Numerical Algorithms (2005) **39:** 349–378

Algorithms for the matrix pth root *

Dario A. Bini [a,**], Nicholas J. Higham [b,***] and Beatrice Meini [a]

[a] *Dipartimento di Matematica, Università di Pisa, via Buonarroti 2, 56127Pisa, Italy*
E-mail: {bini;meini}@dm.unipi.it

[b] *Department of Mathematics, University of Manchester, Manchester, M13 9PL, England*
E-mail: higham@ma.man.ac.uk

Received 10 August 2004; accepted 2 November 2004
Communicated by C. Brezinski

New theoretical results are presented about the principal matrix pth root. In particular, we show that the pth root is related to the matrix sign function and to the Wiener–Hopf factorization, and that it can be expressed as an integral over the unit circle. These results are used in the design and analysis of several new algorithms for the numerical computation of the pth root. We also analyze the convergence and numerical stability properties of Newton's method for the inverse pth root. Preliminary computational experiments are presented to compare the methods.

Keywords: matrix pth root, matrix sign function, Wiener–Hopf factorization, Newton's method, Graeffe iteration, cyclic reduction, Laurent polynomial

AMS subject classification: 15A24, 65H10, 65F30

1. Introduction

Let A be a real or complex matrix of order n with no eigenvalues on $\mathbb{R}^-$ (the closed negative real axis), and let p be a positive integer. Then there exists a unique matrix X such that

1. $X^p = A$.
2. The eigenvalues of X lie in the segment $\{z\colon\ -\pi/p < \arg(z) < \pi/p\}$.

We refer to X as the principal pth root of A and write $X = A^{1/p}$. One application of pth roots is in the computation of the matrix logarithm through the relation [10,19]

$$\log A = p \log A^{1/p},$$

* Numerical Analysis Report 454, Manchester Centre for Computational Mathematics, July 2004.

** This work was supported by MIUR, grant number 2002014121.

*** This work was supported by Engineering and Physical Sciences Research Council grant GR/R22612 and by a Royal Society – Wolfson Research Merit Award.

where p is chosen so that $A^{1/p}$ can be well approximated by a polynomial or rational function. Related to the pth root is the matrix sector function $\mathrm{sect}_p(A) = (A^p)^{-1/p}A$ [20], which arises in control theory. The matrix sector function with $p = 2$ is the matrix sign function.

We briefly survey existing methods for computing pth roots. Hoskins and Walton [18] consider the iteration

$$X_{k+1} = \frac{1}{p}\big[(p-1)X_k + AX_k^{1-p}\big], \qquad X_0 = A, \tag{1.1}$$

which is Newton's method for $X^p - A = 0$ simplified using the commutativity relation $AX_k = X_kA$. They concentrate on the case A symmetric positive definite, in which case X_k converges to $A^{1/p}$. However, for more general A the iteration does not generally converge to $A^{1/p}$, as explained by Smith [27]. Moreover, the iteration is numerically unstable unless A is extremely well conditioned, even for symmetric positive definite A [27, section 6].

Benner et al. [1] prove that if the columns of $U = [U_1^*, \dots, U_p^*]^* \in \mathbb{C}^{pn\times n}$ span an invariant subspace of

$$C = \begin{bmatrix} 0 & I & & & \\ & 0 & I & & \\ & & \ddots & \ddots & \\ & & & \ddots & I \\ A & & & & 0 \end{bmatrix} \in \mathbb{C}^{pn\times pn}, \tag{1.2}$$

that is, $CU = UY$ for some nonsingular $Y \in \mathbb{C}^{n\times n}$, and U_1 is nonsingular, then $X = U_2U_1^{-1}$ is a pth root of A. For an appropriate choice of subspace, X is the principal pth root. This result reduces the pth root problem to that of computing an invariant subspace of a matrix of order pn, for which many methods are available. Note that the matrix C is a block companion matrix for the matrix polynomial $\lambda^p I - A$.

Shieh et al. [26] propose an algorithm for computing $A^{1/p}$ that consists of forming the powers $X_k = G^k[I_n\ 0\ \dots\ 0]^{\mathrm{T}}$ and computing $\lim_{k\to\infty} X_k(1:n, :)X_k(n+1:2n, :)^{-1}$, where $G = C + I$ and C is the matrix (1.2). Thus they are essentially using the result of Benner et al. and computing the invariant subspace by the power method. This method clearly has linear convergence.

Tsay et al. [29] propose another method for pth roots, based on "generalized continued fractions" and with a certain block Toeplitz matrix playing a key role. However, this method appears to require $\mathrm{O}(n^5)$ flops and $\mathrm{O}(n^3)$ storage, and, as admitted in [28], it is numerically unstable!

Tsai et al. [28] derive iterations whose convergence rate is a parameter. Their quadratically convergent iteration for the pth root is

$$\begin{aligned} G_{k+1} &= G_k\big[\big(2I + (p-2)G_k\big)\big(I + (p-1)G_k\big)^{-1}\big]^p, \qquad && G_0 = A, \\ R_{k+1} &= R_k\big(2I + (p-2)G_k\big)^{-1}\big(I + (p-1)G_k\big), && R_0 = I, \end{aligned}$$

for which they state $G_k \to I$ and $R_k \to A^{1/p}$. No convergence proofs are given in [28], but a perturbation analysis in the style of [15] is performed to show that the iterations are numerically stable.

The integral expression

$$A^{1/p} = \frac{p \sin(\pi/p)}{\pi} A \int_0^\infty \left(x^p I + A\right)^{-1} \mathrm{d}x \tag{1.3}$$

can be deduced from a standard identity in complex analysis, as noted in [2, example V.1.10; 21, section 5.5.5]. Hasan et al. [13] propose approximating the integral by Gaussian quadrature, though no details are given.

Finally, Smith [27] derives a Schur method that employs a recurrence for computing the pth root of a triangular matrix, and he proves that the algorithm is numerically stable. This Schur method is the benchmark against which other methods should be compared. A MATLAB implementation is available as function `rootm` in the Matrix Computation Toolbox [14].

Here we present new theoretical results and new algorithms for the matrix pth root. The paper is organized in two parts. Sections 2–6 mainly concern theoretical properties, while sections 7–10 deal with algorithmic results.

In section 2 we represent $A^{1/p}$ in terms of the integral of an analytic function along the unit circle in the complex plane and we show that $A^{1/p}$ can be approximated by means of numerical integration at the Fourier points with an error that decreases as r^{2N}, where N is the number of Fourier points and $r < 1$ is a positive number that depends on p and A.

In section 3 we show that $A^{1/p}$ is a multiple of the (2, 1) block of the matrix sign function $\operatorname{sign}(C)$ of the block companion matrix (1.2), where the multiplicative constant is explicitly known.

In section 4 we show that the Wiener–Hopf factorization of the matrix Laurent polynomial $F(z) = z^{-p/2}((1+z)^p A - (1-z)^p I)$ exists and provides the principal pth root of A. In this way, any algorithm for computing the Wiener–Hopf factorization can be applied in order to compute $A^{1/p}$. A key tool for showing this property is the Cayley transform $x \to z = (1-x)/(1+x)$, which maps the imaginary axis into the unit circle in the complex plane and which relates the factorization of the polynomial $x^p I - A$ with respect to the imaginary axis to the factorization of $F(z)$ with respect to the unit circle (Wiener–Hopf factorization).

Another theoretical result, shown in section 5, relates $A^{1/p}$ with the central coefficients $H_0, \ldots, H_{p/2-1}$ of the matrix Laurent series $H(z) = H_0 + \sum_{i=1}^{+\infty}(z^i + z^{-i})H_i$ such that $H(z)F(z) = I$. More precisely, $A^{1/p}$ is expressed as linear combination of $H_0, \ldots, H_{p/2-1}$ with known coefficients. In this way any algorithm for the computation of $H(z)$ provides a means for computing $A^{1/p}$.

In section 6 we look at $A^{1/p}$ as the inverse of the fixed point of the function $(1/p)[(1+p)X - X^{p+1}A]$ that is obtained by formally applying Newton's iteration to the equation $X^{-p} - A = 0$. We derive sufficient conditions for convergence and numerical stability of the iteration.

Concerning the algorithmic part, in section 7 we present a new algorithm for inverting a general $np \times np$ A-circulant matrix with $n \times n$ blocks in $O(n^3 p \log p + n^2 p \log^2 p)$ operations; it relies on a polynomial interpretation of A-circulant matrices. This algorithm can be used in the computation of $A^{1/p}$ by the matrix sign iteration described in section 3, since when this iteration is applied to the block companion matrix C of (1.2) it generates a sequence of A-circulant matrices.

In section 8 we present two algorithms for computing the central coefficients of the inverse of the Laurent polynomial $F(z)$. The first is based on the evaluation/interpolation technique, while the second, which relies on Graeffe's iteration, exploits the commutativity of the coefficients of $F(z)$. Two algorithms for computing the Wiener–Hopf factorization of $F(z)$ are described in section 9. They rely on applying cyclic reduction, and on inverting a matrix Laurent polynomial. Finally, in section 10 we report the results of some preliminary numerical experiments.

Throughout this paper $A \in \mathbb{C}^{n\times n}$ is assumed to have no eigenvalues on $\mathbb{R}^-$, and X denotes the principal pth root, $A^{1/p}$. Also, i denotes the imaginary unit ($\mathrm{i}^2 = -1$) and for any integer N, $\omega_N = \cos(2\pi/N) + \mathrm{i}\sin(2\pi/N)$ denotes an Nth root of unity.

Except when analyzing the Newton iteration, we assume that $p = 2q$, where $q \in \mathbb{N}$ is odd. This assumption guarantees that there are no pure imaginary pth roots of unity, and that there are exactly q roots with positive real part and q roots with negative real part. There is no loss of generality in this assumption. Indeed, if p is odd we may compute the pth root of the matrix A by computing the $2p$th root of A^2; if $p = 2q$ and q is even, we may compute successive square roots of A until the condition $p = 2q$, q odd, is satisfied.

2. Integral representation

In [13] the integral (1.3) is proposed for approximating the matrix pth root X of A by means of Gaussian quadrature on the positive real axis. In this section, we obtain from (1.3) a representation of X based on complex integration around the unit circle.

Define the matrix polynomial

$$\Psi(z) = (1+z)^p A - (1-z)^p I = \sum_{j=0}^{p} z^j \binom{p}{j} \bigl(A + (-1)^{j+1} I\bigr) \tag{2.1}$$

and observe that $\Psi(z)$ is nonsingular for $|z| = 1$. In fact, z is a singular point of $\Psi(z)$, i.e., $\det\Psi(z) = 0$, if and only if $z \neq -1$ and $\mu = ((1-z)/(1+z))^p$ is eigenvalue of A. For $|z| = 1$, the ratio $(1-z)/(1+z)$ is a pure imaginary number so that μ is real negative, since p is even and not a multiple of 4. Therefore, since A has no real negative eigenvalues, $\Psi(z)$ cannot be singular for $|z| = 1$. Another nice feature of the function $\Psi(z)$ is that $z \neq 0$ is a singular point of $\Psi(z)$ if and only if $1/z$ is singular point of $\Psi(z)$.

These properties imply that the function $\Psi(z)$ and its inverse are analytic in the annulus

$$\mathcal{A} = \left\{ z \in \mathbb{C}\colon\ \rho < |z| < \frac{1}{\rho} \right\} \tag{2.2}$$

where $\rho = \max\{|z|\colon \det \Psi(z) = 0,\ |z| < 1\}$. The analyticity of $\Psi(z)^{-1}$ allows construction of an algorithm for approximating X, based on the following result.

Proposition 2.1. The principal pth root X of A can be represented as

$$X = \frac{p \sin(\pi/p)}{\mathrm{i}\pi} A \int_{|z|=1} (1+z)^{p-2} \Psi(z)^{-1}\, \mathrm{d}z. \tag{2.3}$$

Moreover, we have

$$X = \frac{2p \sin(\pi/p)}{N} A \sum_{i=0}^{N-1} \left(A - \left(\frac{1-\omega_N^i}{1+\omega_N^i} \right)^p I \right)^{-1} \frac{\omega_N^i}{(1+\omega_N^i)^2} + \mathrm{O}\big(r^{2N}\big), \tag{2.4}$$

where $\rho < r < 1$.

Proof. Since p is even, we may rewrite (1.3) as

$$X = \frac{p \sin(\pi/p)}{2\pi} A \int_{-\infty}^{+\infty} \big(x^p I + A\big)^{-1}\, \mathrm{d}x. \tag{2.5}$$

Since p is even and not a multiple of 4, we have $\mathrm{i}^p = -1$. Therefore, making the substitution $x = \mathrm{i}(1-z)/(1+z)$ yields (2.3). Since the integrand is analytic in the annulus (2.2), the Euler–Maclaurin formula [11, p. 137] gives (2.4). □

Formula (2.4) provides a tool for approximating X by numerical integration on the unit circle, and the approximation error decreases exponentially with the number of integration points N. Observe that the speed of convergence is related to the thickness of the annulus $\mathcal{A}$.

Algorithm 2.1 (pth root through numerical integration at the roots of unity).
INPUT: The integers p, n and the matrix $A \in \mathbb{C}^{n\times n}$; an algorithm `sqrt` for computing the principal matrix square root; an integer $N_0 > 0$ and a tolerance $\varepsilon > 0$.
OUTPUT: An approximation to the principal pth root X of A.
COMPUTATION:

1. If p is odd set $p = 2p$ and $A = A^2$; if p is a multiple of 4 then repeat $p = p/2$, $A = \mathtt{sqrt}(A)$, until $p/2$ is odd.

2. Set $N = N_0$.

3. Compute

$$X_N = \frac{2p\sin(\pi/p)}{N} A \sum_{i=0}^{N-1} \left(A - \left(\frac{1-\omega_N^i}{1+\omega_N^i} \right)^p I \right)^{-1} \frac{\omega_N^i}{(1+\omega_N^i)^2}.$$

4. If $\|A - X_N^p\| > \varepsilon$ set $N = 2N$ and repeat from step 3. Otherwise output the approximation X_N to X.

3. Reduction to matrix sign computation

We now explore a connection between the principal pth root and the matrix sign function. Consider the matrix

$$C = \begin{bmatrix} 0 & I & & & \\ & 0 & I & & \\ & & \ddots & \ddots & \\ & & & \ddots & I \\ A & & & & 0 \end{bmatrix} \in \mathbb{C}^{pn \times pn}. \tag{3.1}$$

The result of Benner et al. [1] stated in section 1 shows how to recover a pth root of A from an n-dimensional invariant subspace of C. For $p = 2$, the matrix sign of C provides an explicit expression for the principal square root of A [16]:

$$\operatorname{sign}\left(\begin{bmatrix} 0 & I \\ A & 0 \end{bmatrix} \right) = \begin{bmatrix} 0 & A^{-1/2} \\ A^{1/2} & 0 \end{bmatrix}. \tag{3.2}$$

Here, using the structural properties of C, we generalize this relation to the case $p \geqslant 2$, for p even and not a multiple of 4.

The following result is a straightforward extension to block matrices of a well known result concerning α-circulant matrices [7, theorem 5.1].

Proposition 3.1. Let $\Omega = (\omega_p^{ij})_{i,j=0:p-1}$, let $X = A^{1/p}$ and define the block diagonal matrix $D = \operatorname{diag}(I, X, X^2, \dots, X^{p-1})$. Then

$$C = \frac{1}{p} D(\Omega \otimes I) S\bigl(\overline{\Omega} \otimes I\bigr) D^{-1}$$

where $S = \operatorname{diag}(X, \omega_p X, \omega_p^2 X, \dots, \omega_p^{p-1} X)$ and $\otimes$ denotes the Kronecker product.

An immediate consequence of this proposition is that

$$\operatorname{sign}(C) = \frac{1}{p} D(\Omega \otimes I) \operatorname{sign}(S) \bigl(\overline{\Omega} \otimes I\bigr) D^{-1}, \tag{3.3}$$

where

$$\operatorname{sign}(S) = \operatorname{diag}\bigl(\operatorname{sign}(X), \operatorname{sign}(\omega_p X), \dots, \operatorname{sign}\bigl(\omega_p^{p-1} X\bigr)\bigr).$$

We observe that if X is the principal pth root of A, then, for any integer j, $\omega_p^j X$ is a pth root of A whose eigenvalues are the eigenvalues of X multiplied by ω_p^j. Since multiplication by ω_p^j corresponds to rotation through an angle $2\pi j/p$, and since the eigenvalues of X lie in the sector $\{\rho e^{i\theta}\colon -\pi/p < \theta < \pi/p\}$, we deduce that the eigenvalues of $\omega_p^j X$ lie in the sector $\{\rho e^{i\theta}\colon -\pi/p + 2\pi j/p < \theta < \pi/p + 2\pi j/p\}$. In this way, $\omega_p^j X$ has eigenvalues with positive real part for $j = -\lfloor q/2 \rfloor : \lfloor q/2 \rfloor$, and eigenvalues with negative real part for $j = \lfloor q/2 \rfloor + 1 : \lfloor q/2 \rfloor + q$.

Therefore, if $p = 2q$ where q is an odd integer then, we deduce that $\omega_p^i X$ has eigenvalues with positive real parts for $i = -\lfloor q/2 \rfloor : \lfloor q/2 \rfloor$, and eigenvalues with negative real parts for $i = \lfloor q/2 \rfloor + 1 : \lfloor q/2 \rfloor + q$. Hence $\operatorname{sign}(\omega_p^i X) = -I$ for $i = \lfloor q/2 \rfloor + 1 : \lfloor q/2 \rfloor + q$ and $\operatorname{sign}(\omega_p^i X) = I$ for $i = 1 : \lfloor q/2 \rfloor$ and for $i = \lfloor q/2 \rfloor + q + 1 : p - 1$. Therefore, from (3.3) we deduce the following result.

Proposition 3.2. If $p = 2q$ where q is odd, then the first block column of the matrix $\operatorname{sign}(C)$ is given by

$$V = \frac{1}{p}\begin{bmatrix} \gamma_0 I \\ \gamma_1 X \\ \gamma_2 X^2 \\ \vdots \\ \gamma_{p-1} X^{p-1} \end{bmatrix},$$

where $X = A^{1/p}$ and $\gamma_i = \sum_{j=0}^{p-1} \omega_p^{ij} \theta_j$, $i = 0\colon p-1$, and $\theta_j = -1$ for $j = \lfloor q/2 \rfloor + 1 : \lfloor q/2 \rfloor + q$, $\theta_j = 1$ otherwise.

Proof. From (3.3), the first block column of $\operatorname{sign}(C)$ is

$$\frac{1}{p} D(\Omega \otimes I)\begin{bmatrix} \operatorname{sign}(X) \\ \operatorname{sign}(\omega_p X) \\ \vdots \\ \operatorname{sign}(\omega_p^{p-1} X) \end{bmatrix} = \frac{1}{p} D(\Omega \otimes I)\begin{bmatrix} \theta_0 I \\ \theta_1 I \\ \vdots \\ \theta_{p-1} I \end{bmatrix} = V. \qquad \square$$

The above result allows one to compute the principal pth root of A from the second block entry of the first block column of $\operatorname{sign}(C)$. For $p = 2$ we have $\theta_0 = 1$, $\theta_1 = -1$, $\gamma_0 = 0$, $\gamma_1 = 2$ and proposition 3.2 reproduces the first block column of (3.2).

Based on proposition 3.2 we have the following algorithm for computing the principal matrix pth root of A for a general integer $p \geqslant 2$.

Algorithm 3.1 (pth root through matrix sign function).
INPUT: The integers p, n and the matrix $A \in \mathbb{C}^{n \times n}$; an algorithm `sqrt` for computing the principal matrix square root; an algorithm for computing the matrix sign function.
OUTPUT: The principal pth root X of A.

COMPUTATION:

- If p is odd set $p = 2p$ and $A = A^2$; if p is a multiple of 4 then repeat $p = p/2$, $A = \mathtt{sqrt}(A)$, until $p/2$ is odd.
- Compute $\operatorname{sign}(C)$ and let $V = (V_i)_{i=0:p-1}$ be its first block column.
- Compute $X = (p/(2\sigma))V_1$, where $\sigma = 1 + 2\sum_{j=1}^{\lfloor q/2\rfloor} \cos(2\pi j/p)$ and $q = p/2$.

Observe that if the matrix A is positive definite then its eigenvalues are real and positive, and thus the eigenvalues of X are real and positive. Therefore the real parts of the eigenvalues of $\omega_p^i X$ have the same sign as the real part of ω_p^i. Hence $\operatorname{sign}(\omega_p^i X) = \operatorname{sign}(\omega_p^i)I$, for $i = 0: p-1$, for any $p \geqslant 2$ such that 4 does not divide p (otherwise we would have $\omega_p^{p/4} = \mathrm{i}$ and $\operatorname{sign}(\mathrm{i})$ is not defined). As a consequence, if A is positive definite then proposition 3.2 holds also for *odd* p.

Observe that the result expressed in proposition 3.2 relies on the fact that the sign function applied to the block diagonal matrix S provides a block diagonal matrix having diagonal blocks proportional to the identity matrix I. This fact suggests that we can replace the sign function with any other function having the same feature and which is easily implementable by means of a functional iteration. A candidate is the sector function, defined in section 1, and which for scalars $z \notin \{0\} \cup \bigcup_{j=1,h}\{z \in \mathbb{C}\colon \arg(z) = j\pi/h\}$ satisfies

$$\operatorname{sect}_h(z) = \omega_h^j, \quad \text{if } \arg(z) \in \left((j-1)\frac{\pi}{h}, (j+1)\frac{\pi}{h}\right).$$

In fact, for $h = p$ we have $\operatorname{sect}_p(S) = \operatorname{diag}(I, \omega_p I, \dots, \omega_p^{p-1} I)$, and therefore, as noted in [12],

$$\operatorname{sect}_p(C) = \begin{bmatrix} 0 & X^{-1} & & O \\ \vdots & 0 & \ddots & \\ 0 & \ddots & \ddots & X^{-1} \\ AX^{-1} & 0 & \dots & 0 \end{bmatrix}, \tag{3.4}$$

which generalizes (3.2). It would be interesting to know how the available methods for computing the sector function behave if applied to the block companion matrix C.

4. Reduction to Wiener–Hopf factorization

The splitting of the pth roots of unity with respect to the imaginary axis, implied by the assumption that p is even and not a multiple of 4, is essential for relating the matrix pth root to the Wiener–Hopf factorization of matrix Laurent polynomials.

We recall that a Wiener–Hopf factorization of an $n \times n$ matrix Laurent polynomial $P(z) = \sum_{i=-q}^{q} z^i P_i$ is a factorization of the kind [8]

$$P(z) = U(z)\operatorname{diag}\bigl(z^{\kappa_1}, z^{\kappa_2}, \dots, z^{\kappa_n}\bigr)L\bigl(z^{-1}\bigr), \tag{4.1}$$

where $U(z)$ and $L(z)$ are matrix polynomials that are nonsingular for $|z| \leqslant 1$, and the integers $\kappa_1, \dots, \kappa_n$ are called partial indices. A Wiener–Hopf factorization always exists if $P(z)$ is nonsingular for $|z| = 1$, and the partial indices are uniquely determined, up to the order (see the Gohberg and Krein theorem in [8, p. 189]).

Define the matrix polynomial of degree q

$$Q(x) = \prod_{j=-\lfloor q/2 \rfloor}^{\lfloor q/2 \rfloor} \bigl(xI - \omega_p^j X\bigr) \equiv \sum_{j=0}^{q} x^j Q_j, \tag{4.2}$$

constructed from the matrix pth roots $\omega_p^j X$ having eigenvalues with positive real part. Using the fact that $\omega_p^q = -1$, we find that

$$Q(x)Q(-x) = -\prod_{j=0}^{p-1} \bigl(xI - \omega_p^j X\bigr) = A - x^p I.$$

Moreover, from (4.2) we deduce that $Q(x)$ is singular if and only if x coincides with an eigenvalue of $\omega_p^j X$ for $-q/2 < j < q/2$. Hence $Q(x)$ can be singular only if $\mathrm{Re}(x) > 0$. In other words, the factorization

$$\Phi(x) := A - x^p I = Q(x)Q(-x) = Q(-x)Q(x) \tag{4.3}$$

provides a splitting of the matrix polynomial $\Phi(x)$ with respect to the imaginary axis.

We note that the coefficients of $Q(x)$ are polynomials in X and that

$$Q_q = I, \qquad Q_{q-1} = -\sigma X, \quad \sigma = \sum_{j=-\lfloor q/2 \rfloor}^{\lfloor q/2 \rfloor} \omega_p^j \in \mathbb{R}. \tag{4.4}$$

This fact allows one to express the principal pth root X of A as $X = -\sigma^{-1} Q_{q-1}$.

By applying the Cayley transformation

$$x = \frac{1-z}{1+z}, \qquad z = \frac{1-x}{1+x}, \tag{4.5}$$

which maps the imaginary axis into the unit circle and vice versa, it is straightforward to transform the splitting (4.3) of $\Phi(x)$ with respect to the imaginary axis into a Wiener–Hopf factorization of a suitable matrix polynomial. Observe that under (4.5), infinity is mapped to -1, and -1 to infinity; moreover, the open right half plane is mapped into the open unit disk and vice versa.

Consider the matrix polynomial $\Psi(z)$ of (2.1) and observe that

$$\Psi(z) = (1+z)^p \Phi\left(\frac{1-z}{1+z}\right).$$

Since the matrix coefficients of $\Psi(z)$ are (linear) polynomials in A, they commute. Define the polynomial

$$S(z) = (1+z)^q Q\left(\frac{1-z}{1+z}\right). \tag{4.6}$$

Then we have

$$Q(x) = (1+z)^{-q} S(z), \qquad Q(-x) = \left(1+z^{-1}\right)^{-q} S\left(z^{-1}\right). \tag{4.7}$$

Recalling that $p = 2q$, we find that the factorization (4.3) turns into

$$F(z) := z^{-q}\Psi(z) = S\left(z^{-1}\right)S(z) = S(z)S\left(z^{-1}\right). \tag{4.8}$$

Since $z = (1-x)/(1+x)$ and $\det Q(x) = 0$ only if $\mathrm{Re}(x) > 0$, the matrix polynomial $S(z)$ can be singular only for $|z| < 1$. We conclude that (4.8) is a Wiener–Hopf factorization (4.1) of the Laurent matrix polynomial $F(z)$ with $U(z) = z^q S(z^{-1})$, $L(z) = U(z)$, and null partial indices $\kappa_1 = \cdots = \kappa_n = 0$.

A nice property that follows from (4.8) is that if $\xi \neq 0$ is a zero of $\det \Psi(z)$, then also ξ^{-1} is a zero of $\det \Psi(z)$. If we add k zeros equal to ∞ if $\det \Psi(z)$ has k zeros equal to zero, and if we set $1/\infty = 0$, $1/0 = \infty$, then we may group the zeros of $\det \Psi(z)$ into pairs (ξ_i, ξ_i^{-1}), $i = 1 : qn$, where $0 \leqslant |\xi_i| < 1$.

Observe that, since q is odd, from (4.2) and (4.6) we obtain that

$$S(z) = \prod_{j=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor} \left(I - \omega_p^j X - z\left(I + \omega_p^j X\right)\right) \tag{4.9}$$

$$= G \prod_{j=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor} \left(zI - \left(I + \omega_p^j X\right)^{-1}\left(I - \omega_p^j X\right)\right), \tag{4.10}$$

$$G = -\prod_{j=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor} \left(I + \omega_p^j X\right), \tag{4.11}$$

where $I + \omega_p^j X$ is nonsingular since the eigenvalues of $\omega_p^j X$ have positive real parts.

Summing up, we have seen that the principal pth root X of A can be obtained from the coefficient Q_{q-1} of the matrix polynomial $Q(x)$ of (4.2), and that, by applying the Cayley transformation (4.5), from $Q(x)$ we may derive the Wiener–Hopf factorization (4.8) of $F(z)$. We are now interested in the converse problem. Given a generic Wiener–Hopf factorization of $F(z)$,

$$F(z) = \widehat{U}(z)\widehat{U}\left(z^{-1}\right), \quad \widehat{U}(z) = z^q \widehat{S}\left(z^{-1}\right), \tag{4.12}$$

is it possible to recover the polynomial $Q(x)$ of (4.2), which provides the principal matrix pth root X via (4.4)? The answer is yes. From a classical result, since (4.8) and (4.12) are both Wiener–Hopf factorizations of $F(z)$, there exists a nonsingular matrix W such that $\widehat{S}(z) = S(z)W$ and $WS(z)W = S(z)$. In other words, the factor $S(z)$ is unique up to a right multiplicative matrix factor W such that $WS(z)W = S(z)$. Define the matrix polynomial

$$\widehat{Q}(x) = (1+z)^{-q}\widehat{S}(z), \tag{4.13}$$

which, according to (4.7), corresponds to $\widehat{S}(z)$ by means of the Cayley transformation (4.5). It is immediate to verify from $\widehat{S}(z) = S(z)W$ that $\widehat{Q}(x) = Q(x)W$. Therefore, since the leading block coefficient of $Q(x)$ is the identity matrix, the leading block coefficient of $\widehat{Q}(x)$ is W. Hence the coefficients Q_j, $j = 0:q$, of $Q(x)$ are related to the corresponding coefficients $\widehat{Q}_j$, $j = 0:q$, of $\widehat{Q}(x)$ by the relations

$$Q_j = \widehat{Q}_j\widehat{Q}_q^{-1}, \quad j = 0:q.$$

From (4.4), we therefore obtain

$$X = -\sigma^{-1}\widehat{Q}_{q-1}\widehat{Q}_q^{-1}. \tag{4.14}$$

Now we are ready to prove the following proposition, which relates the principal pth root X with the coefficients of $\widehat{S}(z)$.

Proposition 4.1. Let $\widehat{S}(z)$ be any matrix polynomial such that the Wiener–Hopf factorization (4.12) holds. Then the principal pth root X of A is given by

$$X = -\sigma^{-1}\big(qI + 2\widehat{S}'(-1)\widehat{S}(-1)^{-1}\big), \tag{4.15}$$

where $\sigma = \sum_{j=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor} \omega_p^j = 1 + 2\sum_{j=1}^{\lfloor q/2\rfloor} \cos(2\pi j/p)$.

Proof. Consider $\widehat{Q}(x)$, the matrix polynomial defined in (4.13). Observe that $\widehat{Q}_{q-1} = \widehat{Q}'_R(0)$, $\widehat{Q}_q = \widehat{Q}_R(0)$, where $\widehat{Q}_R(x) = x^q\widehat{Q}(x^{-1})$ is the matrix polynomial obtained by reversing the order of the coefficients of $\widehat{Q}(x)$. Replacing z with $(1-x)/(1+x)$ in (4.13) yields

$$\widehat{Q}(x) = \frac{(x+1)^q}{2^q}\widehat{S}\left(\frac{1-x}{1+x}\right), \qquad \widehat{Q}_R(x) = \frac{(x+1)^q}{2^q}\widehat{S}\left(\frac{x-1}{x+1}\right),$$

whence we obtain $\widehat{Q}_R(0) = (1/2^q)\widehat{S}(-1)$ and $\widehat{Q}'_R(0) = (q/2^q)\widehat{S}(-1) + (2/2^q)\widehat{S}'(-1)$. From (4.14) we obtain the sought expression for X. □

Based on the above result we have the following algorithm for computing $A^{1/p}$ for any integer $p \geqslant 2$.

Algorithm 4.1 (pth root through Wiener–Hopf factorization).
INPUT: The integers p, n and the matrix $A \in \mathbb{C}^{n\times n}$; an algorithm `sqrt` for computing the principal matrix square root; and an algorithm for computing a Wiener–Hopf factorization of a Laurent matrix polynomial with commuting coefficients.
OUTPUT: An approximation to the principal pth root X of A.

COMPUTATION:

- If p is odd set $p = 2p$ and $A = A^2$; if p is a multiple of 4 then repeat $p = p/2$, $A = \mathtt{sqrt}(A)$, until $p/2$ is odd.
- Compute a Wiener–Hopf factorization $F(z) = \widehat{U}(z)\widehat{U}(z^{-1})$ of the Laurent matrix polynomial $F(z) = z^{-q}\Psi(z)$ in (4.8) (see section 9), and set $\widehat{S}(z) = z^q\widehat{U}(z^{-1})$.
- Compute $X = -\sigma^{-1}(qI + 2\widehat{S}'(-1)\widehat{S}(-1)^{-1})$, with $\sigma = \sum_{j=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor}\omega_p^j = 1 + 2\sum_{j=1}^{\lfloor q/2\rfloor}\cos(2\pi j/p)$.

5. Reduction to matrix Laurent polynomial inversion

A different and computationally simpler expression for the pth root X is provided by the next proposition. We first need to observe that the Laurent matrix polynomial $F(z) = z^{-q}\Psi(z)$ is analytic and nonsingular in the annulus (2.2), which can be rewritten in the form

$$\mathcal{A} = \left\{z \in \mathbb{C}\colon\ |\xi_{nq}| < |z| < \frac{1}{|\xi_{nq}|}\right\}, \tag{5.1}$$

where ξ_i, $i = 1:nq$, are the zeros of $\det S(z)$ ordered so that

$$0 \leqslant |\xi_1| \leqslant \cdots \leqslant |\xi_{nq}| < 1.$$

In this way we may define the matrix Laurent series

$$H(z) = F(z)^{-1} = \sum_{j=-\infty}^{+\infty} z^j H_j = H_0 + \sum_{j=1}^{+\infty}\left(z^j + z^{-j}\right)H_j, \tag{5.2}$$

which is analytic for $z \in \mathcal{A}$, and is such that $H_j = H_{-j}$, for $j \neq 0$. The latter property holds since $F(z) = F(z^{-1})$ implies that $H(z) = H(z^{-1})$.

Observe that, if λ_i, $i = 1:n$ are the eigenvalues of the matrix X, then $\{\xi_i\colon\ i = 1:nq\} = \{(1-\omega_p^j\lambda_i)/(1+\omega_p^j\lambda_i)\colon\ i = 1:n,\ j = 1:q\}$, and therefore

$$|\xi_{nq}| = \max_{j=1:q,\ i=1:n}\left|\frac{1-\omega_p^j\lambda_i}{1+\omega_p^j\lambda_i}\right| < 1. \tag{5.3}$$

Equation (5.3) relates the thickness of the annulus $\mathcal{A}$ where $F(z)$ and its inverse are analytic with the location of the eigenvalues of X. In particular, $\mathcal{A}$ is a thin annulus if the eigenvalues of X are very unbalanced in modulus or if they are close to the lines on the boundary between sectors. As we will see later, the thickness of $\mathcal{A}$ is related to the speed of convergence of certain algorithms for computing the pth root.

Proposition 5.1. The principal pth root X of A can be represented as

$$X = 4p\sin\left(\frac{\pi}{p}\right)A\sum_{j=0}^{q-1}\alpha_j H_j, \tag{5.4}$$

where $H(z)$ is the matrix Laurent series of (5.2) and $\alpha_0 = \frac{1}{2}\binom{p-2}{q-1}$, $\alpha_j = \binom{p-2}{q-j-1}$, $j = 1:q-1$.

Proof. In order to prove the representation (5.4) of X consider the Laurent polynomial $t(z) = z^{-q+1}(z+1)^{p-2} = 2\alpha_0 + \sum_{i=1}^{q-1} \alpha_i(z^i + z^{-i})$ and observe that the constant term of the product $W(z) = H(z)t(z)$ is $W_0 = 2\sum_{j=0}^{q-1} \alpha_j H_j$. Therefore it is sufficient to show that $X = 2p\sin(\pi/p)AW_0$. Observe that from the Cauchy integral formula we have

$$W_0 = \frac{1}{2\pi \mathrm{i}} \int_{|z|=1} z^{-1} W(z)\,\mathrm{d}z = \frac{1}{2\pi \mathrm{i}} \int_{|z|=1} F(z)^{-1} z^{-q}(z+1)^{p-2}\,\mathrm{d}z. \tag{5.5}$$

Moreover, from (2.3) and (2.1) we deduce that

$$X = 2p\sin\left(\frac{\pi}{p}\right) A \frac{1}{2\mathrm{i}\pi} \int_{|z|=1} F(z)^{-1} z^{-q}(1+z)^{p-2}\,\mathrm{d}z,$$

whence $X = 2p\sin(\pi/p)AW_0$. □

The representation (5.4), which generalizes a result of Meini [22] that applies for $p = 2$, provides an algorithm for approximating X if a technique for approximating the coefficients H_i, $i = 0:q$ of the Laurent series $H(z) = F(z)^{-1}$ is available.

Summarizing, we have the following algorithm for computing the principal pth root of A, for a general integer $p \geqslant 2$, relying on the inversion of a matrix Laurent polynomial.

Algorithm 5.1 (pth root through matrix Laurent polynomial inversion).
INPUT: The integers p, n and the matrix $A \in \mathbb{C}^{n\times n}$; an algorithm `sqrt` for computing the principal square root; and an algorithm for computing the inverse of a Laurent matrix polynomial with commuting coefficients.
OUTPUT: An approximation to the principal pth root X of A.
COMPUTATION:

- If p is odd set $p = 2p$ and $A = A^2$; if p is a multiple of 4 then repeat $p = p/2$, $A = \mathtt{sqrt}(A)$, until $p/2$ is odd.
- Compute the coefficients $H_0, \dots, H_{q-1}$ of the inverse $H(z) = H_0 + \sum_{i=1}^{+\infty}(z^i + z^{-i})H_i$ of the Laurent matrix polynomial $F(z) = z^{-q}\Psi(z)$ in (4.8) (see section 8).
- Compute $X = 4p\sin(\pi/p)A\sum_{j=0}^{q-1}\alpha_j H_j$, where $\alpha_0 = \frac{1}{2}\binom{p-2}{q-1}$, $\alpha_j = \binom{p-2}{q-j-1}$, $j = 1:q-1$.

6. Newton's iteration for the inverse pth root

The results of this section are valid without any restriction on the integer p. Consider the iteration for computing the *inverse* pth root, $A^{-1/p}$:

$$X_{k+1} = \frac{1}{p}\left[(1+p)X_k - X_k^{p+1}A\right], \qquad X_0 = I. \tag{6.1}$$

This is Newton's method applied to $X^{-p} - A = 0$ with $X_0 = I$, the concise form (6.1) being obtained by exploiting the fact that the iterates are polynomials in A and so commute with A.

Iteration (6.1) contrasts with (1.1), which is Newton's method applied to $X^p - A = 0$ and computes $A^{1/p}$. While (1.1) involves matrix inversion, (6.1) requires only matrix multiplication. As mentioned in section 1, (1.1) has rather unsatisfactory convergence properties. The region of convergence of (1.1) to $a^{1/p}$ for scalars $a \in \mathbb{C}$ is roughly the wedge defined by $\arg a \in (-\pi/p, \pi/p)$, but it has a petal-like boundary intruding inside the wedge for $p > 2$. Consequently, it is difficult to guarantee convergence except for symmetric positive definite A. As we will now show, iteration (6.1) has better convergence properties.

For $p = 1$, iteration (6.1) is the well known Schulz iteration for matrix inversion [25], and it is easily seen that the residual $R_k = I - X_k A$ satisfies $R_{k+1} = R_k^2$. For $p = 2$, the iteration is well known in the scalar case, and it is studied for symmetric positive definite matrices by Philippe [24]. Philippe proves that the residuals $R_k = I - X_k^2 A$ satisfy $R_{k+1} = \frac{3}{4}R_k^2 + \frac{1}{4}R_k^3$ [24, proposition 2.5]. The following result generalizes these residual relations to arbitrary p.

Proposition 6.1. The residuals $R_k = I - X_k^p A$ for (6.1) satisfy, for any choice of X_0,

$$R_{k+1} = \sum_{i=2}^{p+1} a_i R_k^i, \tag{6.2}$$

where the a_i are all positive and $\sum_{i=2}^{p+1} a_i = 1$. Hence if $\|R_0\| < 1$ for some consistent matrix norm then the sequence $\{\|R_k\|\}$ decreases monotonically to 0 as $k \to \infty$.

Proof. We have $X_{k+1} = p^{-1}X_k(pI + R_k)$, which leads to

$$\begin{aligned} R_{k+1} &= I - \frac{1}{p^p}(I - R_k)(pI + R_k)^p = I - \frac{1}{p^p}\left[p^p I + \sum_{i=1}^{p} b_i R_k^i - R_k^{p+1}\right] \\ &= -\frac{1}{p^p}\left[\sum_{i=1}^{p} b_i R_k^i - R_k^{p+1}\right], \end{aligned} \tag{6.3}$$

where

$$\begin{aligned} b_i &= \binom{p}{i} p^{p-i} - \binom{p}{i-1} p^{p-i+1} = p^{p-i}\left[\binom{p}{i} - \binom{p}{i-1} p\right] \\ &= p^{p-i}\left[\frac{p!}{i!(p-i)!} - \frac{p!\,p}{(i-1)!(p-i+1)!}\right] \\ &= p^{p-i}\frac{p!}{(i-1)!(p-i)!}\left[\frac{1}{i} - \frac{p}{(p-i+1)}\right]. \end{aligned}$$

It is easy to see that $b_1 = 0$ and $b_i < 0$ for $i \geqslant 2$. Hence (6.2) holds, with $a_i > 0$ for all i. By setting $R_k \equiv I$ in (6.2) and (6.3) it is easy to see that $\sum_{i=2}^{p+1} a_i = 1$.

If $0 < \|R_0\| < 1$, then taking norms in (6.2) yields

$$\|R_1\| \leqslant \sum_{i=1}^{p+1} |a_i| \, \|R_0\|^i < \|R_0\| \sum_{i=1}^{p+1} |a_i| = \|R_0\|.$$

By induction, the $\|R_k\|$ form a monotonically decreasing sequence that converges to zero. □

An immediate corollary of proposition 6.1 is that iteration (6.1) converges quadratically.

Proposition 6.1 gives the sufficient condition for convergence of the iteration (6.1) that $\|I - A\| < 1$, for some norm. It is a standard result that for any A and any $\delta > 0$ there is a consistent norm such that $\|A\| \leqslant \rho(A) + \delta$, where ρ is the spectral radius [17, problem 6.8]. It follows that a sufficient condition for convergence of (6.1) is that the eigenvalues λ_i of A satisfy

$$\max_i |1 - \lambda_i| < 1, \tag{6.4}$$

that is, the eigenvalues of A lie strictly within the circle of centre 1 and radius 1. For matrices with real, positive eigenvalues we can say more.

Proposition 6.2. Suppose that all the eigenvalues of A are real and positive. Then iteration (6.1) converges to $A^{-1/p}$ if $\rho(A) < p + 1$. If $\rho(A) = p + 1$ the iteration does not converge to the inverse of any pth root of A.

Proof. By standard arguments based on the Jordan canonical form, it suffices to analyze the convergence of the iteration on the eigenvalues of A. We therefore consider the scalar iteration

$$x_{k+1} = \frac{1}{p}\big[(1 + p)x_k - x_k^{p+1} a\big], \qquad x_0 = 1, \tag{6.5}$$

with $a > 0$. Let $y_k = a^{1/p} x_k$. Then

$$y_{k+1} = \frac{1}{p}\big[(1 + p)y_k - y_k^{p+1}\big] =: f(y_k), \qquad y_0 = a^{1/p}$$

and we need to prove that $y_k \to 1$ if $y_0 = a^{1/p} < (p + 1)^{1/p}$. We consider two cases. If $y_k \in [0, 1)$ then

$$y_{k+1} = y_k\left[1 + \frac{1 - y_k^p}{p}\right] > y_k.$$

Moreover, since

$$f(0) = 0, \qquad f(1) = 1, \qquad f'(y) = \frac{p+1}{p}\big(1 - y^p\big) > 0 \quad \text{for } y \in [0, 1),$$

it follows that $f(y) < 1$ for $y \in [0, 1)$. Hence $y_k < y_{k+1} < 1$ and so the y_k form a monotonically increasing sequence tending to 1. Now suppose $y_0 \in (1, (p+1)^{1/p})$. We have $f(1) = 1$ and $f((p+1)^{1/p}) = 0$, and $f'(y) < 0$ for $y > 1$. It follows that f maps $(1, (p+1)^{1/p})$ into $(0, 1)$ and so after one iteration $y_1 \in (0, 1)$ and the first case applies. The last part of the proposition follows from $f((p+1)^{1/p}) = 0$ and the fact that 0 is a fixed point of the iteration. □

For matrices with real, positive eigenvalues the condition $\rho(A) < p + 1$ in proposition 6.2 is clearly much less restrictive than (6.4). It is then natural to ask whether a region of convergence in $\mathbb{C}$ bigger than (6.4) can be identified, and indeed what is the actual region of convergence. We have not been able to answer these questions analytically, so have determined the regions empirically. For a grid of points x_0 in $\mathbb{C}$ we ran 50 iterations and declared convergence to $a^{1/p}$ if x_{50} had relative error less than 10^{-12}. Figure 1 shows the results for $p = 1, 2, 3, 4, 8, 16$; the unit circle is shown (note that the axis scales are not equal). We see that the region of convergence grows with p, extending almost up to $\pm 2i$ for small real parts and approaching the point $p + 1$ on the real axis via a wedge shape.

Analysis of Smith [27] shows that the Newton iteration (1.1) is numerically stable (in the sense that arbitrary perturbations in an iterate do not grow over successive iterations) if

$$\frac{1}{p}\left|(p-1) - \sum_{r=1}^{p-1}\left(\frac{\lambda_i}{\lambda_j}\right)^{r/p}\right| \leqslant 1, \quad i, j = 1\colon n. \tag{6.6}$$

A similar analysis can be done for (6.1), leading to the condition

$$\frac{1}{p}\left|p - \sum_{r=1}^{p}\left(\frac{\lambda_i}{\lambda_j}\right)^{r/p}\right| \leqslant 1, \quad i, j = 1\colon n. \tag{6.7}$$

Condition (6.7) is even more restrictive than (6.6). For example, for symmetric positive definite A and $p = 3$, (6.6) requires $\kappa_2(A) \leqslant 5.74$, whereas (6.7) requires $\kappa_2(A) \leqslant 2.68$. However, if $\max_i |1 - \lambda_i| < 1/2$ holds, say, then the spread of the eigenvalues is narrow enough that instability will be mild or absent. Therefore iteration (6.1) is certainly of interest for A or A^{-1} satisfying $\max_i |1 - \lambda_i| < 1/2$, with an inversion of the limit matrix needed to recover $A^{1/p}$ in the former case.

Another use of (6.1) is for refining an approximate pth root $Y_0 \approx A^{1/p}$ obtained from one of our other algorithms. We can apply (6.1) to A^{-1} with starting matrix Y_0 to obtain

$$Y_{k+1} = \frac{1}{p}\big[(1+p)Y_k - Y_k^{p+1}A^{-1}\big].$$

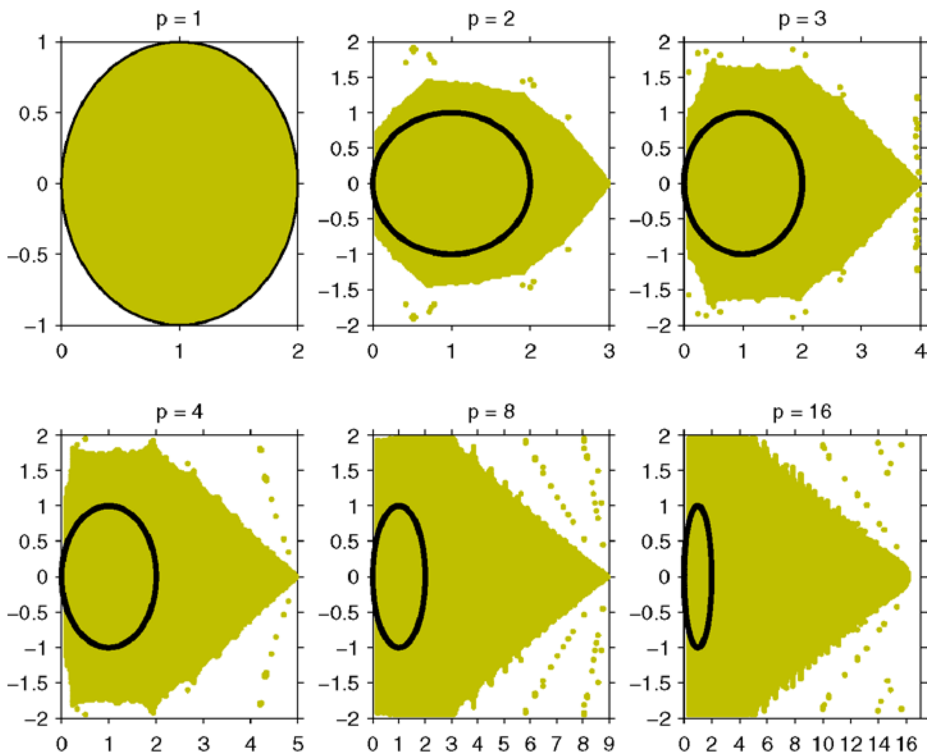

Figure 1. Convergence regions (shaded) in $\mathbb{C}$ for iteration (6.5), together with unit circle. Note differing axis limits.

Since Y_0 is arbitrary, proposition 6.2 does not apply, but proposition 6.1 guarantees quadratic convergence of the Y_k to $A^{1/p}$ if $\|I - Y_0^p A^{-1}\| < 1$.

7. Inverting an A-circulant matrix

In order to compute $\operatorname{sign}(C)$ in algorithm 3.1, where C is the block companion matrix (3.1), we may apply the matrix sign iteration

$$C_{k+1} = \frac{C_k + C_k^{-1}}{2}, \quad k = 0, 1, \ldots, \qquad C_0 = C, \tag{7.1}$$

which converges quadratically to $\operatorname{sign}(C)$. The most expensive part of this iteration is computing C_k^{-1}. In this section we design an algorithm for the fast inversion of C_k that exploits its structural properties.

We recall that the matrix algebra generated by C, i.e., the set of all polynomials in C of the kind $\sum_{i=0}^{p-1}(I \otimes W_i)C^i$, is called the class of A-circulant matrices and is composed of matrices of the form

$$P=\begin{bmatrix} W_0 & W_1 & \dots & W_{p-1} \\ AW_{p-1} & W_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & W_1 \\ AW_1 & \dots & AW_{p-1} & W_0 \end{bmatrix}. \tag{7.2}$$

We call (7.2) the A-circulant matrix associated with $P(x)$.

If P and Q are A-circulant matrices associated with the polynomials $P(x)$ and $Q(x)$, respectively, then $P \operatorname{op} Q$ is the A-circulant matrix associated with the polynomial $P(x) \operatorname{op} Q(x) \bmod x^pI - A$, for op $= +, *$. Similarly, P^{-1} is the A-circulant matrix associated with the polynomial $P(x)^{-1} \bmod x^pI - A$. In particular, the matrix $\operatorname{sign}(C)$ has the structure (7.2), as do the iterates (7.1).

We now analyze the complexity of performing a single step of the matrix sign iteration. Since an A-circulant matrix is defined by its first block row (or its first block column), multiplying two A-circulant matrices reduces to computing the product of an A-circulant matrix and a block vector. This computation can be efficiently performed by means of the FFT in $\mathrm{O}(n^3p + n^2p\log p)$ operations since A-circulant matrices are in particular block Toeplitz [7].

A different analysis is required for the problem of matrix inversion. For this purpose we observe that for any matrix polynomial $P(x) = \sum_{i=0}^{p-1} x^iW_i$ with commuting matrix coefficients the product

$$T(x) = P(x)P(\omega_p x)\cdots P\bigl(\omega_p^{p-1}x\bigr) \tag{7.3}$$

is a polynomial in x^p, say $T(x) = Q(x^p)$. This property holds since, by the commutativity of the coefficients, we have $T(\omega^i x) = T(x)$ for $i = 0\colon p-1$, so that $T(x)$ must necessarily be a polynomial in x^p.

Now observe that if P is the A-circulant matrix associated with $P(x)$, then $D^{-i}PD^i$ is the A-circulant matrix associated with $P(\omega_p^i x)$, where $D = \operatorname{diag}(1, \omega_p, \omega_p^2, \dots, \omega_p^{p-1})$. Moreover, since $C^p = I \otimes A$, the A-circulant matrix associated with $Q(x^p)$ is $I \otimes K$, where $K = Q(A)$ is an $n \times n$ matrix. In this way we may rewrite (7.3) in matrix form as

$$I \otimes K = P\bigl(D^{-1}PD\bigr)\bigl(D^{-2}PD^2\bigr)\cdots\bigl(D^{-p+1}PD^{p-1}\bigr).$$

This relation provides the following inversion formula for P

$$\begin{aligned} P^{-1} &= S\bigl(I \otimes K^{-1}\bigr), \\ S &= \bigl(D^{-1}PD\bigr)\bigl(D^{-2}PD^2\bigr)\cdots\bigl(D^{-p+1}PD^{p-1}\bigr). \end{aligned}$$

In this way the computation of P^{-1} is reduced to computing the A-circulant matrix S and the $n \times n$ matrix K. By making the substitution $B_0 = D^{-1}PD$, we readily find that

$$S = B_0\bigl(D^{-1}B_0D\bigr)\bigl(D^{-2}B_0D^2\bigr)\cdots\bigl(D^{-p+2}B_0D^{p-2}\bigr). \tag{7.4}$$

Moreover, setting $B_1 = B_0(D^{-1}B_0D)$ yields

$$S = B_1\big(D^{-2}B_1D^2\big)\big(D^{-4}B_1D^4\big)\big(D^{-6}B_1D^6\big)\cdots\big(D^{-p+3}B_1D^{p-3}\big), \tag{7.5}$$

where for simplicity we assumed $p-1$ even. That is, the product (7.4) of $p-1$ A-circulant matrices is reduced to the product (7.5) of $(p-1)/2$ A-circulant matrices by just performing a single product of A-circulant matrices.

The following algorithm is based on this reduction and computes the first block column (row) of P^{-1}, which defines the A-circulant matrix P^{-1}.

Algorithm 7.1 (Inverting an A-circulant matrix).
INPUT: The integers p, n, the matrix $A \in \mathbb{C}^{n\times n}$ and the commuting matrices $W_0, \ldots, W_{p-1} \in \mathbb{C}^{n\times n}$ defining the A-circulant matrix P of (7.2).
OUTPUT: The first block column (row) of P^{-1}.
COMPUTATION:

- Represent the integer $p-1$ in base 2 as $p-1 = \sum_{i=0}^{d-1} 2^{m_i}$.
- Set $B_0 = D^{-1}PD$.
- For $i = 0{:}\, m_{d-1}$ compute $B_i = B_{i-1}(D^{-i}B_{i-1}D^i)$.
- Compute

$$V = [I, 0, \ldots, 0]S = [I, 0, \ldots, 0]B_{m_0}D^{-2^{m_0}}B_{m_1}D^{-2^{m_1}}\cdots B_{m_{d-1}}D^{p-1-2^{m_{d-1}}}$$

  by successively multiplying block row vectors and A-circulant matrices.
- Compute K as the first block of VP.
- Output $V(I \otimes K^{-1})$.

The above algorithm relies on the fact that $B_i = (D^{-1}P)^{2^i}D^{2^i}$ and that $S = (D^{-1}P)^{p-1}D^{p-1} = B_{m_0}D^{-2^{m_0}}B_{m_1}D^{-2^{m_1}}\cdots B_{m_{d-1}}$. Its complexity is dominated by $\lceil \log_2(p-1)\rceil$ multiplications of A-circulant matrices, so its computational cost is $\mathrm{O}(n^3p\log p + n^2p\log^2 p)$ operations.

8. Inverting a matrix Laurent polynomial

We present two algorithms for approximating the inverse of the matrix Laurent polynomial $F(z) = z^{-q}\Psi(z)$. The first algorithm is based on the evaluation-interpolation technique, while the second relies on the Graeffe iteration [3,23] extended to matrix polynomials with commuting coefficients. These algorithms can be used for computing $A^{1/p}$ in the light of the results of section 5.

8.1. Evaluation-interpolation

Let us recall that $F(z) = z^{-q}\Psi(z)$ and $H(z)$ are analytic and nonsingular in the annulus

$$\mathcal{A} = \left\{ z \in \mathbb{C}\colon\ |\xi_{nq}| < |z| < \frac{1}{|\xi_{nq}|} \right\}. \tag{8.1}$$

This implies that all the entries of $H(z)$ are analytic functions in $\mathcal{A}$.

We use some basic results from the theory of analytic functions of a complex variable in order to prove a decay property of the coefficients of $H(z)$. Let $f(z) = \sum_{i=-\infty}^{+\infty} z^i f_i$ be a complex valued function analytic in the annulus (8.1). Since the Fourier coefficients of $f(z)$ are given by

$$f_j = \frac{1}{2\pi \mathrm{i}} \int_{|z|=\theta} \frac{f(z)}{z^{j+1}}\,\mathrm{d}z$$

for $|\xi_{nq}| < \theta < 1/|\xi_{nq}|$, we deduce that the sequence f_j decays exponentially to zero. More precisely, for any $\varepsilon > 0$ there exists a constant $\gamma > 0$ such that $|f_j| < \gamma(|\xi_{nq}|+\varepsilon)^j$ for $j > 0$ and $|f_j| < \gamma(1/|\xi_{nq}| - \varepsilon)^j$ for $j < 0$.

Applying this property to each element of the matrix function $H(z)$, it follows that, for any ε, there exists a constant δ such that $\|H_j\|_\infty < \delta n(|\xi_{nq}| + \varepsilon)^j$ for $j > 0$, and $\|H_j\|_\infty < \delta n(1/|\xi_{nq}| - \varepsilon)^j$ for $j < 0$, where $\|\cdot\|_\infty$ denotes the infinity norm. This implies that for a sufficiently large h the Laurent matrix polynomial $K(z) = \sum_{i=-h}^{h} z^i H_i$ well approximates the function $H(z)$. Moreover, due to the exponential bounds on the norms of the H_j, the value of h is not generally large unless $\mathcal{A}$ is a very thin annulus, i.e., $|\xi_{nq}| \approx 1$. The latter situation may happen if the eigenvalues of X have unbalanced moduli or if they are close to the lines which separate the sector (compare with (5.3)). This decay property suggests the following evaluation-interpolation technique for approximating H_i, $i = 0 : q-1$.

Algorithm 8.1 (Inversion by evaluation-interpolation).
INPUT: The coefficients $F_0, \dots, F_p$ of $F(z)$; an integer h such that $\sum_{i>h} \|H_i\|_\infty$ is negligible.
OUTPUT: Approximations to the coefficients H_i, $i = 0 : q-1$, of the matrix Laurent series $H(z) = F(z)^{-1}$.
COMPUTATION: Choose a positive integer $N = 2^\nu$ such that $N > 2h + 1$ and consider the Nth roots of unity ω_N^i, $i = 0 : N-1$, where $\omega_N = \cos(2\pi/N) + \mathrm{i}\sin(2\pi/N)$.

1. Compute $W_i = F(\omega_N^i)$, $i = 0 : N-1$, by means of n^2 FFTs of length N.
2. Compute $V_i = W_i^{-1}$, $i = 0 : N-1$.
3. Interpolate to the values V_i and recover the matrix coefficients K_i of the Laurent matrix polynomial $K(z) = \sum_{i=-h}^{h} z^i K_i$ that interpolates $H(z)$ at the roots of unity, by means of n^2 FFTs of length N.
4. Output the approximations K_i to H_i for $i = 0 : q-1$.

Observe that the cost of the above algorithm is $O(n^3N + n^2N\log N)$ operations. The greater the width of $\mathcal{A}$ the smaller the value of N.

A reduction of the computational cost can be obtained by exploiting the specific structure of $F(z)$. In fact, since the coefficients of $F(z)$ are $\binom{p}{j}(I+(-1)^{j+1}A)$, $j = 0:p$, the matrices W_i at step 1 are given by $W_i = \alpha_i I + \beta_i A$, where $\alpha_i = -\sum_{j=0}^{N-1}\omega_N^{ij}\binom{p}{j}$, $\beta_i = \sum_{j=0}^{N-1}(-1)^j\omega_N^{ij}\binom{p}{j}$. Therefore, only *two* FFTs of length N must be computed. At stage 2, we have to invert the matrices $\alpha_i I + \beta_i A$, for $i = 0:N-1$. It is not clear if this task can be accomplished with a cost lower than $O(n^3N)$. This stage remains the most expensive part of the algorithm.

In the formulation of algorithm 8.1 the value of h, and consequently that of N, must be known a priori. However, by following [5] we may apply a dynamic strategy which performs the computation by repeatedly doubling the values of N until the convergence condition is satisfied. In this way, the algorithm is adaptive and requires neither h nor N as input values. See [5] for more details. As a convergence condition we may require a bound on the residual, say, $\|K(z)F(z) - I\|_\infty \leqslant \varepsilon$, where the infinity norm of a matrix Laurent polynomial is the maximum infinity norm of its coefficients.

It is interesting to note that the nonsingularity of $F(z)$ for $z \in \mathcal{A}$ implies that there exists a positive constant γ such that for any z of modulus 1 the matrix $F(z)$ has a condition number bounded by γ. This property guarantees that the matrices V_i, $i = 0:N-1$, have a condition number independent of N.

8.2. *Graeffe iteration*

The Graeffe iteration is used in [4] for inverting a matrix Laurent polynomial of the form $z^{-1}A_{-1} + A_0 + zA_1$ for $n \times n$ matrices A_{-1}, A_0, A_1. This technique does not apply to general matrix Laurent polynomials of larger degree. However, in our problem, the matrix Laurent polynomial $F(z)$ has an additional feature which is fundamental in order to extend the inversion algorithm of [4]: its coefficients satisfy the commutativity property $F_iF_j = F_jF_i$, $i, j = -q:q$. We now extend the algorithm of [4] to matrix polynomials of the form

$$P(z) = \sum_{j=-q}^{q} z^j P_j, \tag{8.2}$$

where $P_jP_i = P_iP_j$ for any pair (i, j).

Observe that, because the P_i commute, the coefficients of the odd powers in the expression $P(z)P(-z)$ vanish. Therefore $P^{(1)}(z^2) = P(z)P(-z)$ is a matrix Laurent polynomial in z^2 whose coefficients, as polynomial functions of the coefficients of $P(z)$, commute with each other. Inductively, we may define the Graeffe sequence of matrix

Laurent polynomials

$$\begin{aligned} &P^{(0)}(z) = P(z),\\ &P^{(i+1)}\bigl(z^2\bigr) = P^{(i)}(z)P^{(i)}(-z), \quad P^{(i)}(z) = \sum_{j=-q}^{q} z^j P_j^{(i)}, \end{aligned} \tag{8.3}$$

where $P^{(i+1)}(z^2)$ has commuting matrix coefficients.

A fundamental property of the sequence (8.3) is described in the next result.

Proposition 8.1. Assume that the polynomial (8.2) can be factorized as

$$P(z) = U(z)V\bigl(z^{-1}\bigr),$$

where $U(z) = (zI - X_1)(zI - X_2)\cdots(zI - X_q)$, $V(z) = (zI - Y_1)(zI - Y_2)\cdots(zI - Y_q)$, and the matrices X_j, Y_j are such that $\|X_j\|, \|Y_j\| \leqslant \sigma < 1$, $j = 1:q$, for a suitable operator norm $\|\cdot\|$. Moreover assume that $AB = BA$ for any $A, B \in \{X_1, \ldots, X_q\} \cup \{Y_1, \ldots, Y_q\}$. Then the sequence generated by (8.3) is such that

$$\begin{aligned} &\bigl\| P_0^{(i)} - I \bigr\| \leqslant q^2 \sigma^{2\cdot 2^i},\\ &\bigl\| P_j^{(i)} \bigr\| < \binom{q}{j} \sigma^{j2^i} + \mathrm{O}\left(q \binom{q}{j+1} \sigma^{(j+2)2^i} \right). \end{aligned}$$

Proof. Using the commutativity property, we have

$$\begin{aligned} P^{(1)}\bigl(z^2\bigr) = P(z)P(-z) &= \prod_{j=1}^{q} (zI - X_j)(-zI - X_j) \prod_{j=1}^{q} \bigl(z^{-1}I - Y_j\bigr)\bigl(-z^{-1}I - Y_j\bigr)\\ &= \prod_{j=1}^{q} \bigl(z^2 I - X_j^2\bigr)\bigl(z^{-2}I - Y_j^2\bigr). \end{aligned}$$

By induction, we obtain

$$P^{(i)}(z) = \prod_{j=1}^{q} \bigl(zI - X_j^{2^i}\bigr)\bigl(z^{-1}I - Y_j^{2^i}\bigr).$$

By the commutativity of the matrices $X_j^{2^i}$ and $Y_j^{2^i}$ for $j = 1:q$, we may write the block coefficients of the matrix polynomials $E^{(i)}(z) = \sum_{j=0}^{q} z^j E_j^{(i)} = \prod_{j=1}^{q} (zI - X_j^{2^i})$ and $G^{(i)}(z) = \sum_{j=0}^{q} z^j G_j^{(i)} = \prod_{j=1}^{q} (z^{-1}I - Y_j^{2^i})$ in terms of the elementary symmetric functions

$$f_j(x_1, \ldots, x_q) = \sum_{1 \leqslant \sigma_1 < \cdots < \sigma_j \leqslant q} (x_{\sigma_1} \cdots x_{\sigma_j}), \quad j = 0:q.$$

More precisely,

$$E_j^{(i)} = (-1)^j f_{q-j}\left(X_1^{2^i}, \ldots, X_q^{2^i}\right),$$
$$G_j^{(i)} = (-1)^j f_{q-j}\left(Y_1^{2^i}, \ldots, Y_q^{2^i}\right).$$

Moreover, from the relation $P^{(i)}(z) = E^{(i)}(z)G^{(i)}(z^{-1})$ we find that

$$P_k^{(i)} = \sum_{j=0}^{q} E_j^{(i)} G_{k+j}^{(i)}, \quad k = -q:q, \tag{8.4}$$

where we set $E_j^{(i)} = G_j^{(i)} = 0$ for $j < 0$ and for $j > q$. Applying the triangle inequality yields

$$\begin{aligned}\left\|E_j^{(i)}\right\| &= \left\|f_{q-j}\left(X_1^{2^i}, \ldots, X_q^{2^i}\right)\right\| \leqslant f_{q-j}\left(\|X_1\|^{2^i}, \ldots, \|X_q\|^{2^i}\right)\\ &\leqslant f_{q-j}\left(\sigma^{2^i}, \ldots, \sigma^{2^i}\right) = \binom{q}{j}\sigma^{(q-j)2^i}.\end{aligned}$$

Similarly, we have $\|G_j^{(i)}\| \leqslant \binom{q}{j}\sigma^{(q-j)2^i}$. From (8.4) we find that $P_0^{(i)} = I + E_{q-1}^{(i)}G_{q-1}^{(i)} + \cdots + E_0^{(i)}G_0^{(i)}$, whence

$$\begin{aligned}\left\|P_0^{(i)} - I\right\| &\leqslant \left(\binom{q}{1}\sigma^{2^i}\right)^2 + \left(\binom{q}{2}\sigma^{2\cdot 2^i}\right)^2 + \cdots + \left(\binom{q}{0}\sigma^{q\cdot 2^i}\right)^2\\ &= q^2\sigma^{2\cdot 2^i} + \mathrm{O}\left(\left(\binom{q}{2}\sigma^{2\cdot 2^i}\right)^2\right).\end{aligned}$$

Similarly, for $k > 0$ we obtain

$$\begin{aligned}\left\|P_k^{(i)}\right\| &\leqslant \left\|E_{q-k}^{(i)}\right\| \cdot \left\|G_q^{(i)}\right\| + \left\|E_{q-k-1}^{(i)}\right\| \cdot \left\|G_{q-1}^{(i)}\right\| + \cdots + \left\|E_0^{(i)}\right\| \cdot \left\|G_k^{(i)}\right\|\\ &\leqslant \binom{q}{k}\sigma^{k\cdot 2^i} + \mathrm{O}\left(q\binom{q}{k+1}\sigma^{(k+2)\cdot 2^i}\right).\end{aligned}$$

The case $k < 0$ is dealt with analogously. □

The above result is the basis for designing a fast algorithm that computes the central $2q + 1$ coefficients (i.e., the coefficients of z^i, $i = -q:q$) of the matrix Laurent series $H(z) = \sum_{-\infty}^{+\infty} z^i H_i$ such that $F(z)H(z) = I$, where $F(z)$ is the Laurent matrix polynomial of (4.8).

Indeed, from (4.8)–(4.10) and (4.11), it follows that

$$F(z) = V(z)V\left(z^{-1}\right)G^2$$

for $G = -\prod_{i=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor}(I + \omega_p^i X)$, and

$$V(z) = -S(z) = \prod_{i=-\lfloor q/2\rfloor}^{\lfloor q/2\rfloor} \bigl(zI - \bigl(I + \omega_p^i X\bigr)^{-1}\bigl(I - \omega_p^i X\bigr)\bigr).$$

Therefore the polynomial $P(z) = F(z)G^{-2} = V(z)V(z^{-1})$ satisfies the conditions of proposition 8.1 where $X_i = Y_i = (I + \omega_p^{i-\lfloor q/2\rfloor-1}X)^{-1}(I - \omega_p^{i-\lfloor q/2\rfloor-1}X)$, $i = 1:q$.

Observe also that the matrix G as well as the Laurent polynomial $P(z)$ are unknown and that the sequence $\{\widehat{P}_j(z)\}_{j\in\mathbb{N}}$ generated by applying (8.3) to $\widehat{P}_0(z) = F(z)$ is such that

$$\widehat{P}_j(z) = P_j(z)G^{2^{j+1}}.$$

Although this sequence differs from $\{P_j(z)\}_{j\in\mathbb{N}}$ up to a constant matrix factor, its computation would generate overflow since $\rho(G) > 1$.

A method to remove this drawback is to scale the Laurent polynomials generated at each step of (8.3) in the following way:

$$Q^{(0)}(z) = F(z), \qquad Q^{(i+1)}(z) = G_i^{-1}Q^{(i)}(-z)Q^{(i)}(z), \quad i \geqslant 0, \tag{8.5}$$

where G_i^{-1} is the constant coefficient of $Q^{(i)}(-z)Q^{(i)}(z)$. In this way, since the polynomials $P^{(i)}(z)$ and $Q^{(i)}(z)$ differ by a multiplicative matrix factor which is commutative, and since the constant term of $Q^{(i)}(z)$ for $i > 0$ is the identity matrix, we find that

$$P^{(i)}(z) = P_0^{(i)}Q^{(i)}(z),$$

where $P_0^{(i)}$ is the constant coefficient of $P^{(i)}(z)$. Therefore, since $P_0^{(i)} = I + \mathrm{O}(\sigma^{2\cdot 2^i})$, the sequence $\{Q^{(i)}(z)\}_{i\in\mathbb{N}}$ shares the asymptotic properties of the sequence $\{P^{(i)}(z)\}_{i\in\mathbb{N}}$, that is,

$$\bigl\|Q_0^{(i)} - I\bigr\| = \mathrm{O}\bigl(\sigma^{2^i}\bigr), \qquad \bigl\|Q_j^{(i)}\bigr\| = \mathrm{O}\bigl(\sigma^{2^i}\bigr), \quad j \geqslant 1. \tag{8.6}$$

Now we are ready to describe the algorithm for computing the central coefficients of the matrix Laurent series $H(z)$ such that $F(z)H(z) = I$. Pre-multiplying both sides of the equation $F(z)H(z) = I$ by $G_0^{-1}F(-z)$, where G_0 is the constant coefficient of $F(-z)F(z)$, yields

$$Q^{(1)}\bigl(z^2\bigr)H(z) = G_0^{-1}F(-z).$$

Pre-multiplying the above equation by $G_1^{-1}Q^{(1)}(-z^2)$ yields

$$Q^{(2)}\bigl(z^4\bigr)H(z) = G_1^{-1}Q^{(1)}\bigl(-z^2\bigr)F(-z).$$

Repeating this process i times yields

$$Q(i)\bigl(z^{2^i}\bigr)H(z) = G_{i-1}^{-1}Q^{(i-1)}\bigl(-z^{2^{(i-1)}}\bigr)\cdots G_1^{-1}Q^{(1)}\bigl(-z^2\bigr)G_0^{-1}F(-z) =: K^{(i)}(z) \tag{8.7}$$

and, from propositions 8.1 and (8.6), since $Q^{(i)}(z^{2^i})$ converges to the constant polynomial I, we deduce that the matrix coefficients of the matrix Laurent polynomial $K^{(i)}(z)$

on the right-hand side of (8.7) converge doubly exponentially to the corresponding coefficients of $H(z)$.

For the computation of the central $2q+1$ coefficients of $K^{(i)}(z)$ we may apply the technique of [4], which we recall below. Let us define

$$L^{(j)}(z) = G_{i-1}^{-1} Q^{(i-1)}\left(-z^{2^{j-1}}\right) \cdots G_{i-j}^{-1} Q^{(i-j)}(-z)$$

so that $L^{(1)}(z) = G_{i-1}^{-1} Q^{(i-1)}(-z)$ and $L^{(i)}(z) = K^{(i)}(z)$, where we assume $Q^{(0)}(z) = F(z)$. Then the following equations hold:

$$L^{(j)}(z) = L^{(j-1)}\left(z^2\right) G_{i-j}^{-1} Q^{(i-j)}(-z), \quad j = 2:i, \tag{8.8a}$$

$$L^{(1)}(z) = G_{i-1}^{-1} Q^{(i-1)}(-z). \tag{8.8b}$$

Since $L^{(j)}(z)$ is the product of a Laurent polynomial in z^2 with a Laurent polynomial in z that has coefficients in the range $-q:q$, the $2q+1$ central coefficients of this product only depend on the central $2q+1$ coefficients of the two factors. Therefore, in order to compute the $2q+1$ central coefficients of $L^{(i)}(z) = K^{(i)}(z)$ we have to compute only the $2q+1$ central coefficients of the matrix polynomials $L^{(j)}(z)$ for $j = 2:i$. This computation requires computing i products of matrix Laurent polynomials having $2q+1$ coefficients. Each product can be computed by means of FFT-based fast polynomial arithmetic in $\mathrm{O}(n^2 q \log_2 q + qn^3)$ flops, i.e., by applying the evaluation interpolation technique at the Nth roots of 1, where N is the minimum integer power of 2 greater than or equal to $2q+1$.

Below we synthesize the algorithm for the computation of the central coefficients of $H(z) = F(z)^{-1}$ based on the previous arguments.

Algorithm 8.2 (Inversion by Graeffe iteration).
INPUT: The coefficients $F_0, \ldots, F_q$ of $F(z)$; an error tolerance $\varepsilon > 0$.
OUTPUT: Approximations to the coefficients H_i, $i = 0:q-1$, of the matrix Laurent series $H(z) = F(z)^{-1}$.
COMPUTATION:

1. Compute the coefficients $Q_{-q}^{(i)}, \ldots, Q_q^{(i)}$ of the matrix polynomials (8.5) for $i = 0, 1, \ldots, h-1$, together with the matrices G_i, until $\|Q^{(i)}(z) - I\|_\infty \leqslant \varepsilon$.
2. Compute the $2q-1$ central coefficients of the matrix Laurent polynomials $L^{(j)}(z)$ of (8.8a) and (8.8b) for $j = 2:h$.
3. Output the coefficients of $L^{(h)}(z)$.

9. Computing the Wiener–Hopf factorization

The Wiener–Hopf factorization of the matrix Laurent polynomial $F(z)$ of (4.8) used in algorithm 4.1 for approximating $A^{1/p}$, can be computed once the coefficients $H_0, H_1, \ldots, H_q$ of $H(z) = F(z)^{-1}$ have been approximated. In fact, from [6, theorem 4], we deduce the following result.

Proposition 9.1. Consider the Laurent matrix polynomial $F(z) = z^{-q}\Psi(z)$ of (4.8) and let $H(z) = F(z)^{-1} = \sum_{i=-\infty}^{+\infty} z^i H_i$. Then a Wiener–Hopf factorization of $F(z)$,

$$F(z) = \widehat{S}(z^{-1})\widehat{S}(z),$$

is obtained by solving the block $(m+1) \times (m+1)$ Toeplitz system

$$T_m \begin{bmatrix} X_0 \\ X_1 \\ \vdots \\ X_m \end{bmatrix} = \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \qquad T_m = \begin{bmatrix} H_0 & H_1 & \dots & H_m \\ H_1 & H_0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & H_1 \\ H_m & \dots & H_1 & H_0 \end{bmatrix},$$

where $m \geqslant q$, and by setting $\widehat{S}(z) = \sum_{j=0}^{q} z^{q-j} X_j$. Moreover we have $X_j = 0$ for $j = q+1:m$.

In this way the computation of the coefficients of $\widehat{S}(z)$ is reduced to approximating the coefficients H_j of $H(z)$ for $j = 0:m$ and to solving a block Toeplitz system.

An alternative way of computing the Wiener–Hopf factor is by means of cyclic reduction applied to a block tridiagonal block Toeplitz matrix.

Proposition 9.2. Define U, V, W the $(q+1) \times (q+1)$ block Toeplitz matrices

$$\begin{aligned} U &= \left(\binom{p}{q+i-j}\left(A + (-1)^{i-j} I\right)\right)_{i,j=1:q+1}, \\ V &= \left(\binom{p}{p+i-j+1}\left(A + (-1)^{i-j} I\right)\right)_{i,j=1:q+1}, \\ W &= \left(\binom{p}{i-j-1}\left(A + (-1)^{i-j} I\right)\right)_{i,j=1:q+1}, \end{aligned}$$

where $\binom{p}{m} = 0$ if $m < 0$ or $m > p$. Define the sequences

$$\begin{aligned} U_{k+1} &= U_k - V_k U_k^{-1} W_k - W_k U_k^{-1} U_k, \\ V_{k+1} &= -V_k U_k^{-1} V_k, \\ W_{k+1} &= -W_k U_k^{-1} W_k, \end{aligned}$$

for $k = 0, 1, \dots$, where $U_0 = U$, $V_0 = V$, $W_0 = W$, and where we assume that U_k is nonsingular for any k. Then the limit $U^* = \lim_{k\to\infty} U_k$ exists and $U^* = T_q^{-1}$, where T_q is the block Toeplitz matrix defined in proposition 9.1. Moreover, the convergence of U_k to U^* is quadratic.

For proposition 9.1, the first block column of U^* provides the coefficients of a Wiener–Hopf factor $\widehat{S}(z)$.

We may regard the blocks U_k, V_k and W_k of proposition 9.2 as the blocks forming the Schur complement generated at the kth step of cyclic reduction [9] applied to the semi-infinite block tridiagonal block Toeplitz matrix having subdiagonal, diagonal and superdiagonal blocks U, V, W, respectively [3]. This fact allows one to deduce stability and conditioning properties of the sequences U_k, V_k, W_k using linear algebra tools.

Summing up, a Wiener–Hopf factorization can be computed by applying proposition 9.1 or by means of cyclic reduction. In the former case, once the coefficients H_i, $i = 0:q$, have been computed, a block $(q+1) \times (q+1)$ Toeplitz system must be solved. The cost of this computation ranges from $O(q^2n^3)$ flops to $O(qn^3 \log q)$ flops. In our specific case the blocks of the matrix commute and might have further special structures. This property can in principle be exploited for reducing the cost of solving this system. We do not analyze this computational issue in this paper.

Concerning cyclic reduction, at each step a Toeplitz-like matrix must be inverted. This operation has a cost in the same range as the cost of inverting a block Toeplitz matrix. However, the specific structure of the blocks might reduce the complexity of the algorithm. This is a subject of future research. At the moment the algorithm for computing the pth root of a matrix that has the minimum complexity appears to be the one based on inverting a matrix Laurent polynomial by means of Graeffe's iteration.

10. Numerical experiments

We have implemented the algorithms described in the previous sections in Fortran 90 in double precision (unit roundoff $u = 2^{-53} \approx 1.1 \times 10^{-16}$) and compared them with the `rootm` function of [14] which implements the Smith algorithm. More precisely, we report the results of our experiments limited to the following implementations:

- `Int`: Algorithm 2.1, based on equation (2.4).
- `Sign`: Algorithm 3.1 where the sign function is computed by means of the matrix sign iteration $X_{k+1} = \frac{1}{2}(X_k + X_k^{-1})$ and X_k^{-1} is computed by using the appropriate LAPACK subroutine.
- `Li-ei`: Laurent polynomial inversion by means of evaluation/interpolation. This is algorithm 5.1, where the Laurent polynomial inversion is performed with algorithm 8.1 based on evaluation/interpolation.

In these preliminary experiments we have considered the following two test problems:

- Test 1. The matrix A is the unit ε-circulant matrix, that is, the companion matrix associated with the polynomial $x^n - \varepsilon$ where $n = 5$ and $\varepsilon = 10^{-8}$. The eigenvalues of A are the fifth roots of unity multiplied by $\varepsilon^{1/5}$. The matrix is normal and its limit for $\varepsilon \to 0$ has no pth root.
- Test 2. The matrix A is the 5×5 companion matrix associated with the polynomial $\prod_{i=1}^{5}(x - i)$. Its eigenvalues are clearly 1, 2, 3, 4, 5. The matrix is nonnormal.

In figures 2 and 3 we report the infinity norm of the residual error $A - X^p$ for several values of p, where X is the computed approximation of $A^{1/p}$. In the case of

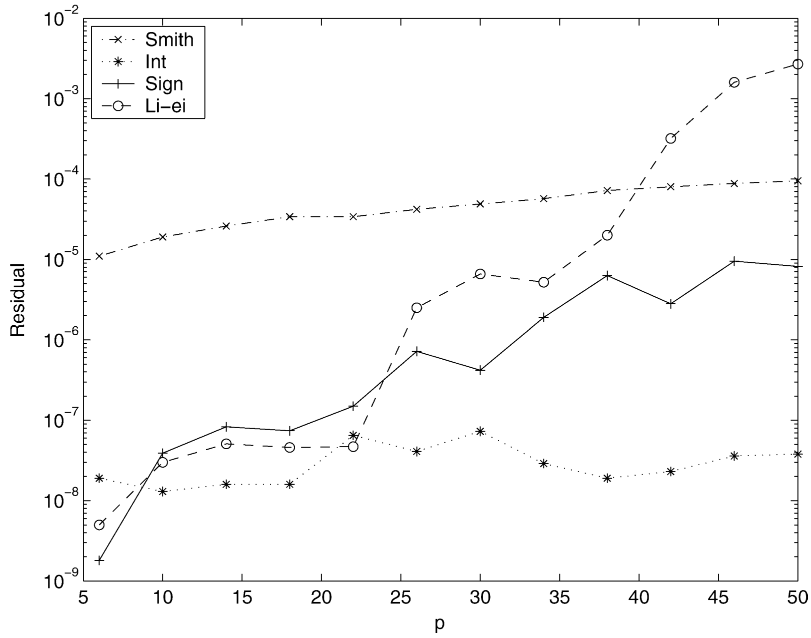

Figure 2. Infinity norm of the residual errors in computing $A^{1/p}$ for an ε-circulant matrix A.

test 1, the residual errors of the methods `Int` and `Sign` are much less than the residual errors of the Smith method, while the method based on Laurent polynomial inversion deteriorates significantly as p grows. This can be explained by the possibly large cancellation which may occur in the computation of the linear combinations of the coefficients of the Laurent series since the constants involved in the combination are binomials.

In the case of the companion matrix the Smith method is more stable even though the performance of both the methods based on numerical integration and on matrix sign are still comparable with Smith's method.

Other tests have been performed concerning the implementation of the methods based on fast A-circulant inversion, Wiener–Hopf factorization, Graeffe iteration and cyclic reduction. The results have shown a deterioration of the numerical behavior as p or n grow. More investigation is needed for these methods.

11. Conclusions and open problems

We have introduced a variety of formulas for expressing the principal pth root of a matrix A in different forms. They reduce the computation of $A^{1/p}$ to numerical integration on the unit circle, to computing the matrix sign function of a block companion matrix, to inverting a matrix Laurent polynomial, to computing a Wiener–Hopf factorization, and to applying a fixed point iteration.

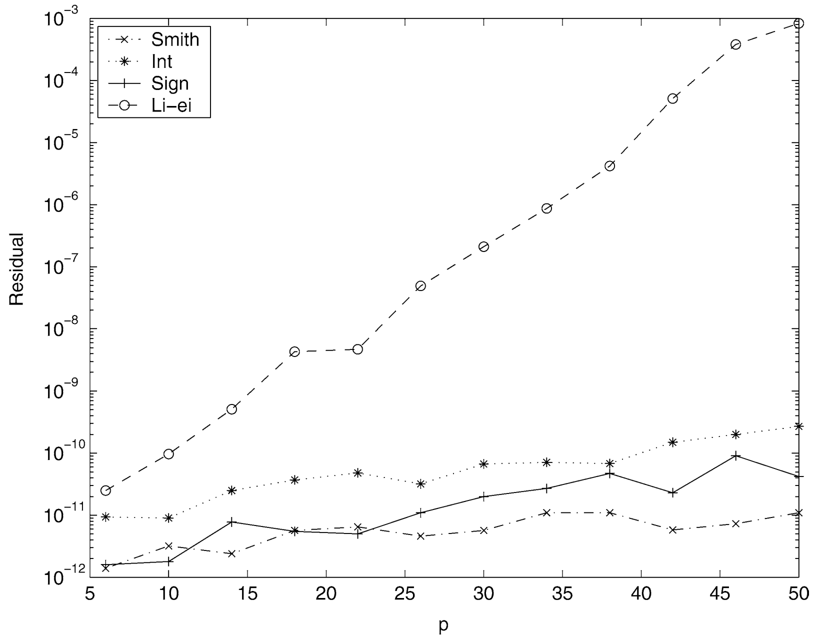

Figure 3. Infinity norm of the residual errors in computing $A^{1/p}$ for the companion matrix A associated with the polynomial $\prod_{i=1}^{5}(x-i)$.

Numerical experiments with preliminary implementations of our new algorithms show some of them to behave well and others to suffer numerical instability. Further analysis and experimentation is needed to understand and improve the finite precision behaviour.

References

[1] P. Benner, R. Byers, V. Mehrmann and H. Xu, A unified deflating subspace approach for classes of polynomial and rational matrix equations, Preprint SFB393/00-05, Zentrum für Technomathematik, Universität Bremen, Bremen, Germany (January 2000).

[2] R. Bhatia, *Matrix Analysis* (Springer, New York, 1997).

[3] D.A. Bini, L. Gemignani and B. Meini, Computations with infinite Toeplitz matrices and polynomials, Linear Algebra Appl. 343/344 (2002) 21–61.

[4] D.A. Bini, L. Gemignani and B. Meini, Solving certain matrix equations by means of Toeplitz computations: Algorithms and applications, in: *Fast Algorithms for Structured Matrices: Theory and Applications*, ed. V. Olshevsky, Contemporary Mathematics, Vol. 323 (Amer. Math. Soc., Providence, RI, 2003) pp. 151–167.

[5] D.A. Bini and B. Meini, Improved cyclic reduction for solving queueing problems, Numer. Algorithms 15(1) (1997) 57–74.

[6] D.A. Bini and B. Meini, Non-skip-free $M/G/1$-type Markov chains and Laurent matrix power series, Linear Algebra Appl. 386 (2004) 187–206.

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

[8] A. Böttcher and B. Silbermann, Introduction to Large Truncated Toeplitz Matrices (Springer, New York, 1999).
[9] B.L. Buzbee, G.H. Golub and C.W. Nielson, On direct methods for solving Poisson's equations, SIAM J. Numer. Anal. 7(4) (1970) 627–656.
[10] S.H. Cheng, N.J. Higham, C.S. Kenney and A.J. Laub, Approximating the logarithm of a matrix to specified accuracy, SIAM J. Matrix Anal. Appl. 22(4) (2001) 1112–1125.
[11] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, 2nd ed. (Academic Press, London, 1984).
[12] M.A. Hasan, A.A. Hasan and K.B. Ejaz, Computation of matrix *n*th roots and the matrix sector function, in: *Proc. of the 40th IEEE Conf. on Decision and Control*, Orlando, FL (2001) pp. 4057–4062.
[13] M.A. Hasan, J.A.K. Hasan and L. Scharenroich, New integral representations and algorithms for computing *n*th roots and the matrix sector function of nonsingular complex matrices, in: *Proc. of the 39th IEEE Conf. on Decision and Control*, Sydney, Australia (2000) pp. 4247–4252.
[14] N.J. Higham, The Matrix Computation Toolbox, `http://www.ma.man.ac.uk/~higham/mctoolbox`.
[15] N.J. Higham, Newton's method for the matrix square root, Math. Comp. 46(174) (1986) 537–549.
[16] N.J. Higham, The matrix sign decomposition and its relation to the polar decomposition, Linear Algebra Appl. 212/213 (1994) 3–20.
[17] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed. (SIAM, Philadelphia, PA, 2002).
[18] W.D. Hoskins and D.J. Walton, A faster, more stable method for computing the *p*th roots of positive definite matrices, Linear Algebra Appl. 26 (1979) 139–163.
[19] C.S. Kenney and A.J. Laub, Condition estimates for matrix functions, SIAM J. Matrix Anal. Appl. 10(2) (1989) 191–209.
[20] Ç.K. Koç and B. Bakkaloğlu, Halley's method for the matrix sector function, IEEE Trans. Automat. Control 40(5) (1995) 944–949.
[21] M.L. Mehta, *Matrix Theory: Selected Topics and Useful Results*, 2nd ed. (Hindustan Publishing, Delhi, 1989).
[22] B. Meini, The matrix square root from a new functional perspective: Theoretical results and computational issues, Technical Report 1455, Dipartimento di Matematica, Università di Pisa (2003), to appear in SIAM J. Matrix Anal. Appl.
[23] M.A. Ostrowski, Recherches sur la méthode de Graeffe et les zeros des polynômes et des séries de Laurent, Acta Math. 72 (1940) 99–257.
[24] B. Philippe, An algorithm to improve nearly orthonormal sets of vectors on a vector processor, SIAM J. Algebra Discrete Methods 8(3) (1987) 396–403.
[25] G. Schulz, Iterative Berechnung der reziproken Matrix, Z. Angew. Math. Mech. 13 (1933) 57–59.
[26] L.-S. Shieh, Y.T. Tsay and R.E. Yates, Computation of the principal *n*th roots of complex matrices, IEEE Trans. Automat. Control 30(6) (1985) 606–608.
[27] M.I. Smith, A Schur algorithm for computing matrix *p*th roots, SIAM J. Matrix Anal. Appl. 24(4) (2003) 971–989.
[28] J.S.H. Tsai, L.S. Shieh and R.E. Yates, Fast and stable algorithms for computing the principal *n*th root of a complex matrix and the matrix sector function, Comput. Math. Appl. 15(11) (1988) 903–913.
[29] Y.T. Tsay, L.S. Shieh and J.S.H. Tsai, A fast method for computing the principal *n*th roots of complex matrices, Linear Algebra Appl. 76 (1986) 205–221.