holds for some \( i \geq 0 \); we have

\[
\psi_{i+1}(x) = J(\psi_i(x)) \mod p(x) \quad \text{(by definition)}
\]

\[= J(z(\phi_i(x))) \mod p(x) \quad \text{(by induction assumption)}
\]

\[= z(S(\phi_i(x))) \mod p(x) \quad \text{(by (6))}
\]

\[= z(\phi_{i+1}(x)) \mod p(x) \quad \text{(by definition)}
\]

which proves that (10) holds for all \( i \).

Observe also that, if \( p(x) = \prod_{i=1}^{n}(x - \xi_i) \), then \( \phi_i(\xi_j) = \phi_0(\xi_j)^{2^i} = S \circ S \circ \cdots \circ S(\phi_0(\xi_j)) \), whereas \( \psi_i(\xi_j) = J \circ J \circ \cdots \circ J(\psi_0(\xi_j)) \), where the functional composition \( \circ \) is applied \( i \) times.

The algorithms can be implemented in the polynomial setting with fast algorithms for polynomial computation [8] or in the matrix setting, where the computation modulo \( p(x) \) automatically results from the application of the square function \( S(\cdot) \) or Joukowski’s function \( J(\cdot) \) to the Frobenius matrix \( F \). In fact, we have \( \phi_i(F) = \phi_{i-1}(F)^2 = \phi_0(F)^{2^i} \), and each step of the repeated squaring iteration costs just one matrix multiplication. We also have \( \psi_i(F) = J(\psi_{i-1}(F)) = J \circ \cdots \circ J(\psi_0(F)) \) and the cost of each iteration with Joukowski’s function is dominated by the cost of one matrix inversion. We observe also that, in its matrix form, Sebastião e Silva’s method can be viewed as an acceleration of Bernoulli’s method which generates the sequence of vectors \( F^i e_1, \ i = 1, 2, \ldots, \ e_1 = (1, 0, \ldots, 0)^T \).

It might seem preferable to apply Sebastião e Silva’s iterations than Cardinal’s iterations since squaring a polynomial modulo \( p(x) \) is less expensive than computing the reciprocal modulo \( p(x) \). However, to apply the Euclidean scheme to \( p(x) \) and \( \phi_i(x) \) is a numerically ill-conditioned problem. In order to overcome this difficulty,
one might consider starting the computation in the domain of the unit circle, compute \( F_i = F_{2i} \) for some large enough \( i \) by means of repeated squaring, then switch to the domain of the imaginary axis by computing \( H_i = (I - F_i)^{-1}(I + F_i) \) and complete the computation there. Unfortunately, the condition number of the matrix \((I - F_i)\) which must be inverted for the computation of \( H_i \) grows exponentially with \( i \) and this makes the mixed approach numerically unstable as well. Therefore, Cardinal’s algorithm, or the equivalent matrix sign iteration, is the more appropriate for polynomial factorization.

This is a nice example of how the application of (2) allows us to relate different existing algorithms.

3.2. Graeffe and Chebyshev

One may also apply the mapping (2) to the variable of a polynomial; this provides us with a different type of association between methods. Take Graeffe’s iterative method \(^1\) [18,29], where a sequence of polynomials is defined as follows:

\[
p_{i+1}(x^2) = (-1)^n p_i(x) p_i(-x), \quad p_0(x) = p(x).
\]

Denote by \( \xi_1, \ldots, \xi_n \) the roots of \( p(x) \). The roots of \( p_i(x) \) are \( \xi_1^{2^i}, \ldots, \xi_n^{2^i} \). If \( k \) roots, \( \xi_1, \ldots, \xi_k \) say, have modulus less than 1, and \( n - k \) roots have modulus greater than 1, then \( k \) roots of the sequence \( \{p_i(x)\} \) tend to zero and \( n - k \) roots tend to infinity, and the convergence is doubly exponential.

This is the approach used in [9,30,31,34] to factor the polynomial \( p(x) \) with respect to the unit circle, i.e., to approximate the coefficients of the polynomials \( \prod_{i=1}^k (x - \xi_i) \) and \( \prod_{i=k+1}^n (x - \xi_i) \). Using the FFT-based fast polynomial arithmetic, the computation of the coefficients of \( p_{i+1}(x) \), given those of \( p_i(x) \), costs \( O(n \log n) \) arithmetic operations.

Let us consider first, for simplicity, the relation

\[
u(x^2) = (-1)^n v(x) v(-x),
\]

where \( u(x) \) and \( v(x) \) are polynomials of degree \( n \). If we replace \( x \) with \( w(t) \) of (2), multiply both sides of the above equation by \((1 + t)^n(1 + t^{-1})^n\) and apply the transformation (3), we find from (5) that

\[(1 + t)^n(1 + t^{-1})^n u(w^2(t)) = (-1)^n V(t)V(1/t),\]

where \( V(t) = \mathcal{P}(v(x)) \). Now, \( w^2(t) = w(J(t)) \) by (6), so that \( u(w^2(t)) = u(w(J(t))) \) and \((1 + J(t))^n u(w(J(t))) = U(J(t))\), where \( U(t) = \mathcal{P}(u(x)) \). Since \((1 + t)(1 + t^{-1})/(1 + J(t)) = 2\), we obtain that

\[U(J(t)) = (-2)^{-n} V(t)V(1/t).\]

By applying the argument above to (11), we obtain the iteration

\(^1\) First discovered by Dandelin and Lobachevski [29].
\[ q_{i+1}(J(t)) = (-2)^{-n} q_i(t) q_i(t^{-1}) \]  

(14)

and the polynomials in that sequence are such that
\[ q_i(t) = \mathcal{P}(p_i(x)) \quad \text{for all } i. \]

If we denote by \( \mu_1^{(i)}, \ldots, \mu_n^{(i)} \) the roots of \( u_i(w) \), then \( \mu_j^{(i+1)} = J(\mu_j^{(i)}) \), \( j = 1, \ldots, n \), and the sequence \( \{\mu_j^{(i)}\} \) converges doubly exponentially to 1 or \(-1\) according to the sign of the real part of \( \mu_j \).

Iteration (14), introduced in [9] and called Chebyshev’s iteration there, can be used to factor a polynomial with respect to the imaginary axis. The computation of the coefficients of \( u_{i+1}(t) \), given the coefficients of \( u_i(t) \), can be performed in \( O(n \log^2 n) \) arithmetic operation; the higher cost (compared to the cost of Graeffe’s iteration) results from the need to apply the evaluation/interpolation technique at the nodes \( \cos \pi i/2n, \ i = 1, \ldots, n \). See [9] for more details.

In Section 5, we will show how the transition from the domain of the unit circle to the domain of the imaginary axis can be performed also in the case of matrix polynomials, and use this fact to devise new iterative methods for the solution of the matrix equation (8). Before doing so, we recall in the following section two classical techniques for solving (1).

4. Past algorithms revisited

CR and LR operate in the domain of the unit circle and can be viewed as two different ways of extending Graeffe’s iteration to matrix polynomials, while the IS algorithm operates in the domain of the imaginary axis and can be viewed as an extension of Cardinal’s algorithm.

Before dealing with the general case, we describe these techniques for positive recurrent QBDs, for which \( n = 2 \) and (1) reduces to
\[ A_0 + A_1 X + A_2 X^2 = 0. \]

(15)

The minimal solution \( G \) of (15) has spectral radius equal to 1. Furthermore, the roots \( \xi_1, \ldots, \xi_{2m} \) of the polynomial \( \det(A_0 + A_1 x + A_2 x^2) \) are such that \( |\xi_1| \leq \cdots \leq |\xi_{2m}| \), \( \xi_m = 1 < |\xi_{m+1}| \) and \( \xi_m \) is simple (here we assume zeros at infinity if \( A_2 \) is singular).

4.1. Logarithmic reduction

We express in polynomial form the algorithm LR of [22] for the solution of (15). Let \( r_i(x) \) be matrix polynomials recursively defined by
\begin{align*}
r_i(z) &= E_{0,i} + E_{1,i} x + E_{2,i} x^2, \\
r_{i+1}(x^2) &= -\tilde{r}_i(x)\tilde{r}_i(-x), \\
\tilde{r}_i(x) &= E_{1,i}^{-1} r_i(x),
\end{align*}

(16)
where \( r_0(x) = A_0 + A_1 x + A_2 x^2 \). The matrix coefficients of \( r_i(x) \) and \( r_{i+1}(x) \) are related by the equations

\[
\begin{align*}
E_{0,i+1} &= -\left( E_{1,i}^{-1} E_{0,i} \right)^2, \\
E_{2,i+1} &= -\left( E_{1,i}^{-1} E_{2,i} \right)^2, \\
E_{1,i+1} &= I - \left( E_{1,i}^{-1} E_{0,i} \right) \left( E_{1,i}^{-1} E_{2,i} \right) - \left( E_{1,i}^{-1} E_{2,i} \right) \left( E_{1,i}^{-1} E_{0,i} \right),
\end{align*}
\]

(17)

and the following result is proved in [22].

**Theorem 1.** The minimal solution \( G \) of \((15)\) is such that \( G = G_i + O(\sigma^{2i}) \) with \( \sigma = |\xi_m|/|\xi_{m+1}| < 1 \), where

\[
G_i = \sum_{j=0}^{i} \left( \prod_{k=0}^{j-1} D_{2,k} \right) D_{0,k},
\]

\[
D_{j,i} = -E_{1,i}^{-1} E_{j,i}, \quad j = 0, 2.
\]

Observe that the computation of (17) can be performed with one matrix inversion and six matrix multiplications (we do not count matrix additions since they have a lower cost \( O(m^2) \) instead of \( O(m^3) \)). In Markov chains applications, the matrices \( E_{1,i} \) are nonsingular M-matrices and their inversion can be performed in a numerically stable way without pivoting by applying Gaussian elimination with the GTH trick of [16]. Actually, these matrices are not explicitly inverted: their LU factorization is computed instead. In this way, multiplying a matrix by \( U^{-1} L^{-1} \) corresponds to solving 2\( m \) triangular systems and can be performed in \( 2m^3 + O(m^2) \) arithmetic operations. The cost of the LU factorization is dominated by \( (2/3)m^3 \) ops, while the cost of matrix multiplication is dominated by \( 2m^3 \) ops. The overall cost, including two matrix multiplications needed to update \( G_i \), is therefore \( (50/3)m^3 + O(m^2) \) ops per iteration. Besides the numerically stable inversion of \( E_{1,i} \), the remaining operations involve multiplications and additions of nonnegative numbers, so that there is no possibility of cancellation errors. Moreover, no overflow/underflow situations are encountered in the computation of \( G \) since \( G_i \) has nonnegative entries less than or equal to 1 which converge in a nondecreasing way [22].

### 4.2. Cyclic reduction

Define the matrix polynomials \( p_i(x) = A_{0,i} + A_{1,i} x + A_{2,i} x^2 \) by

\[
p_{i+1}(x^2) = -p_i(x)A_{1,i}^{-1} p_i(-x),
\]

(18)

with \( p_0(x) = A_0 + A_1 x + A_2 x^2 \). The coefficient matrices \( A_{j,i+1} \) themselves are iteratively defined by
\[ A_{1,i+1} = A_{1,i} - A_{0,i} A_{1,i}^{-1} A_{2,i} - A_{2,i} A_{1,i}^{-1} A_{0,i}, \]
\[ A_{0,i+1} = -A_{0,i} A_{1,i}^{-1} A_{0,i}, \]
\[ A_{2,i+1} = -A_{2,i} A_{1,i}^{-1} A_{2,i}, \]
\[ \text{and the following property is proved in \[5\].} \]

**Theorem 2.** The minimal solution \( G \) of (15) is such that
\[ G = G_i + O(\sigma^2), \]
with \( \sigma = |\xi_m|/|\xi_{m+1}| < 1 \), where \( G_i = -\hat{A}_i A_{0,i} \) and
\[ \hat{A}_{i+1} = \hat{A}_i - A_{2,i} A_{1,i}^{-1} A_{0,i}, \]
for \( i \geq 0 \), with \( \hat{A}_0 = A_1 \).

The computational cost per iteration is one matrix inversion and six matrix multiplications, including the cost of updating \( \hat{A}_i \). The overall cost is therefore (38/3)\( m^3 \) + \( O(m^2) \) ops. The same numerical stability properties hold as for LR. As a matter of fact, the two algorithms are directly related as the following theorem shows (the proof by induction is immediate and omitted).

**Theorem 3.** The blocks ((17) and (19)) generated by LR and CR are such that
\[ E_{j,i} = A_{1,i-1} A_{j,i} \]
for \( j = 0, 1, 2 \) and all \( i \geq 1 \).

### 4.3. Invariant subspace algorithm

In the QBD case, (8) becomes
\[ U_0 + U_1 W + U_2 W^2 = 0, \]
obtained by replacing \( X \) in (15) with \( X = (I + W)(I - W)^{-1} \) and then multiplying on the right by \( (I - W)^2 \). Observe that it is the function \( w(t) \) which is used for mapping the coordinates of the unit circle onto the coordinates of the imaginary axis. The coefficients are
\[ U_0 = A_0 + A_1 + A_2, \]
\[ U_1 = 2A_2 - 2A_0, \]
\[ U_2 = A_0 - A_1 + A_2, \]
and \( U_2 \) is nonsingular. If \( \xi_m = 1 \) and \( |\xi_{m-1}| < 1 \), then the solution \( W = (I + G)^{-1} \)
\( (I - G) \) of (21) which corresponds to the minimal solution \( G \) of (15) has one null eigenvalue and the remaining eigenvalues in the open left half-plane. Because of that null eigenvalue, one may not directly apply Joukowski’s function to \( W \). This difficulty is solved as follows.
The block Frobenius matrix

\[ H = \begin{bmatrix} 0 & I \\ U_2^{-1}U_0 & -U_2^{-1}U_1 \end{bmatrix} \]

has for eigenvalues the roots \( \eta_i, i = 1, \ldots, 2m \), of \( \det(U_0 + U_1t + U_2t^2) \), and \( \eta_i = w(\xi_i) \), so that \( H \) has \( m - 1 \) eigenvalues in \( \mathbb{C}^- \), one equal to zero, and \( m \) in \( \mathbb{C}^+ \).

If \( x \) and \( y \) are such that \( U_0x = 0, y^TU_0 = 0, y^TU_1x = 1 \), then the matrix

\[ \hat{H} = H - \begin{bmatrix} x \\ 0 \end{bmatrix} [y^TU_1 \ y^TU_2] \]

has \( m \) eigenvalues in \( \mathbb{C}^- \) and \( m \) eigenvalues in \( \mathbb{C}^+ \). Therefore, the matrix sign function algorithm [19], that is generated by the repeated application of the Joukowski function, can be applied to \( \hat{H} \), yielding the sequence

\[ \hat{H}_{i+1} = \frac{1}{2}(\hat{H}_i + \hat{H}_i^{-1}) \]

(22)

converging to \( \lim_i \hat{H}_i = K \), which is the matrix sign function of \( \hat{H} \). That limit may be expressed as \( K = TST^{-1} \), where \( S = \text{diag}(s_1, \ldots, s_{2m}) \), \( s_i = \text{sign}(\text{Re}(\eta_i)) \), and \( J = T^{-1}\hat{H}T \) is the Jordan normal form of \( \hat{H} \). Now, it is possible to recover \( W \) from \( K \) as follows: by means of the QR factorization, say, of \( K - I \), compute the \((2m) \times m \) matrix \( K^* \) made up by \( m \) linearly independent columns of \( K - I \); partition \( K^* \) into two \( m \times m \) blocks \( K_1^* \) and \( K_2^* \). The matrix \( W = (K_1^* + K_2^*)(K_1^* - K_2^*)^{-1} \) is the solution of (21) having eigenvalues in \( \mathbb{C}^- \), and \( G = (I + W)(I - W)^{-1} = -K_1^*(K_2^*)^{-1} \) is the minimal nonnegative solution of (15).

The cost of (22) amounts to inverting a \((2m) \times (2m) \) matrix, that is about 16\( m^3 \) ops, to be compared to \((50/3)m^3 \) for LR and to \((38/3)m^3 \) for CR. For the final computation of \( G \), we have to compute a QR factorization of \( \hat{H}_i - I \), for a large enough value of \( i \), obtain approximations of \( K_1^* \) and \( K_2^* \), and compute \( G = -K_1^*(K_2^*)^{-1} \) with one matrix inversion and one matrix product. The cost of the latter computation amounts to \( \frac{4}{3}(2m)^3 + 6m^3 \) ops.

The convergence speed of \( \hat{H}_i \) to \( K \) depends on how fast the values \( J^{(i)}(\eta_j) \) converge to \( \pm 1 \), where \( J^{(i)} \) denotes the composition of Joukowski’s function \( i \) times with itself. Defining \( w_{i+1} = J(w_i) \), one has that \( w_{i+1} \pm 1 = (w_i \pm 1)^2 / 2w_i \), so that one may expect slow convergence if \( w_0 \) is very close to zero or if \( w_0 \) is very large in modulus or if \( w_0 \) is very close to the imaginary axis. In the domain of the unit circle, the three conditions correspond to eigenvalues of \( G \) being close to \( 1, -1 \) or the unit circle. By contrast, the convergence of CR and LR is not slowed down if there are eigenvalues of modulus close to 1 provided that the ratio \( |\xi_m / \xi_{m+1}| \) is sufficiently small. Therefore, we may expect LR and CR to converge faster than IS in general.

Moreover, if some value \( \eta_j \) is close to the imaginary axis, the matrices \( \hat{H}_i \) generated by IS may have a very large condition number and the computation can be affected by large rounding errors [2].
4.4. Arbitrary matrix power series

If \( p(x) = \sum_{i=0}^{+\infty} A_i x^i \) is a matrix power series or a matrix polynomial of degree \( n \) (when \( A_{n+1} = A_{n+2} = \cdots = 0 \)), then LR and CR still apply. For CR, Eq. (18) must be adjusted in the following way [4,5]:

\[
p_{i+1}(x^2) = -p_i(x) K_i(x)^{-1} p_i(-x),
\]

where \( K_i(x^2) = (p_i(x) - p_i(-x))/(2x) \)

with \( p_0(x) = \sum_{j=0}^{+\infty} A_j x^j \).

In this generalization the quadratic matrix polynomials are replaced by the matrix power series \( p_i(x) \) and the matrices \( \hat{A}_i \) of Theorem 2 are replaced by the matrix power series \( \hat{p}_i(x) \) defined below (for more details we refer the reader to [4–6]):

\[
\hat{p}_{i+1}(x) = \hat{p}_i^{(\text{odd})}(x) - p_i^{(\text{even})}(x) K_i(x)^{-1} \hat{p}_i^{(\text{even})}(x),
\]

where

\[
\hat{p}_i(x) = \sum_{j=0}^{+\infty} \hat{A}_{j+1,i} x^j, \quad \hat{p}_0(x) = \sum_{j=0}^{+\infty} A_{j+1} x^j,
\]

\[
p_i^{(\text{even})}(x) = \sum_{j=0}^{+\infty} A_{2j,i} x^j, \quad \hat{p}_i^{(\text{even})}(x) = \sum_{j=0}^{+\infty} \hat{A}_{2(j+1),i} x^j,
\]

\[
\hat{p}_i^{(\text{odd})}(x) = \sum_{j=0}^{+\infty} \hat{A}_{2j+1,i} x^j.
\]

The matrix power series in the sequences \( \{p_i(x)\} \) and \( \{\hat{p}_i(x)\} \) converge in the unit disk, and for this reason can be easily approximated with matrix polynomials of numerical degree \( n_i \) and \( \hat{n}_i \), respectively; here we define the numerical degree of the matrix power series \( f(x) = \sum_{j=0}^{+\infty} F_j z^j \) as the minimum integer \( n \) such that \( \| \sum_{j=n+1}^{+\infty} F_j \|_1 < \epsilon \), where \( \epsilon \) is the machine precision and \( \| \cdot \|_1 \) is the 1-norm defined by \( \| A \|_1 = \max_j \sum_i |a_{i,j}| \). Moreover,

\[
A_0 + \sum_{j=0}^{+\infty} \hat{A}_{j+1,i} G^{1+j2i} = 0
\]

for all \( i \), and the following result holds:

**Theorem 4.** Let the matrix power series \( p(x) = \sum_{i=0}^{+\infty} A_i x^i \) be such that \( I + A_1 \geq 0, A_i \geq 0, i \neq 1, B = I + \sum_{i=0}^{+\infty} A_i \) is stochastic and irreducible. Assume that \( p(x) \) is meromorphic in the complex plane, that \( \sum_{i=1}^{+\infty} i A_i \) is convergent and that the dominating left eigenvector \( b \) of \( B \), normalized by \( b^T e = 1 \), where \( e = (1, \ldots, 1)^T \).
satisfies $b^T \sum_{i=1}^{+\infty} i A_i e < 1$. Then Eq. (1) has one nonnegative solution $G$ with spectral radius $1$ and $G = -\hat{A}_{1,i}^{-1} A_0 + O(\sigma^{-2i})$, where $\hat{A}_{1,i} = \hat{p}_i(0)$ and $\sigma = \min\{|x|: x \in \mathbb{C}, \det p(x) = 0, |x| > 1\}$.

**Proof.** From the hypotheses it follows that there exists a sequence of positive vectors $\pi_j \in \mathbb{R}^m$, $j = 0, 1, 2, \ldots$, such that

$$
\pi_0^T (A_0 + A_1) + \pi_1^T A_0 = 0,
\sum_{j=0}^{i} \pi_j^T A_{i-j} = 0, \quad i = 2, 3, \ldots,
\sum_{j=0}^{+\infty} \pi_j^T e = 1
$$

(see [28]), and $\pi_j = O(\sigma^{-j})$ (see [14]). By following the same argument as in the proof of Theorem 4.1 [5], we find that $\pi_0^T \hat{A}_{j,i} + v_j^T = \pi_{(j-1)2^i}^T$ for $j \geq 2$, where

$$
v_{j,i} = \sum_{k=1}^{j} \pi_k^T A_{j-k,i} + \pi_{(j-1)2^i}^T.
$$

Since the vector $v_{j,i}$ is nonnegative (see [5]), $\pi_0$ is positive and $\hat{A}_{j,i}$ is nonnegative for $j \geq 2$, then we deduce that $\hat{A}_{j,i} = O(\sigma^{-(j-1)2^i})$ for $j = 2, 3, \ldots$ The thesis follows from (25). \qed

A consequence of this theorem is that the numerical degrees $n_i, \hat{n}_i$ are bounded from above by a constant $N$. The computation of the $n_i + 1$ coefficients of $p_i(x)$ can be performed by means of evaluation/interpolation at the roots of 1 at a cost of $O(N m^3 + N m^2 \log m)$ ops (and the same bound holds for the computation of the coefficients of $\hat{p}_i(x)$); in practice, the values of $n_i$ and $\hat{n}_i$ are roughly halved at each step of CR, which makes CR an effective tool for the numerical solution of (1) even for matrix power series. See [6] for details.

It is clear that one may similarly generalize the LR algorithm to the case of matrix power series.

Two special cases are worth mentioning. The first one is when $p(x)$ is a matrix polynomial of degree $n$. Then we may in principle avoid the general form (24) of the iterative procedure by considering the “re-blocked” matrix equation

$$
\mathcal{A}_0 + \mathcal{A}_1 X + \mathcal{A}_2 X^2 = 0,
$$

where $\mathcal{A}_i$ are $(n-1) \times (n-1)$ block matrices with $m \times m$ blocks and: $\mathcal{A}_2$ is the block lower triangular Toeplitz matrix with $A_{n-i+j}$ in position $(i, j)$ for $i \geq j$; $\mathcal{A}_1$ is the block Hessenberg block Toeplitz matrix with $A_{j-i+1}$ in position $(i, j)$ for $j \geq i - 1$; $\mathcal{A}_0$ is the block matrix having null blocks everywhere except for $A_0$ in the upper rightmost corner. We might apply to (26) the CR algorithm (19) or the LR algorithm (17); by following the same technique as in [7], it can be proved that the cost of each iteration is $O(m^3 n \log^2 m)$ ops. We see that the approach based on
power series is generally much more convenient since the numerical degrees of the
matrix power series involved in the computation rapidly decay.

The second case is when the matrix power series \( p(x) = \sum_{i=0}^{+\infty} A_i x^i \) is such
that \( x + p(x) \) is a rational function, i.e., when \( x + p(x) = d(x)^{-1}e(x) \), where \( d(x), e(x) \) are matrix polynomials. Then the equation \( p(X) = 0 \) can be rewritten as \( X = d^{-1}(X)e(X) \) or \( d(X)X - e(X) = 0 \). In this way we reduce the original equation to
a polynomial equation and CR can be applied in its polynomial form.

The IS algorithm can be easily applied to the case of matrix polynomials of any
degree \( n \). The cost of each step is dominated by the inversion of an \( nm \times nm \) matrix
in the algebra generated by the block Frobenius matrix associated with the matrix
polynomial \( p(x) = \sum_{i=0}^{n} A_i x^i \). This is a Toeplitz-like matrix and its inversion costs
\( O(m^3 \log^2 n) \) ops. The evaluation of the invariant subspaces by means of the SVD
costs \( O((nm)^3) \) ops and this is the most expensive part of the overall computation
for large \( n \).

5. Chebyshev’s iteration

We extend here Chebyshev’s iteration (14) in the CR fashion. We give all details
for matrix polynomials of degree \( n = 2 \). The case of matrix polynomials of arbitrary
degree is not given here as it is dealt with in a similar manner, starting from the
general CR iteration (23).

In order to solve (15) in the domain of the unit circle, we apply the operator \( \mathcal{P} \) of
(3) to the matrix polynomial \( p_0(x) \) and solve the equation

\[
Q_0 + Q_1 T + Q_2 T^2 = 0
\]  

(27)
in the domain of the imaginary axis, where

\[
Q_0 = A_0 - A_1 + A_2, \\
Q_1 = 2(A_0 - A_2), \\
Q_2 = A_0 + A_1 + A_2.
\]

If \( T \) is the solution of (27) with all eigenvalues having nonnegative real parts, then
\( G = (T - I)(T + I)^{-1} \) is the solution of (14) with all eigenvalues in the unit circle.

We define the matrix polynomial sequence

\[
q_i(t) = (1 + t)^2 p_i(w(t)) = Q_{0,i} + Q_{1,i}t + Q_{2,i}t^2
\]  

(28)

and, in order to overcome the noncommutativity of matrix multiplication, we need
to find matrices \( K_i \) with the same role of \( -A_{1,i} \) in (18) such that (14) generalizes to
\( q_{i+1}(J(t)) = q_i(t)Mq_i(t^{-1}) \) for some matrix \( M \).

First, we observe that \( -A_{1,i} = (p_i(-x) - p_i(x))/(2x) \). Setting \( x = w(t) \) in the
equation above, we obtain that
\[-A_{1,i} = \frac{t+1}{2(t-1)} \left( \frac{1}{(t-1+1)^2} q_i(t-1) - \frac{1}{(t+1)^2} q_i(t) \right) \]
\[= (Q_{0,i} - Q_{2,i})/2. \tag{29} \]

Now, apply to (18) the same transformations as to (12): replace \(x\) with \(w(t)\) and multiply by \((1+t)^2(1+t^{-1})^2\). This leads to
\[q_{i+1}(J(t)) = \frac{1}{2} q_i(t) K_i q_i(t^{-1}), \tag{30} \]
where
\[K_i = (Q_{0,i} - Q_{2,i})^{-1}. \]

The matrix coefficients of \(q_i(t)\) are related to those of \(q_{i+1}(t)\) by Eqs. (31). For the sake of simplicity in proving (31), we temporarily write \(Q_{0,i}, Q_{1,i}, Q_{2,i}\) and \(K_i\) instead of \(Q_{0,i}, Q_{1,i}, Q_{2,i}\) and \(K_i\).

We define \(\sum_{j=-2}^{2} C_j t^j = q_i(t) K_i q_i(t^{-1})\) and verify by direct inspection that
\[C_1 = Q_1 K Q_0 + Q_2 K Q_1, \quad C_{-1} = Q_0 K Q_1 + Q_1 K Q_2, \]
\[C_2 = Q_2 K Q_0, \quad C_{-2} = Q_0 K Q_2, \]
\[C_0 = Q_0 K Q_0 + Q_1 K Q_1 + Q_2 K Q_2. \]

We have that \(C_1 = C_{-1}\) and \(C_2 = C_{-2}\) since \(Q_1 K (Q_0 - Q_2) = Q_1 = (Q_0 - Q_2) K Q_1\) and \(Q_2 K Q_0 = Q_0 K Q_0 - Q_0 = Q_0 K Q_2\). Therefore, we obtain that
\[\sum_{j=-2}^{2} C_j t^j = C_0 + C_1(t + t^{-1}) + C_2(t^2 + t^{-2}) \]
\[= (C_0 - 2C_2) + \frac{t + t^{-1}}{2} 2C_1 + \left( \frac{t + t^{-1}}{2} \right)^2 (4C_2). \]

Finally, since \(C_0 - 2C_2 = Q_0 - Q_2 + Q_1 K Q_1\), the coefficient matrices \(Q_{0,i}, Q_{1,i}, Q_{2,i}\), of the matrix polynomials \(q_i(t)\) are iteratively determined by the following relations:
\[Q_{0,i+1} = \frac{1}{2} (Q_{0,i} - Q_{2,i} + Q_{1,i} K_i Q_{1,i}), \]
\[Q_{1,i+1} = Q_{2,i} K_i Q_{1,i} + Q_{1,i} K_i Q_{0,i}, \tag{31} \]
\[Q_{2,i+1} = 2 Q_{2,i} K_i Q_{0,i}. \]

In order for this algorithm to be useful, we need to express in the domain of the imaginary axis the successive approximations of \(G\) which are defined in Theorem 2. This means that we need an expression for the matrices \(\hat{A}_i\) of (20) and, unfortunately, it seems that there is no direct polynomial interpretation of these. Thus, we need to explicitly relate the coefficients of \(p_i(x)\) with those of \(q_i(t)\). A simple calculation shows that
\( Q_{0,i} = A_{0,i} - A_{1,i} + A_{2,i}, \quad A_{0,i} = \frac{1}{4}(Q_{0,i} + Q_{1,i} + Q_{2,i}), \)
\( Q_{1,i} = 2(A_{0,i} - A_{2,i}), \quad A_{1,i} = -\frac{1}{2}(Q_{0,i} - Q_{2,i}), \)
\( Q_{2,i} = A_{0,i} + A_{1,i} + A_{2,i}, \quad A_{2,i} = \frac{1}{4}(Q_{0,i} - Q_{1,i} + Q_{2,i}). \)

Defining \( \hat{Q}_i \) as
\[
\hat{Q}_i = \hat{Q}_i + A_{-1,i} A_{0} \quad (Q_{0,i}, i + (Q_{0,i} - Q_{1,i} + Q_{2,i})K_i(Q_{0,i} + Q_{1,i} + Q_{2,i}) / 8,
\]
where \( \hat{Q}_0 = -\frac{1}{2}(Q_0 - Q_2) \), we readily conclude from (20) that \( \hat{Q}_i = \hat{A}_i \) for all \( i \).

Then the matrix sequence \( G_i = -\hat{A}_i^{-1} A_0 \) which quadratically converge to \( G \) (by Theorem 2) can be written as
\[
G_i = -\frac{1}{4} \hat{Q}_i^{-1}(Q_0 + Q_1 + Q_2).
\]
The computational cost at each step of iterations (31) and (32) amounts to one matrix inversion and eight matrix multiplications; this cost is comparable with the one of CR.

In conclusion of this section, we point to the fact that, should it be necessary, Chebyshev’s iterative scheme may be used to approximate the solution \( W = (G + I)^{-1} (G - I) \) of (21) having eigenvalues with nonpositive real parts. Indeed, since \( G_i = -\hat{A}_i^{-1} A_0 \) provides an approximation to \( G \), \( W_i = (G_i + I)^{-1} (G_i - I) \) provides an approximation to \( W \). From (33), a simple calculation shows that
\[
W_i = (-4 \hat{Q}_i + Q_{0,i} + Q_{1,i} + Q_{2,i})^{-1}(4 \hat{Q}_i + Q_{0,i} + Q_{1,i} + Q_{2,i}) \quad (34)
\]
and the following is a direct consequence of Theorem 2.

**Theorem 5.** Let Eq. (21) have a solution \( W \) with eigenvalues \( \eta_1, \ldots, \eta_m \) having nonpositive real parts. Then \( W = W_i + O(\delta^2) \), where \( W_i \) is defined in (31), (32), and (34) and
\[
\delta = \left| \frac{(\eta_m + 1)(\eta_{m+1} - 1)}{(\eta_m - 1)(\eta_{m+1} + 1)} \right| < 1.
\]

6. Numerical experiments

We have implemented in Fortran 95 the three techniques CR, IS and Chebyshev’s iteration (henceforth denoted as ChI) for the solution of (15). We restricted our attention to these algorithms since they have a quadratic convergence and can be related to each other by means of Möbius’ map. For comparisons with different methods having linear convergence like fixed point iterations or having quadratic convergence like the doubling technique of [24], we refer the reader to [5, 6, 25].

We checked the convergence speed of the three algorithms and their numerical performances. We report in Table 1 the number of steps and the residual error for
three test problems. If $\tilde{G}$ is an approximate solution of (15), we define, $\| A_0 + A_1 \tilde{G} + A_2 \tilde{G}^2 \|_{\infty}$ the residual error, where $\| \cdot \|_{\infty}$ denotes the infinity norm.

**Problem 6.** The $m \times m$ matrices $A_0$, $A_1$, $A_2$, where $m = 32$, are given by, $A_0 = M_1^{-1}M_0$, $A_1 = I$, $A_2 = M_1^{-1}M_2$, with $(M_2)_{i,j} = \alpha_i$ if $j = i + 1 \pmod m$, $(M_2)_{m,m} = \sigma$, $(M_2)_{i,j} = 0$ elsewhere, $(M_0)_{i,j} = \beta_i$ if $j = i - 1 \pmod m$, $(M_0)_{i,j} = 0$ elsewhere, $M_1 = \text{diag}(\gamma_1, \ldots, \gamma_m)$, $\gamma_i = -\alpha_i - \beta_i$, $i = 1, \ldots, m - 1$, $\gamma_m = -\alpha_m - \beta_m - \sigma$. The numerical values of the parameters are $\alpha_i = 1$, $\beta_i = 1.2$, for $i = 1, \ldots, 16$, $\beta_i = 0.85$ for $i = 17, \ldots, 32$, $\sigma = 0.001$. For this problem the eigenvalues are close to the unit circle on both sides: it holds $\xi_{m-1} = 0.99\times10^9$, $\xi_m = 1$ and $\xi_{m+1} = 1.00978$. In this case, since $|\xi_m/\xi_{m+1}| < |\xi_{m-1}/\xi_m|$, CR performs better than IS, as seen in Table 1. Moreover, that condition number of $\tilde{H}$ and of $K$ is $1.2 \times 10^{10}$ and $6.4 \times 10^{6}$, respectively, thus leading to instability problems for IS, and to its much larger residual error. The algorithm ChI has the same behavior as CR, as expected.

**Problem 7.** This example is taken from [23, p. 208]. It represents a queueing system in a random environment, where periods of severe overflows alternate with periods of low arrivals. The $m \times m$ matrices $A_0$, $A_1$, $A_2$, where $m = 8$, are given by, $A_0 = M_1^{-1}M_0$, $A_1 = I$, $A_2 = M_1^{-1}M_2$, with $M_2 = \rho \cdot \text{diag}(\alpha_1, \ldots, \alpha_m)$ $M_0 = \text{diag}(\beta_1, \ldots, \beta_m)$ $(M_1)_{i,j} = 1$ if $j = (i \pmod m) + 1$, $(M_1)_{i,j} = -1 - \rho \alpha_i - \beta_i$ if $j = i$, $(M_1)_{i,j} = 0$ elsewhere. The numerical values of the parameters are $\alpha = (0.2, 0.2, 0.2, 0.2, 13, 1, 1, 0.2)$, $\beta_i = 2$ for $i = 1, \ldots, m$, $\rho = 0.99$. Here $\xi_m = 1$, $\xi_{m+1} = 1.00188$, and the remaining zeros are far from the unit circle. This is a critical case for the three algorithms, but CR and ChI converge faster to a more accurate solution than IS. The condition number of $\tilde{H}$ and of $K$ is $5.9 \times 10^8$ and $2.2 \times 10^6$, respectively.

**Problem 8.** This example is taken from [26]. The matrices $A_0$, $A_1$, $A_2$ have all the entries equal to $\alpha = (\rho - 1)/(3(m - 1))$ for $i \neq j$, $(A_0)_{i,j} = -\rho$, $(A_1)_{i,i} = 1$, $(A_2)_{i,i} = 0$ for $i = 1, \ldots, m$, and $m = 32$. We have chosen $\rho = 0.99$, so that $\xi_m = 1$, $\xi_{m+1} = 1.00000003$, and the remaining zeros are equal and far from the unit circle. This is a very difficult problem since $\xi_m/\xi_{m+1}$ is very close to 1. The condition

<table>
<thead>
<tr>
<th>Problem 6</th>
<th>Problem 7</th>
<th>Problem 8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Steps</td>
<td>Residual</td>
<td>Steps</td>
</tr>
<tr>
<td>CR</td>
<td>12</td>
<td>6.7e-16</td>
</tr>
<tr>
<td>IS</td>
<td>16</td>
<td>3.1e-10</td>
</tr>
<tr>
<td>ChI</td>
<td>12</td>
<td>6.3e-15</td>
</tr>
</tbody>
</table>
number of $\hat{H}$ and of $K$ is $4.2 \times 10^{23}$ and $1.8 \times 10^{16}$, respectively. The IS is not able to approximate the solution at all, ChI is not able to reach a really small residual error, and CR still works very well, providing a very accurate result.

7. Conclusions

We considered the problem of solving polynomial matrix equations in terms of matrix polynomial computations. We used Möbius’ mapping of the complex plane and pointed out that it sets a relationship between Joukowski’s and the square functions. This allowed us to relate different existing algorithms to each others and to devise new ones, each algorithm having its own version in the domain of the unit circle and in the domain of the imaginary axis.

In particular, we have seen that the IS algorithm is the matrix version of Cardinal’s method which acts in the domain of the imaginary axis while its formulation in the domain of the unit circle corresponds to Sebastião e Silva’s algorithm. We have also introduced Chebyshev’s iterations for matrix polynomials which correspond, in the domain of the imaginary axis, to the LR and to the CR algorithms.

By comparing CR, Chebyshev’s iteration and IS algorithm we have shown that for problems originated from queueing theory, CR and its new version in the domain of the imaginary axis have better convergence performances and lower cost.

Finally, we pointed out that CR (and LR) have better numerical properties than their counterpart (ChI) in the domain of the imaginary axis. This points to the importance of remaining in the domain of the unit circle where we deal with probabilities: quantities which are positive and less than 1.

Appendix A. Chebyshev’s LR variant

If we apply the transformation (2) to (16) and define $s_i(t) = (1 + t)^2 r_i(w(t)) = S_{0,i} + S_{1,i} t + S_{2,i} t^2$, we obtain

$$E_{1,i} = (r_i(x) - r_i(-x))/(2x) = (S_{2,i} - S_{0,i})/2,$$

where $r_i(x) = E_{0,i} + E_{1,i} x + E_{2,i} x^2$ are defined in (16). Moreover, we have

$$s_{i+1}(J(t)) = \tilde{s}_i(t) \tilde{s}_i(t^{-1}),$$

$$\tilde{s}_i(t) = H_i s_i(t),$$

$$H_i = 2(S_{2,i} - S_{0,i})^{-1},$$

where the initial matrix polynomial is

$$s_0(t) = (A_0 - A_1 + A_2) + 2(A_0 - A_2) t + (A_0 + A_1 + A_2) t^2.$$
\[ = S_0 + S_1 t + S_2 t^2. \]

The coefficients of the matrix polynomials \( s_i(t) \) are related as follows:

\[
S_{0,i}^{i+1} = -(4I + (H_i S_{2,i})^2), \\
S_{1,i}^{i+1} = -2H_i (S_{1,i} H_i S_{0,i} + S_{2,i} H_i S_{1,i}), \\
S_{2,i}^{i+1} = -4H_i S_{2,i} H_i S_{0,i}.
\]  

(A.2)

Theorem 1 provides approximations to \( G \) with \( D_{0,i} = H_i (S_{0,i} + S_{1,i} + S_{2,i}) \), \( D_{2,i} = H_i (S_{0,i} - S_{1,i} + S_{2,i}) \).

References


[34] A. Schönhage, The fundamental theorem of algebra in terms of computational complexity, Math. Dept., University of Tübingen, Tübingen, Germany, 1982.