Quadratically convergent algorithms for solving matrix polynomial equations

Dario A. Bini, Guy Latouche and Beatrice Meini

September 21, 2000

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 many applications, in particular 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. This allows us to compare algorithms like Logarithmic Reduction and Cyclic Reduction, which extend Graeffe's iteration to matrix polynomials, and Matrix Sign Function iterations, which extend Cardinal's algorithm. We devise new iterative techniques having quadratic convergence and present numerical experiments.

1 Introduction

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

$$\sum_{i=0}^{n} A_i X^i = 0. \tag{1}$$

It is encountered in many applicative fields, in particular in the analysis of queueing problems modeled by Markov chains of M/G/1 type [25, 24, 20, 30] in which case $I + A_1 \geq 0$, $A_i \geq 0$ for $i \neq 1$, and $\sum_{i=0}^{n} A_i + I$ is an irreducible stochastic matrix. 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.

In Markov chains applications, one is interested in the minimal nonnegative solution of (1), and different techniques have been introduced in the literature to calculate that minimal solution, usually denoted by G. Algorithms based on functional iterations, having linear convergence, are analyzed in [17, 29, 22, 11, 10]. A few methods with quadratic convergence have also been proposed: the Logarithmic Reduction (LR) technique is defined in [19] for the special case $n = 2$ and a similar iteration, the Cyclic Reduction (CR) method, is extended in [3, 4] to arbitrary values for $n \leq \infty$. The Matrix Sign Function (MSF) algorithm is introduced in [1] for general finite values of n. Finally, we also mention the doubling technique proposed in [21] and Newton's scheme analyzed in [18] — these, however, are less efficient 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 [15].

We pursue here the analysis and critical comparison between CR and MSF performed in [23]. 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 MSF algorithm on the other hand. More precisely, we show that the first two methods are particular extensions of Graeffe's iteration [26, 16] to matrix polynomials and that their quadratic convergence is due to the *implicit* use of the square function $S(z) = z^2$. The MSF algorithm, for its part, coincides with Cardinal's algorithm [9, 8], applied to a suitable Frobenius matrix and its convergence is due to the *explicit* use of 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 Sebastiao e Sylva) and relate them by means of Möbius' mapping.

In Section 4 we show how the LR, CR and MSF iterative procedures are based on generalizations to matrix polynomials of Graeffe's and Cardinal's procedures analyzed 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 MSF procedure. We give some numerical illustration in Section 6 and argue in Section 7 that the Cyclic Reduction 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

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 is characterized by the function

$$z(x) = (1+x)/(1-x), \tag{2}$$

of a complex variable, defined for $x \neq 1$. It maps the unit circle without the point 1 into the imaginary axis and the imaginary axis into the unit circle without the point -1. Moreover, it maps $\mathcal{D}$ into $\mathbb{C}^+$ and the complement of the closed unit disk into $\mathbb{C}^-$. Also, it maps $\mathbb{C}^-$ into $\mathcal{D}$ and $\mathbb{C}^+$ into the complement of the closed unit disk.

The inverse map

$$w(t) = (t-1)/(t+1) \tag{3}$$

defined for $t \neq -1$ has a very similar behavior: it maps the unit circle, without the point -1, into the imaginary axis and the imaginary axis into the unit circle without the point 1; the open unit disk $\mathcal{D}$ is mapped into $\mathbb{C}^-$, while the complement of the closed unit disk is mapped into $\mathbb{C}^+$; $\mathbb{C}^+$ is mapped into $\mathcal{D}$ and $\mathbb{C}^-$ into the complement of the closed unit disk.

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

Remark 1 The mappings z and w are so similar that we might choose the mapping $w(x)$ for changing coordinates from the unit circle to the imaginary axis as it is done in the paper [1]. However, we prefer to use $z(x)$ since this leads to simpler expressions while leaving unchanged the substance of the matter. We return to this comment and motivate our choice in Remarks 2, 3 and 4 of later sections.

Mappings of roots of polynomials The two mappings 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 $\Pi_n = \{p(x) = \sum_{i=0}^n p_i x^i,\ p_i \in \mathbb{C}\}$ of polynomials of degree at most n having complex coefficients, by the following equation

$$\mathcal{P}(p(x)) \equiv q(t) = (1+t)^n p(w(t)). \qquad (4)$$

If $p(x)$ has degree n and $p(x) = \prod_{i=1}^n (x - \xi_i)$, then

$$q(t) = \gamma \prod_{\substack{1 \le i \le n, \\ \xi_i \ne 1}} (t - \eta_i)$$

where $\eta_i = z(\xi_i)$, $\gamma = (-2)^k \prod_{\xi_i \ne 1}(1 - \xi_i)$, and k is the number of the zeros equal to 1. If, on the other hand, $p(x)$ has degree $h < n$ and $p(x) = \prod_{i=1}^h (x - \xi_i)$, then

$$q(t) = \gamma (t+1)^{n-h} \prod_{\substack{1 \le i \le h \\ \xi_i \ne 1}} (t - \eta_i).$$

Thus, if $p(x)$ has k roots in $\mathcal{D}$ and $n-k$ roots in the complement of the closed unit disk, then $q(t)$ has k roots in $\mathbb{C}^+$ and $n-k$ roots in $\mathbb{C}^-$.

Even though in principle there is no difference in the nature of the polynomials $p(x)$ and $q(t)$ in Π_n, we refer to Π_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.

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 in three subsets: the unit circle, its interior $\mathcal{D}$ and its exterior. Sequences obtained from the recursion $x_{i+1} = S(x_i)$ quadratically converge to zero if $|x_0| < 1$ and to ∞ 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 $\mathbb{C}^+$, $\mathbb{C}^-$ and the imaginary axis: sequences obtained from the recursions $t_{i+1} = J(t_i)$ converge to 1 if $t_0 \in \mathbb{C}^+$, to -1 if $t_0 \in \mathbb{C}^-$ and fail to converge if t_0 is on the imaginary axis.

We say that Joukowski's function has its natural environment in the *domain of the imaginary axis* while that of the square function is in the *domain of the unit circle*. Observe that the fixed points 0 and ∞ of $S(\cdot)$ are mapped by $z(x)$ onto 1 and -1 respectively, and that the unit circle and the imaginary axis, where we have no convergence of the sequences obtained by means of the square function and Joukowski's function respectively, are mapped onto one another.

One proves by direct inspection the following properties:

$$w(-t) = \frac{1}{w(t)}, \qquad z(-x) = \frac{1}{z(x)}, \tag{5}$$

$$w(t^{-1}) = -w(t), \qquad z(x^{-1}) = -z(x), \tag{6}$$

$$w(J(t)) = S(w(t)), \qquad z(S(x)) = J(z(x)), \tag{7}$$

$$w(-S(t)) = J(w(t)), \qquad z(J(x)) = -S(z(x)). \tag{8}$$

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. In particular, we see from (5) that the function $f(x) = -x$ is associated to $g(t) = t^{-1}$ and that the function $f(x) = x^{-1}$ is associated to $g(t) = -t$. We see from (8) that the function $S(x)$ is associated to $J(t)$ and $J(x)$ is associated to $-S(t)$.

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.

Remark 2 If we interchange the roles of $z(x)$ and $w(t)$, and use, like in [1], $w(x)$ as the function that maps the coordinates of the unit circle onto the coordinates of the imaginary axis, then the transformation $f(x) \to g(t)$ such that $g(t) = w(f(z(t)))$, maps $S(x)$ into $1/J(t)$ and $J(x)$ into $S(t)$.

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 [12, 13]. Define $p(x) = \sum_{i=0}^{n} A_i x^i$ and denote by $\xi_1, \xi_2, \cdots$ the roots of $\det(p(x))$, ordered in increasing value of their modulus. The eigenvalues of G are the m smallest roots $\xi_1, \ldots, \xi_m$, the eigenvalue ξ_m with largest modulus is real, simple, and $\xi_m \le 1$, furthermore, $|\xi_{m+1}|$ is strictly greater than ξ_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.

The approach followed in [1] consists in applying Möbius' map (2), thus replacing (1) with the equation

$$\sum_{i=0}^{n} A_i(I-W)^{n-i}(I+W)^i = \sum_{i=0}^{n} U_i W^i = 0. \tag{9}$$

The solution W having eigenvalues in $\mathbb{C}^-$ provides the solution G by means of the formula $G = (I-W)^{-1}(I+W)$. 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.

Sebastiao e Sylva and Cardinal The algorithms of Sebastiao e Sylva [32] and of Cardinal [9] are two such applications. The first one uses the square function, the second uses Joukowski's function.

Given a polynomial $p(x)$, Sebastiao e Sylva's procedure is based on the sequence $\phi_{i+1}(x) = \phi_i^2 \bmod p(x)$, 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 $\psi_{i+1}(x) = \frac{1}{2}(\psi_i(x) + \psi_i(x)^{-1}) \bmod p(x)$, 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, Sebastiao e Sylva's algorithm corresponds to applying the power method to the Frobenius matrix

$$F = \begin{pmatrix} 0 & 1 & & \bigcirc \\ & \ddots & \ddots & \\ & & 0 & 1 \\ -p_0 & -p_1 & \dots & -p_{n-1} \end{pmatrix} \tag{10}$$

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 (8) that Cardinal's method may be interpreted as resulting from the application of Möbius' map to the method of Sebastiao e Sylva. In fact, we have that ϕ_i is associated to ψ_i by the mapping defined in Section 2, and

$$\psi_i(x) = z(\phi_i(x)) \bmod p(x), \qquad \text{for all } i, \tag{11}$$

if we choose the initial polynomials $\phi_0(x)$ and $\psi_0(x)$ such that $\psi_0(x) = z(\phi_0(x)) \bmod p(x)$. To see this, assume that (11) holds for some $i \geq 0$; we have

$$\begin{aligned}
\psi_{i+1}(x) &= J(\psi_i(x)) \bmod p(x) && \text{by definition} \\
&= J(z(\phi_i(x)) \bmod p(x) && \text{by induction assumption} \\
&= z(S(\phi_i(x))) \bmod p(x) && \text{by (7)} \\
&= z(\phi_{i+1}(x)) \bmod p(x) && \text{by definition}
\end{aligned}$$

which proves that (11) 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 $\psi_i(\eta_j) = J \circ J \circ \cdots \circ J(\psi_0(\eta_j))$, where the functional composition $\circ$ is applied i times.

The algorithms can be implemented in the polynomial setting with fast algorithms for polynomial computation [7] 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.

Thus, it might seem preferable to apply Sebastiao e Sylva'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 $F_i = F^{2^i}$ 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) and (3) provides algorithmic improvements for solving certain polynomial and matrix computations.

Remark 3 If we use $w(x)$ to map the coordinates of the unit circle onto the coordinates of the imaginary axis, then we are lead through (8) to the following relation between Sebastiao e Sylva's and Cardinal's algorithm: $\psi_i(x) = (-1)^i w(\phi_i(x)) \mathrm{mod}\ p(x)$ instead of (11).

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] [26, 16] where a sequence of polynomials is defined as follows:

$$p_{i+1}(x^2) = (-1)^n p_i(x) p_i(-x), \quad p_0(x) = p(x). \tag{12}$$

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 [8, 27, 28, 31] 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

$$u(x^2) = (-1)^n v(x) v(-x) \tag{13}$$

where $u(x)$ and $v(x)$ are polynomials of degree n. If we replace x with $w(t)$ of (3), multiply both sides of the above equation by $(1+t)^n(1+t^{-1})^n$ and apply the transformation (4), we find from (6) 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 (7), 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

$$U(J(t)) = (-2)^{-n} V(t) V(1/t). \tag{14}$$

By applying the argument above to (12), we obtain the iteration

$$q_{i+1}(J(t)) = (-2)^{-n} q_i(t) q_i(t^{-1}) \tag{15}$$

[1] first discovered by Dandelin and Lobachevski [26]

and the polynomials in that sequence are such that

$$q_i(t) = \mathcal{P}(p_i(x)) \qquad \text{for all } i.$$

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 $\{\mu_j^{(i)}\}_i$ converges doubly exponentially to 1 or to -1 according to the sign of the real part of μ_j.

The iteration (15), introduced in [8] 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 $\cos \frac{\pi i}{2n}$, $i = 1, \ldots, n$. See [8] 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 (9). Before doing so, we recall in the next section two classical technique for solving (1).

Remark 4 If we had used $w(x)$ instead of $z(x)$ to map from the unit circle to the imaginary axis, by replacing $\mathcal{P}$ of (4) with $\mathcal{P}(p(x)) = (1-t)^n p(z(t))$, we would have obtained the following equation

$$U_R(J(t)) = 2^{-n} V(t) V(1/t).$$

in place of (14), where $U_R(t) = t^n U(t^{-1})$ is the polynomial obtained by *reversing* the coefficients of $U(t)$. In this case the iteration (15) would have been replaced by the more complex

$$\begin{aligned} r_{i+1}(J(t)) &= 2^{-n} q_i(t) q_i(t^{-1}) \\ q_{i+1}(t) &= t^n r_{i+1}(t^{-1}) \end{aligned}$$

The iteration above has the same convergence properties and the same computational cost as (15).

4 Past Algorithms Revisited

Cyclic Reduction and Logarithmic Reduction 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 matrix sign iteration 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

$$A_0 + A_1 X + A_2 X^2 = 0. \tag{16}$$

The minimal solution G of (16) 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 $|\xi_1| \le \cdots \le |\xi_{2m}|$, $\xi_m = 1 < |\xi_{m+1}|$ and ξ_m is simple (here we assume zeros at infinity if A_2 is singular).

Logarithmic Reduction We express in polynomial form the algorithm LR of [19] for the solution of (16). Let $r_i(x)$ be matrix polynomials recursively defined by

$$\begin{array}{l} r_i(z) = E_{0,i} + E_{1,i} x + E_{2,i} x^2 \\ r_{i+1}(x^2) = -\widehat{r}_i(x)\widehat{r}_i(-x), \\ \widehat{r}_i(x) = E_{1,i}^{-1} r_i(x), \end{array} \tag{17}$$

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

$$\begin{array}{rcl} E_{0,i+1} & = & -(E_{1,i}^{-1} E_{0,i})^2, \\ E_{2,i+1} & = & -(E_{1,i}^{-1} E_{2,i})^2, \\ E_{1,i+1} & = & I - (E_{1,i}^{-1} E_{0,i})(E_{1,i}^{-1} E_{2,i}) - (E_{1,i}^{-1} E_{2,i})(E_{1,i}^{-1} E_{0,i}) \end{array} \tag{18}$$

and the following result is proved in [19].

Theorem 1 *The minimal solution G of (16) is such that $G = G_i + O(\sigma^{2^i})$ with $\sigma = |\xi_m|/|\xi_{m+1}| < 1$, where $G_i = \sum_{j=0}^{i} \left(\prod_{k=0}^{j-1} D_{2,k} \right) D_{0,k}$, $D_{j,k} = -E_{1,i}^{-1} E_{j,i}$, $j = 0, 2$.*

Observe that the computation of (18) can be performed with one matrix inversion and 6 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 [14]. 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 $2m$ 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.

Cyclic Reduction Define the polynomials $p_i(x) = A_{0,i} + A_{1,i}x + A_{2,i}x^2$ by

$$p_{i+1}(x^2) = -p_i(x)A_{1,i}^{-1}p_i(-x), \tag{19}$$

with $p_0(x) = A_0 + A_1x + A_2x^2$. The coefficient matrices $A_{j,i+1}$, themselves are iteratively defined by

$$\begin{aligned} A_{1,i+1} &= A_{1,i} - A_{0,i}A_{1,i}^{-1}A_{2,i} - A_{2,i}A_{1,i}^{-1}A_{0,i}, \\ A_{0,i+1} &= -A_{0,i}A_{1,i}^{-1}A_{0,i}, \\ A_{2,i+1} &= -A_{2,i}A_{1,i}^{-1}A_{2,i} \end{aligned} \tag{20}$$

and the following property is proved in [4]:

Theorem 2 *The minimal solution G of (16) is such that $G = G_i + O(\sigma^{2^i})$, with $\sigma = |\xi_m|/|\xi_{m+1}| < 1$, where $G_i = -\widehat{A}_i^{-1}A_0$ and*

$$\widehat{A}_{i+1} = \widehat{A}_i - A_{2,i}A_{1,i}^{-1}A_{0,i}, \tag{21}$$

for $i \geq 0$, with $\widehat{A}_0 = A_1$.

The computational cost per iteration is one matrix inversion and 6 matrix multiplications, including the cost of updating $\widehat{A}_i$. 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 (18, 20) generated by LR and CR are such that $E_{j,i} = A_{1,i-1}^{-1}A_{j,i}$, for $j = 0, 1, 2$ and all $i \geq 1$.*

Matrix sign function In the QBD case, (9) becomes

$$U_0 + U_1W + U_2W^2 = 0, \tag{22}$$

obtained by replacing X in (16) with $X = (I-W)^{-1}(I+W)$. 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

$$\begin{aligned} U_0 &= A_0 + A_1 + A_2 \\ U_1 &= 2A_2 - 2A_0 \\ U_2 &= A_0 - A_1 + A_2 \end{aligned}$$

and U_2 is nonsingular. If $\xi_m = 1$ and $|\xi_{m-1}| < 1$, then the solution W of (22) which corresponds to the minimal solution G of (16), has only one null eigenvalue. 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 η_i, $i = 1, \ldots, 2m$, of $\det(U_0 + U_1 t + U_2 t^2)$, 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 $\boldsymbol{x}$ and $\boldsymbol{y}$ are such that $U_0\boldsymbol{x} = 0$, $\boldsymbol{y}^T U_0 = 0$, $\boldsymbol{y}^T U_1 \boldsymbol{x} = 1$, then the matrix

$$\widehat{H} = H - \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{0} \end{bmatrix} \begin{bmatrix} \boldsymbol{y}^T U_1 & \boldsymbol{y}^T U_2 \end{bmatrix}$$

has m eigenvalues in $\mathbb{C}^-$ and m eigenvalues in $\mathbb{C}^+$. Therefore, the MSF iteration can be applied to $\widehat{H}$ and yields the sequence

$$\widehat{H}_{i+1} = \frac{1}{2}(\widehat{H}_i + \widehat{H}_i^{-1}), \tag{23}$$

which converges to $\lim_i \widehat{H}_i = K$. That limit may be expressed as $K = TST^{-1}$, where $S = \mathrm{diag}\,(s_1, \ldots, s_{2m})$, $s_i = \mathrm{sign}\,(\mathrm{Re}\,(\eta_i))$, and $J = T^{-1}\widehat{H}T$ is the Jordan normal form of $\widehat{H}$. 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 (22) having eigenvalues in $\mathbb{C}^-$, and $G = (I+W)(I-W)^{-1} = -K_1^*(K_2^*)^{-1}$ is the minimal nonnegative solution of (16).

The cost of (23) amounts to inverting a $(2m) \times (2m)$ matrix, that is about $16m^3$ ops, to be compared to $50/3m^3$ for LR and to $38/3m^3$ for CR. For the final computation of G, we have to compute a QR factorization of $\widehat{H}_i - I$, 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 $\widehat{H}_i$ to K depends on how fast the values $J^{(i)}(\eta_j)$ converge to ± 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 $w_{i+1} \pm 1 = \frac{(w_i \pm 1)^2}{2w_i}$, 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, to -1 or to 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 MSF in general.

Moreover, if some value η_j is close to the imaginary axis, the matrices $\widehat{H}_i$ generated by MSF may have a very large condition number and the computation can be affected by large rounding errors [2].

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, the equation (19) must be adjusted in the following way [4, 3]:

$$p_{i+1}(x^2) = -p_i(x)K_i(x)^{-1}p_i(-x) \tag{24}$$

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 $\widehat{A}_i$ of Theorem 2 are replaced by the matrix power series $\widehat{p}_i(x)$ defined below (for more details we refer the reader to [4, 3])

$$\widehat{p}_{i+1}(x) = \widehat{p}_i^{(odd)}(x) - p_i^{(even)}(x)K_i(x)^{-1}\widehat{p}_i^{(even)}(x) \tag{25}$$

where $\widehat{p}_i(x) = \sum_{j=0}^{+\infty} \widehat{A}_{j+1,i} x^j$, $\widehat{p}_0(x) = \sum_{j=0}^{+\infty} A_{j+1} x^j$, $p_i^{(even)}(x) = \sum_{j=0}^{+\infty} A_{2j,i} x^j$, $\widehat{p}_i^{(even)}(x) = \sum_{j=0}^{+\infty} \widehat{A}_{2(j+1),i} x^j$, $\widehat{p}_i^{(odd)}(x) = \sum_{j=0}^{+\infty} \widehat{A}_{2j+1,i} x^j$.

The matrix power series in the sequences $\{p_i(x)\}$ and $\{\widehat{p}_i(x)\}$ converge in the unit disk, and for this reason can be easily approximated with polynomials of suitable degree n_i (we call these the numerical degrees of the power series). Moreover,

$$A_0 + \sum_{j=0}^{\infty} \widehat{A}_{j+1,i} G^{1+j2^i} = 0 \tag{26}$$

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 $\sum_{i=1}^{+\infty} iA_i < +\infty$ and that the dominating left eigenvector $\boldsymbol{b}$ of B, normalized by $\boldsymbol{b}^T\boldsymbol{e} = 1$, where $\boldsymbol{e} = (1, \ldots, 1)^T$, satisfies $\boldsymbol{b}^T \sum_{i=1}^{+\infty} iA_i\boldsymbol{e} < 1$. Then the equation (1) has one nonnegative solution G with spectral radius 1 and $G = -(\widehat{A}_{1,i})^{-1}A_0 + O(\sigma^{-2^i})$, where $\widehat{A}_{1,i} = \widehat{p}_i(0)$ 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 $\boldsymbol{\pi}_j \in \mathbb{R}^m$, $j = 0, 1, 2, \ldots$, such that $\boldsymbol{\pi}_0^T(A_0 + A_1) + \boldsymbol{\pi}_1^T A_0 = 0$, $\sum_{j=0}^{i} \boldsymbol{\pi}_j^T A_{i-j} = 0$, $i = 2, 3, \ldots$, $\sum_{j=0}^{\infty} \boldsymbol{\pi}_j^T \boldsymbol{e} = 1$ (see [25]), and $\boldsymbol{\pi}_j = O(\sigma^{-j})$ (see [13]). By following the same argument as in the proof of [4, Theorem 4.1], we find that $\boldsymbol{\pi}_0^T \widehat{A}_{j,i} + \boldsymbol{v}_{j,i} = \boldsymbol{\pi}_{(j-1)2^i}$ for $j \geq 2$, where $\boldsymbol{v}_{j,i} = \sum_{k=1}^{j} \boldsymbol{\pi}_{k2^i}^T A_{j-k,i} + \boldsymbol{\pi}_{(j-1)2^i}$. Since the vector $\boldsymbol{v}_{j,i}$ is nonnegative (see [4]), $\boldsymbol{\pi}_0$ is positive and $\widehat{A}_{j,i}$ is nonnegative for $j \geq 2$, then we deduce that $\widehat{A}_{j,i} = O(\sigma^{-(j-1)2^i})$, for $j = 2, 3, \ldots$. The thesis follows from (26). □

A consequence of this theorem is that the numerical degrees 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 $O(Nm^3 + Nm^2 \log m)$ ops; in practice, the values of 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 [5] 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 to deal with the general form (25) of the iterative procedure by considering the "re-blocked" matrix equation

$$\mathcal{A}_0 + \mathcal{A}_1\mathcal{X} + \mathcal{A}_2\mathcal{X}^2 = 0 \tag{27}$$

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 (27) the CR algorithm (20) or the LR algorithm (18); by following the same technique as in [6], 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, that is, 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 MSF algorithm can be easily applied to the case of 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 $O(m^3 n \log^2 n)$ 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 (15) in the CR fashion. We give all details for matrix polynomials of degree $n = 2$. The case of polynomials of arbitrary degree is not given here as it is dealt with in a similar manner, starting from the general CR iteration (24).

In order to solve (16) in the domain of the unit circle, we apply the operator $\mathcal{P}$ of (4) to the polynomial $p_0(x)$ and solve the equation

$$Q_0 + Q_1 T + Q_2 T^2 = 0 \tag{28}$$

in the domain of the imaginary axis, where

$$\begin{aligned} Q_0 &= A_0 - A_1 + A_2, \\ Q_1 &= 2(A_0 - A_2), \\ Q_2 &= A_0 + A_1 + A_2. \end{aligned}$$

If T is the solution of (28) with all eigenvalues having nonnegative real parts, then $G = (T - I)(T + I)^{-1}$ is the solution of (15) with all eigenvalues in the unit circle.

We define the polynomial sequence

$$q_i(t) = (1+t)^2 p_i(w(t)) \equiv Q_{0,i} + Q_{1,i}t + Q_{2,i}t^2 \tag{29}$$

and, in order to overcome the non-commutativity of matrix multiplication, we need to find matrices K_i with the same role of $-A_{1,i}$ in (19) such that (15) 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

$$\begin{aligned} -A_{1,i} &= \frac{t+1}{2(t-1)}\left(\frac{1}{(t^{-1}+1)^2}q_i(t^{-1}) - \frac{1}{(t+1)^2}q_i(t)\right) \\ &= (Q_{0,i} - Q_{2,i})/2. \end{aligned} \tag{30}$$

Now, apply to (19) the same transformations as to (13): replace x with $w(t)$ and multiply by $(1+t)^2(1+t^{-1})^2$. This leads to

$$q_{i+1}(J(t)) = \frac{1}{2}q_i(t)K_iq_i(t^{-1}) \tag{31}$$

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 the equations (32) below. For the sake of simplicity in proving (32), we temporarily write Q_0, Q_1, Q_2 instead of $Q_{0,i}, Q_{1,i}, Q_{2,i}$ and K instead of K_i.

We define $\sum_{j=-2}^{2} C_j t^j = q_i(t)K_iq_i(t^{-1})$ and verify by direct inspection that

$$\begin{array}{ll} C_1 = Q_1KQ_0 + Q_2KQ_1 & C_{-1} = Q_0KQ_1 + Q_1KQ_2 \\ C_2 = Q_2KQ_0 & C_{-2} = Q_0KQ_2 \\ C_0 = Q_0KQ_0 + Q_1KQ_1 + Q_2KQ_2 & \end{array}$$

We have that $C_1 = C_{-1}$ and $C_2 = C_{-2}$ since $Q_1K(Q_0 - Q_2) = Q_1 = (Q_0 - Q_2)KQ_1$ and $Q_2KQ_0 = Q_0KQ_0 - Q_0 = Q_0KQ_2$. Therefore, we obtain that

$$\begin{aligned} \sum_{j=-2}^{2} C_jt^j &= C_0 + C_1(t+t^{-1}) + C_2(t^2+t^{-2}) \\ &= (C_0 - 2C_2) + \frac{t+t^{-1}}{2}2C_1 + \left(\frac{t+t^{-1}}{2}\right)^2(4C_2). \end{aligned}$$

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 polynomials $q_i(t)$ are iteratively determined by the following relations

$$\begin{aligned} Q_{0,i+1} &= \frac{1}{2}(Q_{0,i} - Q_{2,i} + Q_{1,i} K_i Q_{1,i}), \\ Q_{1,i+1} &= Q_{2,i} K_i Q_{1,i} + Q_{1,i} K_i Q_{0,i}, \\ Q_{2,i+1} &= 2 Q_{2,i} K_i Q_{0,i}. \end{aligned} \tag{32}$$

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 $\widehat{A}_i$ of (21) 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

$$\begin{array}{ll} Q_{0,i} = A_{0,i} - A_{1,i} + A_{2,i}, & A_{0,i} = \frac{1}{4}(Q_{0,i} + Q_{1,i} + Q_{2,i}), \\ Q_{1,i} = 2(A_{0,i} - A_{2,i}), & A_{1,i} = -\frac{1}{2}(Q_{0,i} - Q_{2,i}), \\ Q_{2,i} = A_{0,i} + A_{1,i} + A_{2,i}, & A_{2,i} = \frac{1}{4}(Q_{0,i} - Q_{1,i} + Q_{2,i}). \end{array}$$

Defining $\widehat{Q}_i$ as

$$\widehat{Q}_{i+1} = \widehat{Q}_i + (Q_{0,i} - Q_{1,i} + Q_{2,i}) K_i (Q_{0,i} + Q_{1,i} + Q_{2,i})/8 \tag{33}$$

where $\widehat{Q}_0 = -\frac{1}{2}(Q_0 - Q_2)$, we readily conclude from (21) that $\widehat{Q}_i = \widehat{A}_i$ for all i. Then, the matrix sequence $G_i = -\widehat{A}_i^{-1} A_0$ which quadratically converge to G (by Theorem 2) can be written as

$$G_i = -\frac{1}{4}\widehat{Q}_i^{-1}(Q_0 + Q_1 + Q_2). \tag{34}$$

The computational cost at each step of the iterations (32, 33) amounts to one matrix inversion and 8 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 (22) having eigenvalues with non-positive real parts. Indeed, since $G_i = -\widehat{A}_i^{-1} A_0$ provides an approximation to G, $W_i = (G_i + I)^{-1}(G_i - I)$ provides an approximation to W. From (34), a simple calculation shows that

$$W_i = (-4\widehat{Q}_i + Q_{0,i} + Q_{1,i} + Q_{2,i})^{-1}(4\widehat{Q}_i + Q_{0,i} + Q_{1,i} + Q_{2,i}) \tag{35}$$

and the following is a direct consequence of Theorem 2.

Theorem 5 *Let the equation (22) have a solution W with eigenvalues $\eta_1, \ldots, \eta_m$ having non-positive real parts. Then $W = W_i + O(\delta^{2^i})$, where W_i is defined in (32, 33, 35) 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, MSF and Chebyshev's iteration (henceforth denoted as ChI) for the solution of (16). 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 $\widetilde{G}$ is an approximate solution of (16) we define $||A_0 + A_1\widetilde{G} + A_2\widetilde{G}^2||_\infty$ the residual error, where $||\cdot||_\infty$ denotes the infinity norm.

Problem 1 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, $M_1 = \mathrm{diag}\,(\gamma_1, \ldots, \gamma_m)$, $\gamma_i = -\alpha_i - \beta_i$, $i = 1, \ldots, m-1$, $\gamma_m = -\alpha_m - \beta_m - \sigma$. 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 $\xi_{m-1} = 0.998975$, $\xi_m = 1$ and $\xi_{m+1} = 1.00978$. In this, case, since $|\xi_m/\xi_{m+1}| < |\xi_{m-1}/\xi_m|$, CR performs better than MSF, as seen in Table 1. Moreover that condition number of $\widehat{H}$ and of K is $1.2 \cdot 10^{10}$ and $6.4 \cdot 10^6$, respectively, thus leading to instability problems for MSF, and to its much larger residual error. The algorithm ChI has the same behavior as CR, as expected.

Problem 2 This example is taken from [20, Page 208]. It represents an M/M/1 queue 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 $M_2 = \rho{\cdot}\mathrm{diag}\,(\alpha_1, \ldots, \alpha_m)$ $M_0 = \mathrm{diag}\,(\beta_1, \ldots, \beta_m)$ $(M_1)_{i,j} = 1$ if $j = (i \bmod m)+1$, $(M_1)_{i,j} = -1 - \rho\alpha_i - \beta_i$ if $j = i$, $(M_1)_{i,j} = 0$ elsewhere, The numerical values of the parameters are $\boldsymbol{\alpha} = (0.2, 0.2, 0.2, 0.2, 13, 1, 1, 0.2)$, $\beta_i = 2$, 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 MSF. The condition number of $\widehat{H}$ and of K is $5.9 \cdot 10^8$ and $2.2 \cdot 10^6$, respectively.

	Problem 1		Problem 2		Problem 3	
	steps	residual	steps	residual	steps	residual
CR	12	6.7e-16	14	4.4e-16	29	6.7e-16
MSF	16	3.1e-10	19	2.3e-12	85	1.0e-1
ChI	12	6.3e-15	14	3.4e-16	18	4.9e-12

Table 1: Number of iterations and residual error

Problem 3 This example is taken from [23]. The matrices A_0, A_1, A_2 have all the entries equal to $\alpha = \frac{\rho-1}{3(m-1)}$ for $i \neq j$, $(A_0)_{i,i} = -\rho$, $(A_1)_{i,i} = 1$, $(A_2)_{i,i} = 0$, 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 ξ_m/ξ_{m+1} is very close to 1. The condition number of $\widehat{H}$ and of K is $4.2 \cdot 10^{23}$ and $1.8 \cdot 10^{16}$, respectively. The MSF 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 polynomial computations. We introduced a 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 Akar and Sohrabi's 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 Sebastiao e Sylva's algorithm. We have also introduced Chebyshev's iterations for matrix polynomials which correspond, in the domain of the imaginary axis, to the Logarithmic Reduction and to the Cyclic Reduction algorithms.

By comparing Cyclic Reduction, Chebyshev's iteration and Akar and Sohrabi's method we have shown that for problems originated from queueing theory, Cyclic Reduction and its new version in the domain of the imaginary axis have better convergence performances and lower cost.

In fine, we pointed out that Cyclic Reduction (and Logarithmic Reduction) 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.

A Chebyshev's LR variant

If we apply the transformation (2) to (17) 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 (17). Moreover, we have

$$\begin{aligned} &s_{i+1}(J(t)) = -\widehat{s}_i(t)\widehat{s}_i(t^{-1}) \\ &\widehat{s}_i(t) = H_i s_i(t) \\ &H_i = 2(S_{2,i} - S_{0,i})^{-1} \end{aligned} \tag{36}$$

where the initial 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,

$$\begin{aligned} S_{0,i+1} &= -(4I + (H_i S_{2,i})^2) \\ S_{1,i+1} &= -2H_i(S_{1,i}H_i S_{0,i} + S_{2,i}H_i S_{1,i}) \\ S_{2,i+1} &= -4H_i S_{2,i} H_i S_{0,i}. \end{aligned} \tag{37}$$

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

[1] N. Akar and K. Sohraby. An invariant subspace approach in M/G/1 and G/M/1 type Markov chains. *Commun. Statist. Stochastic Models*, 13:381–416, 1997.

[2] Z. Bai and J. Demmel. Using the matrix sign function to compute invariant subspaces. *SIAM J. Matrix Anal. Appl.*, 19(1):205–225 (electronic), 1998.

[3] D. A. Bini and B. Meini. On cyclic reduction applied to a class of Toeplitz-like matrices arising in queueing problems. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 21–38. Kluwer Academic Publisher, 1995.

[4] D. A. Bini and B. Meini. On the solution of a nonlinear matrix equation arising in queueing problems. *SIAM J. Matrix Anal. Appl.*, 17:906–926, 1996.

[5] D. A. Bini and B. Meini. Improved cyclic reduction for solving queueing problems. *Numerical Algorithms*, 15:57–74, 1997.

[6] D. A. Bini and B. Meini. Using displacement structure for solving non-skip-free M/G/1 type Markov chains. In A. Alfa and S. Chakravarthy, editors, *Advances in Matrix Analytic Methods for Stochastic Models - Proceedings of the 2nd international conference on matrix analytic methods*, pages 17–37. Notable Publications Inc, NJ, 1998.

[7] D. A. Bini and V. Pan. *Matrix and Polynomial Computations, Vol. 1: Fundamental Algorithms*. Birkhäuser, Boston, 1994.

[8] D. A. Bini and V. Y. Pan. Graeffe's, Chebyshev-like, and Cardinal's processes for splitting a polynomial into factors. *J. Complexity*, 12:492–511, 1996.

[9] Jean-Paul Cardinal. On two iterative methods for approximating the roots of a polynomial. In *The mathematics of numerical analysis (Park City, UT, 1995)*, pages 165–188. Amer. Math. Soc., Providence, RI, 1996.

[10] P. Favati and B. Meini. On functional iteration methods for solving M/G/1 type Markov chains. In A. Alfa and S. Chakravarthy, editors, *Advances in Matrix Analytic Methods for Stochastic Models - Proceedings of the 2nd international conference on matrix analytic methods*, pages 39–54. Notable Publications Inc, NJ, 1998.

[11] P. Favati and B. Meini. On functional iteration methods for solving nonlinear matrix equations arising in queueing problems. *IMA J. of Numerical Analysis*, 19:39–49, 1999.

[12] H. R. Gail, S. L. Hantler, and B. A. Taylor. Spectral analysis of $M/G/1$ and $G/M/1$ type Markov chains. *Adv. in Appl. Probab.*, 28:114–165, 1996.

[13] H. R. Gail, S. L. Hantler, and B. A. Taylor. Use of characteristics roots for solving infinite state Markov chains. In W. K. Grassmann, editor, *Computational Probability*, pages 205–255. Kluwer Academic Publishers, 2000.

[14] W. K. Grassman, M. I. Taksar, and D. P. Heyman. Regenerative analysis and steady state distribution for Markov chains. *Oper. Res.*, 33:1107–1116, 1985.

[15] N. J. Higham and H.-M. Kim. Numerical analysis of a quadratic matrix equation. Numerical Analysis Report No. 347, Manchester Centre for Computational Mathematics, Manchester, England, August 1999. To appear in IMA J. of Numerical Analysis.

[16] A. S. Householder. *The Numerical Treatment of a Single Nonlinear Equation*. International Series in Pure and Applied Mathematics. McGraw-Hill Inc., New York, NY, 1970.

[17] G. Latouche. Algorithms for infinite Markov chains with repeating columns. In C.D. Meyer and R.J. Plemmons, editors, *Linear Algebra, Queueing Models and Markov Chains*, pages 231–265. Springer-Verlag, New York, 1993.

[18] G. Latouche. Newton's iteration for non-linear equations in Markov chains. *IMA J. of Numerical Analysis*, 14:583–598, 1994.

[19] G. Latouche and V. Ramaswami. A logarithmic reduction algorithm for Quasi-Birth-Death processes. *J. Appl. Probability*, 30:650–674, 1993.

[20] G. Latouche and V. Ramaswami. *Introduction to Matrix Analytic Methods in Stochastic Modeling*. ASA-SIAM Series on Statistics and Applied Probability 5. SIAM, 1999.

[21] G. Latouche and G.W. Stewart. Numerical methods for M/G/1 type queues. In W. J. Stewart, editor, *Computations with Markov Chains*, pages 571–581. Kluwer Academic Publishers, 1995.

[22] B. Meini. New convergence results on functional iteration techniques for the numerical solution of M/G/1 type Markov chains. *Numer. Math.*, 78:39–58, 1997.

[23] B. Meini. Solving QBD problems: the cyclic reduction algorithm versus the invariant subspace method. *Adv. Perf. Anal.*, 1:215–225, 1998.

[24] M. F. Neuts. *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach*. The Johns Hopkins University Press, Baltimore, MD, 1981.

[25] M. F. Neuts. *Structured Stochastic Matrices of M/G/1 Type and Their Applications*. Marcel Dekk., New York, 1989.

[26] M. A. Ostrowski. Recherches sur la methode de Graeffe et les zeros des polynomes et des series de Laurent. *Acta Math.*, 72:99–257, 1940.

[27] V. Y. Pan. Algebraic complexity of computing polynomial zeros. *Computers & Math. (with Applications)*, 14, 4:285–304, 1987.

[28] V. Y. Pan. Sequential and parallel complexity of approximate evaluation of polynomial zeros. *Computers & Math. (with Applications)*, 14, 8:591–622, 1987.

[29] V. Ramaswami. Nonlinear matrix equations in applied probability - Solution techniques and open problems. *SIAM Review*, 30:256–263, 1988.

[30] V. Ramaswami. A stable recursion for the steady state vector in Markov chains of M/G/1 type. *Commun. Statist. Stochastic Models*, 4:183–188, 1988.

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

[32] J. Sebastiao e Silva. Sur une méthode d'approximation semblable à celle de Gräffe. *Portugaliae Math.*, 2:271–279, 1941.