MODELLING FOR ADDICTIVE BEHAVIOUR, MEDICINE AND ENGINEERING 2010

Instituto de Matemática Multidisciplinar

Edited by: L. Jódar, Instituto de Matemática Multidisciplinar

im2
Instituto de Matemática Multidisciplinar

UNIVERSIDAD POLITECNICA DE VALENCIA

GENERALITAT VALENCIANA

MODELLING FOR ADDICTIVE BEHAVIOUR, MEDICINE AND ENGINEERING 2010

Instituto de Matemática Multidisciplinar
Universidad Politécnica de Valencia
Valencia 46022, SPAIN

Edited by
Lucas Jódar,
Instituto de Matemática Multidisciplinar, Director
I.S.B.N.: 978-84-693-9537-0

Computing matrix exponential to solve coupled differential models in Engineering*

J. Sastre+ †, J. Javier Ibáñezo, Emilio Defez‡ and Pedro Ruizo

(+) iTEAM, (o) I3M, (‡) IMM, Universidad Politécnica de Valencia.

October 10, 2010

1 Introduction

Many scientific and engineering processes are described by systems of linear first-order ordinary differential equations with constant coefficients, whose exact solution is given in terms of the matrix exponential, and a large number of methods for its computation have been proposed [1, 2]. This paper presents the key ideas for a competitive new scaling and squaring algorithm. Throughout this paper $\mathbb{C}^{n\times n}$ denotes the set of complex matrices of size $n\times n$, I denotes the identity matrix for this set, $\rho(A)$ is the spectral radius of matrix A, and $\mathbb{N}$ denotes the set of positive integers. The matrix norm $\|\cdot\|$ denotes any subordinate matrix norm, in particular $\|\cdot\|_1$ is the 1-norm. Next theorem will be used in next section to bound the norm of matrix power series.

Theorem 1.1 *Let $h_l(x)=\sum_{k\geq l} b_k x^k$ be a power series with radius of convergence w, and let $\tilde{h}_l(x)=\sum_{k\geq l}|b_k|x^k$. For any matrix $A\in\mathbb{C}^{n\times n}$ with $\rho(A)<w$, and $p\in\mathbb{N}$, $p\geq 1$, if a_k is an upper bound for $||A^k||$ ($||A^k||\leq a_k$), and $\alpha_p=\max\{(a_k)^{\frac{1}{k}}; k=p,l,l+1,\ldots,l+p-1\}$, then $||h_l(A)||\leq \tilde{h}_l(\alpha_p)$.*

Proof. $||h_l(A)||\leq \sum_{j\geq 0}\sum_{i=l}^{l+p-1}|b_{i+jp}|||A^p||^j||A^i||\leq \sum_{j\geq 0}\sum_{i=l}^{l+p-1}|b_{i+jp}|\alpha_p^{i+pj}=\sum_{k\geq l}|b_k|\alpha_p^k=\tilde{h}_l(\alpha_p)$. □

*This work has been partially supported by the *Universidad Politécnica de Valencia* under the grants PAID-05-09-4338 and PAID-06-08-3307, and by the Spanish *Ministerio de Educación* under the grant MTM2009-08587.

†e-mail:jorsasma@iteam.upv.es

2 Error analysis and algorithm

If we denote $T_m(A) = \sum_{i=0}^{n} A^i/i!$ the truncated matrix exponential Taylor series with Taylor remainder $R_m(A)$, for a scaled matrix $2^{-s}A$ we can write

$$\left(T_m(2^{-s}A)\right)^{2^s} = e^A \left(I + g_{m+1}(2^{-s}A)\right)^{2^s} = e^{A+2^s h_{m+1}\left(2^{-s}A\right)}, \quad s \in \mathbb{N} \cup \{0\}, \tag{1}$$

$$g_{m+1}(2^{-s}A) = -e^{-2^{-s}A} R_m(2^{-s}A),\ h_{m+1}\left(2^{-s}A\right) = \log\left(I + g_{m+1}(2^{-s}A)\right), \tag{2}$$

see [4, sec. 3], where log denotes the principal logarithm and $h_{m+1}(X)$ is defined in the set $\Omega_m = \left\{X \in \mathbb{C}^{n\times n} : \rho\left(e^{-X}T_m(X) - I\right) < 1\right\}$. If we choose s so that $2^{-s}A \in \Omega_m$, then from (1) one gets that $\Delta A = 2^s h_{m+1}\left(2^{-s}A\right)$ and $\Delta E = e^A[(I + g_{m+1}\left(2^{-s}A\right))^{2^s} - I]$ represent the backward and forward errors in exact arithmetic from the approximation of e^A by Taylor series with scaling and squaring, respectively. If s is chosen so that

$$\left\|h_{m+1}\left(2^{-s}A\right)\right\| \le \max\left\{1, ||2^{-s}A||\right\} u, \tag{3}$$

where $u = 2^{-53}$ is the unit roundoff in IEEE double precision arithmetic, then: if $2^{-s}\|A\| \ge 1$, then $\Delta A \le 2^{-s}\|A\| u$ and using (1) one gets $\left(T_m\left(2^{-s}A\right)\right)^{2^s} = e^{A+\Delta A} \approx e^A$, and if $2^{-s}\|A\| < 1$, using (1),(2),(3) and Taylor series one gets

$$\begin{aligned}\left\|R_m\left(2^{-s}A\right)\right\| &= \left\|e^{2^{-s}A} g_{m+1}\left(2^{-s}A\right)\right\| = \left\|e^{2^{-s}A}\left(e^{h_{m+1}\left(2^{-s}A\right)} - I\right)\right\| \\ &= \left\|e^{2^{-s}A}\sum_{k\ge1}\left(h_{m+1}\left(2^{-s}A\right)\right)^k/k!\right\| \le \left\|e^{2^{-s}A}\right\|\sum_{k\ge1} u^k/k! \\ &\approx \left\|T_m\left(2^{-s}A\right)\right\| u\left(1+u/2!+u^2/3!+\cdots\right) \approx \left\|T_m\left(2^{-s}A\right)\right\| u.\end{aligned} \tag{4}$$

Hence, as we are evaluating explicitly $T_m\left(2^{-s}A\right)$, in IEEE double precision arithmetic $T_m\left(2^{-s}A\right) + R_m\left(2^{-s}A\right) \approx T_m\left(2^{-s}A\right)$, and there is no point in increasing m or the scaling to try to get better accuracy. From (2) one gets

$$h_{m+1}\left(2^{-s}A\right) = \sum_{k\ge m+1} b_k^{(m)}\left(2^{-s}A\right)^k, \tag{5}$$

and using MATLAB symbolic Math Toolbox, 200 terms, high precision arith. and a zero finder we obtained the maximal values $\Theta_m = \|2^{-s}A\|$ such that

$$||h_{m+1}\left(2^{-s}A\right)|| \le \tilde{h}_{m+1}\left(||2^{-s}A||\right) = \tilde{h}_{m+1}\left(\Theta_m\right) \le \max\left\{1, \Theta_m\right\} u. \tag{6}$$

We have applied Horner's method to Paterson-Stockmeyer method for the evaluation of matrix polynomial $T_m\left(2^{-s}A\right)$ [2, p. 72-74], modifying it as

$$\begin{aligned}T_m\left(2^{-s}A\right) =& \left(\left(\cdots\left(\frac{A_j}{2^s m} + A_{j-1}\right)/(2^s(m-1)) + A_{j-2}\right)/(2^s(m-2)) + \cdots + A_2\right)/(2^s(m-j+2)) \\ &+ A + 2^s(m-j+1)I)\,\frac{A_j}{2^{2s}(m-j+1)(m-j)} + A_{j-1}\Big)\,/\,(2^s(m-j-1)) \\ &+ A_{j-2})\,/\,(2^s(m-j-2)) + \cdots + A_2)\,/\,(2^s(m-2j+2)) + A \\ &+ 2^s(m-2j+1)I)\,\frac{A_j}{2^{2s}(m-2j+1)(m-2j)} + \cdots + A_2\Big)/(2^s 2) + A + 2^s I\Big)/2^s,\end{aligned} \tag{7}$$

where $A_i = A^i$ are computed as $A_2 = A^2$, $A_4 = A_2^2, \ldots, A_{2k+1} = A_{2k}A$, and we will use a subset of optimal values of m in terms of matrix products, see Table 4.1 of [2, p.74], $m = 4, 6, 9, 12, 16, 20, 25, 30$, with $j = 2, 3, 3, 4, 4, 5, 5, 5$ respectively. (7) saves $O(n^2)$ operations with respect to classic Paterson-Stockmeyer Horner's form and avoids factorials improving numerical results [5]. Similar floating point bounds to those in [5] are applied to the intermediate results in $T_m(2^{-s}A)$ to save matrix products. Then, the scaling algorithm will be as follows: Estimate $\|A^{m+1}\|_1$ using the $O(n^2)$ algorithm of [6]. Then, use Theorem 1.1 and (6), calculating the necessary bounds a_k for $\|A^k\|_1$ using the known matrix power norms, to obtain the initial maximum matrix scaling s_0. Then, try if (3) is satisfied using the bounds for $\|A^k\|_1 \leq a_k$ in

$$\left\|h_{m+1}\left(2^{-s}A\right)\right\|_1 \leq \sum_{k\geq m+1} |b_k^{(m)}| \frac{\|A^k\|_1}{2^{sk}} \approx \sum_{k=m+1}^{m+j+M} \left|b_k^{(m)}\right| \frac{\|A^k\|_1}{2^{sk}} \leq \sum_{k=m+1}^{m+j+M} \left|b_k^{(m)}\right| \frac{a_k}{2^{sk}}, \tag{8}$$

with $s = s_0 - 1$, choosing $M \geq 1$. If it is not satisfied, try then

$$\begin{aligned}\left\|h_{m+1}\left(2^{-s}A\right)\right\|_1 \leq & \left\|\frac{b_{m+1}^{(m)}}{b_{m+2}^{(m)}} 2^s I + A + \frac{b_{m+3}^{(m)}}{b_{m+2}^{(m)}} \frac{A_2}{2^s} + \frac{b_{m+4}^{(m)}}{b_{m+2}^{(m)}} \frac{A_3}{2^{2s}} + \cdots + \frac{b_{m+j}^{(m)}}{b_{m+2}^{(m)}} \frac{A_j}{2^{s(j-1)}}\right\| \\ & \times \frac{\|A^{m+1}\|_1}{2^{s(m+2)}} \left|b_{m+2}^{(m)}\right| + \sum_{k=m+j+1}^{m+j+M} \left|b_k^{(m)}\right| \frac{\|A^k\|_1}{2^{sk}}\end{aligned} \tag{9}$$

(9) is lower or equal than (8) for normal matrices and low bounds for it can be obtained to avoid unnecessary evaluations. Repeat the process with $s = s_0 - 2, s_0 - 3, \ldots$ If the last scaling s where (8) or (9) satisfy (3) is $s \geq 1$ then try if s and previous optimal m also satisfy (3). Return s and the minimum m satisfying (3). The total algorithm consists of using Theorem 1.1, (8) and (9) to try if one of the orders $m = 4, 6, 9, \ldots, m_{max}$ satisfy (3) with $s = 0$, where m_{max} is the max. allowed order. If not, obtain the scaling s using previous algorithm and use (7) and squaring to evaluate $\left(T_m\left(2^{-s}A\right)\right)^{2^s}$.

3 Numerical experiments

133 matrices from [1, 3], MATLAB gallery, and others have been used to compare MATLAB functions `expm` [3] and `expm_new` [4] with an implementation of our algorithm, `dgeexftay`. Table 1 shows that `dgeexftay` average matrix product number is lower than `expm`'s, and slightly greater than `expm_new`, and that `dgeexftay` is more accurate in the majority of cases. Normwise and performance profile figures [3] have shown that all functions perform in a numerically stable way on this test and that `dgeexftay` has better precision performance than `expm` and `expm_new` even since maximum allowed Taylor order $m_{max} = 16$. Now we are applying the new algorithm to Padé method.

References

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

Table 1: Relative error $\mathrm{E} = ||e^A - \tilde{X}||_1/||e^A||_1$, and matrix product number (%) comparison between `dgeexftay`, `expm` and `expm_new`.

Maximum allowed Taylor order m_{max}	16	20	25	30
$E_{\mathtt{dgeexftay}} < E_{\mathtt{expm}}$	74.44	90.98	89.47	88.72
$(P_{\mathtt{dgeexftay}} - P_{\mathtt{expm}})/P_{\mathtt{expm}}$	-15.47	-15.69	-14.95	-14.35
$E_{\mathtt{dgeexftay}} < E_{\mathtt{expm\_new}}$	66.17	87.22	87.22	86.47
$(P_{\mathtt{dgeexftay}} - P_{\mathtt{expm\_new}})/P_{\mathtt{expm\_new}}$	1.31	1.04	1.94	2.65

[2] N.J. Higham. Functions of Matrices: Theory and Computation. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008.

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

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

[5] J. Sastre, J. Ibáñez, E. Defez and P. Ruiz. Efficient scaling-squaring Taylor method for computing matrix exponential. Accepted with modifications in SIAM J. on Scientific Computing.

[6] 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., 21:1185-1201, 2000.