MANCHESTER 1824

The University of Manchester

The $f(A)b$ Problem

Nick Higham
School of Mathematics
The University of Manchester

higham@ma.man.ac.uk
http://www.ma.man.ac.uk/~higham/

ICIAM 2011, Vancouver, July 2011.

The $f(A)b$ Problem

Given

- matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$,
- $A \in \mathbb{C}^{n\times n}$, $b \in \mathbb{C}^n$,

compute $f(A)b$ *without first computing* $f(A)$.

Most important cases

- ~~$f(x) = x^{-1}$~~
- $f(x) = e^x$.
- $f(x) = \log(x)$.
- $f(x) = x^{1/2}$.
- $f(x) = \operatorname{sign}(x)$.

Application: Second Order ODE

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

has solution

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

Application: Second Order ODE

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

has solution

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

But

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

$A^{1/2}b$: Application in Statistics

Chen, Anitescu & Saad (2011): sample $x \sim N(\mu, C)$, $C \in \mathbb{R}^{m\times m}$, $m \in [10^{12}, 10^{15}]$.

Let $x \sim N(0, I)$.

- If $C = LL^T$ is Cholesky factorization, $y = \mu + Lx \sim N(\mu, C)$.
- Cholesky factorization may not be computable or storable.
- $y = \mu + C^{1/2}x \sim N(\mu, C)$.

$A^{1/2}\,b$ via Contour Integration

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

A 5×5 Pascal matrix: $\lambda(A) \in [0.01, 92.3]$, $f(z) = z^{1/2}$.

- Use repeated trapezium rule to integrate around circle centre $(\lambda_{\min} + \lambda_{\max})/2$, radius $\lambda_{\max}/2$.

  Need 32,000 (262,000) points for 2 (13) decimal digits.
- **Hale, H & Trefethen** (2008): conformally map

$$\Omega = \mathbb{C} \setminus \{\,(-\infty, 0] \cup [m, M]\}.$$

  to an annulus: $[m, M] \to$ inner circle, $(-\infty, 0] \to$ outer circle. Now 5 (35) points for 2 (13) decimal digits. Further refinements: 20 points yield 14 digits.

$A^\alpha b$ via Binomial Expansion

Write $A = s(I - C)$.

- If $\lambda_i > 0$, $s = (\lambda_{\min} + \lambda_{\max})/2$ yields $\rho_{\min} = (\lambda_{\max} - \lambda_{\min})/(\lambda_{\max} + \lambda_{\min})$.
- For any A, $s = \mathrm{trace}(A^*A)/\mathrm{trace}(A^*)$ minimizes $\|C\|_F$.

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

So

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

$A^\alpha b$ via Binomial Expansion

Write $A = s(I - C)$.

- If $\lambda_i > 0$, $s = (\lambda_{\min} + \lambda_{\max})/2$ yields $\rho_{\min} = (\lambda_{\max} - \lambda_{\min})/(\lambda_{\max} + \lambda_{\min})$.
- For any A, $s = \mathrm{trace}(A^*A)/\mathrm{trace}(A^*)$ minimizes $\|C\|_F$.

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

So

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

For M-matrices, required splitting with $C \geq 0$ always exists.

$A^\alpha b$ via ODE IVP

$$\frac{dy}{dt} = \alpha(A - I)\big[t(A - I) + I\big]^{-1} y, \qquad y(0) = b$$

has unique solution

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

so

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

Used by Allen, Baglama & Boyd (2000) for $\alpha = 1/2$, spd A.

Exponential Integrators

$$u'(t) = Au(t) + g(t, u(t)), \quad u(0) = u_0, \quad t \geq 0,$$

Solution can be written

$$u(t) = e^{tA}u_0 + \sum_{k=1}^{\infty} \varphi_k\big(tA\big)\, t^k\, u_k,$$

where $u_k = g^{(k-1)}(t, u(t))\,|_{t=0}$ and $\varphi_\ell(z) = \sum_{k=0}^{\infty} z^k/(k+\ell)!$.

Exponential Integrators

$$u'(t) = Au(t) + g(t, u(t)), \quad u(0) = u_0, \quad t \geq 0,$$

Solution can be written

$$u(t) = e^{tA}u_0 + \sum_{k=1}^{\infty} \varphi_k(tA)\, t^k\, u_k,$$

where $u_k = g^{(k-1)}(t, u(t))\,|_{t=0}$ and $\varphi_\ell(z) = \sum_{k=0}^{\infty} z^k/(k+\ell)!$.

$$u(t) \approx \widehat{u}(t) = e^{tA}u_0 + \sum_{k=1}^{p} \varphi_k(tA)\, t^k\, u_k.$$

Exponential time differencing (ETD) Euler ($p = 1$):

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

Saad's Trick (1992)

$$\varphi_1(z) = \frac{e^z - 1}{z}.$$

$$\exp\left(\begin{bmatrix} A & b \\ 0 & 0 \end{bmatrix}\right) = \begin{bmatrix} e^A & \varphi_1(A)b \\ 0 & 1 \end{bmatrix}$$

Theorem (Al-Mohy & H, 2011)

Let $A \in \mathbb{C}^{n\times n}$, $U = [u_1, u_2, \ldots, u_p] \in \mathbb{C}^{n\times p}$, $\tau \in \mathbb{C}$, *and define*

$$B = \begin{bmatrix} A & U \\ 0 & J \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+p)}, \quad J = \begin{bmatrix} 0 & I_{p-1} \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{p\times p}.$$

Then for $X = e^{\tau B}$ *we have*

$$X(1\colon n, n+j) = \sum_{k=1}^{j} \tau^k \varphi_k(\tau A)\, u_{j-k+1}, \quad j = 1\colon p.$$

Theorem (Al-Mohy & H, 2011)

Let $A \in \mathbb{C}^{n\times n}$, $U = [u_1, u_2, \ldots, u_p] \in \mathbb{C}^{n\times p}$, $\tau \in \mathbb{C}$, *and define*

$$B = \begin{bmatrix} A & U \\ 0 & J \end{bmatrix} \in \mathbb{C}^{(n+p)\times(n+p)}, \quad J = \begin{bmatrix} 0 & I_{p-1} \\ 0 & 0 \end{bmatrix} \in \mathbb{C}^{p\times p}.$$

Then for $X = e^{\tau B}$ *we have*

$$X(1\colon n, n+j) = \sum_{k=1}^{j} \tau^k \varphi_k(\tau A)\, u_{j-k+1}, \quad j = 1\colon p.$$

Completely removes the need to evaluate φ_k functions!

Implementation of Exponential Integrators

We compute

$$\widehat{u}(t) = e^{tA}u_0 + \sum_{k=1}^{p} \varphi_k(tA)\, t^k\, u_k$$

as, with $U = [u_p, \ldots, u_1]$,

$$\widehat{u}(t) = \begin{bmatrix} I_n & 0 \end{bmatrix} \exp\left(t \begin{bmatrix} A & \eta U \\ 0 & J \end{bmatrix} \right) \begin{bmatrix} u_0 \\ \eta^{-1} e_p \end{bmatrix}.$$

Choose η so that $\eta\|U\| \approx 1$.

Assumptions

Choice of algorithm depends on assumptions about A.

Examples:

- Can Schur-factorize A.
- "Backslash matrix": can solve $(zI - A)x = b$ by (sparse) direct methods.
- A symmetric, can estimate $[\lambda_{\min}, \lambda_{\max}]$, only matrix–vector products available.
- A is general, only matrix–vector products available.

Formulae for e^A, $A \in \mathbb{C}^{n\times n}$

Taylor series $I + A + \frac{A^2}{2!} + \frac{A^3}{3!} + \cdots$	**Limit** $\lim_{s\to\infty} (I + A/s)^s$	**Scaling and squaring** $(e^{A/2^s})^{2^s}$
Cauchy integral $\frac{1}{2\pi i}\int_\Gamma e^z (zI - A)^{-1}\, dz$	**Jordan form** $Z\mathrm{diag}(e^{J_k})Z^{-1}$	**Interpolation** $\sum_{i=1}^{n} f[\lambda_1, \ldots, \lambda_i] \prod_{j=1}^{i-1} (A - \lambda_j I)$
Differential system $Y'(t) = AY(t),\ Y(0) = I$	**Schur form** $Qe^T Q^*$	**Padé approximation** $p_{km}(A)q_{km}(A)^{-1}$

Krylov methods: Arnoldi fact. $AQ_k = Q_k H_k + h_{k+1,k} q_{k+1} e_k^T$ with Hessenberg H: $e^A b \approx Q_k e^{H_k} Q_k^* b$.

The Sixth Dubious Way

Moler & Van Loan (1978, 2003)

METHOD 6. SINGLE STEP O.D.E. METHODS. Two of the classical techniques for the solution of differential equations are the fourth order Taylor and Runge–Kutta methods with fixed step size. For our particular equation they become

$$x_{j+1} = \left(I + hA + \cdots + \frac{h^4}{4!}A^4\right)x_j = T_4(hA)x_j$$

and

$$x_{j+1} = x_j + \tfrac{1}{6}k_1 + \tfrac{1}{3}k_2 + \tfrac{1}{3}k_3 + \tfrac{1}{6}k_4,$$

where $k_1 = hAx_j$, $k_2 = hA(x_j + \frac{1}{2}k_1)$, $k_3 = hA(x_j + \frac{1}{2}k_2)$, and $k_4 = hA(x_j + k_3)$. A little manipulation reveals that in this case, the two methods would produce identical results were it not for roundoff error. As long as the step size is fixed, the matrix $T_4(hA)$ need be computed just once and then x_{j+1} can be obtained from x_j with just one matrix-vector multiplication. The standard Runge–Kutta method would require 4 such multiplications per step.

Let us consider $x(t)$ for one particular value of t, say $t = 1$. If $h = 1/m$, then

$$x(1) = x(mh) \simeq x_m = [T_4(hA)]^m x_0.$$

Computing $e^A B$

$\underbrace{A}_{n\times n}$, $\underbrace{B}_{n\times n_0}$, $n_0 \ll n$. Exploit, for integer s,

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

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

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

yields $B_s \approx e^A B$.

Computing $e^A B$

$\underbrace{A}_{n\times n}$, $\underbrace{B}_{n\times n_0}$, $n_0 \ll n$. Exploit, for integer s,

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

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

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

yields $B_s \approx e^A B$.

How to choose s and m?

Backward Error Analysis

Lemma (Al-Mohy & H, 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,

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

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

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

Why Use $d_p = \|A^p\|^{1/p}$?

- $\rho(A) \le d_p \le \|A\|$.
- $\|A^k\| \le \|A^{pk}\|^{1/p} \le d_p^k$.
- With $A = \begin{bmatrix} 0.9 & 500 \\ 0 & -0.5 \end{bmatrix}$:

k	2	5	10
$\|A^k\|_1$	2.0e2	2.2e2	1.2e2
$\|A\|_1^k$	2.5e5	3.1e13	9.8e26
$d_2^k = \left(\|A^2\|_1^{1/2}\right)^k$	2.0e2	5.7e5	3.2e11
$d_3^k = \left(\|A^3\|_1^{1/3}\right)^k$	4.5e1	1.3e4	1.9e8

- Cheaply estimate $\|A^k\|_1$, for a few k (H & Tisseur, 2000).

Why Use $d_p = \|A^p\|^{1/p}$? — cont.

- $d_p = \|A^p\|^{1/p}$ provide information about the nonnormality of A.
- Their use helps avoid overscaling.
- What other uses do they have?

Size of Taylor Series Argument

Constant

$$\theta_m := \max \left\{ \theta : \sum_{k=m+1}^{\infty} |c_k| \theta^{k-1} \le \epsilon \right\}.$$

Computed via symbolic, high precision:

m	10	20	30	40	55
single	1.0e0	3.6e0	6.3e0	9.1e0	1.3e1
double	1.4e-1	1.4e0	3.5e0	6.0e0	9.9e0

Choice of s and m

- $\alpha_p(A) := \max\left(d_p, d_{p+1}\right)$

$\dfrac{\|\Delta A\|}{\|A\|} \leq \epsilon$ if $m+1 \geq p(p-1)$ and $s^{-1}\alpha_p(A) \leq \theta_m$.

Computational cost for $B_s \approx e^A B$ is

$$C_m(A) = m \max\left(\lceil \alpha_p(A)/\theta_m \rceil, 1\right)$$

matrix products.

- Cost decreases with m.
- Restrict $2 \leq p \leq p_{\max}$, $p(p-1)-1 \leq m \leq m_{\max}$.
- Minimize cost over p, m.

Preprocessing

Expand the Taylor series about $\mu \in \mathbb{C}$:

$$e^{\mu} \sum_{k=0}^{\infty} (A - \mu I)^k / k!$$

Choose μ so $\|A - \mu I\| \leq \|A\|$.

- Alg is based on 1-norm, but minimizing $\|A - \mu I\|_F$ does better empirically at minimizing $d_p(A - \mu I)$.
- Recover e^A from

$$e^{\mu} \left[T_m(s^{-1}(A - \mu I)) \right]^s, \qquad \left[e^{\mu/s} T_m(s^{-1}(A - \mu I)) \right]^s.$$

  First expression prone to overflow, so prefer second.
- Balancing is an option.

Termination Criterion

In evaluating

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

we accept $T_k(A)B_i$ for the first k such that

$$\frac{\|A^{k-1}B_i\|}{(k-1)!} + \frac{\|A^k B_i\|}{k!} \leq \epsilon \|T_k(A)B_i\|.$$

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

$\mu = \mathrm{trace}(A)/n$
$A = A - \mu I$
$[m_*, s] = \mathrm{parameters}(tA)$ % Includes norm estimation.
$F = B$, $\eta = e^{t\mu/s}$
for $i = 1\colon s$
    $c_1 = \|B\|_\infty$
    for $j = 1\colon m_*$
        $B = tAB/(sj)$, $c_2 = \|B\|_\infty$
        $F = F + B$
        if $c_1 + c_2 \leq \mathrm{tol}\|F\|_\infty$, quit, end
        $c_1 = c_2$
    end
    $F = \eta F$, $B = F$
end

George Forsythe "Pitfalls" (1970)

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots.$$

Since you learned mathematics because it is useful, you might expect to use the series to compute e^x. Suppose—just for illustration—that your floating-point number system F is characterized by $\beta = 10$ and $s = 5$. Let us use the series for $x = -5.5$, as proposed by Stegun and Abramowitz [13]. Here are the numbers we get:

$$\begin{aligned} e^{-5.5} \approx \quad & 1.0000 \\ & - 5.5000 \\ & +15.125 \\ & -27.730 \\ & +38.129 \\ & -41.942 \\ & +38.446 \\ & -30.208 \\ & +20.768 \\ & -12.692 \\ & + 6.9803 \\ & - 3.4902 \\ & + 1.5997 \\ & \quad \vdots \\ \hline & + 0.0026363 \end{aligned}$$

Conditioning of $e^A B$

Relative forward error due to roundoff bounded by

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

- A normal implies $\kappa_{\exp}(A) = \|A\|_2$. Then instability if $e^{\|A\|_2} \gg \|e^A\|_2$.
- A Hermitian implies spectrum of $A - n^{-1}\mathrm{trace}(A)I$ has $\lambda_{\max} = -\lambda_{\min} \;\Rightarrow$ (normwise) stability!

Comparison with the Sixth Dubious Way

Advantages of our method over the one-step ODE integrator:

- Fully exploits the **linearity** of the ODE.
- **Backward error** based; ODE integrator controls local (forward) errors.
- **Overscaling** avoided.

Experiment 1

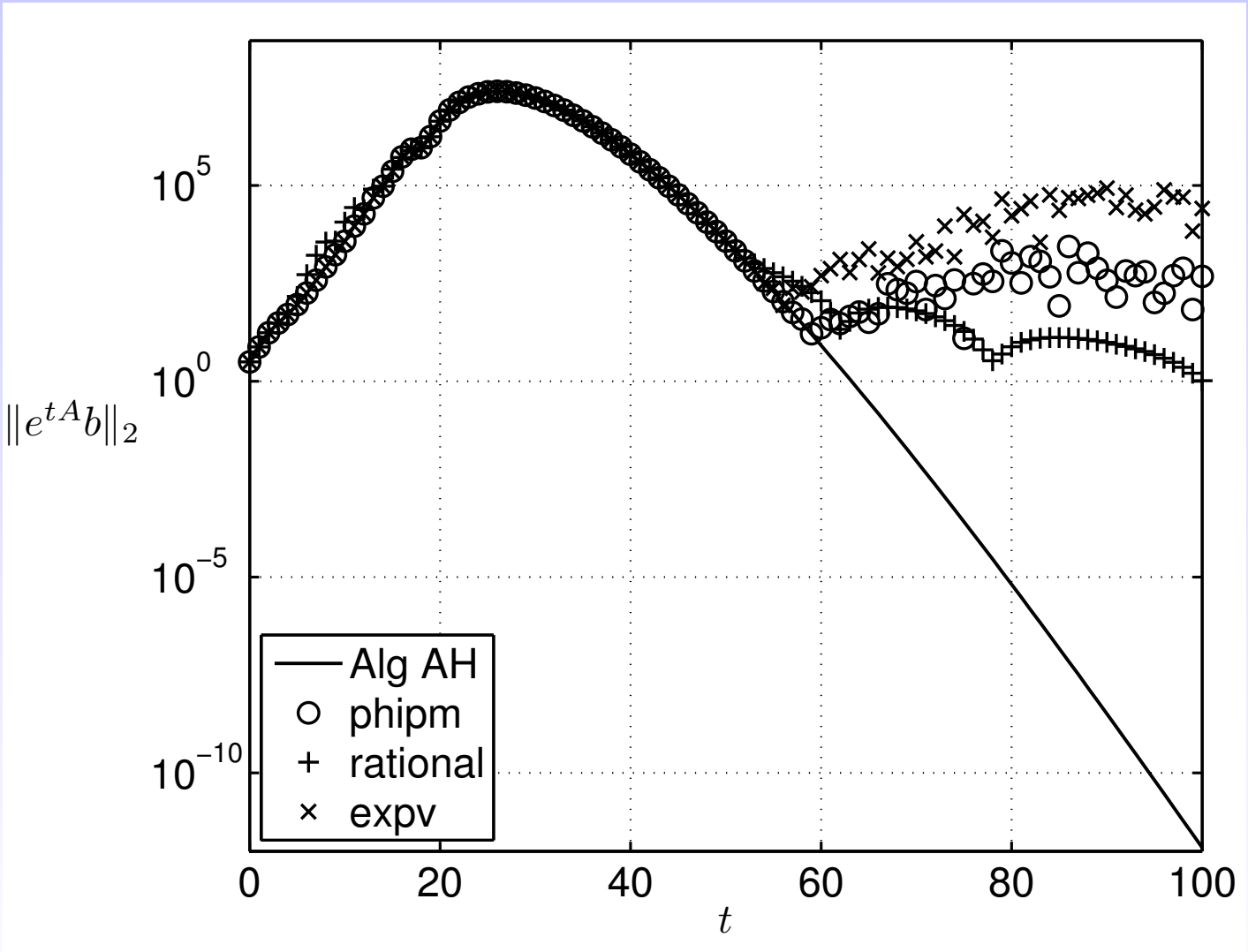
Trefethen, Weideman & Schmelzer (2006):
$A \in \mathbb{R}^{9801\times 9801}$, 2D Laplacian,
`-2500*gallery('poisson',99)`.

Compute $e^{\alpha A}b$, tol $= u_d$.

	$\alpha = 0.02$			$\alpha = 1$		
	speed	mv	diff	speed	mv	diff
Alg AH	1	1010		1	47702	
`expv`	2.8	403	7.7e-15	1.3	8835	4.2e-15
`phipm`	1.1	172	3.1e-15	0.2	2504	4.0e-15
rational	3.8	7	3.3e-14	0.1	7	1.2e-12

Experiment 2

$A =$ `-gallery('triw',20,4.1)`, $b_i = \cos i$, tol $= u_d$.



Conclusions

- $f(A)b$ problem of growing interest.
- Many possible approaches (including Krylov—not discussed here).
- Method design/choice must take into account
  - assumptions about A,
  - desired accuracy.
- New Taylor series-based alg for $e^A B$ very competitive with existing methods. Well suited for "black box" code.

  Al-Mohy & H. Computing the action of the matrix exponential, with an application to exponential integrators. SISC, 2011.

References I


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


E. J. Allen, J. Baglama, and S. K. Boyd.
Numerical approximation of the product of the square root of a matrix with a vector.
Linear Algebra Appl., 310:167–181, 2000.


J. Chen, M. Anitescu, and Y. Saad.
Computing $f(A)b$ via least squares polynomial approximations.
SIAM J. Sci. Comput., 33(1):195–222, 2011.

References II

P. I. Davies and N. J. Higham.
Computing $f(A)b$ for matrix functions f.
In A. Boriçi, A. Frommer, B. Joó, A. Kennedy, and B. Pendleton, editors, *QCD and Numerical Analysis III*, volume 47 of *Lecture Notes in Computational Science and Engineering*, pages 15–24. Springer-Verlag, Berlin, 2005.

G. E. Forsythe.
Pitfalls in computation, or why a math book isn't enough.
Amer. Math. Monthly, 77(9):931–956, 1970.

References III


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


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


N. J. Higham and A. H. Al-Mohy.
Computing matrix functions.
Acta Numerica, 19:159–208, 2010.

References IV


N. J. Higham and F. 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.


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


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

References V

Y. Saad.
Analysis of some Krylov subspace approximations to the matrix exponential operator.
SIAM J. Numer. Anal., 29(1):209–228, Feb. 1992.

L. N. Trefethen, J. A. C. Weideman, and T. Schmelzer.
Talbot quadratures and rational approximations.
BIT, 46(3):653–670, 2006.