Electronic Transactions on Numerical Analysis.
Volume 31, pp. 358-383, 2008.

ISSN 1068-9613.

ETNA
Kent State University
http://etna.math.kent.edu

ALGORITHMS FOR THE MATRIX SECTOR FUNCTION*

BEATA LASZKIEWICZ† AND KRYSTYNA ZIĘTAK†

Abstract. In this paper we consider algorithms for the matrix sector function, which is a generalization of the matrix sign function. We develop algorithms for computing the matrix sector function based on the (real) Schur decompositions, with and without reordering and the Parlett recurrence. We prove some results on the convergence regions for the specialized versions of Newton's and Halley's methods applied to the matrix sector function, using recent results of Iannazzo for the principal matrix pth root. Numerical experiments comparing the properties of algorithms developed in this paper illustrate the differences in the behaviour of the algorithms. We consider the conditioning of the matrix sector function and the stability of Newton's and Halley's methods. We also prove a characterization of the Fréchet derivative of the matrix sector function, which is a generalization of the result of Kenney and Laub for the Fréchet derivative of the matrix sign function, and we provide a way of computing it by Newton's iteration.

Key words. matrix sector function, matrix sign function, matrix pth root, Schur algorithm, Parlett recurrence, Newton's method, Halley's method, stability, conditioning, Fréchet derivative.

AMS subject classifications. 65F30.

1. Introduction.

Matrix functions play an important role in many applications; see, for example, [7, Chapter 2]. In this paper we are concerned with the matrix sector function, developed by Shieh, Tsay and Wang [18], as a generalization of the matrix sign function introduced by Roberts [17]. Let p be a positive integer and let us consider the following sectors of the complex plane $\mathbb{C}$ for $l \in \{0, \ldots, p-1\}$:

$$\Phi_l = \left\{ z \in \mathbb{C} \setminus \{0\} : \frac{2l\pi}{p} - \frac{\pi}{p} < \arg(z) < \frac{2l\pi}{p} + \frac{\pi}{p} \right\}. \tag{1.1}$$

Let $\lambda = |\lambda| e^{i\varphi} \in \mathbb{C} \setminus \{0\}$, where

$$\varphi \in [0, 2\pi), \quad \varphi \neq \frac{2l\pi}{p} + \frac{\pi}{p}, \quad \text{for } l = 0, \ldots, p-1. \tag{1.2}$$

The scalar p-sector function of $\lambda \in \Phi_l$ is defined as

$$s_p(\lambda) = e^{i2\pi l/p}. \tag{1.3}$$

Hence $s_p(\lambda)$ is the pth root of unity, which lies in the same sector Φ_l in which λ lies; therefore it is the pth root of unity nearest to λ. From (1.2) we deduce that the scalar p-sector function is not defined for the pth roots of nonpositive real numbers.

Let $\mathbb{R}^-$ denote the closed negative real axis. The principal pth root $a^{1/p}$ of $a \in \mathbb{C} \setminus \mathbb{R}^-$ lies within Φ_0, therefore $\mathrm{Re}(a^{1/p}) > 0$. As shown in [18], $s_p(\lambda) = \lambda/(\lambda^p)^{1/p}$. The principal pth root of λ^p exists because λ satisfies (1.2).

Any matrix $A \in \mathbb{C}^{n \times n}$ can be expressed in the Jordan canonical form (see, for example, [7, Section 1.2])

$$W^{-1} A W = J, \tag{1.4}$$

*Received January 10, 2008. Accepted February 16, 2009. Published online on September 30, 2009. Recommended by Zdeněk Strakoš.

†Institute of Mathematics and Computer Science, Wrocław University of Technology, 50-370 Wrocław, Poland ({`Beata.Laszkiewicz,Krystyna.Zietak`}`@pwr.wroc.pl`). The work of the second author was partially supported by EC FP6 MC-ToK programme TODEQ, MTKD-CT-2005-030042 of the Institute of Mathematics of the Polish Academy of Sciences and the grant 342346 of the Institute of Mathematics and Computer Science, Wrocław University of Technology.

ETNA
Kent State University
http://etna.math.kent.edu

where W is nonsingular and J is a block diagonal matrix of Jordan blocks of the form

$$\begin{bmatrix} \lambda & 1 & & \\ & \lambda & \ddots & \\ & & \ddots & 1 \\ & & & \lambda \end{bmatrix}.$$

Let $\{\lambda_1, \ldots, \lambda_n\}$ be the spectrum of a nonsingular matrix A, with not necessarily distinct eigenvalues λ_j, satisfying (see (1.2))

$$\arg(\lambda_j) \neq \frac{2l\pi}{p} + \frac{\pi}{p}, \quad \text{for } l = 0, \ldots, p-1, \tag{1.5}$$

and ordered in such a way that the main diagonal of J is $(\lambda_1, \ldots, \lambda_n)$. Then the matrix sector function of A can be defined as (see [13, 18])

$$\operatorname{sect}_p(A) = W \operatorname{diag}(s_p(\lambda_1), \ldots, s_p(\lambda_n)) W^{-1}. \tag{1.6}$$

In principle, we could compute the matrix sector function from the Jordan form (1.4). However, the use of the Jordan form is avoided since it is computationally unattractive because of its computational cost and the possible ill-conditioning of W if A is not Hermitian.

The principal matrix pth root of A, denoted by $A^{1/p}$, is the unique matrix X such that $X^p = A$ and the eigenvalues of X lie within the sector Φ_0; see, for example, [7, Section 7]. If A has no eigenvalue on $\mathbb{R}^-$, then $A^{1/p}$ exists. The matrix sector function can be expressed in the following way (see [18])

$$\operatorname{sect}_p(A) = A(A^p)^{-1/p}, \tag{1.7}$$

where $(A^p)^{1/p}$ is the principal matrix pth root of A^p. Therefore we can compute the matrix sector function using algorithms for principal matrix pth roots. However, we would like to develop algorithms for computing directly the matrix sector function without computing the matrix pth root. This is a goal of this paper.

In the paper we show how some theoretical results and algorithms, known for the matrix sign function and the matrix pth root, can be extended to the matrix sector function. The matrix sector function is interesting for us, because these generalizations are not always easy and possible. Some basic properties of the matrix sector function are recalled in Section 2.

In Section 3 we derive an algorithm for computing the matrix sector function based on the Schur decomposition. We call this algorithm the *complex Schur algorithm*. It is a generalization of the Schur method by Higham [7, Section 5.2] for computing the matrix sign function. In the complex Schur algorithm we use also some ideas from the Smith method for the matrix pth root (see [19, 20]). The complex Schur algorithm is applicable to every A for which the matrix sector function exists. We also propose its modification employing the reordering of the Schur decomposition.

The *real Schur algorithm* for computing the matrix sector function of a real matrix in real arithmetic is stated in Section 3. It employs a real Schur decomposition. The algorithm is partly similar to the method of Smith [20] for computing some real primary matrix pth root of a real matrix (see also [7, Section 7.2]). Unfortunately, the real Schur algorithm fails when A has multiple complex eigenvalues in the sectors Φ_l different from Φ_0 and $\Phi_{p/2}$ (if p is even). The reordered real Schur algorithm is also mentioned. Numerical experiments comparing the above four versions of the Schur algorithm for the matrix sector function are presented in Section 6.

In Section 4 we consider Newton's and Halley's methods for computing the matrix sector function. These methods were formulated in [13] and [18], respectively, but without any information about the regions of convergence. We prove some results on convergence regions, using the recent results of Iannazzo [9, 11] for the matrix principal pth root. Our proof is based on a similar trick, which was used in [2, Section 6]. To our best knowledge the convergence regions for the specialized versions of Newton's and Halley's methods applied to the matrix sector function were unknown up till now. The results of Iannazzo concern Newton's and Halley's scalar iterations for the principal pth root of a number a with starting point equal to 1. For the scalar sector function $s_p(\lambda)$ we have an opposite situation: $a = 1$ and the starting point is equal to λ; see Section 4. The stability of Newton's and Halley's methods for the matrix sector function follows from the general result of Iannazzo [10, Section 4.6] for matrix iterations.

The conditioning of the matrix sector function is considered in Section 5. We generalize to the matrix sector function a characterization of the Fréchet derivative proven by Kenney and Laub for the matrix sign function. We also show that Newton's iteration provides a way of computing the Fréchet derivative of the matrix sector function. It generalizes the result of Kenney and Laub for the matrix sign function (see [7, Section 5.3] and [12]) and it is related to the very recent result of Al-Mohy and Higham [1] for computing the Fréchet derivative of a matrix function by iterative methods.

Numerical tests, presented in Section 6, illustrate the differences in the behaviour of all algorithms developed in this paper. We also include results for the matrix sector function computed directly from (1.6) and from (1.7), where $X^{1/p}$ is computed by the standard `MATLAB` matrix power operator, which uses the eigensystem of X.

In the whole paper we assume that A satisfies the conditions (1.5), i.e., that the matrix sector function exists for A.

2. Properties of the matrix sector function.

A research monograph by Higham [7] presents the theory of matrix functions and numerical methods for computing them, as well as an overview of applications. Some properties of the matrix sector function are common for matrix functions.

Let $S = \mathrm{sect}_p(A)$. The following relations

$$AS = SA, \quad S^p = I, \tag{2.1}$$

lead to the algorithms for computing the matrix sector function. The second equality in (2.1) means that S is some pth root of the identity matrix. However, it is the principal pth root, i.e., the identity matrix I, only when all eigenvalues of A lie in the sector Φ_0.

Using (1.7), we can express A in the following way:

$$A = SN, \tag{2.2}$$

where $N = S^{-1}A = S^{p-1}A = (A^p)^{1/p}$, because S commutes with A and since the spectrum of $S^{-1}A$ lies in the sector Φ_0. We call the expression (2.2) the *matrix sector decomposition*, and we will use it in Section 5 to characterize the Fréchet derivative of the matrix sector function.

If $B = V^{-1}AV$, then $\mathrm{sect}_p(B) = V^{-1}\,\mathrm{sect}_p(A)V$ for arbitrary nonsingular $V \in \mathbb{C}^{n\times n}$. The inverse of $\mathrm{sect}_p(A)$ is equal to the matrix sector function of A^{-1}.

The formula (1.7) is well-known for the matrix sign function ($p = 2$), and the decomposition (2.2) is a generalization of the matrix sign decomposition, introduced by Higham [6].

Some applications of the matrix sector function are mentioned in [18].

ETNA
Kent State University
http://etna.math.kent.edu

3. The Schur algorithms. In this section we apply the Schur decompositions to compute the matrix sector function. The Schur decomposition of $A \in \mathbb{C}^{n\times n}$

$$A = QRQ^H, \quad Q \text{ unitary}, \; R \text{ upper triangular}, \tag{3.1}$$

and the real Schur decomposition of $A \in \mathbb{R}^{n\times n}$

$$A = QRQ^T, \quad Q \text{ orthogonal}, \; R \text{ real upper quasi-triangular}, \tag{3.2}$$

are useful tools for matrix functions, because they can be computed with backward stability by the QR method; see, for example, [4]. Here M^T denotes the transpose of M and M^H denotes the conjugate transpose. In (3.2) the upper quasi-triangular matrix R has $m \le n$ blocks on the main diagonal of orders m_i equal to either 1 or 2. The blocks of orders 2 correspond to complex conjugate eigenvalues pairs. The blocks of R are denoted by R_{ij}. In the Schur decomposition (3.1) all blocks of R are 1×1.

We now show how some ideas of the Schur method by Higham [7, Section 5.2] for computing the matrix sign function, and the Smith method [20] for any primary matrix pth root can be applied to the matrix sector function. Both methods utilize the Schur decompositions and a fast Parlett recursion [16].

The significance of the Schur decompositions is that computing a matrix function $f(A)$ reduces to computing $f(R)$, since $f(A) = Qf(R)Q^H$ when we use the Schur decomposition (3.1), and $f(A) = Qf(R)Q^T$ when we use the real Schur decomposition (3.2). Therefore we focus on computing the matrix sector function of an upper (quasi) triangular matrix R determined in (3.1) and (3.2), respectively. We recall that we have assumed that $\mathrm{sect}_p(A)$ exists, hence $\mathrm{sect}_p(R)$ also exists.

Let $U = \mathrm{sect}_p(R)$. The main diagonal blocks U_{ii} of U are equal to $\mathrm{sect}_p(R_{ii})$. The superdiagonal blocks U_{ij} satisfy the following recurrence relation, derived by Parlett [16] for a matrix function of a block upper triangular matrix,

$$R_{ii}U_{ij} - U_{ij}R_{jj} = U_{ii}R_{ij} - R_{ij}U_{jj} + \sum_{k=i+1}^{j-1} (U_{ik}R_{kj} - R_{ik}U_{kj}), \tag{3.3}$$

for $i < j$. This recurrence comes from equating blocks in the commutativity relation $UR = RU$ (compare (2.1)) and it can be applied to compute superdiagonal blocks U_{ij}, provided that we may evaluate the main diagonal blocks U_{ii}, U_{jj}, and solve the Sylvester equation (3.3) for U_{ij}. The unique solution U_{ij} of (3.3) exists provided that R_{ii} and R_{jj} have no eigenvalues in common.

If the blocks R_{ii} and R_{jj} have a common eigenvalue, then we have to apply the relation $U^p = I$ (compare the second relation in (2.1)) in order to compute the appropriate remaining blocks of the matrix sector function of R. For this purpose we use some ideas from the method of Smith [20] for computing the pth root Y of R. In the method of Smith [20] the superdiagonal blocks of Y are evaluated from the recurrence that follows from the equality $Y^p = R$. A similar recurrence holds also for the matrix sector function U of R, because $U^p = I$. The only differences between these recurrence relations are in computing the main diagonal blocks of Y and U, respectively, and in the superdiagonal blocks of R. In the method of Smith the main diagonal blocks of Y are the appropriate primary pth roots of the main diagonal blocks of R. However, in our algorithm the main diagonal blocks of U are equal to the matrix sector functions of the main diagonal blocks of R. The superdiagonal blocks of R in the Smith recurrence relation are replaced in our algorithm by superdiagonal blocks of I, which are zero. Therefore we write the following generalized Sylvester equation for U_{ij}

ETNA
Kent State University
http://etna.math.kent.edu

without details of derivation, because this is the same as in the proof of Smith [20, Section 4] for the matrix pth roots; see also [19]. We use the same notations as in the formulation of the Smith method in [7, Section 7.2]:

$$\sum_{k=0}^{p-1} V_{ii}^{(p-2-k)} U_{ij} V_{jj}^{(k-1)} = -\sum_{k=0}^{p-2} V_{ii}^{(p-3-k)} B_k, \tag{3.4}$$

where

$$V_{jj}^{(k)} = U_{jj}^{k+1}, \qquad \text{for } k = -1, \ldots, p-2, \tag{3.5}$$

$$B_k = \sum_{l=i+1}^{j-1} U_{il} V_{lj}^{(k)}, \qquad \text{for } k = 0, \ldots, p-2, \tag{3.6}$$

$$V_{ij}^{(k)} = \sum_{l=0}^{k} V_{ii}^{(k-l-1)} U_{ij} V_{jj}^{(l-1)} + \sum_{l=0}^{k-1} V_{ii}^{(k-2-l)} B_l, \qquad \text{for } k = 0, \ldots, p-2. \tag{3.7}$$

We are now in a position to formulate the complex Schur algorithm for the matrix sector function. Let A have the Schur decomposition (3.1), where $R = [r_{ij}]$. Let $U = \mathrm{sect}_p(R) = [u_{ij}]$. Since now all blocks of R and U in (3.3) and (3.4) are of order 1, we replace the blocks by the elements of the matrices R and U in the proper way. The matrix U is upper triangular and $u_{ii} = s_p(r_{ii})$. If $r_{ii} \neq r_{jj}$, then we can solve (3.3) for u_{ij}. If $r_{ii} = r_{jj}$, then the left hand side in (3.4) has the form $\alpha_{ij} u_{ij}$, where

$$\alpha_{ij} = \sum_{k=0}^{p-1} v_{ii}^{(p-2-k)} v_{jj}^{(k-1)} = p u_{ii}^{p-1}$$

and $\alpha_{ij} \neq 0$ because u_{ii} is the scalar sector function. Thus we can compute u_{ij} from (3.4), and all the superdiagonal elements of U can be computed from (3.3) and (3.4), respectively.

Complex Schur algorithm for the matrix sector function

Let $A \in \mathbb{C}^{n\times n}$ have eigenvalues satisfying (1.5). This algorithm computes $\mathrm{sect}_p(A)$.

Step 1. Compute a Schur decomposition $A = QRQ^H$, with $R = [r_{ij}]$ upper triangular, and check if the eigenvalues of R satisfy the assumption (1.5).

Step 2. For $j = 1, \ldots, n$

$u_{jj} = s_p(r_{jj})$

$v_{jj}^{(k)} = u_{jj}^{k+1}, \quad k = -1, \ldots, p-2$

for $i = j-1, j-2, \ldots, 1$

$$b_k = \sum_{l=i+1}^{j-1} u_{il} v_{lj}^{(k)}, \quad k = -1, \ldots, p-2$$

$$u_{ij} = \begin{cases} -\dfrac{1}{p} u_{ii} \displaystyle\sum_{k=0}^{p-2} v_{ii}^{(p-3-k)} b_k, & \text{for } u_{ii} = u_{jj} \\ r_{ij} \dfrac{u_{ii} - u_{jj}}{r_{ii} - r_{jj}} + \dfrac{\sum_{k=i+1}^{j-1} (u_{ik} r_{kj} - r_{ik} u_{kj})}{r_{ii} - r_{jj}}, & \text{for } u_{ii} \neq u_{jj} \end{cases}$$

$$v_{ij}^{(k)} = \sum_{l=0}^{k} v_{ii}^{(k-l-1)} u_{ij} v_{jj}^{(l-1)} + \sum_{l=0}^{k-1} v_{ii}^{(k-2-l)} b_l, \quad k = -1, \ldots, p-2$$

end i

end j

ETNA
Kent State University
http://etna.math.kent.edu

Step 3. $\mathrm{sect}_p(A) = QUQ^H$.

Higham has observed that the matrix sign function of an upper triangular matrix will usually have some zero elements in the upper triangle; see [7, Section 5.2]. The matrix sector function has a similar property. This follows from the general Theorem 4.11 on functions of triangular matrices in [7]. If the diagonal elements of R are grouped according to the sectors Φ_l, then $U_{jj} = \mathrm{sect}_p(R_{jj})$ is the identity matrix multiplied by the pth root of unity lying in the corresponding sector Φ_l, including all eigenvalues of the main diagonal block R_{jj} of R, and we utilize only the Parlett recurrence (3.3) to compute $U = \mathrm{sect}_p(R)$. Thus, computing the main diagonal blocks U_{jj} of U is very cheap and there is no reason to apply the generalized Sylvester equation (3.4) to compute U. Therefore, we propose the reordered complex Schur algorithm, formulated below. We underline that now the orders of the blocks R_{jj} can be large and each block R_{jj} has eigenvalues only in one sector.

Reordered complex Schur algorithm for the matrix sector function

Let $A \in \mathbb{C}^{n\times n}$ have eigenvalues satisfying (1.5). This algorithm computes $\mathrm{sect}_p(A)$.

Step 1. Compute a Schur decomposition $A = QRQ^H$, where $R = [r_{ij}]$ is upper triangular, and check if the eigenvalues of R satisfy (1.5).

Step 2. Determine the sequence of indices $l_1, l_2, \ldots, l_q$ $(0 \le l_1 < l_2 < \cdots < l_q \le p-1)$ of different sectors Φ_{l_k} in which elements r_{jj} lie. For $k = 1, \ldots, q$ compute the number t_k of the elements r_{jj}, which belong to Φ_{l_k}. Determine the vector $w = [w_1, \ldots, w_n]$, where $w_j = k$ if r_{jj} is in Φ_{l_k}.

Step 3. According to the vector w, compute the reordered Schur decomposition $A = \widetilde{Q}\widetilde{R}\widetilde{Q}^H$ and divide the triangular matrix $\widetilde{R}$ into blocks so that the block $\widetilde{R}_{kk}$ on the main diagonal is $t_k \times t_k$ $(k = 1, \ldots, q)$.

Step 4. Compute $\widetilde{U} = \mathrm{sect}_p(\widetilde{R})$ in the following way: for $k = 1, \ldots, q$ compute $\mathrm{sect}_p(\widetilde{R}_{kk})$, which are equal to the identity matrix multiplied by the adequate pth root of unity, and compute the superdiagonal blocks $\widetilde{U}_{ij}$ of $\widetilde{U}$ from equation (3.3).

Step 5. Compute $\mathrm{sect}_p(A) = \widetilde{Q}\widetilde{U}\widetilde{Q}^H$.

Computing t_k in Step 2 of the reordered complex Schur algorithm can be achieved by the function `swapping`, which is included in the function `funm` in `MATLAB` 7.6. The reordered Schur decomposition in Step 3 can be computed by the standard `MATLAB` function `ordschur`, which is available in `MATLAB` 7.6. The vector w determined in Step 2 corresponds to the vector `CLUSTERS` of cluster indices used by `ordschur`, such that all eigenvalues with the same `CLUSTERS` value form one cluster and the specified clusters are sorted in descending order along the diagonal of the triangular $\widetilde{R}$ — the cluster with highest index appears in the upper left corner. The block $\widetilde{R}_{kk}$ has eigenvalues only in Φ_{l_k}, so that $\widetilde{R}_{ii}$ and $\widetilde{R}_{jj}$ do not have a common eigenvalue, hence equation (3.3) has the unique solution $\widetilde{U}_{ij}$. Therefore, the reordered complex Schur algorithm works for all matrices A for which the matrix sector function exists.

If a real matrix A has the complex Schur decomposition (3.1), then the above methods require complex arithmetic. We now derive an algorithm for computing the matrix sector function of a real matrix A in real arithmetic, using the real Schur decomposition (3.2).

Let $A \in \mathbb{R}^{n\times n}$ have the real Schur decomposition (3.2). A formula for $\mathrm{sect}_p(R_{jj})$, for

$$R_{jj} = Z\,\mathrm{diag}(\lambda, \bar{\lambda})Z^{-1} \in \mathbb{R}^{2\times 2}, \quad \mathrm{Im}(\lambda) \neq 0,$$

can be obtained by adapting the approach used in [20] for the matrix pth roots; see also [7,

ETNA
Kent State University
http://etna.math.kent.edu

Section 7.2]. Namely, it is easily seen that

$$U_{jj} = \mathrm{sect}_p(R_{jj}) = aI + \frac{b}{\mathrm{Im}(\lambda)}(R_{jj} - \mathrm{Re}(\lambda)I),$$

where $s_p(\lambda) = a + ib$. If the block R_{jj} is 1×1, then the only element of U_{jj} is equal to the scalar sector function of the only element of R_{jj}.

The generalized Sylvester equation (3.4) can be transformed into the following linear system, applying the operator vec,

$$\sum_{k=0}^{p-1} \left(V_{ii}^{(p-2-k)} \otimes V_{jj}^{(k-1)} \right) \mathrm{vec}(U_{ij}) = -\,\mathrm{vec} \left(\sum_{k=0}^{p-2} V_{ii}^{(p-3-k)} B_k \right), \tag{3.8}$$

which has dimension 1, 2 or 4. We now check when, for $i < j$, the block U_{ij} can be computed from (3.8).

Let $\Lambda(X)$ denote the set of all eigenvalues of a matrix X and let the block U_{ii} have the order $m_i, 1 \le m_i \le 2$. The matrix of the linear system (3.8) has the eigenvalues

$$\alpha_{r,s} = \sum_{k=0}^{p-1} \nu_r^k \theta_s^{p-1-k} = \begin{cases} \dfrac{\nu_r^p - \theta_s^p}{\nu_r - \theta_s}, & \nu_r \neq \theta_s, \\ p\nu_r^{p-1}, & \nu_r = \theta_s, \end{cases}$$

where $1 \le r \le m_i, 1 \le s \le m_j$, while $\nu_r \in \Lambda(U_{ii})$ and $\theta_s \in \Lambda(U_{jj})$ are eigenvalues of the blocks U_{ii}, U_{jj}, respectively. If $\nu_r \neq \theta_s$ for some r, s, then the matrix of the linear system (3.8) is singular because $\nu_r^p = \theta_s^p = 1$. Therefore we can compute the block U_{ij} from (3.8) only if $\nu_r = \theta_s$ for all r, s. This can happen only if all eigenvalues of the blocks R_{ii} and R_{jj} lie in the same sector Φ_l. A pair of conjugate complex eigenvalues belongs to a common sector only if it is Φ_0 or $\Phi_{p/2}$ (if p is even), which covers also the case of real eigenvalues. Therefore we can compute U_{ij} from (3.4) only when a common sector is the sector Φ_0 or $\Phi_{p/2}$. This can happen only in the following cases (we denote by $\lambda(X)$ any eigenvalue of X):

(a) $m_i = m_j = 1$ and $\lambda(R_{ii}) > 0, \lambda(R_{jj}) > 0$;
(b) $m_i = m_j = 1$ and $\lambda(R_{ii}) < 0, \lambda(R_{jj}) < 0$ (if p is even);
(c) $m_i = m_j = 2$ and $\lambda(R_{ii}), \lambda(R_{jj}) \in \Phi_0$ or $\lambda(R_{ii}), \lambda(R_{jj}) \in \Phi_{p/2}$ (if p is even);
(d) $m_i + m_j = 3$ and a real eigenvalue and a pair of conjugate complex eigenvalues of R_{ii} and R_{jj}, respectively, lie in the sector Φ_0 or $\Phi_{p/2}$ (if p is even).

Thus, we can apply (3.4) only in very specific cases, and it is necessary to use also the Sylvester equation (3.3) in order to compute the matrix sector function for more general cases. We recall that the Sylvester equation (3.3) has the unique solution U_{ij} if and only if R_{ii} and R_{jj} have no eigenvalue in common. This holds only when $m_i + m_j = 3$ or if $m_i = m_j$ and $\Lambda(R_{ii}) \cap \Lambda(R_{jj}) = \emptyset$. If $m_i = m_j = 2$ and $\Lambda(R_{ii}) \cap \Lambda(R_{jj}) \neq \emptyset$, then $\Lambda(R_{ii}) = \Lambda(R_{jj})$ and we cannot use (3.3). However, we recall that if the eigenvalue of R_{ii} lies within the sector Φ_0 or within the sector $\Phi_{p/2}$, then we can compute U_{ij} from (3.4). Otherwise we can not apply either (3.4) or (3.3), and the real Schur algorithm, formulated below, does not work. Therefore, we can not apply the real Schur algorithm if A has multiple complex eigenvalues in the sectors different from Φ_0 or $\Phi_{p/2}$. In some cases, when $m_i + m_j = 3$, it is possible to apply both equations (3.4) and (3.3). In such a situation we choose the equation (3.3) because it has a simpler form.

Real Schur algorithm for the matrix sector function
Let $A \in \mathbb{R}^{n\times n}$ have no multiple complex eigenvalues in the sectors different from Φ_0 or $\Phi_{p/2}$ (if p is even) and let all eigenvalues of A satisfy (1.5). This algorithm computes $\mathrm{sect}_p(A)$.

Step 1. Compute a real Schur decomposition $A = QRQ^T$, where R is upper quasi-triangular, that is block upper triangular with m main diagonal blocks R_{jj}, and check if the eigenvalues of the main diagonal blocks of R satisfy the assumption (1.5).

Step 2. For $j = 1, \dots, m$

Compute $U_{jj} = \mathrm{sect}_p(R_{jj})$

Compute $V_{jj}^{(k)} = U_{jj}^{k+1}$ in (3.5), for $k = -1, \dots, p-2$

for $i = j-1, j-2, \dots, 1$

Compute B_k in (3.6), for $k = 0, \dots, p-2$

Compute U_{ij} in the following way

- if $m_i + m_j = 3$ or if $m_i = m_j$ and $\Lambda(R_{ii}) \cap \Lambda(R_{jj}) = \emptyset$, then solve (3.3) for U_{ij}
- if $m_i = m_j = 1$ and $\Lambda(R_{ii}) = \Lambda(R_{jj})$, then solve (3.4) for U_{ij}
- if $m_i = m_j = 2$ and R_{ii} and R_{jj} have common eigenvalues in Φ_0 or $\Phi_{p/2}$ (if p is even), then solve (3.4) for U_{ij}
- if none of the above cases holds, then exit

Compute $V_{ij}^{(k)}$ in (3.7), for $k = 0, \dots, p-2$

end i

end j

Step 3. $\mathrm{sect}_p(A) = QUQ^T$.

The idea of reordering can be applied also to the real Schur algorithm. The way of reordering the eigenvalues should be such that each main diagonal block $\widetilde{R}_{jj}$ of the reordered upper quasi-triangular matrix $\widetilde{R}$ from the reordered real Schur decomposition $A = \widetilde{Q}\widetilde{R}\widetilde{Q}^T$, has the eigenvalues lying only in one of the sectors Φ_l and the conjugate sector $\bar{\Phi}_l = \Phi_{p-l}$ to Φ_l, or just in Φ_0, or in $\Phi_{p/2}$ for even p. The main difference between the real and complex reordered Schur algorithms is in the first part of Step 4. The real reordered Schur algorithm uses, for computing $\mathrm{sect}_p(\widetilde{R}_{jj})$, the real Schur algorithm, in which only (3.3) is applied. Therefore, this step is not as cheap as in the reordered complex Schur algorithm, where $\mathrm{sect}_p(\widetilde{R}_{jj}) = \epsilon_l I$. Other blocks of $\mathrm{sect}_p(\widetilde{R})$ are determined in the reordered real Schur algorithm from (3.3), hence similarly as in the reordered complex Schur algorithm by the Parlett recurrence. With regards to these analogies between algorithms we omit the formulation of the reordered real Schur algorithm for the matrix sector function. The reordered real Schur algorithm works under the same assumptions as the real Schur algorithm.

The complex Schur algorithm for the matrix sector function is a generalization of the Schur method proposed by Higham [7, Section 5.2] for the matrix sign function of a complex matrix having no pure imaginary eigenvalues. The complex Schur algorithm with reordering can be more expensive than the algorithm without reordering, because of the cost of solving the Sylvester equation (3.3) when $\widetilde{R}_{ii}$ and $\widetilde{R}_{jj}$ are of large size, and because of the cost of computing the reordered Schur decomposition.

An application of the reordered Schur decomposition to computing the matrix sign function of a complex matrix is mentioned in [7, Section 5.2], without reporting details or numerical experiments. In the reordered Schur method, the matrix sign function of $\widetilde{R}$ would have only two main diagonal blocks, equal to $\pm I$. Higham writes that the cost of the reordering may or may not be less than the cost of (redundantly) computing zero elements in the upper triangle of the matrix sign function of R; see Algorithm 5.5 in [7]. In the complex Schur algorithm for the matrix sector function, such a situation corresponds to computing the zero element u_{ij} from the first expression in Step 2, what would be redundant (compare Step 4 of the reordered complex Schur algorithm).

The reordering and blocking proposed by Davis and Higham [3] for any matrix function

ETNA
Kent State University
http://etna.math.kent.edu

are different from those developed in the above algorithms for the matrix sector function. Their algorithm has a parameter δ that is used to determine the reordering and blocking of the Schur decomposition to balance the conflicting requirements of producing small diagonal blocks and keeping the separations of the blocks large, and it is intended primarily for functions having a Taylor series.

4. Newton's and Halley's methods. Newton's and Halley's iterative methods are very popular tools for computing matrix functions; see [7]. The matrix sector function $\mathrm{sect}_p(A)$ is a pth root of the identity matrix (see the definition and (2.1)) which depends on the eigenvalues of A, and (1.7) holds. Therefore, it is obvious that there are many links between algorithms for computing the matrix pth root and the matrix sector function.

Computing the matrix sector function of A requires iterative methods for solving the matrix equation $X^p - I = 0$ with the starting matrix $X_0 = A$. For this purpose one can apply Newton's method [18]

$$X_{k+1} = \frac{1}{p}((p-1)X_k + X_k^{1-p}), \qquad X_0 = A, \tag{4.1}$$

or Halley's method [13]

$$X_{k+1} = X_k((p-1)X_k^p + (p+1)I)((p+1)X_k^p + (p-1)I)^{-1}, \qquad X_0 = A. \tag{4.2}$$

It should be noticed that the iteration (4.1) coincides with the customary Newton's method for the matrix equation $X^p - B = 0$, when the latter is defined, because $X_0 = A$ commutes with $B = I$; see [7, Section 7.3].

Newton's and Halley's matrix iterations are related to the following scalar iteration

$$x_{k+1} = g(x_k), \qquad x_0 = \lambda, \tag{4.3}$$

where λ satisfies the assumptions (1.2). For Newton's method

$$g(x) = \frac{(p-1)x^p + 1}{px^{p-1}}, \tag{4.4}$$

and for Halley's method

$$g(x) = x\frac{(p-1)x^p + (p+1)}{(p+1)x^p + (p-1)}. \tag{4.5}$$

Let the sequence (4.3) be convergent to $x_* = s_p(\lambda)$, where g is (4.4) or (4.5). It is easy to verify that x_* is an attractive fixed point. The scalar functions (4.4) and (4.5) do not depend on λ, hence the iterations (4.1) and (4.2) are pure rational matrix iterations defined in [11], because they have the form $Z_{k+1} = \varphi(Z_k)$ $(k = 0, 1, \ldots)$. Therefore, from [11, Theorem 2.4] we obtain the following corollary.

COROLLARY 4.1. *If for every eigenvalue λ_j of A, Newton's (Halley's) scalar method is convergent to the scalar sector function $s_p(\lambda_j)$, then Newton's (Halley's) matrix method is convergent to* $\mathrm{sect}_p(A)$.

Thanks to Corollary 4.1 the problem of the convergence of Newton's (Halley's) matrix iteration to the matrix sector function of A is reduced to the convergence of the corresponding scalar sequences to the scalar sector functions of the eigenvalues of A. Therefore, we can apply to the methods (4.1) and (4.2) some results proven in [9] and [10] for the matrix pth roots, to determine sets of matrices A for which Newton's and Halley's methods are convergent to $\mathrm{sect}_p(A)$.

ETNA
Kent State University
http://etna.math.kent.edu

We first consider Newton's method for the scalar sector function $s_p(\lambda)$ (see (1.3), (4.3), and (4.4))

$$x_{k+1} = \frac{1}{p}\left[(p-1)x_k + x_k^{1-p}\right], \qquad x_0 = \lambda. \tag{4.6}$$

We assume that λ satisfies the condition (1.2) given in the definition of the scalar sector function s_p. Let $u_k = x_k/s_p(\lambda)$, where x_k is determined in (4.6). Then

$$u_{k+1} = \frac{1}{p}\left[(p-1)u_k + u_k^{1-p}\right], \qquad u_0 = (\lambda^p)^{1/p}. \tag{4.7}$$

The iterates x_k converge to $s_p(\lambda)$ if and only if u_k is convergent to 1. Let Newton's method be applied to the scalar equation $y^p - a = 0$, for $a \in \mathbb{C} \setminus \mathbb{R}^-$, with the starting point equal to 1:

$$y_{k+1} = \frac{1}{p}\left[(p-1)y_k + ay_k^{1-p}\right], \qquad y_0 = 1. \tag{4.8}$$

The sequence (4.8) converges to the principal pth root of a if and only if the sequence

$$z_{k+1} = \frac{1}{p}\left[(p-1)z_k + z_k^{1-p}\right], \qquad z_0 = a^{-1/p}, \tag{4.9}$$

converges to 1; see [9]. This property follows from the relation $z_k = y_k a^{-1/p}$. The iterations (4.7) and (4.9) differ only at starting points. Therefore, the sequence (4.7) with $u_0 = (\lambda^p)^{1/p}$ is convergent to 1 if and only if the sequence (4.9) with $z_0 = a^{-1/p}$ is convergent to 1, for a and λ satisfying the relation

$$(\lambda^p)^{1/p} = a^{-1/p}. \tag{4.10}$$

Therefore, if a belongs to a set such that the iteration (4.8) with $y_0 = 1$ is well defined and converges to the principal pth root $a^{1/p}$, then the iteration (4.6) with $x_0 = \lambda$ is convergent to the scalar sector function $s_p(\lambda)$ for λ satisfying the relation (4.10). If λ satisfies (4.10), then

$$\lambda = \epsilon_l a^{-1/p}, \tag{4.11}$$

where

$$\epsilon_l = e^{2l\pi i/p}, \qquad l = 0, \dots, p-1, \tag{4.12}$$

are pth roots of unity. We have assumed that $a \notin \mathbb{R}^-$. Thus $\arg(1/a) \neq \pi$ and, consequently, $\arg(\epsilon_l a^{-1/p})$ satisfies (1.2). Hence $s_p(\lambda)$ exists for λ determined in (4.11).

Halley's iteration has the form

$$y_{k+1} = y_k \frac{(p-1)y_k^p + (p+1)a}{(p+1)y_k^p + (p-1)a}, \quad y_0 = 1, \tag{4.13}$$

for the pth root $a^{1/p}$ with the starting point 1, and

$$x_{k+1} = x_k \frac{(p-1)x_k^p + (p+1)}{(p+1)x_k^p + (p-1)}, \qquad x_0 = \lambda, \tag{4.14}$$

for the scalar sector function $s_p(\lambda)$. By similar arguments as above for Newton's method, we can show that if the iterates (4.13) converge to the pth root of a, then the iterates (4.14)

ETNA
Kent State University
http://etna.math.kent.edu

converge to $s_p(\lambda)$ for λ satisfying (4.11). We omit the details. Thus from the above considerations we obtain the following corollary.

COROLLARY 4.2. *Let Newton's method* (4.8) *(Halley's method (4.13)) be convergent to the principal pth root of* $a \notin \mathbb{R}^-$. *Then Newton's method* (4.6) *(Halley's method (4.14)) is convergent to the sector function* $s_p(\lambda)$ *for* λ *satisfying* (4.11).

Some regions of a for which the scalar Newton's and Halley's iterations, respectively, are convergent to $a^{1/p}$ with starting point 1 are known; see [7, Section 7.3], [9, Theorems 2.1 and 2.3], [10, Theorems 5.3 and 5.20], and [11, Corollary 5.3]. For Newton's iteration (4.8) it is the region

$$a \in \{z \in \mathbb{C} : \mathrm{Re}(z) > 0 \text{ and } |z| \leq 1\} \cup \mathbb{R}^+, \tag{4.15}$$

and for Halley's iteration (4.13)

$$\{z \in \mathbb{C} : \mathrm{Re}(z) > 0\}. \tag{4.16}$$

Using Corollary 4.1 and Corollary 4.2 we obtain the following convergence regions for Newton's and Halley's iterations, respectively, for the matrix sector function. We omit the proof because it is enough to show that $\lambda \in \mathbb{B}_p^{\text{(Newt)}}$ or $\lambda \in \mathbb{B}_p^{\text{(Hal)}}$ if and only if a satisfying the relation (4.11) lies in (4.15) or in (4.16), respectively.

THEOREM 4.3. *Let* $\alpha_l = 2l\pi/p - \pi/(2p)$, $\beta_l = 2l\pi/p + \pi/(2p)$, $l = 0, \ldots, p-1$.

(i) If all eigenvalues of A lie in the region

$$\mathbb{B}_p^{(Newt)} = \bigcup_{l=0}^{p-1} \left[\{z \in \mathbb{C} : |z| \geq 1, \alpha_l < \arg(z) < \beta_l\} \cup \mathbb{R}_l^+\right], \tag{4.17}$$

where $\mathbb{R}_l^+ = \{z \in \mathbb{C} : z = r\epsilon_l, r \in \mathbb{R}^+\}$, *then Newton's matrix iteration* (4.1) *is convergent to* $\mathrm{sect}_p(A)$.

(ii) If all eigenvalues of A lie in the region

$$\mathbb{B}_p^{(Hal)} = \bigcup_{l=0}^{p-1} \{z \in \mathbb{C} : \alpha_l < \arg(z) < \beta_l\}, \tag{4.18}$$

then Halley's iteration (4.2) *is convergent to* $\mathrm{sect}_p(A)$.

Iannazzo shows in [11, Theorem 6.1] that the immediate basin of attraction for the fixed point 1 of the iteration (4.6) contains the set

$$\{z \in \mathbb{C} : |z| \geq 1/2^{1/p},\ |\arg(z)| < \pi/4\}.$$

From his result we obtain the following corollary for the matrix sector function. The proof of the corollary follows from similar considerations as those before Theorem 4.3, therefore we omit it.

COROLLARY 4.4. *If all eigenvalues of A lie in the region*

$$\mathbb{C}_p^{(Newt)} = \bigcup_{l=0}^{p-1} \left\{ z \in \mathbb{C} : \frac{1}{2^{1/p}} \leq |z| \leq 1,\ \frac{2l\pi}{p} - \frac{\pi}{4p} < \arg(z) < \frac{2l\pi}{p} + \frac{\pi}{4p} \right\},$$

then Newton's iteration (4.1) *is convergent to* $\mathrm{sect}_p(A)$.

A set of complex numbers z whose principal arguments satisfy the assumption in the definition of $\mathbb{C}_p^{\text{(Newt)}}$ in Corollary 4.4, but $|z| \geq 1$, is a subset of the set $\mathbb{B}_p^{\text{(Newt)}}$ defined in Theorem 4.3.

ETNA
Kent State University
http://etna.math.kent.edu

Newton's and Halley's methods are stable in the sense considered in [7], i.e., the Fréchet derivative $L_g(X)$ has bounded powers. This follows from Theorem 4.11 in [10] on stability of pure matrix iterations. In Section 5 we consider the Fréchet derivative of the matrix sector function.

In [13] an example is provided, which shows that the sequence (4.14) can be nonconvergent to $s_p(\lambda)$, and it is mentioned that such a situation mostly occurs for matrices whose eigenvalues are near the boundaries of the sectors Φ_l; see (1.1). The starting point in the example given in [13] is not in the region of convergence (4.18) of Halley's method.

In Figure 4.1 we present the convergence regions, determined experimentally, for Newton's method (4.6) for computing $s_p(\lambda)$, and the regions $\mathbb{B}_p^{(\text{Newt})}$ and $\mathbb{C}_p^{(\text{Newt})}$ for $p = 5$. More precisely, each point $x_0 = \lambda$ in the region is coloured according to the pth root ϵ_l of unity (see (4.12)) to which the iteration converges after 100 iterations, i.e., if $|x_{100} - \epsilon_l| < 10^{-5}$. In Figure 4.2 we present the convergence regions, determined experimentally, for Halley's method (4.14) for $p = 4, 5$. We also plot the boundaries of the regions $\mathbb{B}_p^{(\text{Hal})}$.

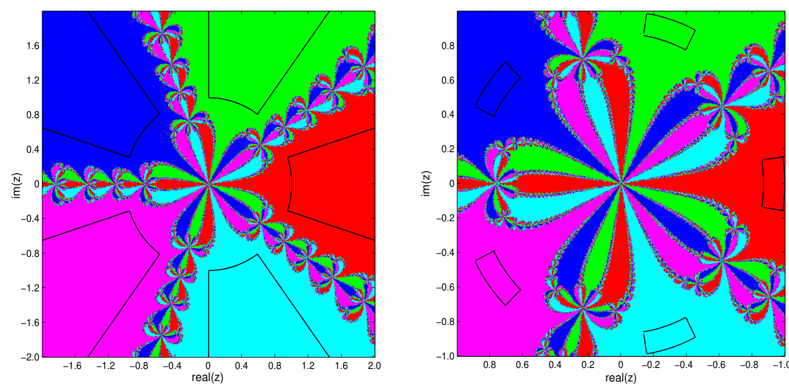


FIGURE 4.1. *Regions of convergence for Newton's method and boundaries of* $\mathbb{B}_p^{(Newt)}$ *(left) and* $\mathbb{C}_p^{(Newt)}$ *(right) for* $p = 5$.

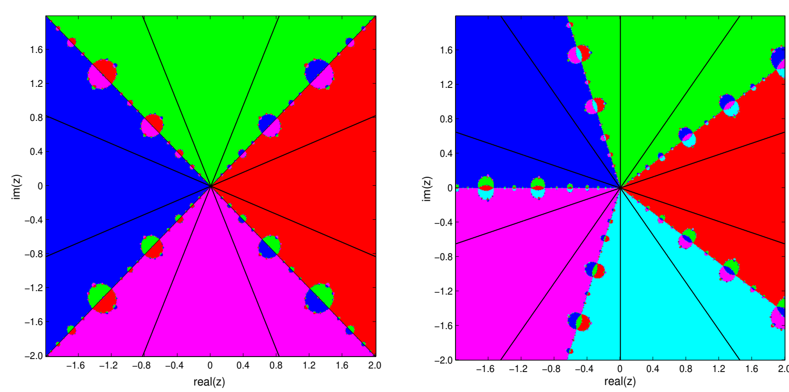


FIGURE 4.2. *Regions of convergence for Halley's method and boundaries of* $\mathbb{B}_p^{(Hal)}$, $p = 4, 5$.

The convergence regions are larger for Halley's method than for Newton's method. In [14] we discuss properties of the Padé family of iterations for the matrix sector function. Halley's method is a particular case of the Padé iterations.

We now concentrate on the rate of convergence of Newton's iterates to the matrix sector function. As mentioned in the Introduction, a generalization to the matrix sector function of some results, known for the matrix sign function, can be complicated. If either the spectral radius $\rho(A)$ of A is much larger than 1, or A has an eigenvalue close to the imaginary axis, then the convergence of Newton's iteration for the matrix sign function will be slow; see [7, Theorem 5.6]. For Newton's iteration for the matrix sector function the situation is even more complicated, as we now show.

The matrices A and $S = \mathrm{sect}_p(A)$ commute; see (2.1). Therefore, from (4.1), by the same arguments as for the matrix pth root in [7, Problem 7.11], we deduce

$$X_{k+1} - S = \frac{1}{p} X_k^{1-p}((p-1)X_k^p - pSX_k^{p-1} + I) = \frac{1}{p} X_k^{1-p}(X_k - S)^2 W_k,$$

where

$$W_k = \sum_{j=0}^{p-2} (j+1) S^{p-2-j} X_k^j. \tag{4.19}$$

This implies, for any consistent norm, the following inequality for $p \geq 2$,

$$\|X_{k+1} - S\| \leq \frac{1}{p} \|X_k^{1-p}\| \, \|X_k - S\|^2 \, \|\sum_{j=0}^{p-2} (j+1) S^{p-2-j} X_k^j\|,$$

and, for sufficiently large k, we have $\|X_{k+1} - S\| \leq c\|X_k - S\|^2$, with

$$c = (p-1)\|S^{1-p}\| \, \|S\|^{p-2} \leq (p-1)[\mathrm{cond}(S)]^{p-1}/\|S\|,$$

under the assumption that Newton's iteration converges. Thus, the convergence of Newton's iteration is asymptotically quadratic. However, the convergence can be slow, as follows from the well-known properties of the scalar Newton's iteration, since the convergence of the matrix iteration is essentially reduced to scalar convergence; see [11]. We now examine the convergence of Newton's iteration in the region $\mathbb{B}_p^{(\mathrm{Newt})}$, determined in Theorem 4.3.

Let p be even. Then

$$X_{k+1} + S = \frac{1}{p} X_k^{1-p}(X_k + S)^2 V_k,$$

where

$$V_k = \sum_{j=0}^{p-2} (-1)^j (j+1) S^{p-2-j} X_k^j. \tag{4.20}$$

Let all eigenvalues λ_j of A lie in the region $\mathbb{B}_p^{(\mathrm{Newt})}$. Then the iterates X_k are convergent to $S = \mathrm{sect}_p(A)$. The region $\mathbb{B}_p^{(\mathrm{Newt})}$ consists of separate subregions $\mathbb{R}_l^+$ and (see (1.1))

$$\Psi_l = \{z \in \mathbb{C} : |z| \geq 1, \alpha_l < \arg(z) < \beta_l\} \subset \Phi_l. \tag{4.21}$$

Modifying the proof of Lemma 2.5 in [9], we can show that an eigenvalue $\lambda_j^{(k)}$ of the Newton iterate X_k stays in Ψ_l, which includes the corresponding eigenvalue λ_j of A. If $\lambda_j \in \mathbb{R}_l^+$, then $\lambda_j^{(k)} \in \mathbb{R}_l^+$. Therefore $\mathrm{sect}_p(X_k) = S$. The matrix $X_k + S$ is nonsingular because

ETNA
Kent State University
http://etna.math.kent.edu

$-s_p(\lambda_j) \notin \Psi_l$, hence $\lambda_j^{(k)} + s_p(\lambda_j) \neq 0$. Analogously as in [7, Theorem 5.6], defining $G_k = (X_k - S)(X_k + S)^{-1}$ we can prove that $I - G_k$ is nonsingular, $X_k = (I - G_k)^{-1}(I + G_k)S$, and (see (4.19), (4.20))

$$G_{k+1} = G_k^2 W_k V_k^{-1} = G_0^{2^{k+1}} \prod_{j=0}^{k} (W_{k-j} V_{k-j}^{-1})^{2^j}, \quad k = 0, 1, \dots. \tag{4.22}$$

For any matrix norm we have

$$\|G_0^{2^k}\| \geq \rho(G_0^{2^k}) = \left(\max_{\lambda \in \Lambda(A)} \frac{|\lambda - s_p(\lambda)|}{|\lambda + s_p(\lambda)|} \right)^{2^k}.$$

Therefore, if the spectral radius $\rho(A)$ is large or small, then the convergence of $G_0^{2^k}$ to the zero matrix will be slow when $k \to \infty$. The spectral radius of A can be small only when all eigenvalues of A lie in the sets $\mathbb{R}_l^+$, because if the eigenvalues lie in Ψ_l, then $\rho(A) \geq 1$ since we have restricted the eigenvalues of A to $\mathbb{B}_p^{(\text{Newt})}$; see (4.21). Of course, the convergence of G_{k+1} to zero depends also on the behaviour of $W_{k-j} V_{k-j}^{-1}$ in (4.22). For $p = 2$ we have $W_{k-j} = V_{k-j} = I$. However, for $p > 2$ the matrix $W_{k-j} V_{k-j}^{-1}$ tends to $(p-1)I$, since we have assumed p is even.

5. Conditioning of the matrix sector function. The sensitivity of the matrix sector function $S = \text{sect}_p(A)$ with respect to perturbations of A can be determined by the norm of its Fréchet derivative. Let $f(X)$ be a matrix function, $X \in \mathbb{C}^{n \times n}$. The Fréchet derivative of f is a linear mapping such that for all $E \in \mathbb{C}^{n \times n}$ we have (see [7, Chapter 3])

$$f(X + E) - f(X) - L(X, E) = o(\|E\|).$$

The notation $L(X, E)$ should be read as "the Fréchet derivative of f at X in the direction E". Then, the absolute and relative condition numbers of $f(X)$ are given by

$$\text{cond}_{\text{abs}}(f, X) = \lim_{\epsilon \to 0} \sup_{\|E\| \leq \epsilon} \frac{\|f(X + E) - f(X)\|}{\epsilon} = \|L(X)\|,$$
$$\text{cond}_{\text{rel}}(f, X) = \text{cond}_{\text{abs}}(f, X) \frac{\|X\|}{\|f(X)\|} = \frac{\|L(X)\| \, \|X\|}{\|f(X)\|},$$

where

$$\|L(X)\| := \max_{Z \neq O} \frac{\|L(X, Z)\|}{\|Z\|}$$

is the norm of $L(X)$.

The Fréchet derivative may not exist, but if it does it is unique. Let $S + \Delta_S = \text{sect}_p(A + \Delta_A)$, where we assume that the sector function is defined on a ball of radius $\|\Delta_A\|$ and center A. The definition of the Fréchet derivative implies that

$$\Delta_S - L(A, \Delta_A) = o(\|\Delta_A)\|). \tag{5.1}$$

Following the ideas from [12] and [7, Section 5.1] for the matrix sign function, applying the relations (2.1) for the matrix sector function and

$$(A + \Delta_A)(S + \Delta_S) = (S + \Delta_S)(A + \Delta_A),$$

ETNA
Kent State University
http://etna.math.kent.edu

we obtain

$$A\Delta_S - \Delta_S A = S\Delta_A - \Delta_A S + o(\|\Delta_A\|), \tag{5.2}$$

since $\Delta_S = O(\|\Delta_A\|)$. Moreover, $(S + \Delta_S)^p = I$ gives

$$S^{p-1}\Delta_S + \Delta_S S^{p-1} + \sum_{k=1}^{p-2} S^k \Delta_S S^{-k-1} + o(\|A\|) = 0. \tag{5.3}$$

Pre-multiplying (5.2) by $S^{p-1} = S^{-1}$, using (5.3) and (2.2), gives

$$N\Delta_S + \left(\sum_{k=0}^{p-2} S^k \Delta_S S^{-k}\right) N = \Delta_A - S^{-1}\Delta_A S + o(\|\Delta_A\|). \tag{5.4}$$

This leads to the following theorem.

THEOREM 5.1. *The Fréchet derivative $L = L(A, \Delta_A)$ of the matrix sector function is the unique solution of the equation*

$$NL + \sum_{k=0}^{p-2} S^k L S^{-k} N = \Delta_A - S^{-1}\Delta_A S, \tag{5.5}$$

where $A = SN$ is the matrix sector decomposition (2.2).

Proof. The idea of the proof of the theorem is the same as of the proof of an analogous theorem for the matrix sign function; see [7, Section 5.1] and [12].

Applying the operator vec to the equation (5.5) we obtain the equation

$$M \operatorname{vec}(L) = \operatorname{vec}(\Delta_A - S^{-1}\Delta_A S). \tag{5.6}$$

Applying the well-known properties of the Kronecker product $\otimes$ to the equation (5.5), we obtain the following expression for the matrix M in (5.6)

$$M = I \otimes N + \sum_{k=0}^{p-2} (S^{-k}N)^T \otimes S^k = I \otimes N + (N^T \otimes I)\left(\sum_{k=0}^{p-2} (S^{-k})^T \otimes S^k\right).$$

The matrix A has the Jordan form (1.4). From the definition of the matrix sector function we obtain $S = WDW^{-1}$, $N = WD^{-1}JW^{-1}$, where (see (1.6) and (1.7)) $D = \operatorname{diag}(d_j)$ and $d_j = s_p(\lambda_j)$. Thus $V = W^{-T} \otimes W$ triangularizes both sides of the sum defining M. Therefore,

$$\widetilde{M} = V^{-1}MV = I \otimes (D^{-1}J) + ((J^T D^{-1}) \otimes I) \sum_{k=0}^{p-2} G_k,$$

where $I \otimes D^{-1}J = \operatorname{diag}(D^{-1}J, \ldots, D^{-1}J)$, $G_k = D^{-k} \otimes D^k = \operatorname{diag}\left(\frac{1}{d_1^k} D^k, \ldots, \frac{1}{d_n^k} D^k\right)$, and $(J^T D^{-1}) \otimes I$ is the block bidiagonal lower triangular matrix with $n \times n$ main diagonal blocks equal to $(\lambda_j^p)^{1/p} I$ for $j = 1, \ldots, n$. Thus $\widetilde{M}$ is block bidiagonal lower triangular. The main diagonal block

$$M_{jj} = D^{-1}J + (\lambda_j^p)^{1/p} \sum_{k=0}^{p-2} \frac{1}{d_j^k} D^k, \quad j = 1, \ldots, n,$$

ETNA
Kent State University
http://etna.math.kent.edu

has diagonal elements $m_{ll}^{(j)}$ equal to

$$m_{ll}^{(j)} = (\lambda_l^p)^{1/p} + (\lambda_j^p)^{1/p} \sum_{k=0}^{p-2} \left(\frac{d_l}{d_j}\right)^k, \qquad l = 1, \ldots, n.$$

Therefore,

$$m_{ll}^{(j)} = \begin{cases} (\lambda_l^p)^{1/p} + (p-1)(\lambda_j^p)^{1/p}, & \text{if } l = j \text{ or } p = 2 \text{ or } d_l = d_j, \\ (\lambda_l^p)^{1/p}(1 - \frac{\lambda_j}{\lambda_l}), & \text{otherwise.} \end{cases}$$

In the first case $m_{ll}^{(j)}$ has positive real part, because the principal pth roots have positive real parts. In the second case $m_{ll}^{(j)} \neq 0$, because $d_j \neq d_l$ implies $\lambda_j \neq \lambda_l$. Thus $\widetilde{M}$ and M are nonsingular and the equation (5.5) has the unique solution L.

The solution L is a linear function of Δ_A and, by (5.4), it differs from

$$\Delta_S = \operatorname{sect}_p(A + \Delta_A) - S$$

by $o(\|\Delta_A\|)$. Thus (5.1) implies that $L = L(A, \Delta_A)$. This completes the proof. ∎

Theorem 3.15 in [7] is applicable to the matrix sector function for diagonalizable matrices A, hence it gives an upper bound for the absolute condition number $\operatorname{cond}_{\text{abs}}(A)$ with respect to the Frobenius norm. In particular, if A is normal then Corollary 3.16 from Theorem 3.15 in [7] implies the following corollary for the matrix sector function.

COROLLARY 5.2. *Let A be normal with the spectral decomposition $A = Q \operatorname{diag}(\lambda_j) Q^H$, where Q is unitary. If all of the eigenvalues λ_j of A lie in the same sector, then $\|L\|_F = 0$. Otherwise,*

$$\operatorname{cond}_{\text{abs}}(A) = \|L\|_F = \max \frac{|s_p(\lambda_i) - s_p(\lambda_j)|}{|\lambda_i - \lambda_j|},$$

where the maximum is taken over all indices i and j such that $\lambda_i \neq \lambda_j$.

We now apply Newton's iterations to computing the Fréchet derivative of the matrix sector function.

THEOREM 5.3. *Let $A \in \mathbb{C}^{n\times n}$ be such that $\operatorname{sect}_p(A)$ exists and Newton's iterates X_k in (4.1) are convergent to $\operatorname{sect}_p(A)$. Let*

$$Y_{k+1} = \frac{1}{p}\left((p-1)Y_k - X_k^{1-p}\left(\sum_{j=0}^{p-2} X_k^{p-2-j} Y_k X_k^j\right) X_k^{1-p}\right), \quad Y_0 = \Delta_A. \tag{5.7}$$

Then the sequence Y_k tends to the Fréchet derivative $L(A, \Delta_A)$ of $\operatorname{sect}_p(A)$, i.e.,

$$\lim_{k\to\infty} Y_k = L(A, \Delta_A).$$

Proof. Analogously to the proof of [7, Theorem 5.7], we denote by Z_k Newton's iterates (4.1) for the matrix $B = \begin{bmatrix} A & \Delta_A \\ 0 & A \end{bmatrix}$. It is easy to show by induction that

$$Z_k = \begin{bmatrix} X_k & Y_k \\ 0 & X_k \end{bmatrix},$$

because

$$\left(\begin{bmatrix} X_k & Y_k \\ 0 & X_k \end{bmatrix}^{p-1}\right)^{-1} = \begin{bmatrix} X_k^{1-p} & -X_k^{1-p}\left(\sum_{j=0}^{p-2} X_k^{p-2-j} Y_k X_k^j\right) X_k^{1-p} \\ 0 & X_k^{1-p} \end{bmatrix}.$$

ETNA
Kent State University
http://etna.math.kent.edu

We have assumed that Newton's iteration is convergent for A, hence also for B, because the eigenvalues of B lie in the same region as A. Thus

$$\lim_{k\to\infty} Z_k = \mathrm{sect}_p(B) = \begin{bmatrix} \mathrm{sect}_p(A) & L(A,\Delta_A) \\ 0 & \mathrm{sect}_p(A) \end{bmatrix},$$

because of Theorem 3.6 of Mathias in [7]; see also [15]. This completes the proof. ☐

The convergence of the sequence (5.7) to the Fréchet derivative of $\mathrm{sect}_p(A)$ can be derived also from the recent more general result by Al-Mohy and Higham; see [1, Theorem 2.2].

Kenney and Laub [12] applied Newton's method to the characterization of the Fréchet derivative of the matrix sign function ($p = 2$), which provides a way of computing the Fréchet derivative; see also [7, Theorem 5.7]. The above Theorem 5.3 is a generalization of their result.

Theorem 5.1 generalizes to the matrix sector function Theorem 5.3 of Kenney and Laub in [7] for the Fréchet derivative of the matrix sign function; see also [12]. For $p = 2$ equation (5.5) reduces to the Sylvester equation.

6. Numerical experiments. We now present numerical experiments performed with the algorithms considered in the previous sections. The computations were done on a personal computer equipped with an Intel ® 1.5 GHz processor and 512 MB memory, using MATLAB 7.6.0 (R2008a). The machine precision is $u = 2.2 \cdot 10^{-16}$. To examine the behaviour of the algorithms we have performed tests for several values of p and test matrices A of different orders.

We compare experimentally the accuracy of the matrix sector function computed by the real Schur algorithm (`rSch`), the complex Schur algorithm (`cSch`), the real Schur algorithm with reordering (`rSch-ord`) and the complex Schur algorithm with reordering (`cSch-ord`), with the accuracy obtained by iterative methods: Newton's method (`Newt`) and two versions of Halley's method. We use the standard `MATLAB` functions `schur` and `ordschur` for computing the Schur and reordered Schur decompositions, respectively.

The cost of algorithms is measured in flops. The flop denotes any of the four elementary scalar operations $+, -, *, /$ performed on real or complex numbers (see [7, Appendix C]). In the number of flops, we give only the leading term as it is, for example, in [7, Table C.1]. The number of flops in the complex and real Schur algorithms for computing the matrix sector function equals $(28 + (p-1)/3)n^3$. The cost of the reordered Schur algorithms may or may not be less than the cost of Schur algorithms without reordering (compare [7, Section 5.2]). Some test matrices are given in the (real) Schur form. Thus the cost of performing the Schur algorithms is less than in the general case.

In Newton's method we have to compute X_k^{1-p}. We first compute the inverse of X_k and then we form the $(p-1)$th power of the computed matrix X_k^{-1}. The cost of computing one iterate in Newton's method mainly consists of inverting and powering a matrix and it is equal to $2\,(\eta\lfloor\log_2(p-1)\rfloor + 1)\,n^3$ flops, where $\eta \in [1,2]$, assuming the pth power is computed by binary powering; see Algorithm 4.1 in [7].

The first version of Halley's method (`Hal1`) was performed according to (4.2). In the second version (`Hal2`) one computes

$$X_{k+1} = \frac{p-1}{p+1}X_k + \frac{4p}{p+1}X_k\left((p+1)X_k^p + (p-1)I\right)^{-1}, \quad X_0 = A. \tag{6.1}$$

This version was inspired by Halley's iteration proposed for the polar decomposition in [8, Section 3, formula (3.2)]. Computing X_{k+1} by Halley's iteration performed according to (4.2) costs $2(\eta\lfloor\log_2(p-1)\rfloor+3)n^3$ flops, $\eta \in [1,2]$. In the second version of Halley's iteration (6.1) the cost is reduced to $2(\eta\lfloor\log_2(p-1)\rfloor + \frac{4}{3})n^3$ flops.

We also compute the matrix sector directly from the formula (2.2): $\mathrm{sect}_p(A) = AN^{-1}$, where $N = (A^p)^{1/p}$ was computed by the standard `MATLAB` matrix power $X^{1/p}$ involving eigensystem. In the tables below this method of computing $\mathrm{sect}_p(A)$ is denoted by `Nroot`. The results obtained by `Nroot` can be incorrect if A is not diagonalizable. However, in all our tests A was chosen to be diagonalizable. The method of computing $\mathrm{sect}_p(A)$ directly from (1.6) is denoted by `Def` in the tables. The scalar p-sector function $s_p(\lambda)$ was computed from (1.3).

The standard `MATLAB` function `inv` was used to compute the inverse of the matrix in the methods `Def`, `Nroot`, `Newt` and `Hal1`. In `Hal2` the formula (6.1) was realized in a different way. Namely, instead of using the inverse directly, the appropriate right-hand-side linear system was solved in each iteration.

In the iterative methods we have applied a simple stopping criterion (for more advanced termination criteria see [7, Section 4.9]), i.e.,

$$\|X_k - X_{k-1}\| \leq 100nu\|X_k\|,$$

where n is the order of the matrix $X_0 = A$ and $\|\cdot\|$ is the spectral norm. Then, $\hat{X} = X_k$ is the computed matrix sector function.

Let $\hat{X}$ denote the computed matrix sector function of A by a given method. For all methods we compute $\|\hat{X}\|$ and

$$\mathrm{res}(\hat{X}) = \frac{\|I - \hat{X}^p\|}{\|\hat{X}\| \|\sum_{i=0}^{p-1} (\hat{X}^{p-1-i})^T \otimes \hat{X}^i\|},$$

called the relative residual. We also compute

$$\|\hat{X}^p - I\|, \quad \|A\hat{X} - \hat{X}A\|, \quad \frac{\|A\hat{X} - \hat{X}A\|}{\|A\| \, \|\hat{X}\|}. \tag{6.2}$$

The relative residual was proposed for the matrix pth root in [5]. Guo and Higham have explained why the relative residual is a more appropriate criterion than the scaled residual for the interpretation of the numerical results. Their arguments are valid also for the matrix sector function. The last two quantities in (6.2) check if the iterations preserve commutativity of A and $\mathrm{sect}_p(A)$. We emphasize that these quantities do not measure the accuracy of the computed matrix sector function.

In the examples presented below, we also include $\mathrm{cond}(A)$ for the spectral norm, the numbers of performed iterations and the `CPU` timings. The function `cputime` was used for computing the execution time for each algorithm. The presented execution times are the averages obtained by repeating one hundred computations for every test matrix A.

We used the standard `MATLAB` `rand` function to generate uniformly distributed random elements of the real or complex matrices or their eigenvalues. The function `triu` was used to obtain the upper triangular part of the random matrix. While generating the test matrices, we took care that the eigenvalues fell into the regions of convergence for Newton's and Halley's methods. In Examples 6.2–6.4 the eigenvalues of A fall into the regions $\mathbb{B}_p^{(\mathrm{Newt})}$ and $\mathbb{B}_p^{(\mathrm{Hal})}$; see Theorem 4.3. In Examples 6.1, 6.5–6.8 the eigenvalues of the matrices lie in the experimentally determined regions of convergence of Newton's and Halley's methods, which are larger than $\mathbb{B}_p^{(\mathrm{Newt})}$ and $\mathbb{B}_p^{(\mathrm{Hal})}$, respectively.

ETNA
Kent State University
http://etna.math.kent.edu

Let $Y = A^{1/p}$ and let us consider the following block companion matrix

$$C = \begin{bmatrix} 0 & I & & & \\ & 0 & I & & \\ & & \ddots & \ddots & \\ & & & \ddots & I \\ A & & & & 0 \end{bmatrix} \in \mathbb{C}^{pn \times pn}. \tag{6.3}$$

Then, (see [2])

$$\mathrm{sect}_p(C) = \begin{bmatrix} 0 & Y^{-1} & & 0 \\ \vdots & 0 & \ddots & \\ 0 & \ddots & \ddots & Y^{-1} \\ AY^{-1} & 0 & \cdots & 0 \end{bmatrix}.$$

In [2] the authors write: *It would be interesting to know how the available methods for computing the matrix sector function behave if applied to the block companion matrix C with random A.* Therefore, we perform numerical experiments also for C. The results are given in Examples 6.7 and 6.8. The eigenvalues of C are the pth roots of the eigenvalues of A. Therefore, C has several eigenvalues of the same modulus.

EXAMPLE 6.1. Let $p = 4$ and A be in real Schur form:

$$A = \begin{bmatrix} 1 & 2 & 0 & 0 \\ -2 & 1 & -450 & 0 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & -3 & 1 \end{bmatrix}.$$

The matrix A has the following eigenvalues: $1 \pm 2i, 1 \pm 3i$. The matrix sector function $S = \mathrm{sect}_p(A)$ is equal to

$$S = \begin{bmatrix} 0 & 1 & 0 & -90 \\ -1 & 0 & -90 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \end{bmatrix}.$$

The results are given in Table 6.1. In the first column we include the norms $\|\hat{X} - S\|$. All results are satisfactory.

EXAMPLE 6.2. Let $A = [a_{ij}]$ be an 8×8 matrix in real Schur form with complex eigenvalues $\frac{-k^2}{10} \pm ik, k = 1, 2, 3, 4$, and elements a_{23}, a_{45}, a_{67} equal to -450; see [5]. The other elements in the upper triangle are zero. The spectral norm of A is equal to $\|A\| = 4.5 \times 10^2$. The matrix A is ill-conditioned. When $p < 21$, only for $p = 3, 4, 7$ the eigenvalues of A lie in the convergence regions $\mathbb{B}_p^{(\text{Newt})}$ and $\mathbb{B}_p^{(\text{Hal})}$. The results are summarized in Table 6.2. For $p = 7$ the commutativity condition is not well satisfied by the matrix sector function computed by `Hal2`, because $\|\hat{X}A - A\hat{X}\|$ is not small; see the first condition in (2.1). We observed this also in some other examples, especially for larger p or n. On the other hand, there are some examples where the situation is the opposite — the first version of Halley's method gives worse results than the second one with respect to the commutativity; see Tables 6.3 and 6.4.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 6.1
Results for A of order $n = 4$ (in real Schur form) from Example 6.1, $\mathrm{cond}(A) = 2.86 \times 10^4$.

$p = 4,\ \|\hat{X}\| = 90,\ iter_{\mathrm{Newt}} = 10,\ iter_{\mathrm{Hal1}} = 7,\ iter_{\mathrm{Hal2}} = 7$

alg.	$\|\hat{X} - S\|$	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|\hat{X}A - A\hat{X}\|$	$\frac{\|\hat{X}A - A\hat{X}\|}{\|\hat{X}\|\|A\|}$
Newt	$1.57e-14$	$5.68e-14$	$2.48e-18$	$6.86e-14$	$1.69e-18$
Hal1	$5.81e-14$	$1.92e-30$	$8.40e-35$	$2.74e-13$	$6.78e-18$
Hal2	$4.38e-14$	$2.84e-14$	$1.24e-18$	$1.27e-13$	$3.14e-18$
cSch	$2.01e-14$	$8.74e-14$	$3.81e-18$	$1.76e-13$	$4.35e-18$
cSch − ord	$1.11e-16$	$2.84e-14$	$1.24e-18$	$5.68e-14$	$1.40e-18$
rSch	$6.12e-17$	$2.20e-14$	$9.62e-19$	0	0
rSch − ord	$6.12e-17$	$2.20e-14$	$9.62e-19$	0	0
Nroot	$6.65e-14$	$9.63e-14$	$4.20e-18$	$2.95e-13$	$7.29e-18$
Def	$1.03e-13$	$2.30e-13$	$1.00e-17$	$2.50e-13$	$6.16e-18$

TABLE 6.2
Results for A of order $n = 8$ (in real Schur form) from Example 6.2, $\mathrm{cond}(A) = 1.4 \times 10^9$.

$p = 3,\ \|\hat{X}\| = 1.8 \times 10^6,\ iter_{\mathrm{Newt}} = 9,\ iter_{\mathrm{Hal1}} = 6,\ iter_{\mathrm{Hal2}} = 6$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|\hat{X}A - A\hat{X}\|$	$\frac{\|\hat{X}A - A\hat{X}\|}{\|\hat{X}\|\|A\|}$
Newt	$6.41e-03$	$9.86e-10$	$1.80e-28$	$3.67e-09$	$4.60e-18$
Hal1	$6.41e-03$	$9.44e-10$	$1.72e-28$	$6.58e-09$	$8.25e-18$
Hal2	$5.00e-03$	$5.85e-10$	$1.07e-28$	$4.99e-08$	$6.26e-17$
cSch	$6.56e-03$	$3.78e-09$	$2.71e-28$	$3.21e-09$	$4.02e-18$
cSch − ord	$4.84e-03$	$2.10e-09$	$3.82e-28$	$4.99e-09$	$6.26e-18$
rSch	$1.13e-02$	$3.01e-09$	$5.49e-28$	$6.64e-10$	$8.34e-19$
rSch − ord	$7.81e-03$	$3.01e-09$	$5.49e-28$	$6.64e-10$	$8.34e-19$
Nroot	$4.69e-04$	$8.14e-09$	$1.48e-27$	$1.61e-08$	$2.02e-17$
Def	$2.50e-03$	$9.65e-09$	$1.76e-27$	$1.33e-08$	$1.66e-17$

$p = 7,\ \|\hat{X}\| = 1.99 \times 10^6,\ iter_{\mathrm{Newt}} = 15,\ iter_{\mathrm{Hal1}} = 9,\ iter_{\mathrm{Hal2}} = 10$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|A\hat{X} - \hat{X}A\|$	$\frac{\|A\hat{X} - \hat{X}A\|}{\|\hat{X}\|\|A\|}$
Newt	$8.91e-03$	$2.06e-09$	$3.21e-28$	$4.39e-09$	$4.89e-18$
Hal1	$1.14e-02$	$1.75e-09$	$2.72e-28$	$6.72e-09$	$7.49e-18$
Hal2	$7.19e-03$	$1.30e-09$	$2.03e-28$	$6.65e-05$	$7.41e-14$
cSch	$8.44e-03$	$6.95e-09$	$1.08e-27$	$2.55e-09$	$2.85e-18$
cSch − ord	$7.97e-03$	$2.38e-09$	$3.71e-28$	$5.28e-09$	$5.88e-18$
rSch	$1.58e-02$	$6.21e-09$	$9.68e-28$	$7.26e-10$	$8.09e-19$
rSch − ord	$1.77e-02$	$6.21e-09$	$9.68e-28$	$7.26e-10$	$8.09e-19$
Nroot	$4.69e-04$	$2.35e-08$	$3.67e-27$	$2.23e-08$	$2.49e-17$
Def	$1.41e-03$	$2.40e-08$	$3.74e-27$	$1.39e-08$	$1.55e-17$

EXAMPLE 6.3. Let now $A = D + T$ be complex upper triangular, where $D = \mathrm{diag}(\lambda_j)$ is complex, $\lambda_j = x_j + iy_j$ for $x_j, y_j \in [-100, 100]$, and T is a real random upper triangular matrix with zero elements on the main diagonal. The nonzero elements of T are generated by `rand` from the interval $[-1, 1]$. In Table 6.3 we present the results for $p = 5$ and $n = 40$. The matrix A is well conditioned and $\|A\| = 1.27e+02$.

EXAMPLE 6.4. Let the matrix $A \in \mathbb{C}^{10\times 10}$ be generated as in Example 6.3. In Table 6.4 we present the results obtained for $p = 4$ and $p = 10$. The results obtained for $p = 4$ by the reordered complex Schur algorithm are worse than without reordering. Newton's method works very well.

EXAMPLE 6.5. Let $A \in \mathbb{C}^{10\times 10}$ be the Grcar matrix, a Toeplitz matrix with sensitive

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 6.3
Results for complex upper triangular A of order $n = 40$ from Example 6.3, $\mathrm{cond}(A) = 9.8$.

$p = 5,\ \|\hat{X}\| = 1.1,\ iter_{\text{Newt}} = 28,\ iter_{\text{Hal1}} = 16,\ iter_{\text{Hal2}} = 16$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|A\hat{X} - \hat{X}A\|$	$\frac{\|A\hat{X}-\hat{X}A\|}{\|\hat{X}\|\|A\|}$
Newt	$4.11e-01$	$7.23e-16$	$1.36e-16$	$5.05e-15$	$3.75e-17$
Hal1	$5.41e-01$	$1.79e-15$	$3.36e-16$	$2.43e-11$	$1.80e-13$
Hal2	$2.31e-01$	$8.26e-16$	$1.55e-16$	$3.58e-15$	$2.66e-17$
cSch	$1.42e-01$	$1.11e-15$	$2.09e-16$	$1.89e-15$	$1.40e-17$
cSch − ord	$3.08e-02$	$8.98e-15$	$1.69e-15$	$8.95e-15$	$6.64e-17$
Nroot	$5.47e-03$	$5.34e-15$	$1.00e-15$	$4.59e-15$	$3.41e-17$
Def	$3.75e-02$	$1.64e-15$	$3.08e-16$	$3.27e-15$	$2.43e-17$

TABLE 6.4
Results for complex upper triangular A of order $n = 10$ from Example 6.4.

$p = 4,\ \mathrm{cond}(A) = 2.8,\ \|\hat{X}\| = 1.01,\ iter_{\text{Newt}} = 22,\ iter_{\text{Hal1}} = 13,\ iter_{\text{Hal2}} = 13$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|A\hat{X} - \hat{X}A\|$	$\frac{\|A\hat{X}-\hat{X}A\|}{\|\hat{X}\|\|A\|}$
Newt	$3.11e-02$	$4.44e-16$	$1.10e-16$	$1.70e-15$	$1.76e-17$
Hal1	$3.94e-02$	$8.88e-16$	$2.19e-16$	$5.20e-15$	$5.38e-17$
Hal2	$1.80e-02$	$5.21e-18$	$1.29e-18$	$1.22e-15$	$1.26e-17$
cSch	$8.28e-03$	$5.20e-18$	$1.28e-18$	$4.66e-16$	$4.82e-18$
cSch − ord	$5.63e-03$	$2.22e-15$	$5.48e-16$	$1.90e-15$	$1.97e-17$
Nroot	$1.09e-03$	$3.12e-15$	$7.70e-16$	$8.31e-16$	$8.59e-18$
Def	$1.56e-03$	$9.28e-16$	$2.29e-16$	$1.09e-15$	$1.13e-17$

$p = 10,\ \mathrm{cond}(A) = 6.4,\ \|\hat{X}\| = 1.02,\ iter_{\text{Newt}} = 51,\ iter_{\text{Hal1}} = 28,\ iter_{\text{Hal2}} = 28$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|A\hat{X} - \hat{X}A\|$	$\frac{\|A\hat{X}-\hat{X}A\|}{\|\hat{X}\|\|A\|}$
Newt	$5.23e-02$	$1.31e-15$	$1.28e-16$	$2.00e-15$	$1.67e-17$
Hal1	$5.25e-02$	$2.09e-15$	$2.04e-16$	$6.04e-08$	$5.07e-10$
Hal2	$2.88e-02$	$8.90e-16$	$8.70e-17$	$1.45e-15$	$1.21e-17$
cSch	$1.20e-02$	$1.28e-15$	$1.25e-16$	$4.68e-16$	$3.93e-18$
cSch − ord	$3.44e-03$	$7.12e-15$	$6.96e-16$	$1.91e-15$	$1.60e-17$
Nroot	$1.56e-04$	$5.36e-15$	$5.24e-16$	$1.09e-15$	$9.15e-18$
Def	$2.66e-03$	$2.17e-15$	$2.12e-16$	$8.01e-16$	$6.72e-18$

eigenvalues generated by the MATLAB command `gallery('grcar',10)`. The matrix A is well conditioned and $\|A\| = 3.11$. The results are presented in Table 6.5. The accuracy of the computed matrix sector is similar for all methods.

EXAMPLE 6.6. Let $A \in \mathbb{R}^{n\times n}$ be in real Schur form. Each block on the main diagonal is either 1×1 with real eigenvalues generated from the interval $[-2, 2]$, or 2×2 having complex conjugate eigenvalues $\lambda_j = x_j \pm iy_j$, $x_j, y_j \in [-2, 2]$. The nonzero elements of A are randomly generated from the interval $[-1, 1]$. The number of pairs of complex eigenvalues was also randomly generated. All eigenvalues belong to the region $\mathbb{B}_p^{(\text{Hal})}$ and to the numerically determined convergence region of Newton's method. The results for $n = 20$ are presented in Table 6.6. The complex eigenvalues lie near the boundaries of the regions of convergence; see Figure 6.1. The matrix A has 16 real eigenvalues: four of them satisfy $0.2 < |\lambda| < 0.5$ and four of them satisfy $|\lambda| < 0.2$. The matrix V of eigenvectors of $A = V\,\mathrm{diag}(\lambda_j)V^{-1}$ is ill-conditioned: $\mathrm{cond}(V) = 2.91 \times 10^5$. Therefore, the matrix sector function computed by `Nroot` and `Def` is less accurate than by other methods. The real Schur algorithm with reordering gives a little better results than without reordering.

ETNA
Kent State University
http://etna.math.kent.edu

TABLE 6.5
Results for Grcar complex matrix A of order $n = 10$ from Example 6.5, $\mathrm{cond}(A) = 2.89$.

$p = 9, \quad \|\hat{X}\| = 2.37, \quad iter_{\text{Newt}} = 13, \quad iter_{\text{Hal1}} = 7, \quad iter_{\text{Hal2}} = 7$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|A\hat{X} - \hat{X}A\|$	$\frac{\|A\hat{X}-\hat{X}A\|}{\|\hat{X}\|\|A\|}$
Newt	$1.48e-02$	$4.24e-15$	$1.39e-17$	$2.03e-15$	$2.76e-16$
Hal1	$2.50e-02$	$4.05e-15$	$1.32e-17$	$3.44e-14$	$4.67e-15$
Hal2	$8.28e-03$	$2.53e-15$	$8.28e-18$	$2.28e-15$	$3.10e-16$
cSch	$3.08e-02$	$3.70e-14$	$1.21e-16$	$1.05e-14$	$1.42e-15$
cSch − ord	$1.81e-02$	$3.71e-14$	$1.21e-16$	$1.02e-14$	$1.38e-15$
rSch	$1.73e-02$	$3.57e-14$	$1.17e-16$	$1.04e-14$	$1.41e-15$
rSch − ord	$1.31e-02$	$2.95e-14$	$9.64e-17$	$1.10e-14$	$1.49e-15$
Nroot	$1.56e-03$	$1.63e-14$	$5.33e-17$	$6.53e-15$	$8.85e-16$
Def	$2.03e-03$	$1.56e-14$	$5.12e-17$	$4.07e-14$	$5.52e-15$

TABLE 6.6
Results for A of order $n = 20$ (in real Schur form) from Example 6.6, $\mathrm{cond}(A) = 6.8 \times 10^5$.

$p = 4, \;\; \|\hat{X}\| = 4.35 \times 10^2, \;\; iter_{\text{Newt}} = 43, \;\; iter_{\text{Hal1}} = 12, \;\; iter_{\text{Hal2}} = 12$

alg.	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|\hat{X}A - A\hat{X}\|$	$\frac{\|\hat{X}A-A\hat{X}\|}{\|\hat{X}\|\|A\|}$
Newt	$7.98e-02$	$2.61e-13$	$7.25e-24$	$3.38e-08$	$2.14e-11$
Hal1	$4.67e-02$	$5.00e-13$	$1.39e-23$	$1.50e-12$	$9.51e-16$
Hal2	$3.00e-02$	$2.04e-13$	$5.68e-24$	$3.59e-12$	$2.27e-15$
cSch	$2.92e-02$	$2.62e-13$	$7.30e-24$	$6.90e-14$	$4.36e-17$
cSch − ord	$1.52e-01$	$6.75e-13$	$1.88e-23$	$1.05e-13$	$5.89e-17$
rSch	$2.28e-02$	$2.07e-11$	$5.76e-22$	$8.25e-14$	$5.22e-17$
rSch − ord	$7.17e-02$	$1.05e-12$	$2.93e-23$	$9.30e-14$	$5.89e-17$
Nroot	$4.84e-03$	$5.94e-09$	$1.65e-19$	$1.46e-10$	$9.25e-14$
Def	$5.94e-03$	$4.23e-10$	$1.18e-20$	$4.99e-11$	$3.16e-14$

EXAMPLE 6.7. Let $A \in \mathbb{R}^{8\times 8}$ be generated as in Example 6.2 and let $C \in \mathbb{R}^{8p\times 8p}$ be determined as (6.3). The matrix C is very ill-conditioned. The results for C are summarized in Table 6.7. Now $\hat{X}$ denotes the computed $\mathrm{sect}_p(C)$. The matrix A has 4 pairs of conjugate complex eigenvalues. The eigenvalues of C are the pth roots of the eigenvalues of A. Therefore, C has 4 groups of eigenvalues with $2p$ eigenvalues with the same modulus in each group. For $p = 3$ and $p = 6$ they are marked in Figure 6.2; see also Figure 6.1. In order to evaluate the accuracy of the computed real Schur decomposition of C by the MATLAB function `schur` we computed the eigenvalues of C in the following three ways:

$\lambda_j^{(\text{eig})}$: eigenvalues of C computed by means of `eig`,

$\lambda_j^{(\text{A})}$: eigenvalues of C computed as the pth roots of the exact eigenvalues of A,

$\lambda_j^{(\text{Sch})}$: eigenvalues of C computed directly from the diagonal blocks of R from the real Schur decomposition of C.

We obtained, for $p = 3$,

$$\max_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{A})}| = 3.62e-11, \qquad \min_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{A})}| = 8.46e-13,$$

$$\max_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{eig})}| = 4.41e-11, \qquad \min_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{eig})}| = 1.88e-12,$$

$$\max_j |\lambda_j^{(\text{A})} - \lambda_j^{(\text{eig})}| = 1.44e-11, \qquad \min_j |\lambda_j^{(\text{A})} - \lambda_j^{(\text{eig})}| = 7.45e-16,$$

and, for $p = 6$,

$$\max_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{A})}| = 1.54e-10, \qquad \min_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{A})}| = 5.90e-12,$$
$$\max_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{eig})}| = 1.46e-10, \qquad \min_j |\lambda_j^{(\text{Sch})} - \lambda_j^{(\text{eig})}| = 2.95e-12,$$
$$\max_j |\lambda_j^{(\text{A})} - \lambda_j^{(\text{eig})}| = 2.83e-11, \qquad \min_j |\lambda_j^{(\text{A})} - \lambda_j^{(\text{eig})}| = 6.13e-13.$$

The inaccuracy in the computed Schur decompositions of C causes the matrix sector function computed by the Schur algorithms to be less accurate; see the values of res in Table 6.7. We notice that the norm of the matrix sector function of C is large. For $n = 48$ and $p = 6$ the matrix V of the eigenvectors of C is very ill-conditioned: $\text{cond}(V) = 4.7 \times 10^7$. Higham [7, Section 4.5] writes that since the conditioning of a matrix function $f(A)$ is not necessarily related to $\text{cond}(V)$, computing a matrix function $f(A)$ via the spectral decomposition of A may be numerically unstable; see the results for `Def`.

TABLE 6.7
Results for real C from Example 6.7, $\text{cond}(C) = 1.4 \times 10^9$.

$n = 24,\ \ p = 3,\ \ \|\hat{X}\| = 1.71 \times 10^6,\ \ iter_{\text{Newt}} = 8,\ \ iter_{\text{Hal1}} = 5,\ \ iter_{\text{Hal2}} = 6$

alg.	CPU	$\|\hat{X}^p - I\|$	$\text{res}(\hat{X})$	$\|C\hat{X} - \hat{X}C\|$	$\frac{\|C\hat{X}-\hat{X}C\|}{\|\hat{X}\|\|C\|}$
Newt	$1.80e-02$	$9.39e-10$	$1.87e-28$	$2.32e-09$	$2.99e-18$
Hal1	$2.97e-02$	$4.07e-09$	$8.09e-28$	$6.11e-09$	$7.89e-18$
Hal2	$2.00e-02$	$1.05e-09$	$2.10e-28$	$7.49e-06$	$9.68e-15$
cSch	$4.22e-02$	$1.34e-06$	$2.67e-25$	$9.98e-08$	$1.29e-16$
cSch − ord	$1.53e-02$	$1.12e-06$	$2.23e-25$	$9.92e-08$	$1.28e-16$
rSch	$7.23e-02$	$1.35e-06$	$2.68e-25$	$9.98e-08$	$1.29e-16$
rSch − ord	$7.30e-02$	$1.34e-06$	$2.66e-25$	$9.97e-08$	$1.29e-16$
Nroot	$2.81e-03$	$1.34e-08$	$2.67e-27$	$5.44e-09$	$7.03e-18$
Def	$1.28e-02$	$6.60e-08$	$1.31e-26$	$2.44e-07$	$3.15e-16$

$n = 48,\ \ p = 6,\ \ \|\hat{X}\| = 8.76 \times 10^5,\ \ iter_{\text{Newt}} = 9,\ \ iter_{\text{Hal1}} = 5,\ \ iter_{\text{Hal2}} = 5$

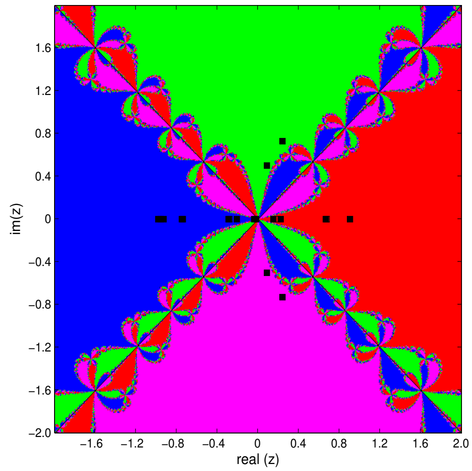
alg.	CPU	$\|\hat{X}^p - I\|$	$\text{res}(\hat{X})$	$\|C\hat{X} - \hat{X}C\|$	$\frac{\|C\hat{X}-\hat{X}C\|}{\|\hat{X}\|\|C\|}$
Newt	$7.39e-02$	$2.99e-09$	$3.16e-28$	$3.29e-09$	$8.30e-18$
Hal1	$8.84e-02$	$3.21e-09$	$3.40e-28$	$1.36e-09$	$3.44e-18$
Hal2	$5.50e-02$	$2.21e-09$	$2.34e-28$	$8.45e-07$	$2.14e-15$
cSch	$2.55e-01$	$6.29e-04$	$6.65e-23$	$3.70e-08$	$9.34e-17$
cSch − ord	$5.70e-02$	$4.87e-03$	$5.15e-22$	$3.63e-08$	$9.17e-17$
rSch	$5.33e-01$	$9.43e-04$	$9.98e-23$	$3.57e-08$	$9.02e-17$
rSch − ord	$2.54e-01$	$9.52e-04$	$1.01e-22$	$3.81e-08$	$9.62e-17$
Nroot	$1.00e-02$	$2.15e-08$	$2.27e-27$	$2.86e-09$	$7.22e-18$
Def	$3.81e-02$	$1.26e-03$	$1.33e-22$	$5.88e-03$	$1.49e-11$

EXAMPLE 6.8. Let $A \in \mathbb{C}^{8\times 8}$, $A = D + T$, where T is generated as in Example 6.3, $D = \text{diag}(\lambda_j)$ is complex and λ_j are generated such that (compare (4.17))

$$|\lambda_j| \geq 1, \quad -\pi/(2p) < \arg(\lambda_j) < \pi/(2p).$$

The eigenvalues of A have different moduli. The results for $C \in \mathbb{C}^{8p\times 8p}$, determined as in (6.3), are summarized in Table 6.8. The matrix C is well-conditioned. The computed matrix sector function $\hat{X}$ has small norm and is computed accurately by all methods. In this example the Schur decomposition of C was computed with good accuracy. The differences between the eigenvalues of C computed by the three methods mentioned in the previous example were of the order 10^{-15} or less.

ETNA
Kent State University
http://etna.math.kent.edu



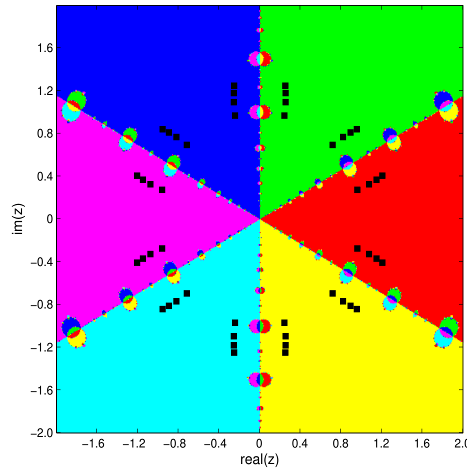


FIGURE 6.1. *Location of the eigenvalues of A from Example 6.6 and the convergence regions for Newton's method for $p = 4$ (left); location of the eigenvalues of C from Example 6.7 and the convergence regions for Halley's method for $p = 6$ (right).*

TABLE 6.8
Results for C from Example 6.8, $\mathrm{cond}(C) = 5.55$.

$n = 48, \quad p = 6, \quad \|\hat{X}\| = 4.17, \quad iter_{\text{Newt}} = 7, \quad iter_{\text{Hal1}} = 5, \quad iter_{\text{Hal2}} = 5$

$alg.$	CPU	$\|\hat{X}^p - I\|$	$\mathrm{res}(\hat{X})$	$\|C\hat{X} - \hat{X}C\|$	$\frac{\|C\hat{X}-\hat{X}C\|}{\|\hat{X}\|\|C\|}$
`Newt`	$1.05e-01$	$9.06e-16$	$1.14e-17$	$1.53e-15$	$6.63e-17$
`Hal1`	$1.83e-01$	$1.24e-15$	$1.57e-17$	$2.32e-15$	$1.00e-16$
`Hal2`	$8.14e-02$	$6.68e-16$	$8.44e-18$	$1.00e-15$	$4.33e-17$
`cSch`	$2.66e-01$	$3.99e-14$	$5.04e-16$	$2.96e-14$	$1.28e-15$
`cSch − ord`	$6.56e-02$	$4.28e-14$	$5.41e-16$	$3.10e-14$	$1.34e-15$
`Nroot`	$1.25e-02$	$2.99e-15$	$3.78e-17$	$3.98e-16$	$1.72e-17$
`Def`	$5.50e-02$	$1.42e-14$	$1.79e-16$	$2.27e-14$	$9.82e-16$

In the above numerical experiments the matrix sector function was computed directly from (1.7) using the standard matrix powering in `MATLAB` for $X^{1/p}$, which can be applied only to diagonalizable matrices. In [14] we present experiments obtained by another approach for computing (1.7), in which we use the method of Guo and Higham [5] for the principal matrix pth root of a real matrix A. Their algorithm is in the Matrix Function Toolbox `mftoolbox` in [7]. The method of Guo and Higham involves the real Schur decomposition, and in our numerical experiments the accuracy of $\mathrm{sect}_p(A)$ computed from (1.7) using their method for the matrix pth root was not better than the accuracy of $\mathrm{sect}_p(A)$ computed by the real and complex Schur algorithms.

7. Conclusions. In this paper we have investigated the properties of some algorithms for computing the matrix sector function. We derived the complex Schur algorithms with and without reordering and blocking. The complex Schur algorithm is applicable to all matrices for which the matrix sector function exists. The developed real Schur algorithm can be applied only to real matrices having no multiple complex eigenvalues in the sectors different from Φ_0 and $\Phi_{p/2}$.

We have determined the regions of convergence for Newton's and Halley's iterations applied to the matrix sector function. From the results known in the general theory of matrix

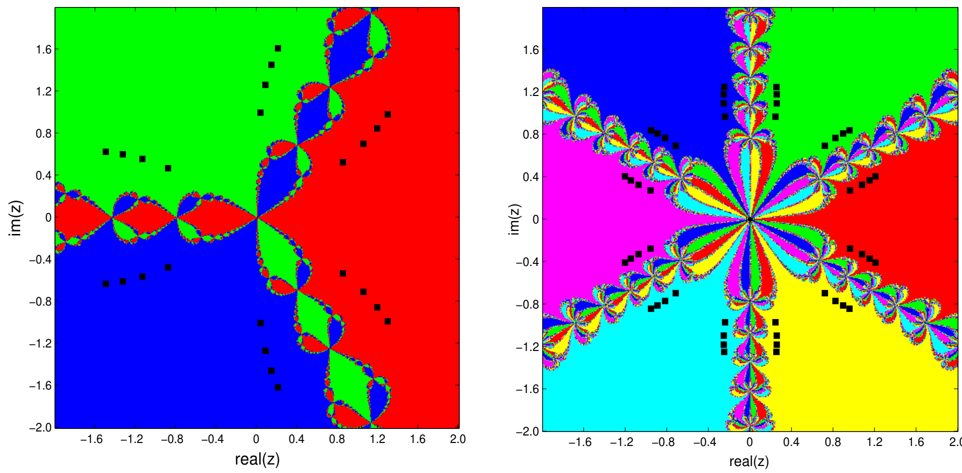


FIGURE 6.2. *Location of the eigenvalues of C from Example 6.7 and the convergence regions for Newton's method, $p = 3$ and $p = 6$.*

functions, we have deduced the stability of Newton's and Halley's methods for computing the matrix sector function in the sense considered in [7, Section 4.9]. Experimental results indicate that these iterative methods compute the matrix sector function with the same or better accuracy than other methods considered in this paper.

Acknowledgements. We are very grateful to Nicholas J. Higham for providing us a working version of his book [7] and for discussions on matrix functions during conferences. We also thank Bruno Iannazzo and the referees for their helpful comments and suggestions.

REFERENCES

[1] A. H. AL-MOHY AND N. J. HIGHAM, *Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation*, SIAM J. Matrix Anal. Appl., 30 (2009), pp. 1639–1657.
[2] D. A. BINI, N. J. HIGHAM, AND B. MEINI, *Algorithms for the matrix pth root*, Numer. Algorithms, 39 (2005), pp. 349–378.
[3] P. I. DAVIES AND N. J. HIGHAM, *A Schur-Parlett algorithm for computing matrix functions*, SIAM J. Matrix Anal. Appl., 25 (2003), pp. 464–485.
[4] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computation*, 3rd ed., Johns Hopkins University Press, Baltimore, 1996.
[5] CH.-H GUO AND N. J. HIGHAM, *A Schur-Newton method for the matrix pth root and its inverse*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 788–804.
[6] N. J. HIGHAM, *The matrix sign decomposition and its relation to the polar decomposition*, Linear Algebra Appl., 212/213 (1994), pp. 3–20.
[7] N. J. HIGHAM, *Functions of Matrices: Theory and Computations*, SIAM, Philadelphia, 2008.
[8] N. J. HIGHAM, D. S. MACKEY, N. MACKEY, AND F. TISSEUR, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM J. Matrix Anal. Appl., 25 (2004), pp. 1178–1192.
[9] B. IANNAZZO, *On the Newton method for the matrix pth root*, SIAM J. Matrix Anal. Appl., 28 (2006), pp. 503–523.
[10] B. IANNAZZO, *Numerical solution of certain nonlinear matrix equations*, Ph.D. thesis, Università degli Studi di Pisa, Facoltà di Scienze Matematiche, Fisiche e Naturali, 2007.
[11] B. IANNAZZO, *A family of rational iterations and its application to the computation of the matrix pth root*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1445–1462.
[12] CH. KENNEY AND A. J. LAUB, *Polar decomposition and matrix sign function condition estimates*, SIAM J. Sci. Stat. Comput., 12 (1991), pp. 488–504.
[13] C. K. KOÇ AND B. BAKKALŎGLU, *Halley method for the matrix sector function*, IEEE Trans. Automat. Control, 40 (1995), pp. 944–949.

[14] B. LASZKIEWICZ AND K. ZIĘTAK, *The Padé family of iterations for the matrix sector function and the matrix pth root*, To appear in Numer. Linear Algebra Appl., 2009.

[15] R. MATHIAS, *A chain rule for matrix functions and applications*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 610–620.

[16] B. N. PARLETT, *A reccurence among the elements of functions of triangular matrices*, Linear Algebra Appl., 14 (1976), pp. 117–121.

[17] J. D. ROBERTS, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, Internat. J. Control, 32 (1980), pp. 677–687.

[18] L. S. SHIEH, Y. T. TSAY, AND C. T. WANG, *Matrix sector functions and their applications to systems theory*, IEEE Proc. 131 (1984), pp. 171–181.

[19] M. I. SMITH, *Numerical Computation of Matrix Functions*, Ph.D. thesis, University of Manchester, Faculty of Science and Engineering, Department of Mathematics, 2002.

[20] M. I. SMITH, *A Schur algorithm for computing matrix pth root*, SIAM J. Matrix Anal. Appl., 24 (2003), pp. 971–989.