Gene Golub SIAM Summer School 2013
Function of Matrices

2013 Summer

Taught by Prof. Nicholas J. Higham

Notes taken by Zhengbo Zhou

May 20, 2023

Contents

1 History and Definitions

The matrix algebra arised by Cayley and Sylvester.

- Matrix algebra developed by Arthur Cayley, FRS (1821–1895) in his paper "Memoir on the Theory of Matrix" (1858).
- Cayley considered matrix square root is his 1858 memoir.
- The term "matrix" coined in 1850 by James Joseph Sylvester, FRS (1814–1897).
- Leguerre (1867) defines the matrix exponential via power series.

We can define the matrix function by substitution. Suppose we want to define $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$, but not elementwise. Given $f(t)$, we can define $f(A)$ by substituting A for t:

$$\begin{aligned} & f(t) = \frac{1+t^2}{1-t} \quad \Rightarrow \quad f(A) = (I-A)^{-1}(I+A^2), \\ & \log(1+x) = x - x^2/2 + x^3/3 - x^4/4 + \cdots, \quad |\,x\,| < 1 \\ \Rightarrow \quad & \log(I+A) = A - A^2/2 + A^3/3 - A^4/4 + \cdots, \quad \rho(A) < 1, \end{aligned}$$

where $\rho(A)$ is the spectral radius of A ($\rho(A) = \max\{|\,\lambda\,| : Ax = \lambda x, \forall x \in \mathbb{C}^n\}$). However, we want a more general definition that works for arbitrary f and arbitrary A.

1.1 Definition via Jordan canonical form

We can reduce any $A \in \mathbb{C}^{n\times n}$ into Jordan canonical form (JCF):

$$A = ZJZ^{-1}, \quad J = \operatorname{diag}(J_1, \ldots, J_p), \quad J_k = s \begin{bmatrix} \lambda_k & 1 & & \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k \times m_k}, \tag{1}$$

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

- $\lambda_1, \ldots, \lambda_s$ the distinct eigenvalues of A,
- n_i is 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

$$f^{(j)}(\lambda_i), \quad j = 0, \ldots, n_i - 1, \quad i = 1, \ldots, s,$$

exist.

▶ **Definition 1.1** (Define via Jordan Canonical Form)**.** Let f be defined on the spectrum of $A \in \mathbb{C}^{n\times n}$ and let A have the JCF (1). Then

$$f(A) \coloneqq Zf(J)Z^{-1}$$

where $f(J) = \operatorname{diag}(f(J_1), \ldots, f(J_p))$ and

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

It is obvious that we require some differentiability of f and it depends on the Jordan structure. If A is diagonalizable, there is no requirement on the differentiability of f. Conversely, if A has a large Jordan block, then lots of derivatives are required. Also, from the definition, we see that we actually don't need the underlying function f to be smooth at all, since we only need the function value at λ_i.

1.1.1 "Deriving" the formula for $f(J_k)$

Write $J_k = \lambda_k I + E_k \in \mathbb{C}^{m_k \times m_k}$. For $m_k = 3$, we have

$$E_k = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, E_k^2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, E_k^3 = 0.$$

Assuming f has Taylor expansion, then

$$f(t) = f(\lambda_k) + f'(\lambda_k)(t - \lambda_k) + \cdots + \frac{f^{(j)}(\lambda_k)(t - \lambda_k)^j}{j!} + \cdots,$$

then

$$f(J_k) = f(\lambda_k)I + f'(\lambda_k)E_k + \cdots + \frac{f^{(m_k-1)}(\lambda_k)E_k^{m_k-1}}{(m_k - 1)!},$$

which is exactly the expression of $f(J_k)$.

1.2 Definition via Interpolation

For $A \in \mathbb{C}^{n\times n}$, the following definition is studied by Sylvester (1886, distinct eigenvalues) and Buckheim (1886, general eigenvalues).

▶ **Definition 1.2** (Define via Hermite Interpolation)**.** Given $A \in \mathbb{C}^{n\times n}$ which has s distinct eigenvalues $\lambda_1, \ldots, \lambda_s$ and n_i is the index of λ_i. Then $f(A) = p(A)$ where p is the unique Hermite interpolating polynomial of degree less than $\sum_{i=1}^{s} n_i$ satisfying: for each eigenvalue λ_k,

$$p^{(j)}(\lambda_k) = f^{(j)}(\lambda_k), \quad j = 0, \ldots, n_k - 1.$$

▶ **Example 1.3.** Let $f(t) = t^{1/2}$, $A = \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix}$ and $\lambda(A) = \{1, 4\}$. Then taking the positive roots, we have

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

which gives

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

Note. This definition gives a unique polynomial to define $f(A)$. Suppose we impose further interpolation conditions which may have nothing todo with the eigenvalues, then we gain a higher degree polynomial but it still interpolates the eigenvalues and it still gives the same result.

For example, suppose we interpolates not only $f(1) = 1$ and $f(4) = 2$, but also $f(-1) = 5$. Then we have

$$r(t) = \frac{(t-4)(t+1)}{-6} + \frac{2(t-1)(t+1)}{15} + \frac{(t-1)(t-4)}{2},$$

and we can define

$$r(A) = \frac{(A-4I)(A+I)}{-6} + \frac{2(A-I)(A+I)}{15} + \frac{(A-I)(A-4I)}{2} = \frac{1}{3}\begin{bmatrix} 4 & 2 \\ 1 & 5 \end{bmatrix}$$

which is exactly the same as using only 2 interpolating data. Further interpolating conditions wouldn't change the resul, this can be useful if we don't know the Jordan form. For example, we can look for the worst case Jordan form and get the interpolating polynomial using lots of data points where some of them my not necessary, but we will eventually get the same result.

The following are properties of matrix function that can be seen by using the definition by interpolation.

- $f(A)$ is a polynomial in A, but the polynomial depends on A.
- $f(A)$ commutes with A, i.e. $Af(A) = f(A)A$.
- $f(A^T) = (f(A))^T$. However, $f(A^*) \neq (f(A))^*$ in general.

The usual Cayley-Hamilton Theorem states

$$p(t) = \det(tI - A) \text{ implies } p(A) = 0.$$

A generalized version of the Cayley-Hamilton theorem is

▶ **Theorem 1.4** (Cayley, 1857)**.** *If $A, B \in \mathbb{C}^{n\times n}$, $AB = BA$, and $f(x, y) = \det(xA - yB)$, then $f(B, A) = 0$.*

Using the Cayley-Hamilton theorem, 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, any power series collapses to a polynomial. For example, $\mathrm{e}^A = \sum_{k=0}^{\infty} A^k/k! = \sum_{k=0}^{n-1} d_k A^k$ where d_k may depends on A.

1.3 Definition via Cauchy Integral Formula

▶ **Definition 1.5** (Define via Cauchy Integral Formula)**.** For $A \in \mathbb{C}^{n\times n}$,

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

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

This definition is useful computationally. Suppose we would like to compute $f(A)b$ for A is large and sparse, then

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

As long as we are able to solve $(zI - A)^{-1}b$ which is a linear system with shifted A, then we are able to do quadrature here.

1.4 Defintion via Schwerdtfeger's formula

▶ **Definition 1.6** (Schwerdtfeger, 1938)**.** For A with distinct eigenvalues $\lambda_1, \ldots, \lambda_s$ with indices n_i,

$$f(A) = \sum_{i=1}^{s} A_i \sum_{j=0}^{n_i-1} \frac{f^{(j)}(\lambda_i)}{j!} (A - \lambda_i I)^j = \sum_{i=1}^{s} \sum_{j=0}^{n_i-1} f^{(j)}(\lambda_i) Z_{ij},$$

where A_i are Frobenius covariants and Z_{ij} depends on A but not f.

1.5 Equivalence of Definition

▶ **Theorem 1.7.** *The four definitions are equivalent, modulo analyticity assumption for the Cauchy definition.*

▶ **Remark.**

- Interpolation: For basic properties such as A and $f(A)$ commute.
- JCF: Solving matrix equations, e.g. $X^2 = A, \mathrm{e}^X = A$ and $W\mathrm{e}^W = A$ where W is the Lambert W function.
- For computation, there exists methods that specific to particular f and A.

1.6 Primary and Nonprimary Functions

1.6.1 Root Oddities

Notice that identity matrix has strange roots

- $B_n^2 = I_n$ where

$$B_4 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & -2 & -3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

This arises in backward differentiation formula solvers for ODEs.

- Turnbull (1927): $A_n^3 = I_n$, where

$$A_4 = \begin{bmatrix} -1 & 1 & -1 & 1 \\ -3 & 2 & -1 & 0 \\ -3 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}.$$

- $C_n^2 = I_n$, where

$$C_n = 2^{-3/2} \begin{bmatrix} 1 & 3 & 3 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -3 & 3 & -1 \end{bmatrix}.$$

1.6.2 Nonprimary Matrix Functions

Consider the Jordan form definition 1.1. However, for example $f(t) = t^{1/2}$, we could take different square root (positive and negative square root) for the same eigenvalues (but different Jordan blocks). Formally, if A is derogatory, and a different branch of f is taken in the two different Jordan blocks for λ, we obtain a nonprimary matrix function for A. For example,

$$\begin{aligned} I_2 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^2 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}^2, \quad \text{primary} \\ &= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}^2 = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ \sin(\theta) & -\cos(\theta) \end{bmatrix}^2, \quad \text{non-primary.} \end{aligned}$$

Primary matrix functions are expressible as a polynomial in A, while nonprimary ones are not. Therefore the above three examples are all nonprimary matrix square roots of I_2.

Sometimes, the nonprimary function are interested. For example, we would like to find the square root of rotations $G(\theta)$ where

$$G(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}.$$

Apparently, $G(\theta/2)$ is a natural square root of $G(\theta)$. However, for $\theta = \pi$,

$$G(\pi) = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad G(\pi/2) = \begin{bmatrix} 0 & 1 \\ 0 & -1 \end{bmatrix},$$

which implies $G(\pi/2)$ is a nonprimary square root of $G(\pi)$.

Sometimes, the family of all nonprimary matrix functions can be written explicitly. For example, we would like to find a matrix X such that

$$X^2 = A = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

A solution is

$$X = \begin{bmatrix} 1 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

In fact, all square roots of A are given by

$$Y = \pm U \begin{bmatrix} 1 & 1/2 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} U^{-1}, \quad U = \begin{bmatrix} a & b & d \\ 0 & a & 0 \\ 0 & e & c \end{bmatrix}$$

where a, b, c, d and e are any numbers that make U nonsingular.

1.7 Principal Logarithm, root and power

Notice that suppose X is a solution of $\mathrm{e}^X = A$, then $X + 2\alpha\pi\mathrm{i}I$ for all $\alpha \in \mathbb{Z}$ is a solution of $\mathrm{e}^X = A$. Therefore, we need to make a choice on the solutions.

Let $A \in \mathbb{C}^{n\times n}$ have no eigenvalues on $\mathbb{R}_-$,

▶ **Definition 1.8** (Principal Log)**.** $X = \log(A)$ denotes the unique X such that

- $e^X = A$,
- $-\pi < \mathrm{Im}(\lambda(X)) < \pi$.

▶ **Definition 1.9** (Principal pth root)**.** For integer $p > 0$, $X = A^{1/p}$ is the unique X such that

- $X^p = A$,
- $-\pi/p < \arg(\lambda(X)) < \pi/p$.

▶ **Definition 1.10** (Principal power)**.** For $s \in \mathbb{R}$, $A^s = \mathrm{e}^{s\log A}$, where $\log A$ is the principal logarithm. It also have the integral representation:

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

2 Application

2.1 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. For the second order differential equation,

$$\frac{\mathrm{d}^2 y}{\mathrm{d}t^2} + Ay = 0, \quad y(0) = y_0, \quad y'(0) = y_0'$$

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 convert to a first order system and solved using the exponential:

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

MATLAB has several preinstalled functions that can take matrix as input, e.g. `funm`, `expm`, `logm` and `sqrtm`.

2.2 Linear Constant Coefficient ODE

In nuclear magnetic resonance (NMR) spectroscopy Solomon equations

$$\frac{\mathrm{d}M}{\mathrm{d}t} = -RM, \quad M(0) = I$$

where $M(t)$ is a matrix of intensities and R is a symmetric relaxation matrix. Thus $M(t) = \mathrm{e}^{-Rt}$.

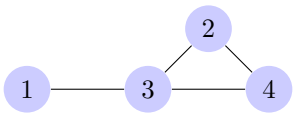
Burnup calculations in nuclear reactor analysis involve

$$\frac{\mathrm{d}X}{\mathrm{d}t} = AX, \quad X(0) = X_0,$$

where A is often upper triangular.

2.3 Application to Complex Networks

Adjacency matrix of the following undirected network:



gives

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

▶ **Definition 2.1** (Network measures)**.**

- Centrality: $(\mathrm{e}^A)_{ii}$ measures how important node i is,
- Communicability: $(\mathrm{e}^A)_{ij}$ measures how well information is transferred between nodes i and j.

▶ **Remark.**

- We can use the resolvent $(I - \alpha A)^{-1}$ in place of e^A.
- $\mathrm{trace}(\cosh(A))/\,\mathrm{trace}(\mathrm{e}^A)$ is a measure of how close a graph is to bipartite.

2.4 The Average Eye

First order character of optical systtem characterized by *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:

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

The usual average $m^{-1}\sum_{i=1}^m T_i$ is not a transference matrix. Harris(2005) propose the average

$$\exp\left(m^{-1}\sum_{i=1}^m \log(T_i)\right)$$

which is always a transference matrix.

2.5 Random Multivariate Samples in Statistics

When we sample from $N(\mu, C)$, we normally sample $x \in N(0, I)$, then compute the Cholesky factorization of $C = LL^T$, then $y = \mu + Lx \sim N(\mu, C)$, $C \in \mathbb{R}^{m\times m}$. In some applications, C has dimension greater than 10^{12} and computing the Cholesky factor is impractical. Chen, Anitescu and Saad propose that $y = \mu + C^{1/2}x \sim N(\mu, C)$. This is practical since we only need to compute $C^{1/2}x$ instead of compute $C^{1/2}$ explicitly, and $C^{1/2}x$ can be computed via such as Krylov method.

3 Properties

We present some basic properties of matrix functions.

(P_1) $f(XAX^{-1}) = Xf(A)X^{-1}$. This is immediate from the Jordan form definition, since similar matrices can be taken to have the same Jordan form. This is also immediate from $f(A)$ is a polynomial of A, then $f(XAX^{-1}) = Xf(A)X^{-1}$ comes from $(XAX^{-1})^k = XA^kX^{-1}$.

(P_2) Eigenvalues of $f(A)$ are $f(\lambda_i)$, where the λ_i are the eigenvalues of A. Notice that A and $f(A)$ are not necessarily have the same Jordan form. For example, define the sign function

$$\operatorname{sign}(z) = \begin{cases} 1 & \operatorname{Re}(z) \geq 0, \\ -1 & \operatorname{Re}(z) \leq 0, \end{cases}$$

then $\operatorname{sign}(A) = Z \operatorname{diag}(\operatorname{sign}(\lambda_i)) Z^{-1}$ where the large Jordan block collapses into several 1×1 Jordan block.

(P_3) If $A = (A_{ij})$ is a block triangular, then $F = f(A)$ is block triangular with the same block structure as A, and $F_{ii} = f(A_{ii})$.

(P_4) If $D = \operatorname{diag}(D_{ii})$, then $f(D) = \operatorname{diag}(f(D_{ii}))$.

(P_5) If $h = f + g$, then $h(A) = f(A) + g(A)$; if $h(t) = f(g(t))$, then $h(A) = f(g(A))$.

(P_6) Polynomial functional relations generalize from the scalar cases. If

$$G(f_1, \ldots, f_m) = 0,$$

where G is a polynomial, then $G(f_1(A), \ldots, f_m(A)) = 0$, such as

$$\begin{aligned} &\sin^2(A) + \cos^2(A) = I, \\ &(A^{1/p})^p = A \quad \text{for any integer } p > 0, \\ &\mathrm{e}^{\mathrm{i}A} = \cos(A) + \mathrm{i}\sin(A). \end{aligned}$$

(P_7) Some relations can fail:

- $f(A^*) \neq (f(A))^*$ in general,
- $\mathrm{e}^{\log(A)} = A$ but $\log(\mathrm{e}^A) \neq A$ in general since it requires A's eigenvalues lie between $\pm\pi$.
- $(AB)^{1/2} \neq A^{1/2} B^{1/2}$ in general.
- $\mathrm{e}^A \neq (\mathrm{e}^{A/\alpha})^\alpha$ in general. This is true for $\alpha \in \mathbb{R}_+$.
- $\mathrm{e}^{(A+B)t} = \mathrm{e}^{At}\mathrm{e}^{Bt}$ for all t if and only if $AB = BA$.

3.1 Function of Triangular Matrices

▶ **Example 3.1** (Function of 2×2 triangular matrix)**.**

$$f\left(\begin{bmatrix} \lambda_1 & t_{12} \\ & \lambda_2 \end{bmatrix}\right) = \begin{bmatrix} f(\lambda_1) & t_{12} f[\lambda_1, \lambda_2] \\ 0 & f(\lambda_2) \end{bmatrix},$$

where

$$f[\lambda_1, \lambda_2] = \begin{cases} \dfrac{f(\lambda_2) - f(\lambda_1)}{\lambda_2 - \lambda_1}, & \lambda_1 \neq \lambda_2 \\ f'(\lambda_1), & \lambda_1 = \lambda_2. \end{cases}$$

Proof. Using the fact that $T = \begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix}$ commute with $F = f(T) = \begin{bmatrix} f(\lambda_1) & F_{12} \\ 0 & f(\lambda_2) \end{bmatrix}$ (this structure is by property P_2 and P_3), we have

$$\begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} f(\lambda_1) & F_{12} \\ 0 & f(\lambda_2) \end{bmatrix} = \begin{bmatrix} f(\lambda_1) & F_{12} \\ 0 & f(\lambda_2) \end{bmatrix} \begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix}.$$

By looking at the $(1, 2)$ entry, we have

$$\lambda_1 F_{12} + t_{12} f(\lambda_2) = f(\lambda_1) t_{12} + F_{12} \lambda_2$$

gives

$$F_{12} = \frac{f(\lambda_1) - f(\lambda_2)}{\lambda_1 - \lambda_2} t_{12},$$

which proves the claim. □

▶ **Theorem 3.2** (Davis, 1973; Descloux, 1963; Van Loan, 1975)**.** *If T is upper triangular, so is $F = f(T)$ and $f_{ii} = f(t_{ii})$,*

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

where $\lambda_i = t_{ii}$.

- *S_{ij} is the set of all strictly increasing sequences of integers starting at i and ending at j. For example, $S_{14} = \{(1,4),(1,2,4),(1,3,4),(1,2,3,4)\}$, and*
- *$f[\lambda_{s_0},\dots,\lambda_{s_k}]$ is the kth order divided difference function.*

Problem of this formula:

1. Complexity of evaluating via computer is increasing exponentially,
2. hard/tricky to implement the divided difference function.

3.2 Diagonalizable Matrices

How to prove $\sin^2(A) + \cos^2(A) = I$?

▶ **Theorem 3.3.** *Let $\mathcal{D}$ be an open subset of $\mathbb{R}$ or $\mathbb{C}$ and let f be $n-1$ times continuously differentiable on $\mathcal{D}$. Then $f(A) = 0$ for all $A \in \mathbb{C}^{n\times n}$ with spectrum in $\mathcal{D}$ if and only if $f(A) = 0$ for all diagonalizable matrix $A \in \mathbb{C}^{n\times n}$ with spectrum in $\mathcal{D}$.*

Using this theorem, $\sin^2(A) + \cos^2(A) = I$ is trivial. And it can be used to prove the following theorem:

▶ **Theorem 3.4.** *For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$,*

$$\log(A) = \int_0^1 (A-I)\left(t(A-I)+I\right)^{-1}\,\mathrm{d}t. \tag{2}$$

4 Fréchet Derivative and Condition Number

▶ **Definition 4.1** (Fréchet Derivative)**.** The Fréchet Derivative of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at a point $X \in \mathbb{C}^{n\times n}$ is a linear mapping $L : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ such that for all $E \in \mathbb{C}^{n\times n}$, where E is an arbitrary perturbation of X,

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

Fréchet Derivative is linear in the second argument by definition.

▶ **Example 4.2.** For $f(X) = X^2$, we have

$$f(X+E) - f(X) = EX + XE + E^2,$$

so $L(X,E) = XE + EX$.

▶ **Example 4.3** (Fréchet Derivative of e^A)**.**

$$L(A,E) = \int_0^1 \mathrm{e}^{A(1-s)} E \mathrm{e}^{As}\,\mathrm{d}s.$$

This can be simplified to $L(A,E) = E\mathrm{e}^A = \mathrm{e}^A E$ when $AE = EA$. Another representation will be

$$L(A,E) = E + \frac{AE+EA}{2!} + \frac{A^2E + AEA + EA^2}{3!} + \cdots.$$

4.1 Condition Number

For E as a perturbation of A, we have

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

measures the maximum changes in f subject to a small change in A in relative sense.

▶ **Lemma 4.4.**

$$\operatorname{cond}(f, A) = \frac{\| L(A) \| \| A \|}{\| f(A) \|},$$

where

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

▶ **Example 4.5** (Condition number of e^A)**.**

$$\kappa_{\exp}(A) = \frac{\| L(A) \| \| A \|}{\| \mathrm{e}^A \|}.$$

Using $\| L(A) \| \geq \| L(A, I) \| = \| \mathrm{e}^A \|$, we have $\kappa_{\exp}(A) \geq \| A \|$.

▶ **Theorem 4.6.**

- *For normal $A \in \mathbb{C}^{n \times n}$, $\kappa_{\exp}(A) = \| A \|_2$.*
- *If $A \in \mathbb{R}^{n \times n}$ is a nonnegative scalar multiple of a stochastic matrix, then in the ∞-norm, $\kappa_{\exp}(A) = \| A \|_\infty$.*

4.2 Computing L_f

We can compute L_f via $2n \times 2n$ matrix: Consider the matrix

$$\begin{bmatrix} A & E \\ 0 & A \end{bmatrix} \in \mathbb{C}^{2n \times 2n},$$

then

$$f\left(\begin{bmatrix} A & E \\ 0 & A \end{bmatrix} \right) = \begin{bmatrix} f(A) & L_f(A, E) \\ 0 & f(A) \end{bmatrix}.$$

Note that $L_f(A, \alpha E) = \alpha L_f(A, E)$, but α may affect algorithm used for the evaluation, since we can let αE be arbitrarily smaller than A which could lead to inaccuracy.

Instead, we can compute the L_f via finite difference method,

$$f'(x) = \frac{f(x+h) - f(x)}{h}.$$

Notice that, we need a little bit of care when choosing h, the accuracy will increase when $h \to 0$, but it will eventually loses its accuracy due to cancellation error (roundoff error).

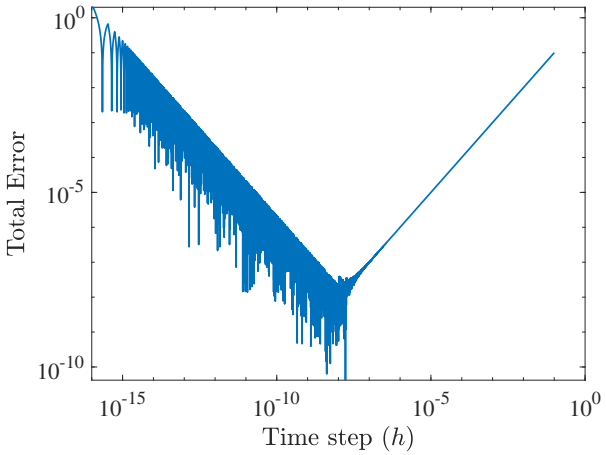


FIG. 1. *The absolute error of $f'(1)$ using finite difference method, where $f(x) = x^2$. (See Appendix A.1)*

We can improve from this method by the following: assume $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ and $A, E \in \mathbb{R}^{n\times n}$. Then

$$f(A + \mathrm{i}hE) - f(A) - \mathrm{i}hL_f(A, E) = o(h).$$

Thus (AI-Moly, Higham, 2010)

$$f(A) \approx \mathrm{Re}\, f(A + \mathrm{i}hE), \qquad L_f(A, E) \approx= \mathrm{Im}\left(\frac{f(A + \mathrm{i}hE)}{h}\right).$$

This can be explained via Taylor expansion,

$$f(X + \mathrm{i}hE) = f(X) + \mathrm{i}hL_f(A, E) - \frac{h^2}{2!}L_f^{(2)}(A, E) + \frac{ih^3}{3!}L_f^{(3)}(A, E) - \cdots$$

where

$$L_f^{(i)}(A, E) = \frac{\mathrm{d}^j}{\mathrm{d}t^j} f(A + tE) \mid_{t=0} .$$

Taking the $\mathrm{Im}(\cdot)$, we have $\frac{f(A+\mathrm{i}hE)}{h}$ is an order $\mathcal{O}(h^2)$ approximation of $L_f(A, E)$.

▶ **Remark.**

- h is not restricted by floating point arithmetic considerations.
- Computing f must not employ complex arithmetic, e.g. we cannot use Schur decomposition which can arise complex values.

4.3 Condition Estimation

The key idea is that: L_f is a *linear* function of E. Then

$$\mathrm{vec}(L_f(A, E)) = K(A)\mathrm{vec}(E) \tag{3}$$

where $K(A) \in \mathbb{C}^{n^2\times n^2}$ is the Kronecker form of the Fréchet Derivative.

▶ **Lemma 4.7.**

$$\| L_f(A) \|_F = \| K(A) \|_2.$$

Proof. Notice that $\| A \|_F = \| \mathrm{vec}(A) \|_2$, then

$$\| L_f(A) \|_F = \max_{E\neq 0} \frac{\| L_f(A, E) \|_F}{\| E \|_F} = \max_{E\neq 0} \frac{\| \mathrm{vec}(L_f(A, E)) \|_2}{\| \mathrm{vec}(E) \|_2} = \max_{E\neq 0} \frac{\| K(A)\mathrm{E} \|_2}{\| \mathrm{vec}(E) \|_2} = \| K(A) \|_2,$$

by definition of 2-norm. □

Algorithm 1 Power method applied to A^*A to produce $\gamma \leq \| A \|_2$

```
Choose a nonzero starting vector z_0 ∈ C^n,
for k = 0 : ∞ do
    w_{k+1} = A z_k
    z_{k+1} = A^* w_{k+1}
    γ_{k+1} = ‖ z_{k+1} ‖_2 / ‖ w_{k+1} ‖_2
    if Converged then
        γ = γ_{k+1}, quit.
    end if
end for
```

For $A = K(A)$, we would like to know how to compute $(K(A))^*x$ and $K(A)x$. These quantities can be calculated using Fréchet Derivative by (3) which gives Algorithm 2.

Algorithm 2 2-norm power method to produce $\gamma \leq \| L_f(A) \|_F$.

Choose a starting nonzero starting matrix $Z_0 \in \mathbb{C}^{n \times n}$.
for $k = 0 : \infty$ **do**
  $W_{k+1} = L_f(A, Z_k)$
  $Z_{k+1} = L_f^*(A, W_{k+1})$
  $\gamma_{k+1} = \| Z_{k+1} \|_F / \| W_{k+1} \|_F$
  if Converged **then**
    $\gamma = \gamma_{k+1}$, quit.
  end if
end for

In practice, we use instead the block 1-norm estimator (Higham & Tisseur, 2000).

5 Problem Classification

5.1 Small/Medium Scale Problem

1. Decomposition
   - Normal A: if we can compute Schur/spectral decomposition $A = QDQ^*, D = \mathrm{diag}(d_i)$, then $f(A) = Q\,\mathrm{diag}(f(d_i))Q^*$.
   - Nonnormal A: if we can compute Schur decomposition $A = QTQ^*$, then use Schur-Parlett method.
2. Matrix iterations: $X_{k+1} = g(X_k), X_0 = A$, for matrix pth root, sign function, polar decomposition. Iterations only require matrix multiplication and solution of multiple right-hand side linear system.
3. Approximation method: polynomial (Taylor expansion) and rational (Padé) approximation.

5.2 Large Scale $f(A)b$ problem

Assume that A is large and sparse, $f(A)$ cannot be stored exactly and the problem is $f(A)b$: action of $f(A)$ on b.

Case 1. We can solve $Ax = b$ by sparse direct methods such as `SparseLU` in MATLAB but cannot compute the Schur decomposition: "backslash matrix".

- Cauchy integral formula can be used (Hale, H & Trefethen, 2008).
- Rational Krylov can be used with direct solves.

Case 2. We can only compute matrix-vector products Ax and maybe A^*x. You may have more information of A such as A is symmetric or $\lambda(A) \subseteq [\lambda_{\min}, \lambda_{\max}]$ with $\lambda_{\min}$ and $\lambda_{\max}$ known.

- Krylov methods.
- Polynomial approximations.

5.3 Accuracy Requirement

- Full double precision.
- Variable tolerance, e.g. within an ODE integrator.
- Given tolerance, where matrix A is subject to measurement error, e.g. $\approx 10^{-4}$ in engineering and healthcare.

6 Methods for $f(A)$

This this second, we will introduce

- Approximation techniques: Taylor series and Padé approximation.
- Decomposition techniques: Similarity transformations and block diagonalizations.
- Iteration methods: Parlett recurrence, BH method, block Parlett recurrence and Schur-Parlett algorithm.
- Finally we analyze the Newton's method for matrix sign function and square root function.

Approximation Methods.

6.1 Taylor Series

Matrix Taylor series converges if eigenvalues of increment matrix lie within radius of convergence of series. Thus for all A,

$$\cos(A) = I - \frac{A^2}{2!} + \frac{A^4}{4!} - \frac{A^6}{6!} + \cdots$$

We have the bound for error in trancated Taylor series in terms of appropriate derivative at matrix argument. However, we need some numerical consideration of computing the truncated Taylor series.

6.2 Padé Approximation

Rational function $r_{km}(x) = p_{km}(x)/q_{km}(x)$ is a $[k, m]$ Padé approximation to $f(x) = \sum_{i=0}^{\infty} \alpha_i x^i$ if p_{km} and q_{km} are polynomials of degree at most k and m respectively, and

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

The idea is: p_{km} is a polynomial of degree k, so it has $k+1$ degree of freedom. Similarly, q_{km} has $m+1$ degree of freedom. Therefore, we have $k+m+2$ degree of freedom in total. However, multiply p_{km} and q_{km} by the same constant will not change r_{km} cause the degree of freedom is actually $k+m+1$. Therefore we expect $f(x) - r_{km}(x) = \mathcal{O}(x^{k+m+1})$ which is the first "unconsidered" term of the Padé approximant $r_{km}(x)$.

▶ **Remark.**

- This is generally more efficient than truncated Taylor series.
- Possible ways of evaluating $r_{km}(x)$
  - Ratio of polynomials.
  - Continued fractions.
  - Partial fractions.

Decomposition methods: Similarity transformations and block diagonalization.

6.3 Similarity Transformations

We can use the formula

$$A = XBX^{-1} \quad \Rightarrow \quad f(A) = Xf(B)X^{-1}$$

provided $f(B)$ is easy to compute.

The problem of this method is that, any error in $f(B)$ will be magnified by a factor up to $\kappa(X) = \| X \| \| X^{-1} \|$: Consider $A = XBX^{-1}$ and $f(A) = Xf(B)X^{-1}$. Suppose we have our computed $f(A)$ as $\widetilde{F} = X(f(B) + \Delta B)X^{-1}$, where $\| \Delta B \| \approx u \| B \|$. Then we have $\widetilde{F} = f(A) + \Delta F$, where

$$\| \Delta F \| \leq u \| B \| \kappa(X).$$

The matrix X can be ill-conditioned which leads to a large error in computed $f(A)$. This is why we prefer unitary X, thus we can use

- Eigendecomposition (diagonal B) when A is normal.
- Schur decomposition (triangular B) in general A.

▶ **Example 6.1.** Suppose we have our `funm_ev`

```
function F = funm_ev(A,fun)
[V,D] = eig(A);
F = V * diag(feval(fun,diag(D))) / V;
```

Then we compare the error

```
>> A = [3,-1;1,1]; X = funm_ev(A,@sqrt)
X =
    1.7678e+00  -3.5355e-01
    3.5355e-01   1.0607e+00
>> norm(A-X*X)       % cond(V) =  9.5e7
ans =
    2.3067e-08
>> Y = sqrtm(A); norm(A-Y*Y)
ans =
    6.4855e-16
```

6.4 Block Diagonalization

We could find a intermediate solution between the Schur form and completely diagonalizing the matrix is the block diagonalization $A = XDX^{-1}$, where

- X is well conditioned.
- $D = \operatorname{diag}(D_{ii})$ is block diagonal.

We start from a Schur decomposition $A = QTQ^*$, then we do further work to compute a block diagonal matrix $D = XTX^{-1} = \operatorname{diag}(D_{ii})$ where $\kappa(X) < tol$. The size of block depends on your tolerance, the smaller then tol is, the bigger the block you will get.

Go back to Schur form and let's see how to compute $F = f(T)$ where T is triangular.

6.5 Parlett's Recurrence

If T is upper triangular, then $F = f(T)$ is upper triangular as well, and $f_{ii} = f(t_{ii})$. Parlett (1976) showes that the off-diagonal entries can be obtained by 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.

However, the recurrence fails when T has repeated eigenvalues, and can suffer from severe loss of accuracy in floating point arithmetic when two eigenvalues t_{ii} and t_{jj} are very close.

This method has been used in MATLAB 6.5 (2002) and eariler.

6.6 Block Parlett Recurrence

Recall that Parlett fails for repeated eigenvalues. We can develop a block version of Parlett recurrence. For T upper triangular, we can partition $T = (T_{ij})$ with square diagonal blocks such that $\lambda(T_{ii}) \cap \lambda(T_{jj}) = \emptyset$. T and $F = f(T)$ has the same block structure. From $FT = TF$, we have the Sylvester equations of F_{ij}

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

and this is guaranteed to have unique solution[1].

[1]For general Sylvester equation $AX + XB = C$, it has unique solution if and only if $\lambda(A) \cap \lambda(-B) = \emptyset$.

6.7 Schur-Parlett Algorithm

Based on the block Parlett algorithm and Schur decomposition, we have the Schur-Parlett algorithm:

1. Given a general A, we get the Schur decomposition $A = QTQ^*$.
2. Re-order T to block triangular form in which the eigenvalues within a block are "close" and those of separate blocks are "well separated".
3. Evaluate $F_{ii} = f(T_{ii})$.
4. Solve the Sylvester equations as shown in Section 6.6

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

5. Finally, $f(A) = QFQ^*$.

▶ **Remark.**

1. Reordering step: Need to choose how we cluster the eigenvalues, for example λ_i and λ_j go into the same set if $|\lambda_i - \lambda_j| \leq \delta = 0.1$. LAPACK carries out the swaps by unitary transformations.
2. Function of atomic blocks: One problem arises in step 3 of the Schur-Parlett algorithm is *how to compute $f(T_{ii})$?* These blocks are called the "atomic block" since they cannot be further split up to smaller block. We can rewrite $T_{ii} = \sigma I + M$ where $\sigma = \text{trace}(T_{ii})/\dim(T_{ii})$ which is the average of the eigenvalues of T_{ii}. Then M is triangular with small eigenvalues. Then we use the Taylor series
$$f(T_{ii}) = f(\sigma I + M) = f(\sigma I) + f'(\sigma I)M + \cdots .$$
M has small eigenvalues on diagonal, we expect $M^k \to 0$ rapidly.
3. Feature of Algorithm:
   - Cost $\mathcal{O}(n^3)$ flops, or upto $n^4/3$ flops if large blocks needed.
   - For blocks of size greater than 1, we needs derivatives.
   - Parameter δ control the blocking and the algorithm may be unstable for any δ.
   - This algorithm is the basis of `funm` in MATLAB.

6.8 Björck & Hammarling Method

Suppose we would like to compute the matrix square root. We can do the following by consider $X^2 = A$,

1. Compute the Schur decomposition $A = QTQ^*$.
2. $X = A^{1/2} = QT^{1/2}Q^* =: QUQ^*$.
3. Solve the problem $U^2 = T$ where T is upper triangular.

Björck and Hammarling (1983) solved this by

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

and then form $X = QUQ^*$. This algorithm can never fail provided A is nonsingular, since we are choosing the principal square root which makes $u_{ii} + u_{jj} > 0$. This has been extended to pth root ($U^p = T$) by Smith (2003).

▶ **Remark.**

- Schur decomposition gives perfect numerical stability.
- Cost is reasonable $\mathcal{O}(pn^3)$. For large p, one can reduce it to $\mathcal{O}(n^3 \log p)$.

- This can be generalized to real Schur decomposition.

▶ **Remark** (Parlett v.s. Björck & Hammarling)**.** Parlett recurrence is not "optimal", from the square root formula: x_{12} is obtained from

$$\text{Parlett} \quad \frac{a_{12}(\sqrt{a_{11}} - \sqrt{a_{22}})}{a_{11} - a_{22}} = \frac{a_{12}}{\sqrt{a_{11}} + \sqrt{a_{22}}} \quad \text{B \& H.}$$

Newton's method for matrix sign function and matrix square root.

6.9 Matrix Sign Function

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

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

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

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

Alternatively,

$$\operatorname{sign}(A) = A(A^2)^{1/2}.$$

Alternatively,

$$\operatorname{sign}(A) = \frac{2}{\pi} A \int_0^\infty (t^2 I + A^2)^{-1} \, \mathrm{d}t. \tag{4}$$

6.9.1 Newton's Method for sign function

Apply Newton's method to $X^2 = I$, we will get

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

Convergence. Let $S \coloneqq \operatorname{sign}(A), G \coloneqq (A - S)(A + S)^{-1}$. Then we have

$$X_k = \left(I - G^{2^k}\right)^{-1} \left(I + G^{2^k}\right) S,$$

where the eigenvalues of G are

$$(\lambda_i - \operatorname{sign}(\lambda_i))/(\lambda_i + \operatorname{sign}(\lambda_i))$$

which is strictly less than 1. Hence $\rho(G) < 1$ gives $G^k \to 0$. Hence $X_k \to S$ as $k \to \infty$.

Programming Aspect. Convergence can be very slow initially. Suppose we have a very large eigenvalue in X_0, then $X_1 = (X_0 + X_0^{-1})/2$ which $\rho(X_1) \approx \rho(X_0)/2$. This leads to linear convergence, therefore in practice we will include some scaling to speed it up.

6.10 Newton's Method for Square Root

Newton's method applies to $X^2 = A$ gives

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

This is not feasible since solving Sylvester equation requires Schur decomposition, but if we have the Schur decomposition, we can instead use the Björck and Hammarling method to compute the square root.

In addition, suppose we assume $AX_0 = X_0 A$, then we can show that

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

▶ **Remark.**

1. For nonsingular A, we expect a local quadratic convergence of full Newton to a primary square root.
2. To which square root do the iteration converge?
3. $(*)$ can converge when full Newton breaks down (Jacobian matrix is singular). Hence the full Newton's method and $(*)$ are not fully equivalent.
4. Lack of symmetry in $(*)$.

6.11 Convergence Analysis of Newton's Method

How do we analyze the convergence of $(*)$?

6.11.1 Jordan form

By Assume $X_0 = p(A)$ for some polynomial p, then all the iterates are all polynomials in A. Let $Z^{-1}AZ = J$ be Jordan canonical form and set $Z^{-1}X_kZ = Y_k$. Then

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

Notice, J is upper triangular, therefore all Y_k are upper triangular. In particular, the diagonal of Y_k have the following iteration form in scalar:

$$\text{Heron:} \quad y_{k+1} = \frac{1}{2}\left(y_k + \frac{\lambda}{y_k}\right), \quad y_0 = \lambda,$$

which converges to $\sqrt{\lambda}$. Therefore the convergence of eigenvalues is immediate from the classical results. What about the convergence of the off-diagonal entries? The result is not immediate, details can be found in (Higham, Function of Matrices, Thm 4.15).

▶ **Remark.** Analysis does not generalize to $AX_0 = X_0A$ where X_0 is not necessarily a polynomial in A. $X_0 = p(A)$ implies $X_0A = AX_0$, and the converse is not true in general. In fact, X_0 is commute with every matrix that commute with A, then $X_0 = p(A)$.

6.11.2 Sign function

▶ **Theorem 6.2.** *Let $A \in \mathbb{C}^{n\times n}$ has no eigenvalues on $\mathbb{R}_-$. The Newton square root iterates X_k with $X_0A = AX_0$ are related to the Newton sign iterates*

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

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 $X_k \to A^{1/2}\text{sign}(S_0)$ quadratically.

Sketch proof. $S_0 = A^{-1/2}X_0$ implies S_0 commute with A since X_0 and $A^{-1/2}$ are polynomials of A, hence S_0 is a polynomial of A with commutes with A.

Assume that $X_k = A^{1/2}S_k$ and $S_kA = AS_k$, then $S_kA^{1/2} = A^{1/2}S_k$ since $A^{1/2}$ is a polynomial in A.

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}A) = \frac{1}{2}(A^{1/2}S_k + S_k^{-1}A^{-1/2}A) = \frac{1}{2}A^{1/2}(S_k + S_k^{-1}) = A^{1/2}S_{k+1}.$$

$S_{k+1}A = AS_{k+1}$ because $S_kA = AS_k$. □

This theorem gives $X_k \to A^{1/2}$ if $\lambda(A^{-1/2}X_0) \subseteq \text{RHP}$.

6.12 Stability of Newton Iteration

▶ **Definition 6.3.** The iteration $X_{k+1} = g(X_k)$ is stable in a neighborhood of a fixed point X if Fréchet Derivative $L_g(X, E)$ has bounded powers.

Let $X_0 = X + E_0$, and $E_k \coloneqq X_k - X$. Then

$$X_{k+1} = g(X_k) = g(X + E_k) + L_g(X, E_k) + o(\| E_k \|).$$

Since $g(X) = X$ and $X_{k+1} = X + E_k$,

$$E_{k+1} = L_g(X, E_k) + o(\| E_k \|).$$

If $\| L_g^i(X, E) \| \leq c$ for all i, then recurring leads to

$$\| E_k \| \leq c\| E_0 \| + kc \cdot o(\| E_0 \|).$$

6.12.1 Stability of Matrix Square Root

- $g(X) = \frac{1}{2}(X + X^{-1}A)$.
- $L_g(X, E) = (E - X^{-1}EX^{-1}A)/2$. Details in Appendix A.2.
- Fixed point: $X = A^{1/2}$.
- $L_g(A^{1/2}, E) = (E - A^{-1/2}EA^{1/2})/2$. And $L_g^i(A^{1/2}, E)$ is bounded if all eigenvalues are in the unit disc. The problem is what are the eigenvalues of $L_g(A^{1/2}, E)$?

  Approach 1. Taking the vec operator:

$$\begin{aligned}\mathrm{vec}(L_g(A^{1/2}, E)) &= \mathrm{vec}((E - A^{-1/2}EA^{1/2})/2) \\ &= \frac{1}{2}\big(\mathrm{vec}(E) - ((A^{1/2})^T \otimes A^{-1/2})\mathrm{vec}(E)\big) \\ &= \frac{1}{2}(I - (A^{1/2})^T \otimes A^{-1/2})\mathrm{vec}(E).\end{aligned}$$

  This is computable.

  Approach 2. Let $Az_i = \lambda_i z_i$ and $z_j^* A = \lambda_j z_j^*$. Define $E = z_i z_j^*$, then

$$L_g(A^{1/2}, E) = \frac{1}{2}\left(z_i z_j^* - A^{-1/2} z_i z_j^* A^{1/2}\right) = \frac{1}{2}(z_i z_j^* - \lambda_i^{-1/2} z_i \lambda_j^{1/2} z_j^*),$$

  which is equivalent to

$$L_g(A^{1/2}, E) = \frac{1}{2}\left(1 - \frac{\lambda_j^{1/2}}{\lambda_i^{1/2}}\right) E.$$

  Therefore, to ensure stability, we need

$$\max_{i,j} \frac{1}{2}| 1 - \frac{\lambda_j^{1/2}}{\lambda_i^{1/2}} | < 1.$$

  For A is Hermitian positive definite, $\kappa_2(A) < 9$.

▶ **Remark** (Advantages of Def. 6.3)**.**

- No additional assumption on A.
- Perturbation analysis is all in the definition.
- General, unifying approach.

6.12.2 Stability of Sign Iteration

▶ **Theorem 6.4.** *Let $X_{k+1} = g(X_k)$ be any superlinearly convergent iteration for $S = \operatorname{sign}(X_0)$. Then $L_g(S, E) = (E - SES)/2$, where $L_g(S, E)$ is the Fréchet Derivative of matrix sign function at S. Hence $L_g(S, E)$ is idempotent ($L_g(S, E) \circ L_g(S, E) = L_g(S, E)$), and the iteration is stable. Since $L_g^i(S, E) = L_g(S, E)$ which is definitely bounded.*

6.13 More Iterations for Sign Function

Starting with Newton's method:

$$x_{k+1} = \frac{1}{2}(x_k + x_k^{-1})$$

- Invert: $x_{k+1} = 2x_k/(x_k^2 + 1)$, which is quadratic convergent.
- Halley: $x_{k+1} = (x_k(3 + x_k^2))/(1 + 3x_k^2)$, which is cubic convergent.
- Newton–Schulz: $x_{k+1} = x_k(3 - x_k^2)/2$, which is quadratic but not globally convergent.

7 Methods for $f(A)b$

Given $A \in \mathbb{C}^{n\times n}, b \in C^n$ and $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$. We would like to compute $f(A)b$ without first computing $f(A)$.

7.1 $A^{1/2}b$ via contour integration

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

By computing the system $(zI - A)^{-1}b = x$ which is a shifted linear system, we can do the quadrature integration method. Suppose we would like to compute a 5×5 Pascal matrix: $\lambda(A) \in [0.01, 92.3], f(z) = z^{1/2}$. If we use the repeated trapezium rule to integrate around circle centred at $(\lambda_{\min} + \lambda_{\max})/2$ with radius $\lambda_{\max}/2$. To get 2 digits accuracy, we need 32,000 integration points, and 262,000 for 13 digits accuracy.

However, from Hale, Higham and Trefethen (2008), they did a conformally mapping from $\mathbb{C}\backslash\{(-\infty, 0]\cup [\lambda_{\min}, \lambda_{\max}]\}$ to an annulus: $[\lambda_{\min}, \lambda_{\max}]$ is mapped to the inner circle and $[-\infty, 0]$ to the outer circle. Using this approach, we only require 5 integration points for 2 digits accuracy and 35 points for 13 digits.

7.2 $A^\alpha b$ via binomial expansion

We cannot directly do binomial expansion on A^α, but let $A = s(I-C)$ where $s \in \mathbb{C}$, then $A^\alpha = s^\alpha(I-C)^\alpha$, and we can do a binomial expansion of $(I - C)^\alpha$. However, this requires $\rho(C) < 1$ to let the binomial expansion to converge.

If $\lambda_i > 0$, then $s = (\lambda_{\min} + \lambda_{\max})/2$ does the job, since

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

From

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

we have

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

7.3 $A^\alpha b$ via ODE IVP

Consider the ODE

$$\frac{\mathrm{d}y}{\mathrm{d}t} = \alpha(A-I)\left[t(A-I)+I\right]^{-1} y, \quad y(0) = b,$$

which has the unique solution

$$y(t) = [t(A-I)+I]^\alpha b,$$

so

$$y(1) = A^\alpha b.$$

Used by Allen, Baglama and Boyd (2000) for $\alpha = 1/2$ and symmetric poisitive definite A.

7.4 Compute $\mathrm{e}^A b$

We have the following formulae for e^A, $A \in \mathbb{C}^{n\times n}$,

1. Taylor series:
$$I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots.$$
2. Limit:
$$\lim_{s\to\infty} (I + A/s)^s.$$
3. Scaling and Squaring
$$\left(\mathrm{e}^{A\cdot 2^{-s}}\right)^{2^s}.$$
4. Cauchy integral:
$$\frac{1}{2\pi \mathrm{i}} \int_\Gamma \mathrm{e}^z (zI - A)^{-1}\, \mathrm{d}z.$$
5. Jordan form:
$$Z\,\mathrm{diag}(\mathrm{e}^{J_k})Z^{-1}.$$
6. Interpolation:
$$\sum_{i=1}^{n} f[\lambda_1, \ldots, \lambda_n] \prod_{j=1}^{i-1} (A - \lambda_j I).$$
7. Differential system:
$$Y'(t) = AY(t), \quad Y(0) = I.$$
8. Schur form:
$$Q\mathrm{e}^T Q^*.$$
9. Padé approximation:
$$p_{km}(A) q_{km}^{-1}(A).$$
10. Krylov method: Using the Arnoldi factorization:
$$AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T$$
with Hessenberg H_k, then
$$\mathrm{e}^A b \approx Q_k \mathrm{e}^{H_k} Q_k^* b.$$

We now want to discuss how we can compute $\mathrm{e}^A B$, where $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times n_0}$ with $n_0 \ll n$. Suppose we are able to compute Ax and A^*x. The problem will be: *Given tol, A and B, we would like to compute* $\mathrm{e}^A B$ *with "error"* $\leq$ *tol.*

7.4.1 AH-Algorithm

For integer s,

$$e^A B = \left(e^{s^{-1}A}\right)^s B = \underbrace{e^{s^{-1}A} \cdots e^{s^{-1}A}}_{s \text{ times}} B.$$

Choose s so that we can get the following truncated Taylor series

$$T_m(s^{-1}A) = \sum_{j=0}^{m} \frac{(s^{-1}A)^j}{j!} \approx e^{s^{-1}A}.$$

Then

$$B_{i+1} = T_m(s^{-1}A)B_i, \quad i = 0 : s-1, \quad B_0 = B$$

yields $B_s \approx e^A B$. The question is how to choose s and m? We could do the following truncation analysis: We know that $T_m(A) \approx e^A$, therefore

$$\log(e^{-A} T_m(A)) =: h_{m+1}(A)$$

we need to show this difference is close to 0, i.e. $h_{m+1}(A) \ll 1$. Take exponential at both sides

$$e^{-A} T_m(A) = e^{h_{m+1}(A)}$$

gives

$$T_m(A) = e^A e^{h_{m+1}(A)} = e^{A + h_{m+1}(A)}$$

since A and $h_{m+1}(A)$ commute. Then we can require $\| h_{m+1}(A) \| \leq tol \| A \|$. Similarly,

$$\left(T_m(A2^{-s})\right)^{2^s} = e^{A + 2^s h_{m+1}(2^{-s}A)} =: e^{A + \Delta A}.$$

The aim is selecting s such that

$$\frac{\| \Delta A \|}{\| A \|} = \frac{\| h_{m+1}(2^{-s}A) \|}{2^{-s}A} \leq u \approx 1.1 \times 10^{-16}.$$

In 2005 paper by Higham, it bound the

$$h_{m+1}(A) = \sum_{k=m+1}^{\infty} c_k A^k$$

by

$$\| h_{m+1}(A) \| \leq \sum_{k=m+1}^{\infty} | c_k | \| A \|^k.$$

This bound is too trivial, can we do anything better?

▶ **Lemma 7.1** (AI-Mohy and Higham, 2009)**.** *$T_m(s^{-1}A)^s B = e^{A+\Delta A} B$, where $\Delta A = s h_{m+1}(s^{-1}A)$ and $h_{m+1}(x) = \log(e^{-x} T_m(x)) = \sum_{k=m+1}^{\infty} c_k x^k$. Moreover*

$$\| \Delta A \| \leq s \sum_{k=m+1}^{\infty} | c_k | \alpha_p(s^{-1}A)^k$$

if $m + 1 \geq p(p-1)$, where

$$\alpha_p(A) = \max(d_p, d_{p+1}), \quad d_p = \| A^p \|^{1/p}.$$

Motivation: Here we bound the norm of $h_{m+1}(X) = \sum_{k=m+1}^{\infty} c_k X^k$. The nonnormality implies $\rho(A) \ll \| A \|$. Notice that

$$\rho(A) \leq \| A^p \|^{1/p} \leq \| A \|, \quad p = 1 : \infty$$

and $\lim_{p \to \infty} \| A^p \|^{1/p} = \rho(A)$. Hence we would like to use $\| A^p \|^{1/p}$ instead of $\| A \|$ in the truncation bounds.

Therefore the key idea of the algorithm is

Algorithm 3 Algorithm for $F = e^{tA}B$

$\mu = \text{trace}(A)/n$
$A = A - \mu I$
$[m_*, s] = \text{parameter}(tA)$
$F = B, \eta = e^{t\mu/s}$
for $i = 1 : s$ **do**
  $c_1 = \| B \|_\infty$
  for $j = 1 : m$ **do**
    $B = tAB/(sj), c_2 = \| B \|_\infty$
    $F = F + B$
    if $c_1 + c_2 \leq tol \| F \|_\infty$, quit, **end**
    $c_1 = c_2$
  end for
  $F = \eta F, B = F.$
end for

- Use the $\| A^p \|^{1/p}$ in the truncation bounds for a few small p.
- Choose optimal m and s for given tolerance.
- Preprocess A by shifting $A \leftarrow A - \mu I$ where $\mu = \text{trace}(A)/n$ to scale A with less condition number.

Conditioning of $e^A B$. How accurate can we expect the answer? Given A, B, we compute $e^A B$, the condition number can be defined as

$$\kappa_{\exp}(A, B) := \lim_{\epsilon \to 0} \sup_{\substack{\| \Delta A \| \leq \epsilon \| A \| \\ \| \Delta B \| \leq \epsilon \| B \|}} \frac{\| e^{A+\Delta A}(B + \Delta B) - e^A B \|}{\epsilon \| e^A B \|}.$$

Then

$$\kappa_{\exp}(A, B) \leq \frac{\| e^A \|_F \| B \|_F}{\| e^A B \|_F} (1 + \kappa_{\exp}(A)).$$

Therefore, the best we can expect is

$$\text{Rel. Error} \leq \kappa_{\exp}(A, B) \cdot tol.$$

From the rounding error analysis, the relative error due to roundoff is bounded by

$$u e^{\| A \|_2} \| B \|_2 / \| e^A B \|_F.$$

The question now becomes

$$\frac{u e^{\| A \|_2} \| B \|_2}{\| e^A B \|_F} \overset{?}{<} u \frac{\| e^A \|_F \| B \|_F}{\| e^A B \|_F} (1 + \kappa_{\exp}(A))$$

Cancelling yields

$$e^{\| A \|_2} \overset{?}{<} \| e^A \|_F (1 + \kappa_{\exp}(A)).$$

▶ **Remark.**

- If A is normal, then $\kappa_{\exp}(A) = \| A \|_2$, then instability arises. Suppose A is Hermitian, then $\| e^A \|_2 = e^{\lambda_{\max}}$, however, $e^{\| A \|_2} = \max(e^{\lambda_{\max}}, e^{-\lambda_{\min}})$. Thus, the instability arises when A is Hermitian and has a large negative eigenvalue.
- However, we have shifted the matrix A by μI which gives $\lambda_{\max} = -\lambda_{\min}$ which implies $e^{\| A \|_2} = \| e^A \|_2$ which implies stability for Hermitian A.

TABLE 1. *Advantages of AH-algorithm over Krylov method.*

AH-Algorithm	Krylov Method
Most time spent in matrix-vector product.	Krylov recurrence and e^H can be significant.
"Direct method", hence the cost is predictable.	Need stopping test.
No parameter to estimate.	Select Krylov subspace size.
Storage: 2 vectors.	Storage: Krylov basis.
Handle with B which is a matrix.	To handle B, we need block Krylov method.

A Supplimentary

A.1 Figure 1

```
% we would like to find the total error (truncation error + roundoff
% error) of f(x) = x^2 near x = 1. f'(x) = 2x.
clc; clear; close all;
power = -1:-1e-3:-16;
h = 10; h = h.^power;
error = zeros(length(h),1);
ref = 2;
MC = 50;
for i = 1:length(h)
    acc = 0;
    for j = 1:MC
        acc = acc + ((1+h(i))*(1+h(i)) - 1)/h(i) - 2;
    end
    error(i) = acc/MC;
end
loglog(h,abs(error));
xlabel("Time step $(h)$");
ylabel("Total Error");
set(gca,"FontSize",20);
```

A.2 Proof in Section 6.12.1

We would like to prove for $g(X) = \frac{1}{2}(X + X^{-1}A)$. The Fréchet Derivative is given by $L_g(X, E) = (E - X^{-1}EX^{-1}A)/2$.

Proof. $g(X + E) - g(X) = (E + (X + E)^{-1}A - X^{-1}A)/2$. Remain to expand $(X + E)^{-1}$: Consider the expansion of $(X + E)^{-1}$,

$$\begin{aligned}(X + E)^{-1} &= (X(I + X^{-1}E))^{-1} = (I + X^{-1}E)^{-1}X^{-1} \\ &= (f(I) + f'(I)X^{-1}E + \mathcal{O}(\| E \|^2))X^{-1} \\ &= (I + (-I^2)X^{-1}E)X^{-1} + \mathcal{O}(\| E \|^2) \\ &= X^{-1} - X^{-1}EX^{-1} + \mathcal{O}(\| E \|^2).\end{aligned}$$

Going back to $L_g(X, E)$, we have

$$g(X + E) - g(X) = (E + (X^{-1} - X^{-1}EX^{-1} + \mathcal{O}(\| E \|^2))A - X^{-1}A)/2,$$

expanding out yields the result. □