Scaling and modified squaring method for the matrix exponential

Shinsuke Nakamura¹, Kazufumi Ozawa¹ and Chiaki Hirota¹

¹ Faculty of Systems Science and Technology, Akita Prefectural University, 84-4 Aza, Ebinokuchi, Tsuchiya, Yurihonjo, Akita 015-0055, Japan
E-mail shinsuke@akita-pu.ac.jp
Received March 15, 2016, Accepted July 18, 2016

Abstract
In recent years, many discrete variable methods using the exponential operator, called exponential integrator, have been presented, and various computational methods of the matrix exponential are also proposed. Especially the combination of the Padé approximant and the scaling and squaring method is most powerful and widely used. However, the squaring process is susceptible to roundoff errors. We propose a modified squaring process for the scaling and squaring method. From the numerical results, an accuracy improvement about 1/100 is obtained in the best case.

Keywords matrix exponential, scaling and squaring, roundoff error analysis


1. Introduction
The matrix exponential $\exp(A)$ is particularly useful in the numerical analysis of linear ordinary equations. Various computational techniques for $\exp(A)$ were proposed, such as series methods, Padé approximation methods, characteristic polynomial methods, and so on [1]. Usually these methods were used with scaling and squaring technique to accelerate the convergences.

Higham [2] analyzed the scaling and squaring technique based on the Padé approximation and evaluated its efficiency.

In this letter, we propose a modified squaring technique for improving the accuracy of the scaling and squaring process of the Padé approximant.

2. Algorithms for matrix exponential
The scaling and squaring method combined with the diagonal Padé approximation is the most widely used algorithm for the computation of the matrix exponential [2], and is implemented as MATLAB function $\text{expm}$ [1]. In this section we review the algorithm.

2.1 Padé approximation
Among all the entries of the Padé approximants to $\exp(x)$, the diagonal ones are the most frequently used approximants. The diagonal approximant of degree $m$ is given by

$$ r_m(x) = \frac{p_m(x)}{q_m(x)}, \quad (1) $$

where

$$ p_m(x) = \sum_{j=0}^{k} \frac{(2m-j)!}{(2m)!} \frac{x^j}{(m-j)!} j!, \quad q_m(x) = q_m(-x). \quad (2) $$

The truncation error of the approximant is given by $r_m(x) - \exp(x) = O(x^{2m+1})$. The approximant satisfies

$$ r_m(x) = 1/r_m(-x), $$

which shares the property that $\exp(x) = 1/\exp(-x)$ [3].

Moreover, the approximant has the well-known favorite property that for any complex $z$ with $\text{Re}(z) \leq 0$, that is, for any $z$ satisfying $|\exp(z)| \leq 1$, $r_m(z)$ also satisfies $|r_m(z)| \leq 1$ (see e.g. [3]).

By replacing the scalar argument $x$ with a matrix argument $A$ in (1), we have

$$ r_m(A) = \{q_m(A)^{-1}p_m(A)\}. \quad (3) $$

To accelerate the convergence of the Padé approximant, the technique called scaling and squaring is usually used. We will describe it briefly in the next subsection.

2.2 Scaling and Squaring of Padé approximant
First we choose a sufficiently large integer $\sigma > 0$ so that $A/\sigma$ has a norm of order 1 or less, then compute the approximant to $\exp(A/\sigma)$, and next compute its $\sigma$th power. If we take an integer $s > 0$ and set $\sigma = 2^s$, then we can easily raise the power by $s$ repeated squarings, since $\exp(A) = \{\exp(A/2^s)\}^{2^s}$. This process is summarized in Algorithm 1. In this algorithm and the modified algorithm to be described later, we will call the index $i$ scaling level.

To keep the truncation error of $r_m(A/2^s)$ as small as possible with minimal cost, Higham [2] proposed a criterion for choosing the initial scaling level $s$ and the degree $m$ of the approximant. Here we describe it briefly.

Let $G$ be the perturbation to $A$ which satisfies the relation

$$ \{r_m(A/2^s)\}^{2^s} = \exp(A + G). \quad (6) $$
Algorithm 1 Scaling and squaring method [2]

Scaling: Choose appropriate integers $s$ and $m$, and compute an approximation to \( \exp(A/2^s) \) by the Padé approximant and set it \( E_s := r_m(A/2^s) \).

\[
E_s := r_m(A/2^s). \tag{4}
\]

Squaring: Repeat the iteration

\[
E_{i-1} = E_i^2, \quad i = s, s - 1, \ldots, 1
\]

to obtain \( E_0 \approx \exp(A) \).

For several \( m \), Higham calculates the bound for \( \|A/2^s\| \), say \( \theta_m \), such that

\[
\text{if } 2^{-s}\|A\| \leq \theta_m, \quad \text{then } \|G\| \leq u\|A\|,
\]

where \( u \) is the unit roundoff of IEEE 754 double precision arithmetic, and obtain the number of matrix multiplications, say \( M_m \), for evaluating \( p_m \) and \( q_m \). Using this information, Higham concluded that \( m = 13 \) is the best choice in the sense that the total cost \( s + M_m \) is minimum at \( m = 13 \).

In this letter, we consider the influence of the rounding errors in (5), and propose a modified algorithm.

3. Modified squaring

Higham [4] pointed out that squaring of matrices is an unstable process. Therefore in [2] the strategy to lower the scaling level has been adopted. However the strategy is almost the same as that of Algorithm 1.

Consider the calculation of \( (1 + \varepsilon)^{2^s} \) by \( s \) repeated squarings of \( (1 + \varepsilon) \). For a finite precision floating-point arithmetic, if \( \varepsilon \) is a small number such that \( 0 < |\varepsilon| \ll 1 \), then a part of the information on \( 2\varepsilon + \varepsilon^2 \) is lost due to the “large” constant 1. To avoid such a loss of accuracy, consider the iteration

\[
\varepsilon_{i-1} = \varepsilon_i^2 + 2\varepsilon_i, \quad \varepsilon_s = \varepsilon, \quad i = s, \ldots, 1, \tag{7}
\]

and finally compute \( 1 + \varepsilon_0 \), which is the required quantity. We call this algorithm modified squaring. Brent [5] proposed to use this technique to compute \( \exp(x) \) by setting \( \varepsilon = \exp(2^{-s}x) - 1 \) (see [5, Exercise 4.8]).

For the case of the matrix exponential, if the scaling level \( s \) is properly chosen, then we can expect that

\[
\varepsilon_i = r_m(A/2^s) \approx I + A/2^s,
\]

and \( |a_{ii}/2^s| \ll 1 \), where \( a_{ii} \) are the diagonal elements of \( A \). If we square \( r_m(A/2^s) \) then a similar phenomenon described above might occur at the diagonal elements. We extend the modified squaring to the matrix exponential.

3.1 Modified squaring for matrix case

Here we introduce the new symbols as

\[
E_i^1 := E_i - I, \quad \exp^1(A) := \exp(A) - I.
\]

Using these we present the modified algorithm in Algorithm 2. First we consider the cost of the modified algorithm.

Algorithm 2 Scaling and modified squaring method

Scaling: Choose appropriate integers \( s \) and \( m \) and evaluate \( E_s(r = E_s - I) \).

Modified Squaring: For \( i = s, s - 1, \ldots, 1 \), repeat

\[
E_{i-1}^1 = E_i^1 \cdot E_i^1 + 2E_i^1 \approx \exp^1(A/2^{i-1}). \tag{8}
\]

Postprocessing: \( E_0 = E_0^1 + I \).

Since \( p_m(x) = q_m(-x) \), we have

\[
p_m(x) = u_m(x) + v_m(x), \quad q_m(x) = -u_m(x) + v_m(x),
\]

where \( u_m \) and \( v_m \) are the sums of odd and even powers of \( x \) in \( p_m \), respectively. From this relation, we have for \( E_s^1 \)

\[
E_s^1 = r_m(A/2^s) - I
\]

\[
= \{q_m(A/2^s)\}^{-1}(p_m(A/2^s) - q_m(A/2^s))
\]

\[
= 2\{q_m(A/2^s)\}^{-1}u_m(A/2^s) \tag{10}
\]

Higham [2] proposed an efficient method to compute \( p_m(A) \) and \( q_m(A) \) by separating these into \( u_m \) and \( v_m \). Using the method, we can evaluate the two polynomials \( p_m \) and \( q_m \) by about \( m/2 \) matrix multiplications. For the present case, since \( E_s^1 \) is constructed only from \( u_m \) and \( v_m \), we can also apply Higham’s technique for \( E_s^1 \) with the same cost. After the calculation of \( E_s^1 \) we repeat (8), which needs one matrix multiplication per iteration as in (5). Therefore, the total cost of the modified algorithm is almost the same as that of Algorithm 1.

In the next section, we analyze the roundoff errors introduced in the modified algorithm.

4. Rounding error analysis

For the roundoff analysis we introduce new symbols. Let \( \hat{E}_{i} \) and \( \hat{E}_{i}^1 \) be the computed matrices corresponding to \( E_{i} \) and \( E_{i}^1 \), respectively, and let \( \Delta_i \) and \( \Delta_i^1 \) be the errors of these matrices, that is,

\[
\Delta_i := \hat{E}_i - E_i, \quad \Delta_i^1 := \hat{E}_i^1 - E_i^1, \quad i = 0, \ldots, s.
\]

When evaluating the bounds for rounding errors, the two constants

\[
\gamma_k = \frac{ku}{1 - ku}, \quad \tilde{\gamma}_k = \frac{cku}{1 - cku}
\]

are often used, where \( c \) is a small integer constant whose exact value is unimportant. To estimate the roundoff errors \( \Delta_i \) and \( \Delta_i^1 \), the following two theorems on the errors in Gaussian elimination and an evaluation of polynomials are useful.

Theorem 1 (Higham [6]) Let \( A \in \mathbb{R}^{n \times n} \) and suppose that Gaussian elimination produces the computed LU factors \( \hat{L}, \hat{U} \), and the computed solution \( \hat{x} \) to \( \hat{A}\hat{x} = \hat{b} \). Then,

\[
(A + \Delta A)\hat{x} = \hat{b}, \quad \|\Delta A\| \leq \gamma_{\operatorname{nn}}\|\hat{L}\|\|\hat{U}\|,
\]

where \( \| \cdot \| \) denotes the matrix or vector with the elements replaced by their absolute values.
Theorem 2 (Higham [4]) Let \( \hat{p}_m(X) \) be the computed polynomial for \( p_m(X) = \sum_{k=0}^{m} b_k X^k \). For the evaluation of the polynomial, if we use Horner’s rule, explicit powers, or the Paterson–Stockmeyer method, then the bound for the error is given by

\[
|p_m - \hat{p}_m| \leq \gamma_m \hat{p}_m(|X|),
\]

where \( \hat{p}_m(X) = \sum_{k=0}^{m} b_k X^k \).

Hereafter we put \( X = A/2^n \) for brevity. Applying these two theorems we have the following result.

Theorem 3 The roundoff errors of the Padé approximants satisfy

\[
|\Delta_s| \leq T_m(|X|) \exp(|X|), \quad |\Delta_s^+| \leq T_m(|X|) \exp^+(|X|),
\]

where

\[
T_m(|X|) = |q_m(|X|)^{-1} \{(2 \gamma_{mn} v_m(|X|)) + \gamma_{3n} [\tilde{L}][\tilde{U}]\} |,
\]

and \( \tilde{L}\tilde{U} \) is the computed \( LU \)-factorization of \( q_m(X) \).

Proof Since all the coefficients of \( p_m \) are positive, which is shown in (2), and \( q_m(x) = p_m(-x) \), we can find, from Theorem 2, that the computed polynomials \( \hat{p}_m(X) \) and \( \hat{q}_m(X) \) satisfy

\[
|\Delta p_m| \leq \gamma_{mn} p_m(|X|), \quad |\Delta q_m| \leq \gamma_{mn} q_m(|X|), \quad (12)
\]

where \( \Delta p_m = \hat{p}_m - p_m \) and \( \Delta q_m = \hat{q}_m - q_m \). From Theorem 1, the computed approximant \( \hat{r}_m(X) \) satisfies

\[
\{q_m(X) + \Delta^{(1)} q_m(X)\} \hat{r}_m(X)_{s,j} = \{p_m(X)\} s,j,
\]

where \( [\cdot]_{s,j} \) denotes the \( j \)th column of the enclosed matrix, and the perturbation matrix \( \Delta^{(1)} q_m(X) \) satisfies inequality (11) with \( \hat{L}\hat{U} \approx q_m(X) \). From (13), \( q_m(X) \hat{r}_m(X)_{s,j} = [p_m(X)] s,j \), and \( \Delta_s = \hat{r}_m(X) - r_m(X) \), we have

\[
|\Delta_s| \leq q_m^{-1}(X) \{ |\Delta p_m| \hat{r}_m(X)_{s,j} - |\Delta q_m| q_m(X) \hat{r}_m(X)_{s,j} \}.
\]

From Theorem 1 and inequality (12)

\[
|\Delta_s| \leq |q_m^{-1}(X)| \{ \gamma_{mn} p_m(|X|) + \gamma_{3n} \{L\tilde{U}\} \hat{r}_m(X) \}
\]

\[
\leq |q_m^{-1}(X)| \{ 2 \gamma_{mn} v_m(|X|) + \gamma_{3n} \} \exp(|X|),
\]

where \( v_m \) is the polynomial given in (9), and the relations \( p_m(|X|) = q_m(|X|) r_m(|X|) \) and \( r_m(|X|) \approx \exp(|X|) \) are used. Following the similar process and applying

\[
|u_m(X) - \hat{u}_m(X)| \leq \gamma_{mn} u_m(|X|),
\]

which is obtained from Theorem 2, the inequality for \( |\Delta_s^{+}| \) can also be obtained.

(QED)

From Theorem 3 it can be expected that the modified initial approximant (10) is being accurate than the normal one (4), since the inequality

\[
\exp(|A/2^n|) \leq \exp(|A/2^n|)
\]

always holds, in particular in the diagonal elements this holds without equal sign.

Next we consider the roundoff errors in the squaring processes. For the errors in matrix product the following theorem is known:

Theorem 4 (Higham [6]) Let \( A, B \in \mathbb{R}^{n \times n} \), then the computed product \( \hat{C} \) of \( C = AB \) satisfies

\[
\|C - \hat{C}\| \leq \gamma_n \|A\|_p \|B\|_p, \quad p = 1, \infty, F.
\]

About the accumulation of the roundoff errors, we have the following theorem:

Theorem 5 Let \( R_i \) be the error introduced in the products of the squaring (5), that is,

\[
R_i := \text{fl}(E_i^2) - \hat{E}_i^2,
\]

and let \( M_i^+ \) and \( N_i^- \) be the errors introduced in the squaring and the addition in (8), respectively, that is,

\[
M_i^+ := \text{fl}(E_i^{+2}) - (\hat{E}_i^2)^2,
\]

\[
N_i^- := \hat{E}_i - (\text{fl}(E_i^{+2}) + 2\hat{E}_i^2).
\]

Therefore the error in the \( i \)th step in (8) is given by

\[
R_i := N_i^+ + M_i^+.
\]

The final errors of the two squarings satisfy the following bounds:

\[
\begin{align*}
&\|\Delta_0\| \leq \sum_{i=1}^{s} 2^{i-1} w_i \|R_i\| + 2^s w_{s+1} \|\Delta_s\|, \\
&\|\Delta_s^{+}\| \leq \sum_{i=1}^{s} 2^{i-1} w_i \|R_i\| + 2^s w_{s+1} \|\Delta_s^{+}\|,
\end{align*}
\]

where we put \( w_i = \prod_{j=1}^{i-0} |E_j| \) and use the convention that \( \prod_{j=1}^{0} |E_j| = 1 \).

Proof By applying the result of Theorem 4 to the \( i \)th step of (5) and neglecting the \( O(\|\Delta_s^{+}\|^2) \) term, we have

\[
\Delta_{i-1} = \hat{E}_{i-1} - \text{fl}(E_i^{+2}) = \Delta E_i + E_i \Delta_i + R_i.
\]

Thus we have

\[
\|\Delta_{i-1}\| \leq 2\|E_i\| \|\Delta_s^{+}\| + \|R_i\|.
\]

From this we have the first inequality of (15).

On the other hand, for the modified squaring (8), also neglecting \( O(\|\Delta_s\|^2) \) term, we have

\[
\hat{E}_{i-1} = \text{fl}(\hat{E}_i^{+2}) + 2\hat{E}_i^2
\]

\[
= E_i^{+2} + 2E_i^2 + R_i^2
\]

\[
= E_i^{+2} + 2E_i^2 + \Delta E_i^2 + E_i^2 + \Delta_i^2 + 2\Delta_i^2 + R_i^2.
\]

From this we have the following relation similar to (16):

\[
\Delta_{i-1} = E_i \Delta_i + E_i \Delta_i^+ + R_i^2.
\]

Thus the second inequality of (15) is also derived.

(QED)

Theorem 5 shows that the accuracies of the both algorithms depend on the size of the local errors as well as on those of the initial errors \( \Delta_s \) and \( \Delta_s^{+} \). They can be estimated readily by applying Theorem 4, and they can be obtained from the following theorem.

Theorem 6 Let \( N_i^\pm \) be the errors introduced in the squaring and the addition in (8), respectively, that is,

\[
N_i^\pm := \text{fl}(E_i^{+2}) - \hat{E}_i^2.
\]

Theorem 6 shows that the accuracies of the both algorithms depend on the size of the local errors as well as on those of the initial errors \( \Delta_s \) and \( \Delta_s^{+} \). They can be estimated readily by applying Theorem 4, and they can be obtained from the following theorem.

Theorem 6 Let \( N_i^\pm \) be the errors introduced in the squaring and the addition in (8), respectively, that is,

\[
N_i^\pm := \text{fl}(E_i^{+2}) - \hat{E}_i^2.
\]

Theorem 6 shows that the accuracies of the both algorithms depend on the size of the local errors as well as on those of the initial errors \( \Delta_s \) and \( \Delta_s^{+} \). They can be estimated readily by applying Theorem 4, and they can be obtained from the following theorem.

Theorem 6 Let \( N_i^\pm \) be the errors introduced in the squaring and the addition in (8), respectively, that is,

\[
N_i^\pm := \text{fl}(E_i^{+2}) - \hat{E}_i^2.
\]

Theorem 6 shows that the accuracies of the both algorithms depend on the size of the local errors as well as on those of the initial errors \( \Delta_s \) and \( \Delta_s^{+} \). They can be estimated readily by applying Theorem 4, and they can be obtained from the following theorem.

Theorem 6 Let \( N_i^\pm \) be the errors introduced in the squaring and the addition in (8), respectively, that is,

\[
N_i^\pm := \text{fl}(E_i^{+2}) - \hat{E}_i^2.
\]

Theorem 6 shows that the accuracies of the both algorithms depend on the size of the local errors as well as on those of the initial errors \( \Delta_s \) and \( \Delta_s^{+} \). They can be estimated readily by applying Theorem 4, and they can be obtained from the following theorem.
Thus, if the inequality
\[ \| \exp(A/2^i) \|_p \approx \| \tilde{E}_i \|_p \ll \| \tilde{E}_i \|_p \approx \| \exp(A/2^i) \|_p \]
holds, then it is expected \( \| R_i \|_p < \| R_i \|_p \) so that it can be expected that the modified squaring (8) has better accuracy.

5. Numerical experiments

Here we briefly describe the outline of our experiment. First of all, we shift the eigenvalues of \( A \) by the transformation \( A \leftarrow A - \mu I \), where \( \mu = \text{trace}(A)/n \), in order to reduce the spectral radius of \( A \), and perform the balancing of the transformed matrix. Next we scale the new \( A \) so as to satisfy \( 2^{-s} \| A \| \leq 10^{-13} \approx 5 \), since \( m = 13 \) is being optimal. Using the resulting \( A \), we evaluate the polynomials \( q_m \) and \( u_m \) by Higham’s method [2] and solve the matrix equation \( q_mE_i^2 = 2u_m \), and repeat (8) \( s \) times. Finally add the identity matrix \( I \) and multiply by \( \exp(\mu) \).

The matrices used in the experiment are shown in Table 1. All the matrices, which are generated by MATLAB gallery function [7], are of degree 400. We compute two approximations to \( \exp(A) \) and \( \exp(-A) \) for each \( A \), and compare the results of the two methods using the relative errors
\[ \frac{\| E_0 - \exp(A) \|_F}{\| \exp(A) \|_F}, \quad \frac{\| E_i^2 + I - \exp(A) \|_F}{\| \exp(A) \|_F}. \]

All the calculations are done by the IEEE double precision arithmetic of g++ 4.4.5 compiler. The exact values are computed by the multiple precision libraries MPACK 0.8.0 and GMP 4.3.2, in which the mantissa length is set to 256 bit. The results are shown in Table 2.

We can find from Table 2 that the modified algorithm is more accurate than Higham’s one, or at least comparable, in almost all cases. Remarkable results are obtained when \( A \) is chebvand and orthog for both \( \exp(A) \) and \( \exp(-A) \), and cauchy and randcorr for \( \exp(-A) \). Here we consider the reason.

For these four matrices, we can see from the ranges of the shifted and scaled eigenvalues \( \lambda_i/2^s \) (\( \lambda_i = \lambda_i - \mu \)) in Table 3 that many eigenvalues are located near the origin. Therefore the inequality (19) is likely to hold even when the scaling level \( i \) is low, that is, \( i \) is small. Moreover for two matrices cauchy and randcorr, the remarkable fact is that the farthest eigenvalues from the origin are positive. Let \( \lambda_{\text{max}} \) be the farthest eigenvalue of the shifted matrix. The both sides of the inequality (19) strongly depend on the values \( \exp(\lambda_{\text{max}}/2^i) \) and \( \exp(\lambda_{\text{max}}/2^i) \), which are close to each other when \( \lambda_{\text{max}} \) is positive large value. So the accuracies of the two algorithms are almost the same when computing \( \exp(A) \) for these matrices. On the other hand, for \( -A \) the farthest eigenvalues from the origin are negative, that is \( \lambda_{\text{max}} < 0 \), which means that \( \exp(\lambda_{\text{max}}/2^i) \) and \( \exp(\lambda_{\text{max}}/2^i) \) are very small. As a result the both sides of (19) depend on the other eigenvalues near the origin.

6. Conclusions

We have presented a modified squaring technique in the scaling and squaring algorithm for the matrix exponential. For many matrices in our experiment, our technique has been shown to be more accurate than Higham’s algorithm. Moreover our algorithm has the same complexity with that of Higham’s one. For future works we will apply our algorithm to the computation of the other matrix functions.

Acknowledgments

The authors are very grateful to the anonymous referee, whose comments greatly improved this paper.

References