Matrix Functions: A Short Course

Higham, Nicholas J. and Lijing, Lin

2013

MIMS EPrint: **2013.73**

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

Matrix Functions: A Short Course*

Nicholas J. Higham† Lijing Lin†

Contents

*Version of November 20, 2013. To appear in Matrix Functions and Matrix Equations, Zhaojun Bai, Weiguo Gao and Yangfeng Su (eds.), Series in Contemporary Applied Mathematics, World Scientific Publishing.

†School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk, http://www.maths.man.ac.uk/~higham, lijing.lin@manchester.ac.uk, http://www.maths.manchester.ac.uk/~lijing).

1 Introduction

A summary is given of a course on functions of matrices delivered by the first author (lecturer) and second author (teaching assistant) at the Gene Golub SIAM Summer School 2013 at Fudan University, Shanghai, China, July 22–26 2013 [35]. This article covers some essential features of the theory and computation of matrix functions. In the spirit of course notes the article is not a comprehensive survey and does not cite all the relevant literature. General references for further information are the book on matrix functions by Higham [32] and the survey by Higham and Al-Mohy [36] of computational methods for matrix functions.

2 History

Matrix functions are as old as matrix algebra itself. The term "matrix" was coined in 1850 [58] by James Joseph Sylvester, FRS (1814–1897),

while the study of matrix algebra was initiated by Arthur Cayley, FRS (1821–1895) in his "*A Memoir on the Theory of Matrices*" (1858) [11]. In that first paper, Cayley considered matrix square roots.

Notable landmarks in the history of matrix functions include:

- Laguerre (1867) [45] and Peano (1888) [53] defined the exponential of a matrix via its power series.
- Sylvester (1883) stated the definition of $f(A)$ for general f via the interpolating polynomial [59]. Buchheim (1886) [10], [34] extended Sylvester's interpolation formula to arbitrary eigenvalues.
- Frobenius (1896) [22] showed that if f is analytic then $f(A)$ is the sum of the residues of $(zI - A)^{-1}f(z)$ at the eigenvalues of A, thereby anticipating the Cauchy integral representation, which was used by Poincaré (1899) [54].
- The Jordan form definition was used by Giorgi (1928) [23], and Cipolla (1932) [14] extended it to produce nonprimary matrix functions.
- The first book on matrix functions was published by Schwerdtfeger (1938) [56].
- Frazer, Duncan and Collar published the book *Elementary Matrices and Some Applications to Dynamics and Differential Equations* [21] in 1938, which was "the first book to treat matrices as a branch of applied mathematics" [15].
- A research monograph on functions of matrices was published by Higham (2008) [32].

3 Theory

3.1 Definitions

We are concerned with functions $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ that are defined in terms of an underlying scalar function f. Given $f(t)$, one can define $f(A)$ by substituting A for t: e.g.,

$$f(t) = \frac{1+t^2}{1-t} \quad \Rightarrow \quad f(A) = (I-A)^{-1}(I+A^2),$$

$$\log(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \cdots, \quad |x| < 1,$$

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

This way of defining $f(A)$ works for f a polynomial, a rational function, or a function having a convergent power series (see section 6.1). Note

that f is not evaluated elementwise on the matrix A, as is the case in some programming languages.

For general f, there are various equivalent ways to formally define a matrix function. We give three definitions, based on the Jordan canonical form, polynomial interpolation, and the Cauchy integral formula.

3.1.1 Definition via Jordan canonical form

Any matrix $A \in \mathbb{C}^{n\times n}$ can be expressed in the Jordan canonical form

$$Z^{-1}AZ = J = \mathrm{diag}(J_1, J_2, \ldots, J_p), \tag{3.1a}$$

$$J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k\times m_k}, \tag{3.1b}$$

where Z is nonsingular and $m_1 + m_2 + \cdots + m_p = n$. Denote by

- $\lambda_1, \ldots, \lambda_s$ the distinct eigenvalues of A,
- n_i the order of the largest Jordan block in which λ_i appears, which is called the *index* of λ_i.

We say the function f is *defined on the spectrum* of A if the values

$$f^{(j)}(\lambda_i), \qquad j = 0\colon n_i - 1, \quad i = 1\colon s, \tag{3.2}$$

exist.

Definition 3.1 (matrix function via Jordan canonical form)**.** *Let f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$ and let A have the Jordan canonical form* (3.1)*. Then*

$$f(A) := Zf(J)Z^{-1} = Z\mathrm{diag}(f(J_k))Z^{-1}, \tag{3.3}$$

where

$$f(J_k) := \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \cdots & \dfrac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}. \tag{3.4}$$

The definition yields a matrix $f(A)$ that can be shown to be independent of the particular Jordan canonical form.

In the case of multivalued functions such as $\sqrt{t}$ and $\log t$ it is implicit that a single branch has been chosen in (3.4) and the resulting function is called a *primary* matrix function. If an eigenvalue occurs in more than one Jordan block and a different choice of branch is made in two different blocks then a *nonprimary* matrix function is obtained (see section 3.1.5).

3.1.2 Definition via interpolating polynomial

Before giving the second definition, we recall some background on polynomials at a matrix argument.

- The *minimal polynomial* of $A \in \mathbb{C}^{n\times n}$ is defined to be the unique monic polynomial ϕ of lowest degree such that $\phi(A) = 0$. The existence of the minimal polynomial is proved in most textbooks on linear algebra.
- By considering the Jordan canonical form it is not hard to see that $\phi(t) = \prod_{i=1}^{s}(t-\lambda_i)^{n_i}$, where $\lambda_1, \dots, \lambda_s$ are the distinct eigenvalues of A and n_i is the index of λ_i. It follows immediately that ϕ is zero on the spectrum of A (that is, the values (3.2) are all zero for $f(t) = \phi(t)$).
- Given any polynomial p and any matrix $A \in \mathbb{C}^{n\times n}$, p is clearly defined on the spectrum of A and $p(A)$ can be defined by substitution.
- For polynomials p and q, $p(A) = q(A)$ if and only if p and q take the same values on the spectrum [32, Thm. 1.3]. Thus the matrix $p(A)$ is completely determined by the values of p on the spectrum of A.

The following definition gives a way to generalize the property of polynomials in the last bullet point to arbitrary functions and define $f(A)$ in terms of the values of f on the spectrum of A.

Definition 3.2 (matrix function via Hermite interpolation)**.** *Let f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$. Then $f(A) := p(A)$, where p is the unique polynomial of degree less than $\sum_{i=1}^{s} n_i$ (which is the degree of the minimal polynomial) that satisfies the interpolation conditions*

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

The polynomial p specified in the definition is known as the Hermite interpolating polynomial.

For an example, let $f(t) = t^{1/2}$ (the principal branch of the square root function, so that $\operatorname{Re} t^{1/2} \ge 0$), $A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}$, $\lambda(A) = \{1, 4\}$. Seeking $p(t)$ with $p(1) = f(1)$ and $p(4) = f(4)$, we obtain

$$p(t) = f(1)\frac{t-4}{1-4} + f(4)\frac{t-1}{4-1} = \frac{1}{3}(t+2).$$

$$\Rightarrow \quad A^{1/2} = p(A) = \frac{1}{3}(A + 2I) = \frac{1}{3}\begin{bmatrix} 4 & 2 \\ 1 & 5 \end{bmatrix}.$$

Several properties follow immediately from this definition:

- $f(A) = p(A)$ is a polynomial in A, where the polynomial p depends on A.
- $f(A)$ commutes with A.
- $f(A^T) = f(A)^T$.

Because the minimal polynomial divides the characteristic polynomial, $q(t) = \det(tI - A)$, it follows that $q(A) = 0$, which is the Cayley–Hamilton theorem. Hence A^n can be expressed as a linear combination of lower powers of A: $A^n = \sum_{k=0}^{n-1} c_k A^k$. Using this relation recursively we find that any power series collapses to a polynomial. For example, $e^A = \sum_{k=0}^{\infty} A^k/k! = \sum_{k=0}^{n-1} d_k A^k$ (but the d_k depend on A).

3.1.3 Definition via Cauchy integral theorem

Definition 3.3 (matrix function via Cauchy integral)**.** *For $A \in \mathbb{C}^{n\times n}$,*

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

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

3.1.4 Multiplicity and equivalence of definitions

Definitions 3.1, 3.2, and 3.3 are equivalent, modulo the analyticity assumption for the Cauchy integral definition [32, Thm. 1.12]. Indeed this equivalence extends to other definitions, as noted by Rinehart [55]:

> "There have been proposed in the literature since 1880 eight distinct definitions of a matric function, by Weyr, Sylvester and Buchheim, Giorgi, Cartan, Fantappiè, Cipolla, Schwerdtfeger and Richter .... All of the definitions, except those of Weyr and Cipolla are essentially equivalent."

The definitions have different strengths. For example, the interpolation definition readily yields some key basic properties (as we have already seen), the Jordan canonical form definition is useful for solving matrix equations (e.g., $X^2 = A$, $e^X = A$) and for evaluation when A is normal, and the Cauchy integral definition can be useful both in theory and in computation (see section 7.2).

3.1.5 Nonprimary matrix functions

Nonprimary matrix functions are ones that are not obtainable from our three definitions, or that violate the single branch requirement in Definition 3.1. Thus a nonprimary matrix function of A is obtained from Definition 3.1 if A is derogatory and a different branch of f is taken in

two different Jordan blocks for λ. For example, the 2×2 identity matrix has two primary square roots and an infinity of nonprimary square roots:

$$\begin{aligned} I_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}^2 && \text{primary} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^2 = \begin{bmatrix} \cos\theta & \sin\theta \\ \sin\theta & -\cos\theta \end{bmatrix}^2 && \text{nonprimary.} \end{aligned}$$

In general, primary matrix functions are expressible as polynomials in A, while nonprimary ones are not. The 2×2 zero matrix 0_2 is its own primary square root. Any nilpotent matrix of degree 2 is also a nonprimary square root of 0_2, for example $\left[\begin{smallmatrix} 0 & 1 \\ 0 & 0 \end{smallmatrix}\right]$, but the latter matrix is not a polynomial in 0_2.

The theory of matrix functions is almost exclusively concerned with primary matrix functions, but nonprimary functions are needed in some applications, such as the embeddability problem in Markov chains [32, Sec. 2.3].

3.1.6 Principal logarithm, root, and power

Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{R}^-$ (the closed real axis). We need the following definitions.

Principal log: $X = \log A$ denotes the unique X such that $e^X = A$ and $-\pi < \operatorname{Im}\lambda_i < \pi$ for every eigenvalue λ_i of X.

Principal pth root: For integer $p > 0$, $X = A^{1/p}$ is the unique X such that $X^p = A$ and $-\pi/p < \arg\lambda_i < \pi/p$ for every eigenvalue λ_i of X.

Principal power: For $s \in \mathbb{R}$, the principal power is defined as $A^s = e^{s\log A}$, where $\log A$ is the principal logarithm. An integral representation is also available:

$$A^s = \frac{\sin(s\pi)}{s\pi} A \int_0^\infty (t^{1/s} I + A)^{-1} dt, \qquad s \in (0,1).$$

3.2 Properties and formulas

Three basic properties of $f(A)$ were stated in section 3.1.2. Some other important properties are collected in the following theorem.

Theorem 3.4 ([32, Thm. 1.13])**.** *Let $A \in \mathbb{C}^{n\times n}$ and let f be defined on the spectrum of A. Then*

(a) $f(XAX^{-1}) = Xf(A)X^{-1}$;

(b) *the eigenvalues of $f(A)$ are $f(\lambda_i)$, where the λ_i are the eigenvalues of A;*

(c) *if X commutes with A then X commutes with $f(A)$;*
(d) *if $A = (A_{ij})$ is block triangular then $F = f(A)$ is block triangular with the same block structure as A, and $F_{ii} = f(A_{ii})$;*
(e) *if $A = \mathrm{diag}(A_{11}, A_{22}, \dots, A_{mm})$ is block diagonal then $f(A) = \mathrm{diag}(f(A_{11}), f(A_{22}), \dots, f(A_{mm}))$.*

Some more advanced properties are as follows.

- $f(A) = 0$ if and only if (from Definition 3.1 or 3.2) $f^{(j)}(\lambda_i) = 0$, $j = 0\colon n_i - 1$, $i = 1\colon s$.
- The sum, product, composition of functions work "as expected":
  - $(\sin + \cos)(A) = \sin A + \cos A$,
  - $f(t) = \cos(\sin t) \quad \Rightarrow \quad f(A) = \cos(\sin A)$.
- Polynomial functional relations generalize from the scalar case. For example: if $G(f_1, \dots, f_m) = 0$, where G is a polynomial, then $G(f_1(A), \dots, f_m(A)) = 0$. E.g.,
  - $\sin^2 A + \cos^2 A = I$,
  - $(A^{1/p})^p = A$ for any integer $p > 0$,
  - $e^{iA} = \cos A + i \sin A$.
- However, other plausible relations can fail:
  - $f(A^*) \neq f(A)^*$ in general,
  - $e^{\log A} = A$ but $\log e^A \neq A$ in general,
  - $e^A \neq (e^{A/\alpha})^\alpha$ in general,
  - $(AB)^{1/2} \neq A^{1/2}B^{1/2}$ in general,
  - $e^{(A+B)t} = e^{At}e^{Bt}$ for all t if and only if $AB = BA$.

  Correction terms involving the matrix unwinding function can be introduced to restore equality in the second to fourth cases [7].

3.3 Fréchet derivative and condition number

3.3.1 Relative condition number

An important issue in the computation of matrix functions is the conditioning. The data may be uncertain and rounding errors from finite precision computations can often be interpreted via backward error analysis as being equivalent to perturbations in the data. So it is important to understand the sensitivity of $f(A)$ to perturbations in A. Sensitivity is measured by the condition number defined as follows.

Definition 3.5. *Let $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ be a matrix function. The relative condition number of f is*

$$\mathrm{cond}(f, A) := \lim_{\epsilon\to 0} \sup_{\|E\| \le \epsilon\|A\|} \frac{\|f(A+E) - f(A)\|}{\epsilon\|f(A)\|},$$

where the norm is any matrix norm.

3.3.2 Fréchet derivative

To obtain explicit expressions for $\mathrm{cond}(f, A)$, we need an appropriate notion of derivative for matrix functions. The *Fréchet derivative* of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at a point $A \in \mathbb{C}^{n\times n}$ is a linear mapping $L_f(A, \cdot) : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ such that for all $E \in \mathbb{C}^{n\times n}$

$$f(A + E) = f(A) + L_f(A, E) + o(\|E\|).$$

It is easy to show that the condition number $\mathrm{cond}(f, A)$ can be characterized as

$$\mathrm{cond}(f, A) = \frac{\|L_f(A)\|\,\|A\|}{\|f(A)\|},$$

where

$$\|L_f(A)\| := \max_{Z\neq 0} \frac{\|L_f(A, Z)\|}{\|Z\|}.$$

3.3.3 Condition number estimation

Since L_f is a linear operator,

$$\mathrm{vec}(L_f(A, E)) = K_f(A)\mathrm{vec}(E)$$

where $K_f(A) \in \mathbb{C}^{n^2\times n^2}$ is a matrix independent of E known as the *Kronecker form* of the Fréchet derivative. It can be shown that $\|L_f(A)\|_F = \|K_f(A)\|_2$ and that $\|L_f(A)\|_1$ and $\|K_f(A)\|_1$ differ by at most a factor n. Hence estimating $\mathrm{cond}(f, A)$ reduces to estimating $\|K_f(A)\|$ and this can be done using a matrix norm estimator, such as the block 1-norm estimator of Higham and Tisseur [39].

4 Applications

Functions of matrices play an important role in many applications. Here we describe some examples.

4.1 Toolbox of matrix functions

In software we want to be able to evaluate interesting f at matrix arguments as well as scalar arguments. For example, trigonometric matrix functions, as well as matrix roots, arise in the solution of second order differential equations: the initial value problem

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

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. On the other hand, the differential equation can be converted to a first order system and then solved using the exponential:

$$\begin{bmatrix} y' \\ y \end{bmatrix} = \exp\left(\begin{bmatrix} 0 & -tA \\ tI_n & 0 \end{bmatrix}\right)\begin{bmatrix} y_0' \\ y_0 \end{bmatrix}.$$

4.2 Nuclear magnetic resonance

In nuclear magnetic resonance (NMR) spectroscopy, the Solomon equations

$$\frac{dM}{dt} = -RM, \quad M(0) = I$$

relate a matrix of intensities $M(t)$ to a symmetric, diagonally dominant matrix R (known as the relaxation matrix). Hence $M(t) = e^{-Rt}$. NMR workers need to solve both forward and inverse problems:

- in simulations and testing, compute $M(t)$ given R;
- determine R from observed intensities: estimation methods are used since not all the m_{ij} are observed.

4.3 Phi functions and exponential integrators

The φ functions are defined by the recurrence $\varphi_{k+1}(z) = \frac{\varphi_k(z)-1/k!}{z}$ with $\varphi_0(z) = e^z$, and are given explicitly by

$$\varphi_k(z) = \sum_{j=0}^{\infty} \frac{z^j}{(j+k)!}.$$

They appear in explicit solutions to certain linear differential equations:

$$\begin{aligned} \frac{dy}{dt} = Ay, \quad y(0) = y_0 \quad &\Rightarrow \quad y(t) = e^{At}y_0, \\ \frac{dy}{dt} = Ay + b, \quad y(0) = 0 \quad &\Rightarrow \quad y(t) = t\varphi_1(tA)b, \\ \frac{dy}{dt} = Ay + ct, \quad y(0) = 0 \quad &\Rightarrow \quad y(t) = t^2\varphi_2(tA)c, \end{aligned}$$

and more generally provide an explicit solution for a differential equation with right-hand side $Ay + p(t)$ with p a polynomial.

Consider an initial value problem written in the form

$$u'(t) = Au(t) + g(t, u(t)), \quad u(t_0) = u_0, \quad t \geq t_0, \tag{4.1}$$

where $u(t) \in \mathbb{C}^n$, $A \in \mathbb{C}^{n\times n}$, and g is a nonlinear function. Spatial semidiscretization of partial differential equations leads to systems in this form with A representing a discretized linear operator. Thus A may be large and sparse. The solution can be written as [47, Lem. 5.1]

$$u(t) = e^{(t-t_0)A}u_0 + \sum_{k=1}^{\infty} \varphi_k\big((t-t_0)A\big)(t-t_0)^k\, u_k, \tag{4.2}$$

where

$$u_k = \frac{d^{k-1}}{dt^{k-1}} g(t, u(t)) \,|_{t=t_0}\ .$$

By truncating the series in (4.2), we obtain the approximation

$$u(t) \approx \widehat{u}(t) = e^{(t-t_0)A}u_0 + \sum_{k=1}^{p} \varphi_k\big((t-t_0)A\big)(t-t_0)^k\, u_k. \tag{4.3}$$

Exponential integrator methods are obtained by employing suitable approximations to the vectors u_k [40]. The simplest method is the exponential time differencing (ETD) Euler method [46]

$$y_n = e^{hA}y_{n-1} + h\varphi_1(hA)g(t_{n-1}, y_{n-1}).$$

Clearly, implementing an exponential integrator involves computing matrix–vector products involving the exponential and the φ functions.

4.4 Complex networks

Let A be an adjacency matrix of an undirected network. Certain characteristics of the network are defined in terms of the matrix exponential [20]

$$e^A = I + A + \frac{1}{2}A^2 + \frac{1}{3!}A^3 + \frac{1}{4!}A^4 + \cdots\ .$$

The *centrality* of node i, defined as $(e^A)_{ii}$, measures how important that node is. The resolvent $(I - \alpha A)^{-1}$ can be used in place of e^A, as was originally done by Katz [41]. The *communicability* between nodes i and j, defined as $(e^A)_{ij}$, measures how well information is transferred between the nodes.

4.5 Random multivariate samples in statistics

Suppose we wish to generate random vectors distributed as $y \sim N(\mu, C)$, where $N(\mu, C)$ denotes the multivariate normal distribution with mean μ and covariance matrix C. This can be done by generating standard normally distributed vectors $x \sim N(0, I)$ and setting $y = \mu + Lx$, where

$C = LL^T$ is the Cholesky factorization. However, in some applications C has dimension greater than 10^{12} and computing the Cholesky factor is impractical. Chen, Anitescu, and Saad [12] note that one may instead generate y as $y = \mu + C^{1/2}x$. Methods are available for computing $C^{1/2}x$ that require only matrix–vector products with C (see sections 7.2 and 7.3) and so this computation is feasible, even if computing the Cholesky factor is not.

4.6 The average eye in optics

The first order character of an optical system is characterized by the *transference matrix*

$$T = \begin{bmatrix} S & \delta \\ 0 & 1 \end{bmatrix} \in \mathbb{R}^{5\times 5},$$

where $S \in \mathbb{R}^{4\times 4}$ is symplectic, that is,

$$S^T J S = J = \begin{bmatrix} 0 & I_2 \\ -I_2 & 0 \end{bmatrix}.$$

A straightforward average $m^{-1}\sum_{i=1}^m T_i$ is not in general a transference matrix. Harris [27] proposes as a suitable average the matrix $\exp\left(m^{-1}\sum_{i=1}^m \log T_i\right)$, where log is the principal logarithm, which is always a transference matrix.

5 Problem classification

The choice of method to compute $f(A)$ should take into account the properties of A, the size of the problem, the type of function, and accuracy requirements. We classify the problems according to their size and then describe some of the main methods for each class in sections 6 and 7.

5.1 Small/medium scale $f(A)$ problems

For this class of problems it is possible to compute a decomposition of A and to store $f(A)$.

For a normal matrix A we can compute the Schur (spectral) decomposition $A = QDQ^*$, with Q unitary and $D = \mathrm{diag}(d_i)$, and then form $f(A) = Q\mathrm{diag}(f(d_i))Q^*$. If A is nonnormal we can compute a Schur decomposition $A = QTQ^*$, with Q unitary and T upper triangular, and then the Schur–Parlett algorithm described in section 6.7 can be used.

For a number of matrix functions, such as matrix roots, the matrix sign function, and the unitary polar factor, $f(A)$ can be computed by

a matrix iteration $X_{k+1} = g(X_k)$, $X_0 = A$, where g is some nonlinear function. Usually, g is rational and so the iterations require only matrix multiplication and the solution of multiple right-hand side linear systems.

Another important tool for evaluating matrix functions is approximation, whereby $f(A)$ is approximated by $r(A)$, where $r(x)$ is a polynomial or rational approximation to $f(x)$, such as a truncated Taylor series or a Padé approximant. In this case some preprocessing is needed to get A into a region where the approximant is sufficiently accurate.

5.2 Large scale $f(A)b$ problems

If A is sufficiently large and sparse then it will be undesirable or impossible to compute a Schur decomposition of A or store $f(A)$. Therefore the problem of interest is to compute $f(A)b$, the action of $f(A)$ on b, without first computing $f(A)$. There are in general two different cases.

- Case 1: we are able to solve $Ax = b$ by a sparse direct method. Then methods based on the Cauchy integral formula can be used. Also, rational Krylov methods can be used with direct solves.
- Case 2: we can only compute matrix-vector products Ax (and perhaps A^*x). Exponential integrators for sufficiently large problems are contained in this case. Krylov methods based on the Arnoldi or Lanczos process, or methods employing polynomial approximations, can be used.

5.3 Accuracy requirements

The desired accuracy is an important question when choosing a method. This may range from full double precision accuracy (about 16 significant decimal digits) to just 3 or 4 digits if the matrix A is subject to large measurement errors, as may be the case in problems in engineering or healthcare.

Some methods will accept the error tolerance as a parameter, while others are designed always to aim for full precision. Some methods work best when less than full precision is required.

A further consideration is that for testing purposes we may want to compute a very accurate solution that can be taken as the "exact solution". Thus we may need a method that can deliver a computed solution correct to quadruple precision, or even higher precision, when implemented in high precision arithmetic.

6 Methods for $f(A)$

6.1 Taylor series

The Taylor series is a basic tool for approximating matrix functions. If f has a Taylor series expansion

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

with radius of convergence r then for $A \in \mathbb{C}^{n\times n}$ the series

$$f(A) = \sum_{k=0}^{\infty} a_k (A-\alpha I)^k \tag{6.1}$$

converges if $|\lambda_i - \alpha| < r$ for every eigenvalue λ_i of A [32, Thm. 4.7]. The error in a truncated Taylor series with terms up to $(A-\alpha I)^{m-1}$ in (6.1) can be bounded in terms of the mth derivative of f at matrix arguments. Just as for scalar Taylor series, numerical cancellation must be avoided by restricting the size of $\|A-\alpha I\|$.

6.2 Padé approximation

For a given scalar function $f(x)$, the rational function $r_{km}(x) = p_{km}(x)/q_{km}(x)$ is a $[k/m]$ *Padé approximant* of f if r_{km} has numerator and denominator polynomials of degrees at most k and m, respectively, $q_{km}(0) = 1$, and

$$f(x) - r_{km}(x) = O(x^{k+m+1}).$$

If a $[k/m]$ Padé approximant exists then it is unique. Padé approximants tend to be more efficient for matrix arguments than truncated Taylor series in that they can deliver similar accuracy at lower cost. For some important functions, such as the exponential and the logarithm, Padé approximants are explicitly known.

The choice of method to evaluate a Padé approximant at a matrix argument is based on balancing numerical stability and computational cost. Possibilities are

- a ratio of polynomials: $r_{km}(A) = q_{km}(A)^{-1} p_{km}(A)$, with an appropriate way to evaluate $p_{km}(A)$ and $q_{km}(A)$ (for example, Horner's method or the Paterson–Stockmeyer method [32, Sec. 4.2], [52]),
- continued fraction form, with bottom-up or top-down evaluation,
- partial fraction form.

6.3 Similarity transformations

Given a factorization $A = XBX^{-1}$, we can use the formula $f(A) = Xf(B)X^{-1}$, provided that $f(B)$ is easily computable, for example if $B = \mathrm{diag}(\lambda_i)$. However, any error ΔB in $f(B)$ can be magnified by as much as $\kappa(X) = \|X\|\|X^{-1}\| \geq 1$ in $f(A)$. Therefore we prefer to work with unitary X. Hence we typically use an eigendecomposition (diagonal B) when A is normal ($AA^* = A^*A$), or a Schur decomposition (triangular B) in general.

We could also take $B = \mathrm{diag}(B_i)$ block diagonal and require X to be well conditioned. Such a decomposition can be computed by starting with a Schur decomposition and then eliminating off-diagonal blocks using similarity transformations obtained by solving Sylvester equations. This approach needs a parameter: the maximum allowed condition number of individual transformations. The larger that parameter, the more numerous and smaller the diagonal blocks will be. The diagonal blocks B_i are triangular but have no particular eigenvalue distribution, so computing $f(B_i)$ is nontrivial. Block diagonalization has not proved to be a popular approach.

6.4 Schur method for matrix roots

Björck and Hammarling [9] show that a square root X of a matrix $A \in \mathbb{C}^{n\times n}$ can be computed by computing a Schur decomposition $A = QTQ^*$, solving $U^2 = T$ for the upper triangular matrix U by the recurrence

$$u_{ii} = \sqrt{t_{ii}}, \quad u_{ij} = \frac{t_{ij} - \sum_{k=i+1}^{j-1} u_{ik}u_{kj}}{u_{ii} + u_{jj}}, \tag{6.2}$$

and then forming $X = QUQ^*$. This method has essentially optimal numerical stability. It was extended to use the real Schur decomposition for real matrices by Higham [29] and to compute pth roots by Smith [57]. The recurrences for $p > 2$ are substantially more complicated than those for $p = 2$.

Recently, Deadman, Higham, and Ralha [18] have developed blocked versions of the recurrence (6.2) that give substantially better performance on modern computers.

6.5 Parlett's recurrence

If T is upper triangular then $F = f(T)$ is upper triangular and the diagonal elements of F are $f_{ii} = f(t_{ii})$. Parlett [51] shows that the off-diagonal elements of F can be obtained from the recurrence, derived

from $FT = TF$,

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

which enables F to be computed a column or a superdiagonal at a time. The recurrence fails when T has repeated eigenvalues and can suffer severe loss of accuracy in floating point arithmetic when two eigenvalues t_{ii} and t_{jj} are very close. A way around these problems is to employ a block form of this recurrence.

6.6 Block Parlett recurrence

For upper triangular T, Parlett [50] partitions $T = (T_{ij})$ with square diagonal blocks. Then F has same block upper triangular structure and $F_{ii} = f(T_{ii})$. The equation $FT = TF$ leads to Sylvester equations

$$T_{ii}F_{ij} - F_{ij}T_{jj} = F_{ii}T_{ij} - T_{ij}F_{jj} + \sum_{k=i+1}^{j-1} (F_{ik}T_{kj} - T_{ik}F_{kj}),$$

which are nonsingular as long as no two different diagonal blocks T_{ii} and T_{jj} have an eigenvalue in common. Thus F can be computed a block superdiagonal or a block column at a time. We can expect numerical difficulties when two blocks have close spectra.

6.7 Schur–Parlett algorithm

Davies and Higham [16] build from the block Parlett recurrence a general purpose algorithm for computing $f(A)$. The key ideas are to reorder and re-block in an attempt to produce well conditioned Sylvester equations and to evaluate $f(T_{ii})$ from a Taylor series (unless a more specific method is available for the given f).

The outline of the Schur–Parlett algorithm is:

- Compute a Schur decomposition $A = QTQ^*$.
- Reorder T to block triangular form in which eigenvalues within a diagonal block are "close" and those of different diagonal blocks are "well separated".
- Evaluate $F_{ii} = f(T_{ii})$ for each i.
- Solve in an appropriate order the Sylvester equations

$$T_{ii}F_{ij} - F_{ij}T_{jj} = F_{ii}T_{ij} - T_{ij}F_{jj} + \sum_{k=i+1}^{j-1} (F_{ik}T_{kj} - T_{ik}F_{kj}).$$

- Undo the unitary transformations.

Reordering step. The eigenvalues are split into sets such that λ_i and λ_j go in the same set if, for some parameter $\delta > 0$, $|\lambda_i - \lambda_j| \leq \delta$. An ordering of sets on the diagonal is chosen and a sequence of swaps of diagonal elements to produce that ordering determined. The swaps are effected by unitary transformations [8].

Function of atomic diagonal block. Let $U \in \mathbb{C}^{m\times m}$ represent an atomic diagonal block of the reordered Schur form. Assume f has a Taylor series with an infinite radius of convergence and that all the derivatives are available. We write $U = \sigma I + M$, where $\sigma = \mathrm{trace}(U)/m$ is the mean of the eigenvalues, and then take the Taylor series about σ:

$$f(U) = \sum_{k=0}^{\infty} \frac{f^{(k)}(\sigma)}{k!} M^k.$$

Because the convergence of the series can be very nonmonotonic we truncate it based on a strict error bound.

The key features of the algorithm are as follows.

- It usually costs $O(n^3)$ flops, but can cost up to $n^4/3$ flops if large blocks are needed (which can happen only when there are many close or clustered eigenvalues).
- It needs derivatives if there are blocks of size greater than 1. This is the price to pay for treating general f and nonnormal A (see (3.4)).
- The choice of $\delta = 0.1$ for the blocking parameter δ performs well most of time. However, it is possible for the algorithm to be unstable for all δ.
- This is the best general $f(A)$ algorithm and is the benchmark for comparing with other $f(A)$ algorithms, both general and specific.
- The algorithm is the basis of the MATLAB function `funm` and the NAG codes F01EK, F01EL, F01EM (real arithmetic) and F01FK, F01FL, F01FM (complex arithmetic).

6.8 (Inverse) scaling and squaring for the logarithm and exponential

The most popular approaches to computing the matrix exponential and the matrix logarithm are the (inverse) scaling and squaring methods, which employ Padé approximants $r_m \equiv r_{mm}$ together with initial transformations that ensure that the Padé approximants are sufficiently ac-

curate. The basic identities on which the methods are based are

$$e^A \approx r_m(A/2^s)^{2^s}, \qquad r_m(x) \approx e^x,$$
$$\log(A) \approx 2^s r_m(A^{1/2^s} - I), \qquad r_m(x) \approx \log(1+x),$$

respectively. In designing algorithms the key questions are how to choose the integers s and m. Originally a fixed choice was made, based on a priori truncation error bounds, but the state of the art algorithms use a dynamic choice that depends on A. Truncation errors from the Padé approximants are accounted for by using backward error bounds that show the approximate logarithm or exponential to be the true logarithm or exponential of a matrix within normwise relative distance u of A (these backward error bounds do not take into account rounding errors).

It is beyond our scope to describe the algorithms here. Instead we give a brief historical summary.

6.8.1 Matrix exponential

1967 The scaling and squaring method is suggested by Lawson [46].

1977 Ward [62] uses Padé degree $m = 8$ and chooses s so that $\|2^{-s}A\|_1 \le 1$.

1978–2005 Based on error analysis of Moler and Van Loan [49], MATLAB function `expm` uses $m = 6$ and chooses s so that $\|2^{-s}A\|_\infty \le 1/2$.

2005 Higham [31], [33] develops a dynamic choice of parameters allowing $m \in \{3, 5, 7, 9, 13\}$ and with $\|2^{-s}A\|_1 \le \theta_m$ for certain parameters θ_m. This algorithm is incorporated in MATLAB R2006a.

2009 Al-Mohy and Higham [2] improve the method of Higham [31] by using sharper truncation error bounds that depend on terms $\mu_k = \|A^k\|_1^{1/k}$ instead of $\|A\|_1$. The quantities μ_k are estimated for several small values of k using a matrix norm estimator [39]. This algorithm is implemented in NAG Library routines F01EC/F01FC (Mark 25).

6.8.2 Matrix logarithm

1989 Kenney and Laub [42, App. A] introduce the inverse scaling and squaring method, used with a Schur decomposition. Square roots are computed using the Schur method of section 6.4. Based on forward error bounds in [43], Kenney and Laub take $m = 8$ and require $\|I - A^{1/2^s}\| \le 0.25$.

1996 Dieci, Morini, and Papini [19] also use a Schur decomposition and take $m = 9$ and require $\|I - A^{1/2^s}\| \le 0.35$.

2001 Cheng, Higham, Kenney, and Laub [13] propose a transformation-free form of the inverse scaling and squaring method that takes as a parameter the desired accuracy. Square roots are computed by the product form of the Denman–Beavers iteration (6.4), with the Padé degree m chosen dynamically. Padé approximants are evaluated using a partial fraction representation [30].

2008 Higham [32, Sec. 11.5] develops two inverse scaling and squaring algorithms, one using the Schur decomposition and one transformation-free. Both choose the Padé degree and the number of square roots dynamically. Like all previous algorithms these are based on forward error bounds for the Padé error from [43].

2012 Al-Mohy and Higham [5] develop backward error bounds expressed in terms of the quantities $\mu_k = \|A^k\|_1^{1/k}$ and incorporate them into both Schur-based and transformation-free algorithms. They also use special techniques to compute the argument $A^{1/2^s} - I$ of the Padé approximant more accurately. This work puts the inverse scaling and squaring method on a par with the scaling and squaring algorithm of [2] for the matrix exponential. The Schur-based algorithm is implemented in NAG routine F01FJ (Mark 25).

2013 Al-Mohy, Higham, and Relton [6] develop a version of the algorithm of Al-Mohy and Higham [5] that works entirely in real arithmetic when the matrix is real, by exploiting the real Schur decomposition. The algorithm is implemented in NAG routine F01EJ (Mark 25).

6.9 Matrix iterations

For some matrix functions f that satisfy a nonlinear matrix equation it is possible to derive an iteration producing a sequence X_k of matrices that converges to $f(A)$ for a suitable choice of X_0.

The practical utility of iterations for matrix functions can be destroyed by instability due to growth of errors in floating point arithmetic, so we need an appropriate definition of stability that can be used to distinguish between "good" and "bad" iterations. Let $L^i(X)$ denote the ith power of a Fréchet derivative L at X, defined as i-fold composition; thus $L^3(X, E) \equiv L(X, L(X, L(X, E)))$. Consider an iteration $X_{k+1} = g(X_k)$ with a fixed point X. Assume that g is Fréchet differentiable at X. The iteration is defined to be *stable* in a neighborhood of X if the Fréchet derivative $L_g(X)$ has bounded powers, that is, there exists a constant c such that $\|L_g^i(X)\| \le c$ for all $i > 0$. For a stable iteration sufficiently small errors introduced near a fixed point have a bounded effect, to first order, on succeeding iterates. Note the useful standard result that a

linear operator on $\mathbb{C}^{n\times n}$ is power bounded if its spectral radius is less than 1 and not power bounded if its spectral radius exceeds 1.

Let $A \in \mathbb{C}^{n\times n}$ have no pure imaginary eigenvalues and have the Jordan canonical form

$$A = Z \begin{bmatrix} J_1 & 0 \\ 0 & J_2 \end{bmatrix} Z^{-1},$$

where $J_1 \in \mathbb{C}^{p\times p}$ and $J_2 \in \mathbb{C}^{q\times q}$ have spectra in the open left half-plane and right half-plane, respectively. The *matrix sign function* is defined by

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

It can be verified that the matrix sign function can also be expressed as

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

An iteration for computing the matrix sign function can be derived by applying Newton's method to $X^2 = I$:

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

To prove convergence, let $S = \operatorname{sign}(A)$ and $G = (A - S)(A + S)^{-1}$. It can be shown that

$$X_k = (I - G^{2^k})^{-1}(I + G^{2^k})S.$$

The eigenvalues of G are of the form $(\lambda_i - \operatorname{sign}(\lambda_i))/(\lambda_i + \operatorname{sign}(\lambda_i))$, where λ_i is an eigenvalue of A. Hence $\rho(G) < 1$ and $G^k \to 0$, implying that $X_k \to S$. It is easy to show that

$$\|X_{k+1} - S\| \le \frac{1}{2}\|X_k^{-1}\|\|X_k - S\|^2,$$

and hence the convergence is quadratic.

Consider a superlinearly convergent iteration $X_{k+1} = g(X_k)$ for $S = \operatorname{sign}(X_0)$. It can be shown that the Fréchet derivative $L_g(S, E) = \frac{1}{2}(E - SES)$, which does not depend on g. It follows that $L_g(S)$ is idempotent ($L_g^2(S) = L_g(S)$) and the iteration is stable. Hence essentially all sign iterations of practical interest are stable.

Applying Newton's method to $X^2 = A \in \mathbb{C}^{n\times n}$ yields, with X_0 given, the iteration

$$\left.\begin{aligned} \text{Solve } X_k E_k + E_k X_k &= A - X_k^2 \\ X_{k+1} &= X_k + E_k \end{aligned}\right\} \quad k = 0, 1, 2, \ldots.$$

If X_0 commutes with A and all the iterates are defined then this iteration simplifies to

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}A). \tag{6.3}$$

To analyze convergence, assume $X_0 = A$. Let $Z^{-1}AZ = J$ be a Jordan canonical form and set $Z^{-1}X_kZ = Y_k$. Then

$$Y_{k+1} = \frac{1}{2}(Y_k + Y_k^{-1}J), \qquad Y_0 = J.$$

The diagonal elements $d_i^{(k)} = (Y_k)_{ii}$ obey Heron's iteration

$$d_i^{(k+1)} = \frac{1}{2}(d_i^{(k)} + \lambda_i/d_i^{(k)}), \quad d_i^{(0)} = \lambda_i,$$

so $d_i^{(k)} \to \lambda_i^{1/2}$ assuming A has no eigenvalues on $\mathbb{R}^-$. It can also be shown that the off-diagonal elements converge and hence that $Y_k \to J^{1/2}$, or equivalently $X_k \to A^{1/2}$. However, this analysis does not generalize to general X_0 that do not commute with A.

A more general convergence result can be obtained by relating the iteration to the Newton iteration for the matrix sign function.

Theorem 6.1. *Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{R}^-$. The Newton square root iterates X_k with $X_0A = AX_0$ are related to the Newton sign iterates*

$$S_{k+1} = \frac{1}{2}(S_k + S_k^{-1}), \qquad S_0 = A^{-1/2}X_0$$

by $X_k \equiv A^{1/2}S_k$. Hence, provided $A^{-1/2}X_0$ has no pure imaginary eigenvalues the X_k are defined and $X_k \to A^{1/2}\mathrm{sign}(S_0)$ quadratically.

The Newton iteration (6.3) is unstable, as was pointed out by Laasonen [44], who stated that

> "Newton's method if carried out indefinitely, is not stable whenever the ratio of the largest to the smallest eigenvalue of A exceeds the value 9."

Higham [28] gives analysis that explains the instability for diagonalizable A. For general A, the instability can be analyzed using our definition. The iteration function is $g(X) = (X + X^{-1}A)/2$ and its Fréchet derivative is $L_g(X, E) = (E - X^{-1}EX^{-1}A)/2$. The relevant fixed point is $X = A^{1/2}$, for which $L_g(A^{1/2}, E) = (E - A^{-1/2}EA^{1/2})/2$. The eigenvalues of $L_g(A^{1/2})$ (i.e., the eigenvalues of $(I - A^{1/2^T} \otimes A^{-1/2})/2$, where $\otimes$ denotes the Kronecker product) are $(1 - \lambda_i^{-1/2}\lambda_j^{1/2})/2$. For stability we need $\max_{i,j} \frac{1}{2}\left|1 - \lambda_i^{-1/2}\lambda_j^{1/2}\right| < 1$. For Hermitian positive definite A this reduces to the condition $\kappa_2(A) < 9$ stated by Laasonen.

Fortunately, the instability of (6.3) is not intrinsic to the method, but depends on the equations used to compute X_k. The iteration can be rewritten in various ways as a stable coupled iteration, for example as the product form of the Denman–Beavers iteration [13],

$$M_{k+1} = \frac{1}{2}\left(I + \frac{M_k + M_k^{-1}}{2}\right), \qquad M_0 = A, \tag{6.4a}$$

$$X_{k+1} = \frac{1}{2}X_k(I + M_k^{-1}), \qquad X_0 = A, \tag{6.4b}$$

for which $X_k \to A^{1/2}$ and $M_k \to I$.

7 Methods for $f(A)b$

For $A \in \mathbb{C}^{n\times n}, b \in \mathbb{C}^n$, we now consider the problem of computing $f(A)b$ without first computing $f(A)$. Cases of interest include

- $f(x) = e^x$,
- $f(x) = \log(x)$,
- $f(x) = x^\alpha$ with arbitrary real α,
- $f(x) = \operatorname{sign}(x)$.

A minimal assumption is that matrix–vector products with A can be formed. It may also be possible to solve linear systems with A, by direct methods or iterative methods.

7.1 Krylov subspace method

A general purpose approach is to run the Arnoldi process on A and b to obtain the factorization

$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T,$$

where $Q_k = [q_1, q_2, \ldots, q_k]$ with $q_1 = b/\|b\|_2$ has orthonormal columns and H_k is $k \times k$ upper Hessenberg. Then we approximate

$$f(A)b \approx Q_k f(H_k) Q_k^* b = \|b\|_2 Q_k f(H_k) e_1.$$

Typically, k will be chosen relatively small in order to economize on storage ($k < 100$, say). Hence any method for dense matrices can be used for $f(H_k)e_1$. Care must be taken to guard against loss of orthogonality of Q_k in the Arnoldi process.

Among the large literature on Krylov methods for matrix functions we cite just the recent survey by Güttel [25].

7.2 $f(A)b$ via contour integration

For general f, we can use the Cauchy integral formula

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

where f is analytic on and inside a closed contour Γ that encloses the spectrum of A. Assume that we can solve linear systems with A, preferably by a direct method.

We can take for the contour Γ a circle enclosing the spectrum, for example with centre $(\lambda_1 + \lambda_n)/2$ and radius $\lambda_1/2$ for a symmetric positive definite matrix with spectrum in $[\lambda_n, \lambda_1]$. Then we can apply the repeated trapezium rule. However, this is inefficient unless A is very well conditioned.

Hale, Higham and Trefethen [26] use a conformal mapping, carefully constructed based on knowledge of the extreme points of the spectrum and any branch cuts or singularities of f, and then apply the repeated trapezium rule. For $A^{1/2}b$ and A with real spectrum in $[\lambda_n, \lambda_1]$ they conformally map $\mathbb{C} \setminus \{(-\infty, 0] \cup [\lambda_n, \lambda_1]\}$ to an annulus: $[\lambda_n, \lambda_1]$ is mapped to the inner circle and $(-\infty, 0]$ to the outer circle. Compared with taking a circle as described above, the conformal mapping approach reduces the number of quadrature points from 32,000 to 5 when two digits of accuracy are required or 262,000 to 35 for 13 digits.

7.3 $A^\alpha b$ via binomial expansion

Write $A = s(I - C)$, where we wish to choose s so that $\rho(C) < 1$. This is certainly possible if A is an M-matrix. If A has real, positive eigenvalues then $s = (\lambda_{\min} + \lambda_{\max})/2$ minimizes the spectral radius $\rho(C)$ with

$$\rho_{\min}(C) = (\lambda_{\min} - \lambda_{\max})/(\lambda_{\min} + \lambda_{\max}).$$

For any A, the value $s = \mathrm{trace}(A^*A)/\mathrm{trace}(A^*)$ minimizes $\|C\|_F$ but may or may not achieve $\rho(C) < 1$. From

$$(I - C)^\alpha = \sum_{j=0}^{\infty} \binom{\alpha}{j} (-C)^j, \qquad \rho(C) < 1,$$

we have

$$A^\alpha b = s^\alpha \sum_{j=0}^{\infty} \binom{\alpha}{j} (-C)^j b,$$

and this series can be truncated to approximate $A^\alpha b$.

7.4 $e^A b$

One of Moler and Van Loan's "nineteen dubious ways" to compute the matrix exponential [48] applies a fourth order Runge–Kutta method with fixed step size to the ODE $y' = Ay$. This produces an approximation $e^A \approx T_4(A/m)^m$, where $T_4(x)$ is a degree 4 truncation of the Taylor series for e^x. Al-Mohy and Higham [4] develop an algorithm that can be thought of as an extension of this idea to use degree s truncated Taylor series polynomials, where the degree s and the scaling parameter m are chosen to minimize the cost while ensuring a backward error of order the unit roundoff, u. It is interesting to compare this algorithm with a Krylov method (in general, Arnoldi-based):

Al-Mohy and Higham algorithm	Krylov method
Most time spent in matrix–vector products.	Cost of Krylov recurrence and computing e^H can be significant.
A "direct method", so its cost is predictable.	An iterative method; needs a stopping test.
No parameters to estimate.	Must select maximum size of Krylov subspace.
Storage: 2 vectors	Storage: Krylov basis
Can evaluate e^{At} at multiple points on the interval.	Degree of Krylov subspace will depend on t.
Works directly for $e^A B$ with a matrix B.	Need a block Krylov method for $e^A B$.
Cost tends to increase with $\|A\|$.	$\|A\|^{1/2}$-dependence of cost for symmetric negative definite A.

8 Concluding remarks

We note that this treatment is not comprehensive. For example, we have given few details about methods for the matrix exponential and matrix logarithm and not described iterative methods for pth roots [24], methods for arbitrary matrix powers [37], [38], or methods for computing Fréchet derivatives [1], [3], [6], [38].

We have also said very little about software. A catalogue of available software is given by Deadman and Higham [17].

We finish by emphasizing the importance of considering the underlying assumptions and requirements when selecting a method, as discussed in section 5. These include the desired accuracy; whether the matrix A is known explicitly or is available only in the form of a black box that returns Ax, and possibly A^*x, given x; and whether it is possible to solve $Ax = b$ by a (sparse) direct method.

References

[1] Awad H. Al-Mohy and Nicholas J. Higham. Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation. *SIAM J. Matrix Anal. Appl.*, 30(4):1639–1657, 2009.

[2] Awad H. Al-Mohy and Nicholas J. Higham. A new scaling and squaring algorithm for the matrix exponential. *SIAM J. Matrix Anal. Appl.*, 31(3):970–989, 2009.

[3] Awad H. Al-Mohy and Nicholas J. Higham. The complex step approximation to the Fréchet derivative of a matrix function. *Numer. Algorithms*, 53(1):133–148, 2010.

[4] Awad H. Al-Mohy and Nicholas J. Higham. Computing the action of the matrix exponential, with an application to exponential integrators. *SIAM J. Sci. Comput.*, 33(2):488–511, 2011.

[5] Awad H. Al-Mohy and Nicholas J. Higham. Improved inverse scaling and squaring algorithms for the matrix logarithm. *SIAM J. Sci. Comput.*, 34(4):C153–C169, 2012.

[6] Awad H. Al-Mohy, Nicholas J. Higham, and Samuel D. Relton. Computing the Fréchet derivative of the matrix logarithm and estimating the condition number. *SIAM J. Sci. Comput.*, 35(4):C394–C410, 2013.

[7] Mary Aprahamian and Nicholas J. Higham. The matrix unwinding function, with an application to computing the matrix exponential. MIMS EPrint 2013.21, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, May 2013. 22 pp. Revised October 2013. To appear in SIAM J. Matrix Anal. Appl.

[8] Zhaojun Bai and James W. Demmel. On swapping diagonal blocks in real Schur form. *Linear Algebra Appl.*, 186:73–95, 1993.

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

[10] A. Buchheim. An extension of a theorem of Professor Sylvester's relating to matrices. *Phil. Mag.*, 22(135):173–174, 1886. Fifth series.

[11] Arthur Cayley. A memoir on the theory of matrices. *Philos. Trans. Roy. Soc. London*, 148:17–37, 1858.

[12] Jie Chen, Mihai Anitescu, and Yousef Saad. Computing $f(A)b$ via least squares polynomial approximations. *SIAM J. Sci. Comput.*, 33(1):195–222, 2011.

[13] 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.

[14] M. Cipolla. Sulle matrice espressione analitiche di un'altra. *Rendiconti Circolo Matematico de Palermo*, 56:144–154, 1932.

[15] A. R. Collar. The first fifty years of aeroelasticity. *Aerospace (Royal Aeronautical Society Journal)*, 5:12–20, 1978.

[16] 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.

[17] Edvin Deadman and Nicholas J. Higham. A catalogue of software for matrix functions. MIMS Eprint, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2013. In preparation.

[18] Edvin Deadman, Nicholas J. Higham, and Rui Ralha. Blocked Schur algorithms for computing the matrix square root. In *Applied Parallel and Scientific Computing: 11th International Conference, PARA 2012, Helsinki, Finland*, P. Manninen and P. Öster, editors, volume 7782 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 2013, pages 171–182.

[19] Luca Dieci, Benedetta Morini, and Alessandra Papini. Computational techniques for real logarithms of matrices. *SIAM J. Matrix Anal. Appl.*, 17(3):570–593, 1996.

[20] Ernesto Estrada and Desmond J. Higham. Network properties revealed through matrix functions. *SIAM Rev.*, 52(4):696–714, 2010.

[21] R. A. Frazer, W. J. Duncan, and A. R. Collar. *Elementary Matrices and Some Applications to Dynamics and Differential Equations*. Cambridge University Press, 1938. xviii+416 pp. 1963 printing.

[22] G. Frobenius. Über die cogredienten Transformationen der bilinearen Formen. *Sitzungsber K. Preuss. Akad. Wiss. Berlin*, 16:7–16, 1896.

[23] G. Giorgi. Nuove osservazioni sulle funzioni delle matrici. *Atti Accad. Lincei Rend.*, 6(8):3–8, 1928.

[24] Chun-Hua Guo and Nicholas J. Higham. A Schur–Newton method for the matrix pth root and its inverse. *SIAM J. Matrix Anal. Appl.*, 28(3):788–804, 2006.

[25] Stefan Güttel. Rational Krylov approximation of matrix functions: Numerical methods and optimal pole selection. *GAMM-Mitteilungen*, 36(1):8–31, 2013.

[26] Nicholas Hale, Nicholas J. Higham, and Lloyd N. Trefethen. Computing A^α, $\log(A)$, and related matrix functions by contour integrals. *SIAM J. Numer. Anal.*, 46(5):2505–2523, 2008.

[27] W. F. Harris. The average eye. *Opthal. Physiol. Opt.*, 24:580–585, 2005.

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

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

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

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

[32] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. xx+425 pp. ISBN 978-0-898716-46-7.

[33] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Rev.*, 51(4):747–764, 2009.

[34] Nicholas J. Higham. Arthur Buchheim (1859-1888). `http://nickhigham.wordpress.com/2013/01/31/arthur-buchheim/`, 2013.

[35] Nicholas J. Higham. Gene Golub SIAM Summer School 2013. `http://nickhigham.wordpress.com/2013/08/09/gene-golub-siam-summer-school-2013/`, 2013.

[36] Nicholas J. Higham and Awad H. Al-Mohy. Computing matrix functions. *Acta Numerica*, 19:159–208, 2010.

[37] Nicholas J. Higham and Lijing Lin. A Schur–Padé algorithm for fractional powers of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(3):1056–1078, 2011.

[38] Nicholas J. Higham and Lijing Lin. An improved Schur–Padé algorithm for fractional powers of a matrix and their Fréchet derivatives. *SIAM J. Matrix Anal. Appl.*, 34(3):1341–1360, 2013.

[39] Nicholas J. Higham and Françoise Tisseur. A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra. *SIAM J. Matrix Anal. Appl.*, 21(4):1185–1201, 2000.

[40] Marlis Hochbruck and Alexander Ostermann. Exponential integrators. *Acta Numerica*, 19:209–286, 2010.

[41] Leo Katz. A new status index derived from sociometric analysis. *Psychometrika*, 18(1):39–43, 1953.

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

[43] Charles S. Kenney and Alan J. Laub. Padé error estimates for the logarithm of a matrix. *Internat. J. Control*, 50(3):707–730, 1989.

[44] Pentti Laasonen. On the iterative solution of the matrix equation $AX^2 - I = 0$. *M.T.A.C.*, 12:109–116, 1958.

[45] Edmond Nicolas Laguerre. Le calcul des systèmes linéaires, extrait d'une lettre adressé à M. Hermite. In *Oeuvres de Laguerre*, Ch. Hermite, H. Poincaré, and E. Rouché, editors, volume 1, Gauthier–Villars, Paris, 1898, pages 221–267. The article is dated 1867 and is "Extrait du Journal de l'École Polytechnique, LXII$^{\text{e}}$ Cahier".

[46] J. Douglas Lawson. Generalized Runge-Kutta processes for stable systems with large Lipschitz constants. *SIAM J. Numer. Anal.*, 4 (3):372–380, 1967.

[47] Borislav V. Minchev and Will M. Wright. A review of exponential integrators for first order semi-linear problems. Preprint 2/2005, Norwegian University of Science and Technology, Trondheim, Norway, 2005. 44 pp.

[48] 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.

[49] 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.

[50] Beresford N. Parlett. Computation of functions of triangular matrices. Memorandum ERL-M481, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, November 1974. 18 pp.

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

[52] 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.

[53] G. Peano. Intégration par Séries des équations différentielles linéaires. *Math. Annalen*, 32:450–456, 1888.

[54] H. Poincaré. Sur les groupes continus. *Trans. Cambridge Phil. Soc.*, 18:220–255, 1899.

[55] R. F. Rinehart. The equivalence of definitions of a matric function. *Amer. Math. Monthly*, 62:395–414, 1955.

[56] Hans Schwerdtfeger. *Les Fonctions de Matrices. I. Les Fonctions Univalentes*. Number 649 in *Actualités Scientifiques et Industrielles*. Hermann, Paris, France, 1938. 58 pp.

[57] Matthew I. Smith. A Schur algorithm for computing matrix pth roots. *SIAM J. Matrix Anal. Appl.*, 24(4):971–989, 2003.

[58] J. J. Sylvester. Additions to the articles, "On a New Class of Theorems," and "On Pascal's Theorem". *Philosophical Magazine*, 37: 363–370, 1850. Reprinted in [60, pp. 1451–151].

[59] J. J. Sylvester. On the equation to the secular inequalities in the planetary theory. *Philosophical Magazine*, 16:267–269, 1883. Reprinted in [61, pp. 110–111].

[60] *The Collected Mathematical Papers of James Joseph Sylvester*, volume 1 (1837–1853). Cambridge University Press, 1904. xii+650 pp.

[61] *The Collected Mathematical Papers of James Joseph Sylvester*, volume IV (1882–1897). Chelsea, New York, 1973. xxxvii+756 pp. ISBN 0-8284-0253-1.

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