MANCHESTER 1824

The University of Manchester

A Schur–Padé Algorithm for Fractional Powers of a Matrix

Nick Higham
School of Mathematics
The University of Manchester

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

Joint work with Lijing Lin

Dynamical Systems, Control and Optimization
Paul Van Dooren 60
Oct. 2010

MATLAB: Exponentiation

```
>> A = [1 2; 1 1];

>> 2.^A
ans =
       2       4
       2       2

>> 2^A
ans =
   3.0404e+000   3.2384e+000
   1.6192e+000   3.0404e+000

>> expm( log(2)*A )
ans =
   3.0404e+000   3.2384e+000
   1.6192e+000   3.0404e+000
```

MATLAB: Fractional Powers

```
>> A = [1 1e-8; 0 1]
A =
   1.0000e+000   1.0000e-008
             0   1.0000e+000

>> A^0.1
ans =
      1     0
      0     1

>> expm(0.1*logm(A))
ans =
   1.0000e+000   1.0000e-009
             0   1.0000e+000
```

MATLAB: Integer Powers

```
>> e = 1e-1; A = [1 1+e; 1-e 1]
A =
    1.0000    1.1000
    0.9000    1.0000

>> X = A^(-3); Y = inv(A)^3;

>> Z = double(vpa(A)^(-3));
>> norm(X-Z,1)/norm(Z,1), norm(Y-Z,1)/norm(Z,1)
ans =
  1.2412e-009
ans =
  6.6849e-016
```

Arbitrary Power

Definition

For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ and $p \in \mathbb{R}$, $A^p = e^{p \log A}$, where $\log A$ is the principal logarithm.

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

Arbitrary Power

Definition

For $A \in \mathbb{C}^{n\times n}$ with no eigenvalues on $\mathbb{R}^-$ and $p \in \mathbb{R}$, $A^p = e^{p \log A}$, where $\log A$ is the principal logarithm.

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

Applications:

- Pricing American options (Berridge & Schumacher, 2004).
- Finite element discretizations of fractional Sobolev spaces (Arioli & Loghin, 2009).
- Computation of geodesic-midpoints in neural networks (Fiori, 2008).

Email from a Power Company

The problem has arisen through proposed methodology on which the company will incur charges for use of an electricity network.

$\vdots$

I have the use of a computer and Microsoft Excel.

$\vdots$

I have an Excel spreadsheet containing the transition matrix of how a company's [Standard & Poor's] credit rating changes from one year to the next. I'd like to be working in eighths of a year, so the aim is to find the ***eighth root of the matrix****.*

HIV to Aids Transition

- Estimated 6-month transition matrix.
- Four AIDS-free states and 1 AIDS state.
- 2077 observations (Charitos et al., 2008).

$$P = \begin{bmatrix} 0.8149 & 0.0738 & 0.0586 & 0.0407 & 0.0120 \\ 0.5622 & 0.1752 & 0.1314 & 0.1169 & 0.0143 \\ 0.3606 & 0.1860 & 0.1521 & 0.2198 & 0.0815 \\ 0.1676 & 0.0636 & 0.1444 & 0.4652 & 0.1592 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Want to estimate the **1-month transition matrix**.

$$\Lambda(P) = \{1, 0.9644, 0.4980, 0.1493, -0.0043\}.$$

N. J. Higham and L. Lin.
On *p*th roots of stochastic matrices, LAA, 2010.

Basic Idea

Assume $p \in (-1, 1)$, A has no nonpositive real ei'vals.

- For suitable p, write

$$\boxed{A^p = (A^{1/2^k})^{p \cdot 2^k}} \equiv (I - X)^{p \cdot 2^k}.$$

- Choose suitable Padé approximant $r_m(x)$ to $(1 - x)^p$.
- Compute $r_m(X)^{2^k}$.

Questions:

- Existence of r_m.
- Choice of k and m.
- How to evaluate r_m.
- How to handle general $p \in \mathbb{R}$ $(A^{3.4} = A^3 A^{0.4} = A^4 A^{-0.6})$.

Evaluating r_m

$$r_m(x) = 1 + \cfrac{c_1 x}{1 + \cfrac{c_2 x}{1 + \cfrac{c_3 x}{\cdots \cfrac{}{1 + \cfrac{c_{2m-1} x}{1 + c_{2m} x}}}}},$$

$$c_1 = -p, \quad c_{2j} = \frac{-j+p}{2(2j-1)}, \quad c_{2j+1} = \frac{-j-p}{2(2j+1)}, \quad j = 1, 2, \ldots .$$

Bottom-up evaluation:

$Y_{2m} = c_{2m} X$
for $j = 2m-1\colon -1\colon 1$
    Solve $(I + Y_{j+1}) Y_j = c_j X$ for Y_j
end
$r_m = I + Y_1$

Choice of m

Variation of result of Kenney & Laub (1989).

Theorem (H & Lin, 2010)

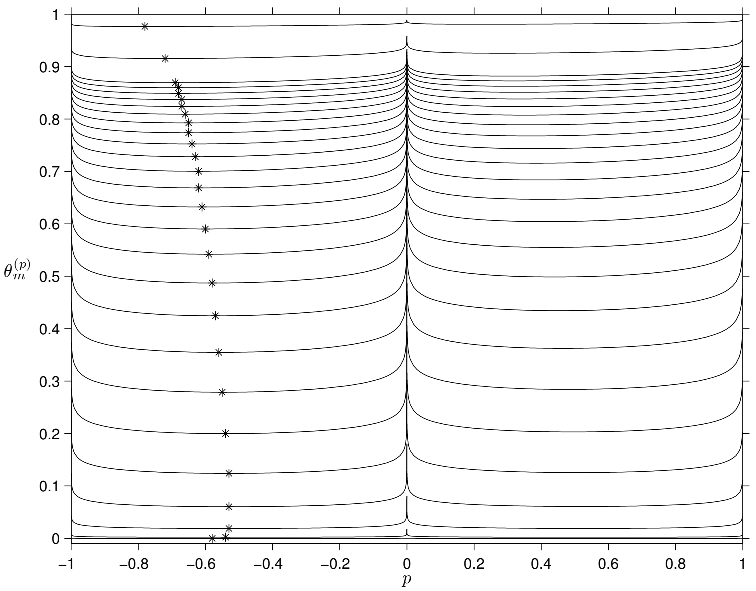
For $p \in (-1,1)$ and $\|X\| < 1$,

$$\|(I-X)^p - r_m(X)\| \le |(I - \|X\|)^p - r_m(\|X\|)|.$$

Let

$$\theta_m^{(p)} := \max\{\, \|X\| : |(I - \|X\|)^p - r_m(\|X\|)| \le u \,\}.$$

$\theta_m^{(p)}$ for $p \in (-1, 1)$, $m \in [1, 64]$



Strategy

- Initial Schur decomposition.
- Keep taking square roots until $\|I - T^{1/2^k}\| \le \theta_7 = 0.279$.

  Logic based on

  $$\|I - T^{1/2}\| = \|(I + T^{1/2})^{-1}(I - T)\| \approx \tfrac{1}{2}\|I - T\|.$$

- Squaring phase forms $r_m(T^{1/2^k})^{2^j} \approx (I - T^{1/2^k})^{p/2^{k-j}}$, $j = 1\colon k$.
  - Compute diagonal and first superdiagonal **exactly!**

Diag and Superdiag

$$F = \begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix}^p = \begin{bmatrix} \lambda_1^p & t_{12}(\lambda_2^p - \lambda_1^p)/(\lambda_2 - \lambda_1) \\ 0 & \lambda_2^p \end{bmatrix}.$$

Diag and Superdiag

$$F = \begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix}^p = \begin{bmatrix} \lambda_1^p & t_{12}(\lambda_2^p - \lambda_1^p)/(\lambda_2 - \lambda_1) \\ 0 & \lambda_2^p \end{bmatrix}.$$

$$f_{12} = t_{12} \exp\left(\tfrac{p}{2}\left(\log\lambda_2 + \log\lambda_1\right)\right) \times \frac{2\sinh\left(p\left(\operatorname{atanh}(z) + \pi i\,\mathcal{U}(\log\lambda_2 - \log\lambda_1)\right)\right)}{\lambda_2 - \lambda_1},$$

where $z = (\lambda_2 - \lambda_1)/(\lambda_2 + \lambda_1)$,
$\mathcal{U}(z)$ is the **unwinding number** of $z \in \mathbb{C}$,

$$\mathcal{U}(z) := \frac{z - \log(e^z)}{2\pi i} = \left\lceil \frac{\operatorname{Im} z - \pi}{2\pi} \right\rceil \in \mathbb{Z}.$$

General $p \in \mathbb{R}$

$$\begin{aligned} p &= \lfloor p \rfloor + p_1, \qquad & p_1 > 0, \\ p &= \lceil p \rceil + p_2, \qquad & p_2 < 0. \end{aligned}$$

If A is Hermitian positive definite then

$$\kappa_{x^p}(A) \gtrsim \begin{cases} |p|\kappa_2(A)^{1-p}, & p \ge 0, \\ |p|\kappa_2(A), & p \le 0, \end{cases}$$

where $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \lambda_1/\lambda_n$.

- Choose p_1 if $\kappa_2(A) \ge \exp(p_1^{-1}\log\bigl(p_1/(1-p_1)\bigr))$.
- For general A, use $\kappa_2(A) \ge \max_i |t_{ii}|/\min_i |t_{ii}|$.

A^p for $p = -k \in \mathbb{Z}^-$

Power then invert.

$Y = A^k$ by binary powering
$X = Y^{-1}$ via GEPP

Invert then power.

$Y = A^{-1}$ via GEPP
$X = Y^k$ by binary powering

Repeated triangular solves.

Compute a factorization $PA = LU$ by GEPP.
$X_0 = I$
for $i = 0\colon k-1$
    Solve $LX_{i+1/2} = PX_i$
    Solve $UX_{i+1} = X_{i+1/2}$
end
$X = X_k$

Numerical Experiments

`powerm` essentially

```
[V,D] = eig(A);
X = V*diag(diag(D).^p)/V;
```

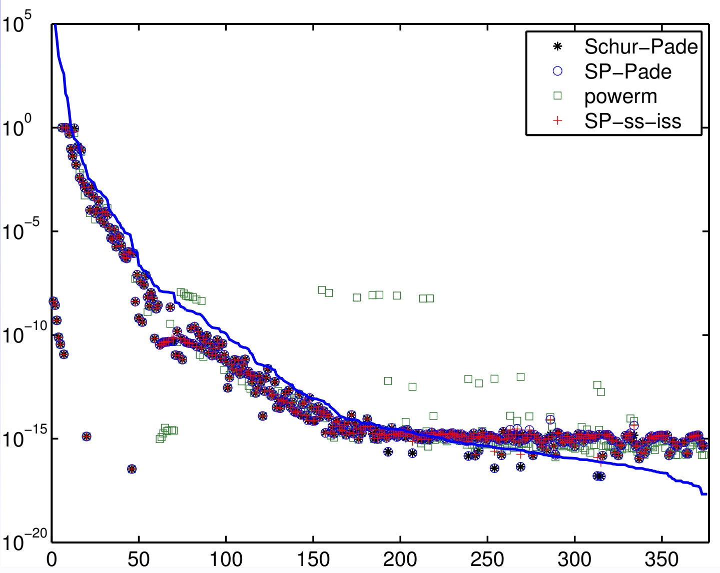
`Schur-Pade` New algorithm.

`SP-Pade:` Mod. of `funm` using `Schur-Pade` on the diagonal blocks.

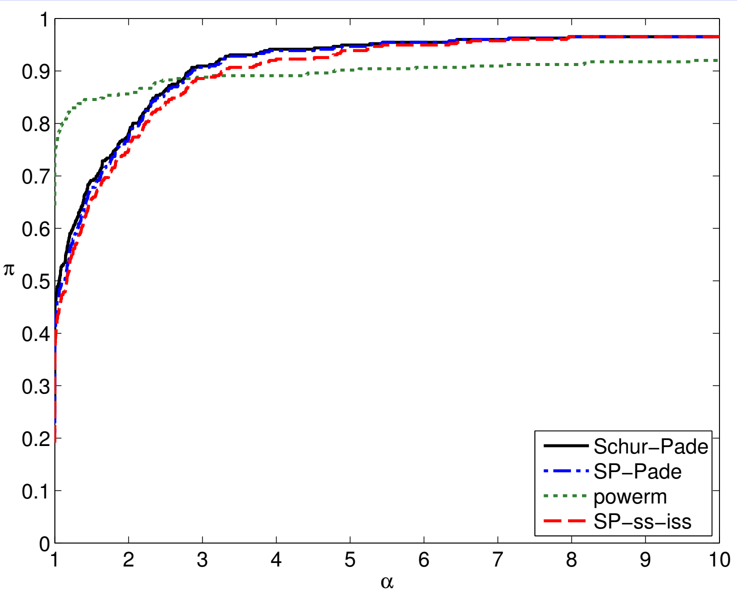
`SP-ss-iss` Mod. of `funm` with evaluation of $\exp(p \log(T_{ii}))$ by ISS for log and SS for exp.

Experiment 1: Relative Errors

195 10×10 matrices, $p \in \{1/52,\ 1/12,\ 1/3,\ 1/2\}$.

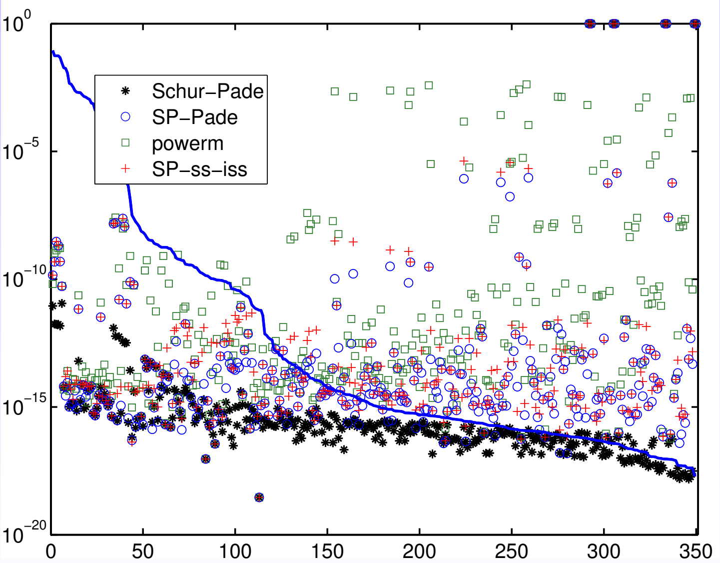


Performance Profile

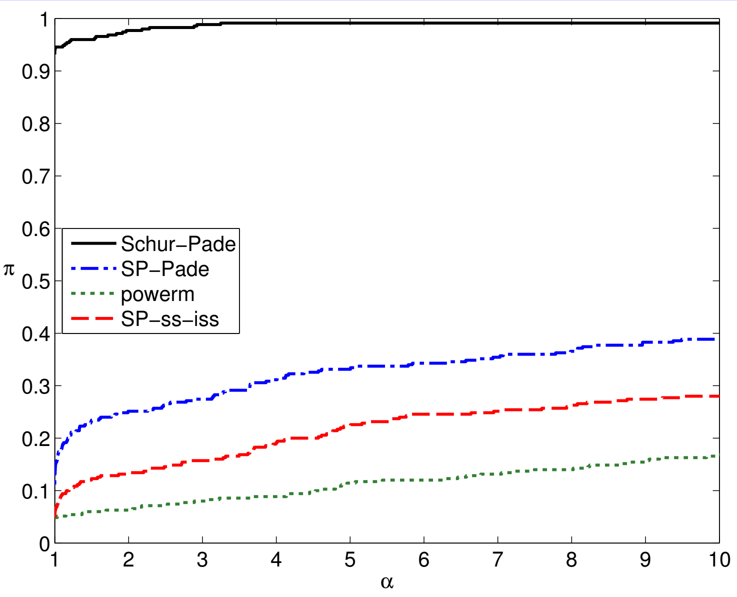


Experiment 2: Relative Errors

Triangular QR factors of previous test set, $r_{ii} \leftarrow |r_{ii}|$.

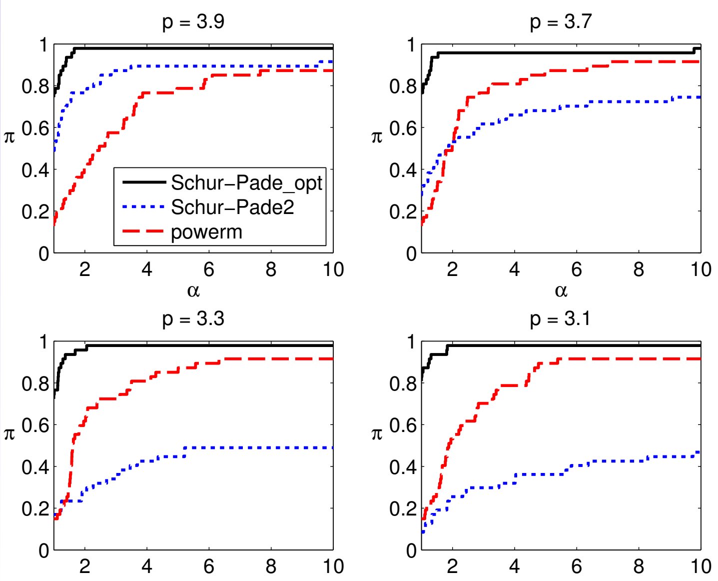


Performance Profile



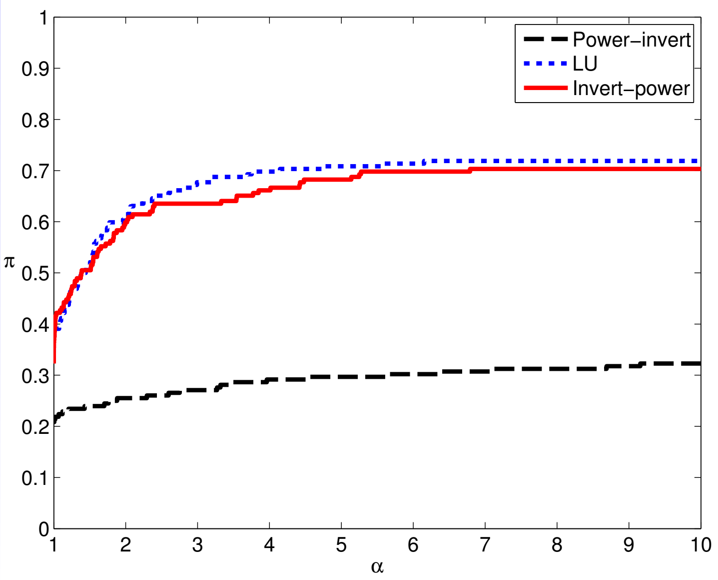
Experiment 3: $p \notin (-1, 1)$

Same matrices; $p = 3.9, 3.7, 3.3, 3.1$.



Experiment 4: Negative Integer *p*

Same matrices; $p = -3, -5, -7, -9$.



Conclusions

- MATLAB `A^p` is unreliable
  - for fractional p due to use of eigendecomposition,
  - for negative integer p due to “power then invert”.
- New Schur–Padé alg works for any $p \in \mathbb{R}$.
  - Schur decomposition,
  - square roots and Padé approximant,
  - choice of parameters balancing speed & accuracy,
  - exact computation of diag and superdiag,
  - superior to direct use of $\exp(p \log(A))$.
- Various alternative methods available when $p^{-1} \in \mathbb{Z}$.

N. J. Higham and L. Lin.
MIMS EPrint 2010.91, October 2010.

References I

M. Arioli and D. Loghin.
Discrete interpolation norms with applications.
SIAM J. Numer. Anal., 47(4):2924–2951, 2009.

S. Berridge and J. M. Schumacher.
An irregular grid method for high-dimensional free-boundary problems in finance.
Future Generation Computer Systems, 20:353–362, 2004.

T. Charitos, P. R. de Waal, and L. C. van der Gaag.
Computing short-interval transition matrices of a discrete-time Markov chain from partially observed data.
Statistics in Medicine, 27:905–921, 2008.

References II

S. Fiori.
Leap-frog-type learning algorithms over the Lie group of unitary matrices.
Neurocomputing, 71:2224–2244, 2008.

N. J. Higham.
The Matrix Function Toolbox.
`http://www.ma.man.ac.uk/~higham/mftoolbox.`

References III

- 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 L. Lin.
A Schur–Padé algorithm for fractional powers of a matrix.
MIMS EPrint 2010.91, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, Oct. 2010.
25 pp.

C. S. Kenney and A. J. Laub.
Padé error estimates for the logarithm of a matrix.
Internat. J. Control, 50(3):707–730, 1989.