Functions of Matrices

Higham, Nicholas J.

2005

MIMS EPrint: **2005.21**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

F.13 Functions of Matrices[1]

Matrix functions are used in many areas of linear algebra and arise in numerous applications in science and engineering. The most common matrix function is the matrix inverse; it is not treated specifically in this article, but is covered in Chapter B. This article is concerned with general matrix functions as well as the specific cases of matrix square roots, trigonometric functions, and the exponential and logarithmic functions.

The specific functions just mentioned can all be defined via power series or as the solution of nonlinear systems. For example, $\cos(A) = I - A^2/2! + A^4/4! - \cdots$. However, a general theory exists from which a number of properties possessed by all matrix functions can be deduced and which suggest computational methods. This article treats general theory, then specific functions, and finally outlines computational methods.

Proofs of the facts in this article can be found in one or more of [Hig], [HJ91] or [LT85], unless otherwise stated.

1 General Theory

Definitions:

A function of a matrix can be defined in several ways, of which the following three are the most generally useful.

- *Jordan canonical form definition.* Let $A \in \mathbb{C}^{n\times n}$ have the Jordan canonical form

[1]Nicholas J. Higham, School of Mathematics, The University of Manchester, Sackville Street, Manchester, M60 1QD, UK (higham@ma.man.ac.uk, http://www.ma.man.ac.uk/~higham/). For Handbook of Linear Algebra, edited by Hogben, Brualdi, Greenbaum, and Mathias, CRC Press. Version dated November 7, 2005.

$Z^{-1}AZ = J_A = \operatorname{diag}(J_1(\lambda_1), J_2(\lambda_2), \ldots, J_p(\lambda_p))$, where Z is nonsingular,

$$J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k \times m_k}, \tag{1}$$

and $m_1 + m_2 + \cdots + m_p = n$. Then

$$f(A) := Z f(J_A) Z^{-1} = Z \operatorname{diag}(f(J_k(\lambda_k))) Z^{-1}, \tag{2}$$

where

$$f(J_k(\lambda_k)) := \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \ldots & \dfrac{f^{(m_k-1)})(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}. \tag{3}$$

- *Polynomial interpolation definition.* Denote by $\lambda_1, \ldots, \lambda_s$ the distinct eigenvalues of A and let n_i be the **index** of λ_i, that is, the order of the largest Jordan block in which λ_i appears. Then $f(A) := r(A)$, where r is the unique Hermite interpolating polynomial of degree less than $\sum_{i=1}^{s} n_i$ that satisfies the interpolation conditions

$$r^{(j)}(\lambda_i) = f^{(j)}(\lambda_i), \qquad j = 0\colon n_i - 1, \quad i = 1\colon s. \tag{4}$$

Note that in both these definitions the derivatives in (4) must exist in order for $f(A)$ to be defined. The function f is said to be **defined on the spectrum of** A if all the derivatives in (4) exist.

- *Cauchy integral definition.*

$$f(A) := \frac{1}{2\pi i} \int_\Gamma f(z)(zI - A)^{-1}\, dz, \tag{5}$$

where f is analytic inside a closed contour Γ that encloses $\sigma(A)$.

When the function f is multivalued and A has a repeated eigenvalue occurring in more than one Jordan block (i.e., A is derogatory), the Jordan canonical form definition has more than one interpretation. Usually, for each occurrence of an eigenvalue in different Jordan blocks the same branch is taken for f and its derivatives. This gives a **primary matrix function**. If different branches are taken for the same eigenvalue in two different Jordan blocks then a **nonprimary matrix function** is obtained. A nonprimary matrix function is not expressible as a polynomial in the matrix, and if such a function is obtained from the Jordan canonical form definition (2) then it depends on the matrix Z. In most applications it is primary matrix functions that are of interest. For the rest of this article $f(A)$ is assumed to be a primary matrix function, unless otherwise stated.

Facts:

1. The Jordan canonical form and polynomial interpolation definitions are equivalent. Both definitions are equivalent to the Cauchy integral definition when f is analytic.

2. $f(A)$ is a polynomial in A and the coefficients of the polynomial depend on A.

3. $f(A)$ commutes with A.

4. $f(A^T) = f(A)^T$.

5. For any nonsingular X, $f(XAX^{-1}) = Xf(A)X^{-1}$.

6. If A is diagonalizable, with $Z^{-1}AZ = D = \operatorname{diag}(d_1, d_2, \ldots, d_n)$, then $f(A) = Zf(D)Z^{-1} = Z\operatorname{diag}(f(d_1), f(d_2), \ldots, f(d_n))Z^{-1}$.

7. $f(\operatorname{diag}(A_1, A_2, \ldots, A_m)) = \operatorname{diag}(f(A_1), f(A_2), \ldots, f(A_m))$.

8. Let f and g be functions defined on the spectrum of A. (a) If $h(t) = f(t) + g(t)$ then $h(A) = f(A) + g(A)$. (b) If $h(t) = f(t)g(t)$ then $h(A) = f(A)g(A)$.

9. Let $G(u_1, \dots, u_t)$ be a polynomial in $u_1, \dots, u_t$ and let $f_1, \dots, f_t$ be functions defined on the spectrum of A. If $g(\lambda) = G(f_1(\lambda), \dots, f_t(\lambda))$ takes zero values on the spectrum of A then $g(A) = G(f_1(A), \dots, f_t(A)) = 0$. For example, $\sin^2(A) + \cos^2(A) = I$, $(A^{1/p})^p = A$, and $e^{iA} = \cos A + i \sin A$.

10. Suppose f has a Taylor series expansion

$$f(z) = \sum_{k=0}^{\infty} a_k (z - \alpha)^k \qquad \left(a_k = \frac{f^{(k)}(\alpha)}{k!} \right)$$

with radius of convergence r. If $A \in \mathbb{C}^{n \times n}$ then $f(A)$ is defined and is given by

$$f(A) = \sum_{k=0}^{\infty} a_k (A - \alpha I)^k$$

if and only if each of the distinct eigenvalues $\lambda_1, \dots, \lambda_s$ of A satisfies one of the conditions

(a) $|\lambda_i - \alpha| < r$,

(b) $|\lambda_i - \alpha| = r$ and the series for $f^{n_i - 1}(\lambda)$, where n_i is the index of λ_i, is convergent at the point $\lambda = \lambda_i$, $i = 1\colon s$.

11. [Dav73], [Des63], [GVL96, 1996, Thm. 11.1.3]. Let $T \in \mathbb{C}^{n \times n}$ be upper triangular and suppose that f is defined on the spectrum of T. Then $F = f(T)$ is upper triangular with $f_{ii} = f(t_{ii})$ and

$$f_{ij} = \sum_{(s_0, \dots, s_k) \in S_{ij}} t_{s_0, s_1} t_{s_1, s_2} \dots t_{s_{k-1}, s_k} f[\lambda_{s_0}, \dots, \lambda_{s_k}],$$

where $\lambda_i = t_{ii}$, S_{ij} is the set of all strictly increasing sequences of integers that start at i and end at j, and $f[\lambda_{s_0}, \dots, \lambda_{s_k}]$ is the kth order divided difference of f at $\lambda_{s_0}, \dots, \lambda_{s_k}$.

Examples:

1. For $\lambda_1 \neq \lambda_2$,
$$f\left(\begin{bmatrix} \lambda_1 & \alpha \\ 0 & \lambda_2 \end{bmatrix}\right) = \begin{bmatrix} f(\lambda_1) & \alpha \dfrac{f(\lambda_2) - f(\lambda_1)}{\lambda_2 - \lambda_1} \\ 0 & f(\lambda_2) \end{bmatrix}.$$
For $\lambda_1 = \lambda_2 = \lambda$,
$$f\left(\begin{bmatrix} \lambda & \alpha \\ 0 & \lambda \end{bmatrix}\right) = \begin{bmatrix} f(\lambda) & \alpha f'(\lambda) \\ 0 & f(\lambda) \end{bmatrix}.$$

2. Compute e^A for the matrix
$$A = \begin{bmatrix} -7 & -4 & -3 \\ 10 & 6 & 4 \\ 6 & 3 & 3 \end{bmatrix}.$$
We have $A = XJ_AX^{-1}$, where $J_A = [0] \oplus \left[\begin{smallmatrix} 1 & 1 \\ 0 & 1 \end{smallmatrix}\right]$ and
$$X = \begin{bmatrix} 1 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 3 \end{bmatrix}.$$
Hence, using the Jordan canonical form definition, we have
$$\begin{aligned} e^A &= Xe_A^JX^{-1} = X([1] \oplus \left[\begin{smallmatrix} e & e \\ 0 & e \end{smallmatrix}\right])X^{-1} \\ &= \begin{bmatrix} 1 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & e & e \\ 0 & 0 & e \end{bmatrix} \begin{bmatrix} 6 & 3 & 2 \\ 2 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix} \\ &= \begin{bmatrix} 6-7e & 3-4e & 2-3e \\ -6+10e & -3+6e & -2+4e \\ -6+6e & -3+3e & -2+3e \end{bmatrix}. \end{aligned}$$

3. Compute $\sqrt{A}$ for the matrix in Example 1. To obtain the square root, we use the polynomial interpolation definition. The eigenvalues of A are 0 and 1, with indices

1 and 2, respectively. The unique polynomial r of degree at most 2 satisfying the interpolation conditions $r(0) = f(0)$, $r(1) = f(1)$, $r'(1) = f'(1)$ is

$$r(t) = f(0)(t-1)^2 + t(2-t)f(1) + t(t-1)f'(1).$$

With $f(t) = t^{1/2}$, taking the positive square root, we have $r(t) = t(2-t)+t(t-1)/2$, and therefore

$$A^{1/2} = A(2I - A) + A(A - I)/2 = \begin{bmatrix} -6 & -3.5 & -2.5 \\ 8 & 5 & 3 \\ 6 & 3 & 3 \end{bmatrix}.$$

4. Consider the $m_k \times m_k$ Jordan block $J_k(\lambda_k)$ in (1). The polynomial satisfying the interpolation conditions (4) is

$$r(t) = f(\lambda_k) + (t-\lambda_k)f'(\lambda_k) + \frac{(t-\lambda_k)^2}{2!}f''(\lambda_k) + \cdots + \frac{(t-\lambda_k)^{m_k-1}}{(m_k-1)!}f^{(m_k-1)}(\lambda_k),$$

which of course is the first m_k terms of the Taylor series of f about λ_k. Hence, from the polynomial interpolation definition,

$$\begin{aligned} f(J_k(\lambda_k)) &= r(J_k(\lambda_k)) \\ &= f(\lambda_k)I + (J_k(\lambda_k) - \lambda_k I)f'(\lambda_k) + \frac{(J_k(\lambda_k) - \lambda_k I)^2}{2!}f''(\lambda_k) + \cdots \\ &\quad + \frac{(J_k(\lambda_k) - \lambda_k I)^{m_k-1}}{(m_k-1)!}f^{(m_k-1)}(\lambda_k). \end{aligned}$$

The matrix $(J_k(\lambda_k) - \lambda_k I)^j$ is zero except for 1s on the jth superdiagonal. This expression for $f(J_k(\lambda_k))$ is therefore equal to that in (3), confirming the consistency of the first two definitions of $f(A)$.

2 Matrix Square Root

Definitions:

Let $A \in \mathbb{C}^{n\times n}$. Any X such that $X^2 = A$ is a **square root** of A.

Facts:

1. If $A \in \mathbb{C}^{n\times n}$ has no eigenvalues on $\mathbb{R}^-$ (the closed negative real axis) then there is a unique square root X of A all of whose eigenvalues lie in the open right half-plane, and it is a primary matrix function of A. This is the **principal square root** of A and is written $X = A^{1/2}$. If A is real then $A^{1/2}$ is real. An integral representation is

$$A^{1/2} = \frac{2}{\pi} A \int_0^\infty (t^2 I + A)^{-1}\, dt.$$

2. A Hermitian positive definite matrix $A \in \mathbb{C}^{n\times n}$ has a unique Hermitian positive definite square root.

3. [CL74] A singular matrix $A \in \mathbb{C}^{n\times n}$ may or may not have a square root. A necessary and sufficient condition for A to have a square root is that in the "ascent sequence" of integers d_1, d_2, ... defined by

$$d_i = \dim(\ker(A^i)) - \dim(\ker(A^{i-1}))$$

no two terms are the same odd integer.

4. $A \in \mathbb{R}^{n\times n}$ has a real square root if and only if A satisfies the condition in the previous fact and A has an even number of Jordan blocks of each size for every negative eigenvalue.

5. The $n \times n$ identity matrix I_n has 2^n diagonal square roots $\mathrm{diag}(\pm 1)$. Only two of these are primary matrix functions, namely I and $-I$. Nondiagonal but symmetric nonprimary square roots of I_n include any Householder matrix $I - 2\mathbf{v}\mathbf{v}^T/(\mathbf{v}^T\mathbf{v})$ ($\mathbf{v} \neq 0$) and the identity matrix with its columns in reverse order. Nonsymmetric

square roots of I_n are easily constructed in the form XDX^{-1}, where X is nonsingular but nonorthogonal and $D = \text{diag}(\pm 1) \neq \pm I$.

Examples:

1. The Jordan block $\left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$ has no square root. The matrix

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

has ascent sequence $2, 1, 0, \ldots$ and so does have a square root—for example, the matrix

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}.$$

3 Matrix Exponential

Definitions:

The exponential of $A \in \mathbb{C}^{n \times n}$, written e^A or $\exp(A)$, is defined by

$$e^A = I + A + \frac{A^2}{2!} + \cdots + \frac{A^k}{k!} + \cdots.$$

Facts:

1. $e^{(A+B)t} = e^{At}e^{Bt}$ holds for all t if and only if $AB = BA$.

2. The differential equation in $n \times n$ matrices

$$\frac{dY}{dt} = AY, \qquad Y(0) = C, \qquad A, Y \in \mathbb{C}^{n \times n},$$

has solution $Y(t) = e^{At}C$.

3. The differential equation in $n \times n$ matrices

$$\frac{dY}{dt} = AY + YB, \qquad Y(0) = C, \qquad A, B, Y \in \mathbb{C}^{n\times n},$$

has solution $Y(t) = e^{At}Ce^{Bt}$.

4. $A \in \mathbb{C}^{n\times n}$ is unitary if and only if it can be written $A = e^{iH}$, where H is Hermitian. In this representation H can be taken to be Hermitian positive definite.

5. $A \in \mathbb{R}^{n\times n}$ is orthogonal with $\det(A) = 1$ if and only if $A = e^S$ with $S \in \mathbb{R}^{n\times n}$ skew-symmetric.

Examples:

1. Fact 5 is illustrated by the matrix

$$A = \begin{bmatrix} 0 & \alpha \\ -\alpha & 0 \end{bmatrix},$$

for which

$$e^A = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix}.$$

4 Matrix Logarithm

Definitions:

Let $A \in \mathbb{C}^{n\times n}$. Any X such that $e^X = A$ is a **logarithm** of A.

Facts:

1. If A has no eigenvalues on $\mathbb{R}^-$ then there is a unique logarithm X of A all of whose eigenvalues lie in the strip $\{\, z : -\pi < \mathrm{Im}(z) < \pi \,\}$. This is the **principal logarithm** of A, and is written $X = \log A$. If A is real then $\log A$ is real.

2. If $\rho(A) < 1$,
$$\log(I + A) = A - \frac{A^2}{2} + \frac{A^3}{3} - \frac{A^4}{4} + \cdots.$$

3. $A \in \mathbb{R}^{n\times n}$ has a real logarithm if and only if A is nonsingular and A has an even number of Jordan blocks of each size for every negative eigenvalue.

4. $\exp(\log A) = A$ holds when log is defined on the spectrum of $A \in \mathbb{C}^{n\times n}$. But $\log(\exp(A)) = A$ does not generally hold unless the spectrum of A is restricted.

5. If $A \in \mathbb{C}^{n\times n}$ is nonsingular then $\det(A) = \exp(\mathrm{tr}(\log A))$, where $\log A$ is any logarithm of A.

Examples:

For the matrix
$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
we have
$$\log(A) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

5 Matrix Sine and Cosine

Definitions:

The sine and cosine of $A \in \mathbb{C}^{n\times n}$ are defined by
$$\cos(A) \;=\; I - \frac{A^2}{2!} + \cdots + \frac{(-1)^k}{(2k)!}A^{2k} + \cdots,$$

$$\sin(A) \;=\; A - \frac{A^3}{3!} + \cdots + \frac{(-1)^k}{(2k+1)!}A^{2k+1} + \cdots.$$

Facts:

1. $\cos(2A) = 2\cos^2(A) - I$.

2. $\sin(2A) = 2\sin(A)\cos(A)$.

3. $\cos^2(A) + \sin^2(A) = I$.

4. The differential equation

$$\frac{d^2y}{dt^2} + Ay = 0, \qquad y(0) = y_0, \quad y'(0) = y_0'$$

has solution

$$y(t) = \cos(\sqrt{A}t)y_0 + (\sqrt{A})^{-1}\sin(\sqrt{A}t)y_0',$$

where $\sqrt{A}$ denotes any square root of A.

Examples:

1. For

$$A = \begin{bmatrix} 0 & i\,\alpha \\ i\,\alpha & 0 \end{bmatrix},$$

we have

$$e^A = \begin{bmatrix} \cos\alpha & i\sin\alpha \\ i\sin\alpha & \cos\alpha \end{bmatrix}.$$

2. For

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

we have

$$\cos(A) = \cos(1)I, \qquad \sin(A) = \begin{bmatrix} \sin(1) & \sin(1) & \sin(1) & \sin(1) \\ 0 & -\sin(1) & -2\sin(1) & -3\sin(1) \\ 0 & 0 & \sin(1) & 3\sin(1) \\ 0 & 0 & 0 & -\sin(1) \end{bmatrix}$$

and $\sin^2(A) = \sin(1)^2 I$, so $\cos(A)^2 + \sin(A)^2 = I$.

6 Matrix Sign Function

Definitions:

If $A = ZJ_AZ^{-1} \in \mathbb{C}^{n\times n}$ is a Jordan canonical form arranged so that

$$J_A = \begin{bmatrix} J_A^{(1)} & 0 \\ 0 & J_A^{(2)} \end{bmatrix},$$

where the eigenvalues of $J_1^{(1)} \in \mathbb{C}^{p\times p}$ lie in the open left half-plane and those of $J_2^{(2)} \in \mathbb{C}^{q\times q}$ lie in the open right half-plane, with $p + q = n$, then

$$\operatorname{sign}(A) = Z \begin{bmatrix} -I_p & 0 \\ 0 & I_q \end{bmatrix} Z^{-1}.$$

Alternative formulae are

$$\begin{aligned} \operatorname{sign}(A) &= A(A^2)^{-1/2}, \\ \operatorname{sign}(A) &= \frac{2}{\pi} A \int_0^\infty (t^2 I + A^2)^{-1}\, dt. \end{aligned} \tag{6}$$

If A has any pure imaginary eigenvalues then $\operatorname{sign}(A)$ is not defined.

Facts:

Let $S = \operatorname{sign}(A)$ be defined. Then

1. $S^2 = I$ (S is involutory).

2. S is diagonalizable with eigenvalues ± 1.

3. $SA = AS$.

4. If A is real then S is real.

5. If A is symmetric positive definite then $\operatorname{sign}(A) = I$.

Examples:

1. For the matrix A in Example 2 of the previous subsection we have $\operatorname{sign}(A) = A$, which follows from (6) and the fact that A is involutory.

7 Computational Methods for General Functions

Many methods have been proposed for evaluating matrix functions. Three general approaches of wide applicability are outlined here. They have in common that they do not require knowledge of Jordan structure and are suitable for computer implementation.

1. *Polynomial and Rational Approximations*

Polynomial approximations

$$p_m(X) = \sum_{k=0}^{m} b_k X^k, \qquad b_k \in \mathbb{C},\ X \in \mathbb{C}^{n\times n},$$

to matrix functions can be obtained by truncating or economizing a power series representation, or by constructing a best approximation (in some norm) of a given degree. How to most efficiently evaluate a polynomial at a matrix argument is a nontrivial question. Possibilities include Horner's method, explicit computation of the powers of the matrix, and a method of Paterson and Stockmeyer [GVL96, Sec. 11.2.4], [PS73] that is a combination of these two methods that requires fewer matrix multiplications.

Rational approximations $r_{mk}(X) = p_m(X)q_k(X)^{-1}$ are also widely used, particularly those arising from Padé approximation, which produces rationals matching as many terms of the Taylor series of the function at the origin as possible. The evaluation of rationals at matrix arguments needs careful consideration in order to find the best compromise between speed and accuracy. The main possibilities are

- Evaluating the numerator and denominator polynomials and then solving a multiple right-hand side linear system.
- Evaluating a continued fraction representation (in either top-down or bottom-up order).
- Evaluating a partial fraction representation.

Since polynomials and rationals are typically accurate over a limited range of matrices, practical methods involve a reduction stage prior to evaluating the polynomial or rational.

2. *Factorization Methods*

Many methods are based on the property $f(XAX^{-1}) = Xf(A)X^{-1}$. If X can be found such that $B = XAX^{-1}$ has the property that $f(B)$ is easily evaluated, then an obvious method results. When A is diagonalizable, B can be taken to be diagonal, and evaluation of $f(B)$ is trivial. In finite precision arithmetic, though, this approach is reliable only if X is well conditioned, that is, if the condition number $\kappa(X) = \|X\|\|X^{-1}\|$ is not too large. Ideally, X will be unitary, so that in the 2-norm $\kappa_2(X) = 1$. For Hermitian A, or more generally normal A, the spectral decomposition $A = QDQ^*$ with Q unitary and D diagonal is always possible, and if this decomposition can be computed then the formula $f(A) = Qf(D)Q^*$ provides an excellent way of computing $f(A)$.

For general A, if X is restricted to be unitary then the furthest that A can be reduced

is to Schur form: $A = QTQ^*$, where Q is unitary and T upper triangular. This decomposition is computed by the QR algorithm. Computing a function of a triangular matrix is an interesting problem. While Fact 11 of Subsection 1 gives an explicit formula for $F = f(T)$, the formula is not practically viable due to its exponential cost in n. Much more efficient is a recurrence of Parlett [Par76]. This is derived by starting with the observation that since F is representable as a polynomial in T, F is upper triangular, with diagonal elements $f(t_{ii})$. The elements in the strict upper triangle are determined by solving the equation $FT = TF$. Parlett's recurrence is:

Algorithm 1. Parlett's recurrence.

$f_{ii} = f(t_{ii})$, $i = 1\colon n$

for $j = 2\colon n$

  for $i = j-1\colon -1\colon 1$

$$f_{ij} = t_{ij}\frac{f_{ii} - f_{jj}}{t_{ii} - t_{jj}} + \Big(\sum_{k=i+1}^{j-1} f_{ik}t_{kj} - t_{ik}f_{kj}\Big) \,/\, (t_{ii} - t_{jj})$$

  end

end

This recurrence can be evaluated in $2n^3/3$ operations. The recurrence breaks down when $t_{ii} = t_{jj}$ for some $i \neq j$. In this case, T can be regarded as a block matrix $T = (T_{ij})$, with square diagonal blocks, possibly of different sizes. T can be reordered so that no two diagonal blocks have an eigenvalue in common; reordering means applying a unitary similarity transformation to permute the diagonal elements whilst preserving triangularity. Then a block form of the recurrence can be employed. This requires the evaluation of the diagonal blocks $F_{ii} = f(T_{ii})$, where T_{ii} will typically be of small dimension. A general way to obtain F_{ii} is via a Taylor series. The use of the block Parlett recurrence in combination

with a Schur decomposition represents the state of the art in evaluation of $f(A)$ for general functions [DH03].

3. *Iteration Methods*

Several matrix functions f can be computed by iteration:

$$X_{k+1} = g(X_k), \qquad X_0 = A, \tag{7}$$

where, for reasons of computational cost, g is usually a polynomial or a rational function. Such an iteration might converge for all A for which f is defined, or just for a subset of such A. A standard means of deriving matrix iterations is to apply Newton's method to an algebraic equation satisfied by $f(A)$. The iterations most used in practice are quadratically convergent, but iterations with higher orders of convergence are known.

4. *Contour Integration*

The Cauchy integral definition (5) provides a way to compute or approximate $f(A)$ via contour integration. While not suitable as a practical method for all functions or all matrices, this approach can be effective when numerical integration is done over a suitable contour using the repeated trapezium rule, whose high accuracy properties for periodic functions integrated over a whole period are beneficial [DH05], [TW05].

8 Computational Methods for Specific Functions

Some methods specialized to particular functions are now outlined.

1. *Matrix Exponential*

A large number of methods have been proposed for the matrix exponential, many of them of pedagogic interest only or of dubious numerical stability. Some of the more computationally useful methods are surveyed in [MVL03]. Probably the best general-

purpose method is the scaling and squaring method. In this method an integral power of 2, $\sigma = 2^s$ say, is chosen so that A/σ has norm not too far from 1. The exponential of the scaled matrix is approximated by an $[m/m]$ Padé approximant, $e^{A/2^s} \approx r_{mm}(A/2^s)$, and then s repeated squarings recover an approximation to e^A: $e^A \approx r_{mm}(A/2^s)^{2^s}$. Symmetries in the Padé approximant permit an efficient evaluation of $r_{mm}(A)$. The scaling and squaring method was originally developed in [MVL78] and [War77], and it is the method employed by MATLAB's `expm` function. How best to choose σ and m is described in [Hig05].

2. *Matrix Logarithm*

The (principal) matrix logarithm can be computed using an inverse scaling and squaring method based on the identity $\log A = 2^k \log A^{1/2^k}$, where A is assumed to have no eigenvalues on $\mathbb{R}^-$. Square roots are taken to make $\|A^{1/2^k} - I\|$ small enough that an $[m/m]$ Padé approximant approximates $\log A^{1/2^k}$ sufficiently accurately, for some suitable m. Then $\log A$ is recovered by multiplying by 2^k. To reduce the cost of computing the square roots and evaluating the Padé approximant, a Schur decomposition can be computed initially so that the method works with a triangular matrix. For details, see [CHKL01], [Hig01], [KL89, App. A].

3. *Matrix Cosine and Sine*

A method analogous to the scaling and squaring method for the exponential is the standard method for computing the matrix cosine. The idea is again to scale A to have norm not too far from 1 and then compute a Padé approximant. The difference is that the scaling is undone by repeated use of the double-angle formula $\cos(2A) = 2\cos^2 A - I$, rather than by repeated squaring. The sine function can be obtained as $\sin(A) = \cos(A - \frac{\pi}{2}I)$. See [SB80], [HS03], [HH05].

4. *Matrix Square Root*

The most numerically reliable way to compute matrix square roots is via the Schur decomposition, $A = QTQ^*$ [BH83]. Rather than use the Parlett recurrence, a square root U of the upper triangular factor T can be computed by directly solving the equation $U^2 = T$. The choices of sign in the diagonal of U, $u_{ii} = \sqrt{t_{ii}}$, determine which square root is obtained. When A is real, the real Schur decomposition can be used to compute real square roots entirely in real arithmetic [Hig87].

Various iterations exist for computing the principal square root when A has no eigenvalues on $\mathbb{R}^-$. The basic Newton iteration,

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}A), \qquad X_0 = A, \tag{8}$$

is quadratically convergent, but is numerically unstable unless A is extremely well conditioned and its use is not recommended [Hig86]. Stable alternatives include the Denman–Beavers iteration [DB76]

$$\begin{aligned} X_{k+1} &= \frac{1}{2}\left(X_k + Y_k^{-1}\right), \qquad X_0 = A, \\ Y_{k+1} &= \frac{1}{2}\left(Y_k + X_k^{-1}\right), \qquad Y_0 = I, \end{aligned}$$

for which $\lim_{k\to\infty} X_k = A^{1/2}$ and $\lim_{k\to\infty} Y_k = A^{-1/2}$, and the Meini iteration [Mei04]

$$\begin{aligned} Y_{k+1} &= -Y_k Z_k^{-1} Y_k, \qquad Y_0 = I - A, \\ Z_{k+1} &= Z_k + 2Y_{k+1}, \qquad Z_0 = 2(I + A), \end{aligned}$$

for which $Y_k \to 0$ and $Z_k \to 4A^{1/2}$. Both of these iterations are mathematically equivalent to (8) and hence are quadratically convergent.

An iteration not involving matrix inverses is the Schulz iteration

$$\begin{aligned} Y_{k+1} &= \tfrac{1}{2}Y_k(3I - Z_kY_k), \qquad Y_0 = A, \\ Z_{k+1} &= \tfrac{1}{2}(3I - Z_kY_k)Z_k, \qquad Z_0 = I, \end{aligned}$$

for which $Y_k \to A^{1/2}$ and $Z_k \to A^{-1/2}$ quadratically provided that $\|\operatorname{diag}(A-I, A-I)\| < 1$, where the norm is any consistent matrix norm [Hig97].

5. *Matrix sign function*

The standard method for computing the matrix sign function is the Newton iteration

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \qquad X_0 = A,$$

which converges quadratically to $\operatorname{sign}(A)$, provided A has no pure imaginary eigenvalues. In practice, a scaled iteration

$$X_{k+1} = \frac{1}{2}(\mu_k X_k + \mu_k^{-1} X_k^{-1}), \qquad X_0 = A.$$

is used, where the scale parameters μ_k are chosen to reduce the number of iterations needed to enter the regime where asymptotic quadratic convergence sets in. See [Bye87], [KL92].

The Newton–Schulz iteration

$$X_{k+1} = \frac{1}{2} X_k(3I - X_k^2), \qquad X_0 = A,$$

involves no matrix inverses but convergence is guaranteed only for $\|I - A^2\| < 1$.

A Padé family of iterations

$$X_{k+1} = X_k p_{\ell m}(1 - X_k^2) q_{\ell m}(1 - X_k^2)^{-1}, \qquad X_0 = A,$$

is obtained in [KL91], where $p_{\ell m}(\xi)/q_{\ell m}(\xi)$ is the $[\ell/m]$ Padé approximant to $(1-\xi)^{-1/2}$. The iteration is globally convergent to $\operatorname{sign}(A)$ for $\ell = m-1$ and $\ell = m$, and for $\ell \geq m-1$ is convergent when $\|I - A^2\| < 1$, with order of convergence $\ell + m + 1$ in all cases.

References

[BH83] Åke Björck and Sven Hammarling. A Schur method for the square root of a matrix. *Linear Algebra Appl.*, 52/53:127–140, 1983.

[Bye87] Ralph Byers. Solving the algebraic Riccati equation with the matrix sign function. *Linear Algebra Appl.*, 85:267–279, 1987.

[CHKL01] Sheung Hun Cheng, Nicholas J. Higham, Charles S. Kenney, and Alan J. Laub. Approximating the logarithm of a matrix to specified accuracy. *SIAM J. Matrix Anal. Appl.*, 22(4):1112–1125, 2001.

[CL74] G. W. Cross and P. Lancaster. Square roots of complex matrices. *Linear and Multilinear Algebra*, 1:289–293, 1974.

[Dav73] Chandler Davis. Explicit functional calculus. *Linear Algebra Appl.*, 6:193–199, 1973.

[DB76] Eugene D. Denman and Alex N. Beavers, Jr. The matrix sign function and computations in systems. *Appl. Math. Comput.*, 2:63–94, 1976.

[Des63] Jean Descloux. Bounds for the spectral norm of functions of matrices. *Numer. Math.*, 15:185–190, 1963.

[DH03] Philip I. Davies and Nicholas J. Higham. A Schur–Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. Appl.*, 25(2):464–485, 2003.

[DH05] Philip I. Davies and Nicholas J. Higham. Computing $f(A)b$ for matrix functions f. In Artan Boriçi, Andreas Frommer, Báalint Joó, Anthony Kennedy, and Brian Pendleton, editors, *QCD and Numerical Analysis III*, volume 47 of

Lecture Notes in Computational Science and Engineering, pages 15–24. Springer-Verlag, Berlin, 2005.

[GVL96] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

[HH05] Gareth I. Hargreaves and Nicholas J. Higham. Efficient algorithms for the matrix cosine and sine. Numerical Analysis Report 461, Manchester Centre for Computational Mathematics, Manchester, England, February 2005. To appear in Numerical Algorithms.

[Hig] Nicholas J. Higham. *Functions of a Matrix: Theory and Computation*. Book in preparation.

[Hig86] Nicholas J. Higham. Newton's method for the matrix square root. *Math. Comp.*, 46(174):537–549, April 1986.

[Hig87] Nicholas J. Higham. Computing real square roots of a real matrix. *Linear Algebra Appl.*, 88/89:405–430, 1987.

[Hig97] Nicholas J. Higham. Stable iterations for the matrix square root. *Numerical Algorithms*, 15(2):227–242, 1997.

[Hig01] Nicholas J. Higham. Evaluating Padé approximants of the matrix logarithm. *SIAM J. Matrix Anal. Appl.*, 22(4):1126–1135, 2001.

[Hig05] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM J. Matrix Anal. Appl.*, 26(4):1179–1193, 2005.

[HJ91] Roger A. Horn and Charles R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[HS03] Nicholas J. Higham and Matthew I. Smith. Computing the matrix cosine. *Numerical Algorithms*, 34:13–26, 2003.

[KL89] Charles S. Kenney and Alan J. Laub. Condition estimates for matrix functions. *SIAM J. Matrix Anal. Appl.*, 10(2):191–209, 1989.

[KL91] Charles S. Kenney and Alan J. Laub. Rational iterative methods for the matrix sign function. *SIAM J. Matrix Anal. Appl.*, 12(2):273–291, 1991.

[KL92] Charles S. Kenney and Alan J. Laub. On scaling Newton's method for polar decomposition and the matrix sign function. *SIAM J. Matrix Anal. Appl.*, 13(3):688–706, 1992.

[LT85] Peter Lancaster and Miron Tismenetsky. *The Theory of Matrices*. Academic Press, London, second edition, 1985.

[Mei04] Beatrice Meini. The matrix square root from a new functional perspective: Theoretical results and computational issues. *SIAM J. Matrix Anal. Appl.*, 26(2):362–376, 2004.

[MVL78] Cleve B. Moler and Charles F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.*, 20(4):801–836, 1978.

[MVL03] Cleve B. Moler and Charles F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49, 2003.

[Par76] B. N. Parlett. A recurrence among the elements of functions of triangular matrices. *Linear Algebra Appl.*, 14:117–121, 1976.

[PS73] Michael S. Paterson and Larry J. Stockmeyer. On the number of nonscalar multiplications necessary to evaluate polynomials. *SIAM J. Comput.*, 2(1):60–66, 1973.

[SB80] Steven M. Serbin and Sybil A. Blalock. An algorithm for computing the matrix cosine. *SIAM J. Sci. Statist. Comput.*, 1(2):198–204, 1980.

[TW05] L. N. Trefethen and J. A. C. Weideman. The fast trapezoid rule in scientific computing. Paper in preparation, 2005.

[War77] Robert C. Ward. Numerical computation of the matrix exponential with accuracy estimate. *SIAM J. Numer. Anal.*, 14(4):600–610, 1977.