MATHEMATICS OF COMPUTATION
Volume 82, Number 283, July 2013, Pages 1577–1596
S 0025-5718(2013)02677-4
Article electronically published on March 13, 2013

COMPUTING EXPONENTIALS OF ESSENTIALLY NON-NEGATIVE MATRICES ENTRYWISE TO HIGH RELATIVE ACCURACY

JUNGONG XUE AND QIANG YE

Abstract. A real square matrix is said to be essentially non-negative if all of its off-diagonal entries are non-negative. It has recently been shown that the exponential of an essentially non-negative matrix is determined entrywise to high relative accuracy by its entries up to a condition number intrinsic to the exponential function (Numer. Math. 110 (2008), 393–403). Thus the smaller entries of the exponential may be computed to the same relative accuracy as the bigger entries. This paper develops algorithms to compute exponentials of essentially non-negative matrices entrywise to high relative accuracy.

1. Introduction

Matrix exponential is an important theoretical and numerical tool in sciences and engineering. Efficient and accurate computations of matrix exponentials have been studied extensively. A wide range of different numerical methods have been developed; see [1, 3, 7, 16, 19, 20, 21, 23, 26]. The state of the arts is surveyed in the classical paper [18] by Moler and Van Loan with a recent update in [19]. The scaling and squaring method coupled with the Padé approximation as well as the Schur decomposition method are in general considered to be most competitive overall; see [17, 19]. As they are at best normwise backward stable [19], the computed matrix exponentials have relative errors that are bounded in norm and dependent on a condition number for normwise perturbations. This condition number is well bounded for normal matrices and generator matrices of Markov chains, but in general, it may grow very quickly as the norm of the matrix increases [17, 22, 24]. Moreover, even for problems where this condition number is well bounded, the entrywise relative accuracy is not guaranteed by these two methods; namely smaller entries of the exponential matrix may be computed with lower relative accuracy. It turns out to be quite common for exponentials of matrices (and many other functions of matrices) to have some very small entries relative to others. For example, the entries of the exponential of a banded matrix decay exponentially away from the main diagonal [5]. More generally, the entries of the exponential of a sparse matrix can have very large variations in their order of magnitude [4].

This paper considers the computation of exponentials of essentially non-negative matrices. A matrix $A = [a_{ij}] \in \mathbf{R}^{n\times n}$ is said to be essentially non-negative [25, Sec. 8.2] if all of its off-diagonal entries are non-negative, i.e., $a_{ij} \geq 0$ for $i \neq j$.

Received by the editor September 27, 2011.

2010 *Mathematics Subject Classification.* Primary 65F60; Secondary 65F35, 15A12.

The first author's research was supported in part by NSFC under Grant 10971036 and Laboratory of Mathematics for Nonlinear Science, Fudan University.

The second author's research was supported in part by NSF under Grant DMS-0915062.

We note that essentially non-negative matrices form a natural class of matrices in the study of matrix exponentials because a matrix A is essentially non-negative if and only if $\exp(tA)$ is non-negative for all $t \geq 0$ [25, Sec. 8.2]. Related to this, for linear dynamical systems

$$x'(t) = Ax(t), \qquad x(0) = x_0, \tag{1.1}$$

the matrix A is essentially non-negative if and only if it has the property that $x(0) \geq 0$ implies $x(t) \geq 0$. Clearly, many physical systems of (1.1) would have such a property and hence have an essentially non-negative A.

For the exponential of an essentially non-negative matrix A, we have recently obtained an entrywise perturbation analysis in [28] showing that, if E is a small perturbation to A such that $|E| \leq \epsilon|A|$, then we have

$$|\exp(A+E) - \exp(A)| \leq \kappa_{\exp}(A) e^{\kappa_{\exp}(A)\epsilon/(1-\epsilon)} \frac{\epsilon}{1-\epsilon} \exp(A), \tag{1.2}$$

where $\kappa_{\exp}(A) := n - 1 + \rho(A - dI) + \max_i |a_{ii}|$, $\rho(A - dI)$ is the spectral radius of $A - dI$, and $d = \min a_{ii}$. Here the absolute value and inequalities on matrices are entrywise. With the upper bound (1.2) attainable, $\kappa_{\exp}(A)$ is a condition number for the entrywise perturbation; see [28]. Indeed, it is intrinsic to the exponential function itself in the sense that it is present in the perturbation of the exponential function of a real variable when $n = 1$ (see [28]).

The perturbation bound suggests that it is possible to compute all entries of the exponential of an essentially non-negative matrix with a relative accuracy in the order of machine precision up to a scaling by the condition number $\kappa_{\exp}(A)$. However, popular methods such as the Padé approximation and the Schur decomposition method may not achieve this. We note that for a given method to achieve entrywise relative accuracy, we first need to identify an approximation to $\exp(A)$ (such as a rational function $R_{p,q}(A)$ in the Padé approximation) that has entrywise small relative errors, and then we need to be able to compute that approximation ($R_{p,q}(A)$ in the Padé approximation) with entrywise relative accuracy. Both turn out to be challenging problems for the existing methods. For example, in Padé approximations, computing the rational function $R_{p,q}(A)$ requires inverting a polynomial of A with both additions and subtractions of powers of A, which may not be computed accurately in the entrywise sense. A more subtle difficulty concerns determining the degrees of the Padé approximation p, q to ensure that $R_{p,q}(A)$ has desired entrywise relative accuracy; a problem exists even if we assume exact arithmetic or use higher precision arithmetic for computing $R_{p,q}(A)$. We have therefore studied other existing algorithms for exponentials [19] and our investigations have led us to two algorithms, namely the Taylor series method and the polynomial method. We shall show in this paper that some carefully implemented algorithms of these two methods, when combined with a suitable shifting and a proper scaling and squaring, can compute exponentials of essentially non-negative matrices entrywise with a relative accuracy in the order of $\kappa_{\exp}(A)\mathbf{u}$ where $\mathbf{u}$ is the machine roundoff unit.

The paper is organized as follows. In §2, we provide an error analysis for the shifting to show that the subtraction involved in shifting does not cause significant errors in the exponential. In §3 we briefly discuss various numerical issues in implementing the Taylor series method combined with a shifting to achieve best entrywise relative accuracy possible. We note that the Taylor series method combined with a shifting has been proposed before as a more accurate method in [6, 8, 13] as well as in the applied probability community [14, 15] and our contributions here are a rigorous truncation criterion and an entrywsie error analysis. The Taylor series method, however, may be expensive as the number of terms required for the series truncation cannot be bounded *a priori*. For matrices whose eigenvalues are real (such as symmetric matrices or triangular matrices), we also develop in §4 a more efficient algorithm based on the polynomial method [19] that collapses the Taylor series using the characteristic polynomial of the matrix. Finally, in §5, we present some numerical examples to demonstrate the entrywise relative accuracy of the new algorithms.

We remark that the need to compute all entries of the exponential of an essentially non-negative matrix accurately is critical in some applications. An important problem where the exponentials of essentially non-negative matrices arise is continuous-time Markov chains. In a continuous-time Markov chain, the exponential of its generator matrix, which is essentially non-negative with zero row sums, is the transition matrix whose entries are the transition probabilities between the states. One is often interested in computing the transition matrix from the generator matrix. In this context, small entries of the exponential, representing small probabilities, are often the ones of interest and are required to be computed accurately.

A current application where entrywise relative accuracy is needed concerns the use of matrix exponential for characterizations of networks [10, 11, 12]. Consider a network as represented by an undirected graph and let $A \in \mathbf{R}^{n\times n}$ be its adjacency matrix, which has (i, j) element equal to 1 if nodes i and j are connected and 0 otherwise. Then A is symmetric and non-negative. The entries of $\exp(A)$ can be used to characterize or measure various network properties [10, 11, 12], among which is the *betweenness* of a node r as defined by

$$\frac{1}{(n-1)^2-(n-1)} \sum_{\substack{i,j \\ i\neq j, i\neq r, j\neq r}} \frac{(\exp(A)-\exp(A-E_r))_{ij}}{(\exp(A))_{ij}}, \tag{1.3}$$

where E_r is zero except in row and column r, where it agrees with A. As the graph is usually very sparse, some entries of $\exp(A)$ could be very tiny. With E_r being a low rank perturbation, relative accuracy of $(\exp(A))_{ij}$ is needed in order to compute quantities like (1.3) with any accuracy; see Example 3 in §5.

Notation. Throughout, inequalities and absolute values of matrices and vectors are entrywise. Namely, for a matrix $X = [x_{ij}]$, we denote by $|X|$ the matrix of entries $|x_{ij}|$ and we write $X \geq Y$, where $Y = [y_{ij}]$ is of the same dimension as X, if $x_{ij} \geq y_{ij}$ for all i and j. Especially, $X \geq 0$ means that every entry of X is non-negative. We denote by $\rho(B)$ the spectral radius of a square matrix B. Given a real number γ, $\lceil\gamma\rceil$ is the smallest integer no less than γ and $\lfloor\gamma\rfloor$ denotes the largest integer no more than γ. We use $fl(z)$ to denote the computed result of the expression z. We also assume the following standard model of floating point arithmetic

$$fl(a \text{ op } b) = (a \text{ op } b)(1+\delta), \quad |\delta| \leq \mathbf{u},$$

where op $= +, -, *$ or $/$ and $\mathbf{u}$ is the machine roundoff unit; see [9, p. 12] for example. Throughout, we assume that no underflow or overflow occurs in computations.

2. Shifting

To compute $\exp(A)$ accurately for an essentially non-negative matrix $A = [a_{ij}]$, we first shift the matrix into a non-negative matrix $A_\alpha := A - \alpha I$ with a proper α. Then we compute $\exp(A_\alpha)$ and obtain $\exp(A)$ via the relation $\exp(A) = e^\alpha \exp(A_\alpha)$. We now consider the effects of possible subtractions and cancellations in forming A_α.

It turns out that subtractions and possible cancellations in forming A_α does not affect the accuracy of computing $\exp(A_\alpha)$, because by a perturbation theorem of [28], the absolute accuracy on the diagonals of A_α is sufficient for determining the entrywise relative accuracy of $\exp(A_\alpha)$.

Lemma 2.1 ([28, Theorem 2])**.** *Let A be an $n \times n$ essentially non-negative matrix. Let $E = [e_{ij}]$ be a perturbation to A satisfying*

$$|e_{ij}| \le \epsilon_1 a_{ij}, \quad i \ne j \;\; \textit{and} \;\; |e_{ii}| \le \epsilon_2, \quad 1 \le i \le n,$$

where $0 \le \epsilon_1 < 1$ and $\epsilon_2 \ge 0$. Letting $A_d := A - dI$, where

$$d = \min a_{ii}, \tag{2.1}$$

we have

$$|\exp(A+E) - \exp(A)| \le c\left(\epsilon_2 + (n - 1 + \rho(A_d))\frac{\epsilon_1}{1-\epsilon_1}\right)\exp(A), \tag{2.2}$$

where $c = \exp\left(\epsilon_2 + (n-1+\rho(A_d)\frac{\epsilon_1}{1-\epsilon_1}\right)$.

From this, we derive the following error analysis associated with the shifting.

Lemma 2.2. *Let A be an $n \times n$ essentially non-negative matrix and let $\widehat{A}_\alpha$ be the computed result of $A_\alpha := A - \alpha I$. Then*

$$|\exp(\widehat{A}_\alpha) - \exp(A_\alpha)| \le \left((2\max|a_{ii}| + 2|\alpha| + n - 1 + \rho(A_d))\mathbf{u} + O(\mathbf{u}^2)\right)\exp(A_\alpha), \tag{2.3}$$

where $\mathbf{u}$ is the machine precision.

Proof. Denote $B = \widehat{A}_\alpha = [b_{ij}]$. Then, for $i \ne j$, we have $b_{ij} = fl(a_{ij}) = a_{ij} + e_{ij}a_{ij}$ where $|e_{ij}| \le \mathbf{u}$. For $i = j$, we have

$$b_{ii} = fl(fl(a_{ii}) - fl(\alpha)) = (a_{ii}(1+\delta_{i,1}) - \alpha(1+\delta_{i,2}))(1+\delta_{i,3}) = a_{ii} - \alpha + e_{ii}$$

where $|\delta_{i,\ell}| \le \mathbf{u}$ (for $1 \le \ell \le 3$) and hence $|e_{ii}| \le 2(|a_{ii}| + |\alpha|)\mathbf{u} + O(\mathbf{u}^2) \le 2(\max|a_{ii}| + |\alpha|)\mathbf{u} + O(\mathbf{u}^2)$. Then, (2.3) follows from Lemma 2.1 with $\epsilon_1 = \mathbf{u}$, $\epsilon_2 = 2(\max|a_{ii}| + |\alpha|)\mathbf{u}$. □

Provided that α is of order $\kappa_{\exp}(A)$ (see (1.2)), Lemma 2.2 shows that the entrywise relative error caused from the errors in computing $A - \alpha I$ is of order $\kappa_{\exp}(A)\mathbf{u}$ and is thus comparable to the rounding errors already made in rounding A to $fl(A)$.

3. Taylor series method

Let $A_d := A - dI$ with $d = \min a_{ii}$. Then $A_d \ge 0$. We can compute $\exp(A)$ through computing $\exp(A_d)$ using the Taylor series as

$$e^A = e^d e^{A_d} = e^d \sum_{k=0}^{\infty} \frac{A_d^k}{k!}. \tag{3.1}$$

As A_d is non-negative, the computation of the Taylor series involves no subtractions and can therefore have entrywise high relative accuracy, modulo possible accumulations of small errors in the summation. This approach has been suggested as a more stable way for computing e^A in the context of the transient matrix computations in the continuous Markov chain in [14, 15] and more generally in [6, 13]. Here, we rigourously examine various numerical issues arising in this process and provide an entrywise error analysis. Specifically, we need to consider effects of rounding errors in forming A_d as well as to determine the truncation of the Taylor series to ensure entrywise accuracy.

Let $\widehat{A}_d$ be the computed A_d. The rounding errors caused by forming A_d has been discussed in Lemma 2.2; namely, we have

$$\begin{aligned} |\exp(\widehat{A}_d) - \exp(A_d)| &\leq \left((4\max|a_{ii}| + n - 1 + \rho(A_d))\mathbf{u} + O(\mathbf{u}^2)\right)\exp(A_d) \\ &\leq \left(4\kappa_{\exp}(A)\mathbf{u} + O(\mathbf{u}^2)\right)\exp(A_d). \end{aligned} \tag{3.2}$$

We now turn to computing $\exp(\widehat{A}_d)$ accurately by the Taylor series. In this regard, we need to determine when to truncate the series, that is, given a tolerance (tol) for entrywise relative errors, we are to determine a positive integer m such that the error of approximating $\exp(\widehat{A}_d)$ by $T_m(\widehat{A}_d) := \sum_{k=0}^{m-1} \frac{\widehat{A}_d^k}{k!}$ satisfies

$$|\exp(\widehat{A}_d) - T_m(\widehat{A}_d)| \leq tol \cdot T_m(\widehat{A}_d). \tag{3.3}$$

This can be done using the following theorem, which is proved for triangular non-negative matrices in [8, Theorem 2] but is easily extended to general non-negative matrices. A similar normwise bound is also obtained in [6, 13].

Theorem 3.1. *Let m be such that $\rho(\widehat{A}_d/(m+1)) < 1$. Then*

$$|\exp(\widehat{A}_d) - T_m(\widehat{A}_d)| \leq \frac{\widehat{A}_d^m}{m!}\left(I - \frac{\widehat{A}_d}{m+1}\right)^{-1}. \tag{3.4}$$

It follows that we can guarantee (3.3) by checking the criterion

$$\frac{\widehat{A}_d^m}{m!}\left(I - \frac{\widehat{A}_d}{m+1}\right)^{-1} \leq tol \cdot T_m(\widehat{A}_d). \tag{3.5}$$

Since $I - \widehat{A}_d/(m+1)$ is an M-matrix, its inverse can be computed entrywise to high relative accuracy by the GTH-like algorithm (Algorithm 1 of [2]). Then the upper bound (3.4) can be computed entrywise accurately.

Checking the condition (3.5) requires solving the linear systems with n right-hand sides. To reduce this expensive operation, we should check the simpler condition $\frac{\widehat{A}_d^m}{m!} \leq tol \cdot T_m(\widehat{A}_d)$ first and only when this is satisfied, we should proceed to check (3.5), because $\frac{\widehat{A}_d^m}{m!} \leq \frac{\widehat{A}_d^m}{m!}\left(I - \frac{\widehat{A}_d}{m+1}\right)^{-1}$. The condition of Theorem 3.1 can be satisfied by the scaling and squaring that we discuss now.

The scaling and squaring method (see [19]) is a basic tool that is used in combination with many existing methods for computing exponentials. If implemented carefully, it can even enhance the accuracy of certain methods; see the recent work [16]. Here, it can also be used to reduce the norm (or spectral radius) of the matrix so as to accelerate convergence of the Taylor series. Furthermore, it may also have the potential benefit of avoiding underflow/overflow.

Let ρ be a rough estimate of the spectral radius $\rho(\widehat{A}_d)$, which can be obtained by a few iterations of the power methods. In practice, using $\rho = \|\widehat{A}_d\|_\infty$ should be sufficient in most cases. Let $p = \lfloor \log_2 \rho \rfloor + 1$. Then, we evaluate the exponential through the following scaling and repeated squaring

$$\begin{aligned} \exp(A) &= e^d \exp(A_d) \approx e^d \exp(\widehat{A}_d) \\ &= \left(e^{d/2^p} \exp(\widehat{A}_d/2^p)\right)^{2^p} \approx \left(e^{d/2^p} T_m(\widehat{A}_d/2^p)\right)^{2^p} . \end{aligned} \tag{3.6}$$

Note that the scaling ensures $\rho(\widehat{A}_d/2^p) < 1$ (or $\|\widehat{A}_d/2^p\|_\infty < 1$ if $\rho = \|\widehat{A}_d\|_\infty$), which accelerates convergence of the Taylor series.

We now discuss the forward entrywise relative errors of the exponential computed this way by breaking down the errors and their propagations as follows. We omit a precise statement of error bounds.

(1) $\exp(\widehat{A}_d)$ approximates $\exp(A_d)$ with entrywise relative errors bounded by $4\kappa_{\exp}(A)\mathbf{u} + O(\mathbf{u}^2)$ (see (3.2)).

(2) $T_m(\widehat{A}_d/2^p)$ is computed with entrywise relative error in the order of $mn\mathbf{u}$. Note that the entrywise relative error of multiplying m non-negative matrices is of order $mn\mathbf{u} + O(\mathbf{u}^2)$ and the summation does not amplify the errors.

(3) With (3.5), the entrywise relative errors of approximating $\exp(\widehat{A}_d/2^p)$ by $T_m(\widehat{A}_d/2^p)$ is of order $\mathbf{u}$ (assuming *tol* is set to $\mathbf{u}$).

(4) Multiplication by $e^{d/2^p}$ introduces an entrywise error bounded by $\mathbf{u}$.

(5) Combining these errors, we have that the computed $e^{d/2^p} T_m(\widehat{A}_d/2^p)$ approximates $e^{d/2^p} \exp(\widehat{A}_d/2^p)$ with entrywise relative error of order $mn\mathbf{u}$. The repeated squaring will amplify this error by 2^p times, which results in entrywise relative error of approximating $\exp(\widehat{A}_d)$ in the order of $mn2^p\mathbf{u}$. Consequently, the entrywise relative error of approximating $\exp(A)$ is in the order of $(mn2^p + 4\kappa_{\exp}(A))\mathbf{u} \sim mn\kappa_{\exp}(A)\mathbf{u}$.

From the above discussion, we conclude that the computed exponential has entrywise relative errors in the order of $mn\kappa_{\exp}(A)\mathbf{u}$. We remark that it has been observed in [6, 13] that the Taylor series method can produce entrywise high relative accuracy but only normwise error analysis was carried out there. While there are some similarities in the normwise and entrywise analysis, some substantial differences exist such as the need to truncate the series to have entrywise small relative errors (3.5) and the need to compute the quantities involved in the bound accurately.

We finally state a full algorithm for easy references.

Algorithm 1. *Taylor Series Method*

```
Input: an $n \times n$ essentially non-negative matrix $A$, tol (error tolerance),
       iter (max iteration)
Output: $E$
1.   $d = \min a_{ii}$; $A_d = A - dI$;
2.   Compute $\rho$ to be an estimate of $\rho(\widehat{A}_d)$ or $\|A_d\|_\infty$;
3.   $p = \lfloor \log_2(\rho) \rfloor + 1$; $B = A_d/2^p$
4.   $E = I + B$; $W = B$;
5.   For $m = 2 : iter$,
6.       $W = BW/m$
```

```
        $E = E + W$;
        If $W \le tol \cdot E$,
            compute $\left(I - \frac{B}{m+1}\right)^{-1}$ by Algorithm 1 of [2]
            If (3.5) is true, BREAK, End
        End
    End
    $E = e^{d/2^p} E$
    For $i = 1 : p$, $E = E^2$, End;
```

4. Polynomial method

One disadvantage of the Taylor series method is that m required to satisfy (3.5) cannot be bounded in *a priori*. This is a distinct difficulty arising in achieving entrywise accuracy. For example, if A is irreducible tridiagonal, the $(n, 1)$ entry of $T_m(A)$ is 0 for $m \le n - 2$. Then for $T_m(A)$ to have any relative accuracy in the $(n, 1)$ entry, m should be at least $n - 1$ no matter how small the norm of A is. The cost of checking condition (3.5) is another disadvantage of the Taylor series method. In this section, we develop a method that computes $\exp(A_\alpha/2^p)$, for some proper choice of α and p, with high entrywise relative accuracy. The method is based on the polynomial method [19], which collapses the Taylor series to a polynomial of degree $n - 1$. In particular, this method applies to symmetric matrices or any matrices whose eigenvalues are real (such as triangular matrices).

We remark that symmetric essentially non-negative matrices arise often in applications. For example, the generator matrix of a reversible Markov chain can be similarly transformed to a symmetric one by a diagonal matrix. The adjacency matrix of an undirected graph is a symmetric non-negative matrix.

The basic framework of the polynomial method can be described as follows. Let the characteristic polynomial of A be

$$\det(zI - A) = z^n + \sum_{i=0}^{n-1} (-1)^{n-i} \gamma_{n-i} z^i,$$

where γ_k (for $1 \le k \le n$) can be expressed in terms of the eigenvalues of A, λ_i $(1 \le i \le n)$, as

$$\gamma_k = \sum_{1 \le i_1 < i_2 < \cdots < i_k \le n} \lambda_{i_1} \lambda_{i_2} \ldots \lambda_{i_k}.$$

From the Cayley-Hamilton theorem, we have $A^n = \gamma_1 A^{n-1} - \gamma_2 A^{n-2} + \gamma_3 A^{n-3} - \ldots + (-1)^{n-1} \gamma_n I$. It follows that any power A^m, $m \ge n$, can be expressed in terms of $I, A, \ldots, A^{n-1}$ as

$$A^m = \beta_{m,n-1} A^{n-1} - \beta_{m,n-2} A^{n-2} + \beta_{m,n-3} A^{n-3} - \ldots + (-1)^{n-1} \beta_{m,0} I. \tag{4.1}$$

Based on the coefficients γ_i's of the characteristic polynomial, $\beta_{m,k}$'s can be generated by:

$$\beta_{n,k} = \gamma_{n-k}, \qquad 0 \le k \le n - 1;$$

and for $m > n$,

$$\beta_{m,0} = \gamma_n \beta_{m-1,n-1}, \tag{4.2}$$

$$\beta_{m,k} = \gamma_{n-k} \beta_{m-1,n-1} - \beta_{m-1,k-1}, \quad 1 \le k \le n - 1. \tag{4.3}$$

Then $\exp(A)$ can be expressed as

$$\exp(A) = \sum_{k=0}^{n-1} \frac{\alpha_k}{k!} A^k, \tag{4.4}$$

where

$$\alpha_k = 1 + (-1)^{n-k-1} \sum_{m=n}^{\infty} \frac{k!}{m!} \beta_{m,k}. \tag{4.5}$$

We will show in this section how (4.4) will produce an entrywise accurate $\exp(A)$. In §4.1, we show that, if the eigenvalues of A are positive, then all the $\beta_{m,k}$'s are positive. In §4.2, we give a sufficient condition for α_k's to be positive. We also discuss in these two subsections accurate computations of $\beta_{m,k}$ and α_k respectively. In §4.3, we discuss how to accurately compute the characteristic polynomial. In §4.4, we describe how to shift A and apply the scaling and squaring strategy on the shifted matrix to ensure that the conditions mentioned above are all met. We summarize the process in an algorithm in §4.5.

4.1. **Positivity and computation of $\beta_{m,k}$.**

Suppose all the eigenvalues of A are positive, namely $0 < \lambda_1 \le \lambda_2 \le \cdots \le \lambda_n$, then we show in this section that all the $\beta_{m,k}$'s are positive and, if $\frac{\lambda_1}{\lambda_n}$ is not too small, cancellations do not occur in its computation (4.3).

For $m \ge n$, define the polynomial

$$p_m(x) = \beta_{m,n-1}x^{n-1} - \beta_{m,n-2}x^{n-2} + \beta_{m,n-3}x^{n-3} - \ldots + (-1)^{n-1}\beta_{m,0}.$$

Then, $p_m(x)$ is the polynomial of degree $n-1$ which agrees with x^m at points $\lambda_1, \ldots, \lambda_n$; see (4.1). Define

$$f_m(x) := x^m$$

and denote by $f_m[x_1, x_2, \ldots, x_k]$ the k-th divided difference of the function f_m with respect to $x_1, x_2, \ldots, x_k$. Specifically, let $f_m[x_1] = x_1^m$. By Newton's interpolatory divided-difference formula, $p_m(x)$ can be written as

$$p_m(x) = \lambda_1^m + \sum_{k=2}^{n} f_m[\lambda_1, \lambda_2, \ldots, \lambda_k](x-\lambda_1)(x-\lambda_2)\cdots(x-\lambda_{k-1}). \tag{4.6}$$

The following result gives an explicit formula for the divided difference $f_m[x_1, x_2, \ldots, x_k]$.

Lemma 4.1. *Define*

$$S_l(x_1, x_2, \ldots, x_k) := \sum_{i_1+i_2+\ldots+i_k=l} x_1^{i_1} x_2^{i_2} \ldots x_k^{i_k}$$

with the convention

$$S_0(x_1, x_2, \ldots, x_k) = 1.$$

Then for $m \ge k-1$,

$$f_m[x_1, x_2, \ldots, x_k] = S_{m-k+1}(x_1, x_2, \ldots, x_k).$$

Proof. We assume that $x_i \ne x_j$ for $i \ne j$. By taking limits we can show that the result holds for general x_i's.

We prove by induction on k. Obviously it holds for $k = 1$. Suppose that it holds for some k with $k \leq m$. Noting

$$S_{m-k+1}(x_1, x_2, \ldots, x_k) = S_{m-k+1}(x_1, x_2, \ldots, x_{k+1}) - x_{k+1}S_{m-k}(x_1, x_2, \ldots, x_{k+1})$$

and

$$S_{m-k+1}(x_2, x_3, \ldots, x_{k+1}) = S_{m-k+1}(x_1, x_2, \ldots, x_{k+1}) - x_1S_{m-k}(x_1, x_2, \ldots, x_{k+1}),$$

we have

$$\begin{aligned} f_m[x_1, x_2, \ldots, x_{k+1}] =& \frac{f_m[x_1, x_2, \ldots, x_k] - f_m[x_2, \ldots, x_{k+1}]}{x_1 - x_{k+1}} \\ =& \frac{S_{m-k+1}(x_1, x_2, \ldots, x_k) - S_{m-k+1}(x_2, x_3, \ldots, x_{k+1})}{x_1 - x_{k+1}} \\ =& S_{m-k}(x_1, x_2, \ldots, x_{k+1}). \end{aligned}$$

We have proved for the case of $k + 1$. □

For ease of exposing, we shall simply write $S_k(\lambda_1, \ldots, \lambda_n)$ as S_k. By Lemma 4.1 and (4.6), we have

$$\beta_{m,n-1} = f_m[\lambda_1, \lambda_2, \ldots, \lambda_n] = S_{m-n+1},$$

and then by (4.2)

$$\beta_{m,0} = \gamma_n\beta_{m-1,n-1} = \gamma_nS_{m-n}.$$

The following lemma gives some monotonicity property of the coefficients of the polynomial that interpolates x^m. It will play a key role in the proof that all $\beta_{m,k}$'s are positive.

Lemma 4.2. *For $m \geq k - 1$, let*

$$q_m(x) = \delta_{m,k-1}x^{k-1} - \delta_{m,k-2}x^{k-2} + \delta_{m,k-3}x^{k-3} - \ldots + (-1)^{k-1}\delta_{m,0}$$

and

$$\widehat{q}_m(x) = \widehat{\delta}_{m,k-1}x^{k-1} - \widehat{\delta}_{m,k-2}x^{k-2} + \widehat{\delta}_{m,k-3}x^{k-3} - \ldots + (-1)^{k-1}\widehat{\delta}_{m,0}$$

be the polynomials of degree $k - 1$ that interpolate x^m at $0 < x_1 \leq x_2 \leq \ldots \leq x_k$ and at $0 < \widehat{x}_1 \leq \widehat{x}_2 \leq \ldots \leq \widehat{x}_k$ respectively. If

$$\widehat{x}_j \geq x_j, \qquad 1 \leq j \leq k,$$

with strict inequality holds for at least one j, then

$$\widehat{\delta}_{m,j} > \delta_{m,j}, \qquad 0 \leq j \leq k - 1.$$

Proof. Without loss of generality, we assume that $x_j = \widehat{x}_j$ for $1 \leq j \leq k - 1$ and $x_k < \widehat{x}_k$. For the general case, we can prove the result step by step and at each step change one x_j to $\widehat{x}_j$.

By Newton's interpolatory divided-difference formula,

$$q_m(x) = x_1^m + \sum_{j=2}^{k} f_m[x_1, \ldots, x_j](x - x_1) \cdots (x - x_{j-1})$$

and

$$\begin{aligned} \widehat{q}_m(x) =& x_1^m + \sum_{j=2}^{k-1} f_m[x_1, \ldots, x_j](x - x_1) \cdots (x - x_{j-1}) \\ &+ f_m[x_1, \ldots, x_{k-1}, \widehat{x}_k](x - x_1) \cdots (x - x_{k-1}). \end{aligned}$$

Then

$$\widehat{q}_m(x) - q_m(x) = \delta(x - x_1)(x - x_2)\cdots(x - x_{k-1}),$$

where

$$\delta = f_m[x_1, \ldots, x_{k-1}, \widehat{x}_k] - f_m[x_1, \ldots, x_{k-1}, x_k].$$

By Lemma 4.1,

$$\delta = S_{m-k+1}(x_1, \ldots, x_{k-1}, \widehat{x}_k) - S_{m-k+1}(x_1, \ldots, x_{k-1}, x_k) > 0,$$

which implies

$$\widehat{\delta}_{m,j} - \delta_{m,j} = \delta \sum_{1 \le i_1 < i_2 < \ldots < i_{k-1-j} \le k-1} x_{i_1} x_{i_2} \cdots x_{i_{k-1-j}} > 0.$$

The proof is completed. □

The following is the main result of this subsection.

Theorem 4.3. *For $m \ge n+1$ and $1 \le k \le n-1$, we have $\beta_{m,k} > 0$ and*

$$\frac{\gamma_{n-k}\beta_{m-1,n-1}}{\beta_{m-1,k-1}} \ge 1 + \frac{\lambda_1}{(n-1)\lambda_n}. \tag{4.7}$$

Proof. Let

$$\gamma'_k := \sum_{2 \le i_1 < i_2 < \cdots < i_k \le n} \lambda_{i_1}\lambda_{i_2}\cdots\lambda_{i_k}, \quad 1 \le k \le n-1$$

and let

$$\widetilde{p}_{m-1}(x) = \widetilde{\beta}_{m-1,n-2}x^{n-2} - \widetilde{\beta}_{m-1,n-3}x^{n-3} + \widetilde{\beta}_{m-1,n-4}x^{n-4} - \ldots + (-1)^{n-2}\widetilde{\beta}_{m-1,0}$$

be the polynomial of degree $n-2$ that agrees with x^{m-1} at $\lambda_2, \lambda_3, \ldots, \lambda_n$. By Newton's divided-difference formula, $p_{m-1}(x)$, the polynomial of degree $n-1$ that agrees with x^{m-1} at $\lambda_1, \lambda_2, \ldots, \lambda_n$, can be expressed as

$$p_{m-1}(x) = \widetilde{p}_{m-1}(x) + \beta_{m-1,n-1}(x - \lambda_2)(x - \lambda_3)\cdots(x - \lambda_n),$$

which gives

$$\beta_{m-1,k} = \gamma'_{n-k-1}\beta_{m-1,n-1} - \widetilde{\beta}_{m-1,k}, \qquad 0 \le k \le n-2. \tag{4.8}$$

Note that $p_m(x)$ is the polynomial of degree $n-1$ that agrees with x^m at $\lambda_1, \lambda_2, \ldots, \lambda_n$, then $f(x) = x^m - p_m(x)$ has zeros $\lambda_1, \lambda_2, \ldots, \lambda_n$, and thus $f'(x) = mx^{m-1} - p'_m(x)$ has zeros $\widehat{\lambda}_1, \widehat{\lambda}_2, \ldots, \widehat{\lambda}_{n-1}$, with $\widehat{\lambda}_j \in [\lambda_j, \lambda_{j+1}]$. Then

$$\frac{1}{m}p'_m(x) = \sum_{k=1}^{n-1}(-1)^{n-k-1}\frac{k}{m}\beta_{m,k}x^{k-1}$$

is the polynomial of degree $n-2$ that agrees with x^{m-1} at $\widehat{\lambda}_1, \widehat{\lambda}_2 \ldots, \widehat{\lambda}_{n-1}$. Comparing the interpolatory points of $p'_m(x)/m$ to those of $\widetilde{p}_{m-1}(x)$ and using Lemma 4.2, we obtain

$$\frac{k}{m}\beta_{m,k} \le \widetilde{\beta}_{m-1,k-1}. \tag{4.9}$$

From (4.8), we have

$$\widetilde{\beta}_{m-1,k-1} = \gamma'_{n-k}\beta_{m-1,n-1} - \beta_{m-1,k-1}. \tag{4.10}$$

Substituting $\beta_{m,k} = \gamma_{n-k}\beta_{m-1,n-1} - \beta_{m-1,k-1}$ and (4.10) into (4.9) and using the fact that $\beta_{m-1,n-1} = S_{m-n} > 0$, we obtain

$$\gamma_{n-k}\beta_{m-1,n-1} - \beta_{m-1,k-1} \le \frac{m}{k}(\gamma'_{n-k}\beta_{m-1,n-1} - \beta_{m-1,k-1}).$$

From this, we can deduce that, if $\beta_{m-1,k} > 0$ (for $0 \le k \le n-1$),

$$\frac{\beta_{m-1,k-1}}{\gamma_{n-k}\beta_{m-1,n-1}} \le \frac{k}{m-k}\left(\frac{m}{k}\frac{\gamma'_{n-k}}{\gamma_{n-k}} - 1\right) \le \frac{\gamma'_{n-k}}{\gamma_{n-k}} < 1, \quad 1 \le k \le n-1. \tag{4.11}$$

It follows from this and an induction that $\beta_{m,k} > 0$ for all m, k, where we note that $\beta_{n,k} > 0$, $\beta_{m,0} = \gamma_n\beta_{m-1,n-1}$ and $\beta_{m,k} = \gamma_{n-k}\beta_{m-1,n-1} - \beta_{m-1,k-1}$ for $k \ge 1$.

We now show that $\frac{\gamma_k}{\gamma'_k} \ge 1 + \frac{\lambda_1}{(n-1)\lambda_n}$ for $1 \le k \le n-1$. This clearly holds for $k = 1$. For $2 \le k \le n-1$, we have

$$\gamma_k = \gamma'_k + \lambda_1\gamma_{k-1} \ge \gamma'_k + \lambda_1\gamma'_{k-1}.$$

Note that

$$(n-1)\lambda_n\gamma'_{k-1} \ge (\lambda_2 + \cdots + \lambda_n)\gamma'_{k-1} \ge \gamma'_k,$$

from which it follows that

$$\frac{\gamma_k}{\gamma'_k} \ge 1 + \frac{\lambda_1}{(n-1)\lambda_n}.$$

Now, combining this with (4.11), we obtain (4.7). □

If $\frac{\lambda_1}{\lambda_n}$ is not too small, the ratio in (4.7) is bounded away from 1 and hence $\beta_{m,k} = \gamma_{n-k}\beta_{m-1,n-1} - \beta_{m-1,k-1} > 0$ is computed with no cancelation.

Remark 4.4. If the eigenvalues λ_i, $1 \le i \le n$, are complex with positive real parts, then γ_k, $0 \le k \le n-1$, of the characteristic polynomial are positive. However, $\beta_{m,k}$ may not be positive. For example, if A is a 3×3 matrix with characteristic polynomial $(x-1)(x^2 - ax + b)$ with $a > 0, b > 4a^2$, then

$$\beta_{4,2} = (1-a)^2 - 4b$$

and $\beta_{4,2} < 0$ for sufficiently large b. Therefore the algorithm developed in this section is for essentially non-negative matrices with real eigenvalues.

4.2. **Positivity and computation of α_k.** By (4.5), for those k such that $n-k-1$ is even, α_k is non-negative and is computed with addition operations only. For those k such that $n-k-1$ is odd, a subtraction is involved. However, the following result shows that this subtraction does not lead to any cancelation and α_k remains positive if $\lambda_n < \frac{\sqrt{5}-1}{2}$.

Theorem 4.5. *Let $0 < \lambda_1 \le \cdots \le \lambda_n$ be the eigenvalues of A. If $\lambda_n < \frac{\sqrt{5}-1}{2}$, then*

$$\sum_{m=n}^{\infty} \frac{k!}{m!}\beta_{m,k} \le \frac{\lambda_n^2}{2(1-\lambda_n)} < \frac{1}{2}.$$

In particular, $\alpha_k > 0$ for $0 \le k \le n-1$.

Proof. We have, for $1 \le k \le n$,

$$\gamma_k = \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} \lambda_{i_1}\lambda_{i_2}\ldots\lambda_{i_k} \le \frac{n!}{k!(n-k)!}\lambda_n^k.$$

Furthermore, it follows from (4.3) and Theorem 4.3 that, for $m > n$,

$$\beta_{m,k} \le \gamma_{n-k}\beta_{m-1,n-1} \quad \text{and} \quad \beta_{m-1,n-1} \le \gamma_1\beta_{m-2,n-1}.$$

Repeatedly using the second inequality above and noting $\beta_{n,n-1} = \gamma_1$, we have

$$\beta_{m,k} \leq \gamma_{n-k}\gamma_1^{m-n}, \qquad 0 \leq k \leq n-1.$$

Hence, for k such that $0 \leq k \leq n-1$ and $n-k-1$ is odd, we have

$$\begin{aligned}\sum_{m=n}^{\infty} \frac{k!}{m!}\beta_{m,k} \leq & k! \sum_{m=n}^{\infty} \frac{\gamma_{n-k}\gamma_1^{m-n}}{m!} \\ \leq & k! \sum_{m=n}^{\infty} \frac{n! n^{m-n}}{k!(n-k)!m!}\lambda_n^{m-k} \\ \leq & k! \sum_{m=n}^{\infty} \frac{1}{k!(n-k)!}\lambda_n^{m-k} \\ \leq & \frac{1}{(n-k)!} \sum_{m=n}^{\infty} \lambda_n^{m-k} \\ = & \frac{1}{(n-k)!} \frac{\lambda_n^{n-k}}{1-\lambda_n} \\ \leq & \frac{\lambda_n^2}{2(1-\lambda_n)} < \frac{1}{2},\end{aligned}$$

where we note that $n-k-1$ is at least 1. Then α_k is positive and the theorem is proved. □

It follows from the theorem that the computation of α_k involves no cancelation. We now discuss how to truncate the series in α_k. If we truncate the series at $m = m_0$, α_k is approximated by

$$\alpha_{k,m_0} := 1 + (-1)^{n-k-1} \sum_{m=n}^{m_0} \frac{k!}{m!}\beta_{m,k}.$$

Then, we can bound the error as

$$\begin{aligned} err_{m_0} := & |\alpha_k - \alpha_{k,m_0}| \\ = & \sum_{m=m_0+1}^{\infty} \frac{k!}{m!}\beta_{m,k} \\ \leq & \sum_{m=m_0+1}^{\infty} \frac{k!}{m!}\gamma_{n-k}\gamma_1^{m-n} \\ = & \frac{k!}{(m_0+1)!}\gamma_{n-k}\gamma_1^{m_0+1-n} \sum_{m=m_0+1}^{\infty} \frac{\gamma_1^{m-m_0-1}}{m(m-1)\cdots(m_0+2)} \\ \leq & \frac{k!\gamma_{n-k}\gamma_1^{m_0+1-n}}{(m_0+1)!} \sum_{m=m_0+1}^{\infty} \frac{(n\lambda_n)^{m-m_0-1}}{m(m-1)\cdots(m_0+2)} \\ \leq & \frac{1}{1-\lambda_n} \frac{k!\gamma_{n-k}\gamma_1^{m_0+1-n}}{(m_0+1)!}.\end{aligned}$$

We therefore determine the truncation term m_0 by

$$\frac{k!\gamma_{n-k}\gamma_1^{m_0+1-n}}{(m_0+1)!} \leq tol \cdot (1-\lambda_n)\alpha_{k,m_0}, \quad 0 \leq k \leq n-1$$

such that

$$|\alpha_{k,m_0} - \alpha_k| \leq (tol + O(tol^2)) \cdot \alpha_k.$$

4.3. **Characteristic polynomial.** The computations of $\beta_{m,k}$'s begins with γ_i, $1 \leq i \leq n$, the coefficients of the characteristic polynomial

$$\det(zI - A) = z^n + \sum_{i=0}^{n-1}(-1)^{n-i}\gamma_{n-i}z^i.$$

Let

$$Q_k(z) = (z-\lambda_1)(z-\lambda_2)\cdots(z-\lambda_k) = z^k - \sigma_1^{(k)}z^{k-1} + \sigma_2^{(k)}z^{k-2} - \cdots + (-1)^k\sigma_k^{(k)}.$$

The coefficients of $Q_{k+1}(z)$ can be obtained from those of $Q_k(z)$ via the relation

$$\sigma_j^{(k+1)} = \sigma_j^{(k)} + \lambda_{k+1}\sigma_{j-1}^{(k)}, \quad 1 \leq j \leq k+1$$

with the convention that $\sigma_0^{(k)} = 1$ and $\sigma_{k+1}^{(k)} = 0$. This results in the following recurrence algorithm for γ_i, $1 \leq i \leq n$.

Algorithm 2. *Compute the Coefficients of Characteristic Polynomial*

```
Input:      $\lambda_1, \lambda_2, \ldots, \lambda_n$
Initialize: $\gamma_1 = \lambda_1$, set $\gamma_0 = 1$
for $i = 2 : n$,
    $\gamma_i = \lambda_i\gamma_{i-1}$
    for $j = 1 : i-1$,
        $\gamma_{i-j} = \gamma_{i-j} + \lambda_i\gamma_{i-j-1}$
    end
end
```

Thus $\gamma_k' s$ can be obtained in $O(n^2)$ operations. Noting that if $\lambda_i \geq 0$ (for all i), no substraction is involved in the above process. Consequently, all γ_i are computed accurately.

4.4. **Accurate implementation.** When applying the polynomial method to compute $\exp(A)$ via (4.4), the entrywise relative accuracy is guaranteed if A is non-negative, and α_k's are non-negative and accurately computed. We show in this subsection how to achieve this for symmetric or triangular essentially non-negative matrices by combining the polynomial method with the shifting as well as the scaling and squaring techniques.

We first choose an α and shift A to $A_\alpha = A - \alpha I$. Then we scale A_α to $A_\alpha/2^p$ and compute $\exp(A_\alpha/2^p)$ by the polynomial method. Finally, square $\exp(A_\alpha/2^p)$ repeatedly p times and obtain $\exp(A)$ via $\exp(A) = e^\alpha \exp(A_\alpha)$. The key of this method is the choice of α and p, which should ensure that $\exp(A_\alpha/2^p)$, as computed via the polynomial method, satisfies the conditions as discussed in previous subsections, as well as that the entrywise relative error caused by forming A_α and magnified in the squaring process is of order $\kappa_{\exp}(A)\mathbf{u}$.

Let $\lambda_1^{(d)} \leq \lambda_2^{(d)} \leq \cdots \leq \lambda_n^{(d)}$ be the eigenvalues of the non-negative matrix $A_d = A - dI$, where $d = \min_i a_{ii}$. Note that $\lambda_n^{(d)} = \rho(A_d)$ is the Perron eigenvalue of

A_d. If A is triangular, then $\lambda_k^{(d)}$, $1 \le k \le n$, are explicitly known. If A is symmetric, we compute the eigenvalues by the QR algorithm in roughly $\frac{4n^3}{3}$ flop operations with absolute error in the order of $\mathbf{u}\|A_d\|_2$, i.e., $\mathbf{u}\rho(A_d)$ as $\rho(A_d) = \|A_d\|_2$. To make A_α non-negative, α is no greater than d. We write $\alpha = d - c\rho(A_d)$ with $c \ge 0$ to be determined. Denote by $\lambda_1^{(\alpha)} \le \lambda_2^{(\alpha)} \le \cdots \le \lambda_n^{(\alpha)}$ the eigenvalues of A_α. Then, $\lambda_k^{(\alpha)} = \lambda_k^{(d)} + c\rho(A_d)$, $1 \le k \le n$, and $\lambda_1^{(\alpha)}/2^p \le \lambda_2^{(\alpha)}/2^p \le \cdots \le \lambda_n^{(\alpha)}/2^p$ are the eigenvalues of $A_\alpha/2^p$. To make $\beta_{m,k}$'s positive and computed with high relative accuracy when applying the polynomial method to $A_\alpha/2^p$, we need

$$0 < \lambda_1^{(\alpha)}/2^p \le \lambda_2^{(\alpha)}/2^p \le \cdots \le \lambda_n^{(\alpha)}/2^p$$

and $\lambda_n^{(\alpha)}/\lambda_1^{(\alpha)}$ is well bounded. We choose $c = 2$. Then all the eigenvalues of A_α are in the interval $[\rho(A_d), 3\rho(A_d)]$ and can be computed with relative error in the order of $O(\mathbf{u})$. To make α_k's non-negative and computed with high relative accuracy, $(\lambda_n^{(\alpha)}/2^p)^2/2(1-\lambda_n^{(\alpha)}/2^p)$ is less than and bounded away from 1 by Theorem 3. We then choose p as the largest non-negative integer such that

$$\frac{3\rho(A_d)}{2^p} < \frac{\sqrt{5}-1}{2}$$

and by Theorem 4.5,

$$\frac{(\lambda_n^{(\alpha)}/2^p)^2}{2(1-\lambda_n^{(\alpha)}/2^p)} < \frac{1}{2}.$$

Note that if $\rho(A_d)$ is small, then $p = 0$. We point out that we can also use a smaller α to increase the eigenvalues which can prevent possible underflows in computing γ_k's due to small eigenvalues.

4.5. Complete algorithm.

We summarize the discussion of this section in the following algorithm for computing the exponential of a symmetric, or triangular essentially non-negative matrix.

Algorithm 3. *Polynomial Method for* $\exp(A)$

```
Input: an $n \times n$ symmetric or triangular essentially non-negative matrix $A$,
       tol (error tolerance);
Output: $E$
    $d = \min a_{ii}$; $A_d = A - dI$;
    Compute the eigenvalues $\lambda_1^{(d)} < \lambda_2^{(d)} < \cdots < \lambda_n^{(d)}$ of $A_d$ and let $\rho = \lambda_n^{(d)}$;
    $p = \max(0, \lceil \log_2(6\rho/(\sqrt{5}-1)) \rceil)$; $B = (A_d + 2\rho I)/2^p$;
    Compute $(\lambda_k^{(d)} + 2\rho)/2^p,\ 1 \le k \le n$, the eigenvalues of $B$,
    and let $\tau = (\lambda_n^{(d)} + 2\rho)/2^p$;
    Compute $\gamma_1, \gamma_2, \cdots, \gamma_n$ for $B$ by Algorithm 2;
    For $k = 0 : n-1$, $\beta_k = \gamma_{n-k}$; End
    For $k = 0 : n-1$, $\alpha_k = 1 + (-1)^{n-k-1}\frac{k!}{n!}\beta_k$; End
    $m = n$;
    While $\frac{k!\gamma_{n-k}\gamma_1^{m+1-n}}{(m+1)!} \ge (1-\tau)tol \cdot \alpha_k$ for any $0 \le k \le n-1$;
        $m = m + 1$ and $t = \beta_{n-1}$;
        For $k = n-1 : 1$, $\beta_k = \gamma_{n-k}t - \beta_{k-1}$; End
        $\beta_0 = \gamma_n t$;
        For $k = 0 : n-1$, $\alpha_k = \alpha_k + (-1)^{n-k-1}\frac{k!}{m!}\beta_k$; End
```

```
14. End
15. $E = \sum_{k=0}^{n-1} \alpha_k B^k/k!$;
16. $E = e^{(d+\rho)/2^p} e^{-\tau} E$;
17. For $i = 1 : p$, $E = E^2$, End.
```

Remark 4.6. (a) The process is very similar to the Taylor series method except that we compute α_k and fix the degree of the polynomial at $n-1$. Since the coefficients α_k are computed accurately as discussed, an error analysis for this algorithm will be similar to that for the Taylor method. In particular, the entrywise relative errors will be in the order of $n^2\kappa_{\exp}(A)\mathbf{u}$. (b) With a slight modification, this algorithm applies to any essentially non-negative matrices which are diagonalizable and whose eigenvalues are real, since the eigenvalues can be computed by the QR algorithm with absolute error $O(\mathbf{u})$, modulus a scaling by the condition numbers for the eigenvalues. (c) The computational complexity of this algorithm is $O(n^3)$ when applied to sparse matrices with $O(n)$ nonzero entries, which is comparable to other methods for computing matrix exponentials, e.g., the Padé approximation. (d) If A is dense, the main cost of the polynomial methods is the matrix-matrix multiplications. However, it can be significantly reduced by a strategy used in [16]. For simplicity of exposing, suppose that $n = ml$ with $m \le l$. Then $\exp(A)$ can be written as

$$\exp(A) = F(m,0) + A^m F(m,1) + A^{2m} F(m,2) + \cdots + A^{(l-1)m} F(m,l-1), \tag{4.12}$$

where

$$F(m,k) = \alpha_{km} I + \alpha_{km+1} A + \cdots + \alpha_{km+m-1} A^{m-1}, \quad 0 \le k \le l-1.$$

Then, it totally costs $m+l-2$ matrix multiplications: $m-1$ matrix multiplications in forming $A^2,\ A^3, \ldots, A^m$ and $l-1$ matrix multiplications by applying the Horner's algorithm to (4.12).

5. Numerical examples

In this section, we present three numerical examples of computing exponentials of essentially non-negative matrices to demonstrate the entrywise high relative accuracy achieved by the two new algorithms. We compare them with `expm` of MATLAB.

Example 1. We first test the algorithms on symmetric tridiagonal matrices. Consider the $n \times n$ tridiagonal matrix $-T_n$ whose diagonal elements are -2 and the off-diagonal elements are 1 (i.e., T_n is the discretization of 1-dimensional Laplacian operator). The matrix has the known eigenvalue decomposition $T_n = Z\Lambda Z^T$ with $Z = \left[\sqrt{\frac{2}{n+1}} \sin\frac{jk\pi}{n+1}\right]_{j,k=1}^{n}$ and $\Lambda = \mathrm{diag}\{2\left(1 - \cos\frac{j\pi}{n+1}\right)\}$; see [9, p. 268]. We compute $\exp(-T_n) = Z\exp(-\Lambda)Z^T$ using MATLAB's Symbolic Toolbox with 100-digit arithmetic and consider the result E exact for comparisons. We then compute $\exp(T_n)$ by the Taylor series method (Algorithm 1), the Polynomial method (Algorithm 3), and MATLAB's `expm` and we compare them with E. Denote the three computed results by E_1, E_2 and E_3 respectively, and the largest entrywise relative error by err_1, err_2 and err_3.

In Figure 1, we plot the entries of E for $n = 50$ as a mesh surface in (a) and, for each of the three methods tested (Taylor, Polynomial, and `expm`, resp.), we plot `abs(E − Ei)./E` (i.e., the matrix of the entrywise relative errors of the computed result E_i) in logarithmic scale in (b), (c) and (d), respectively.

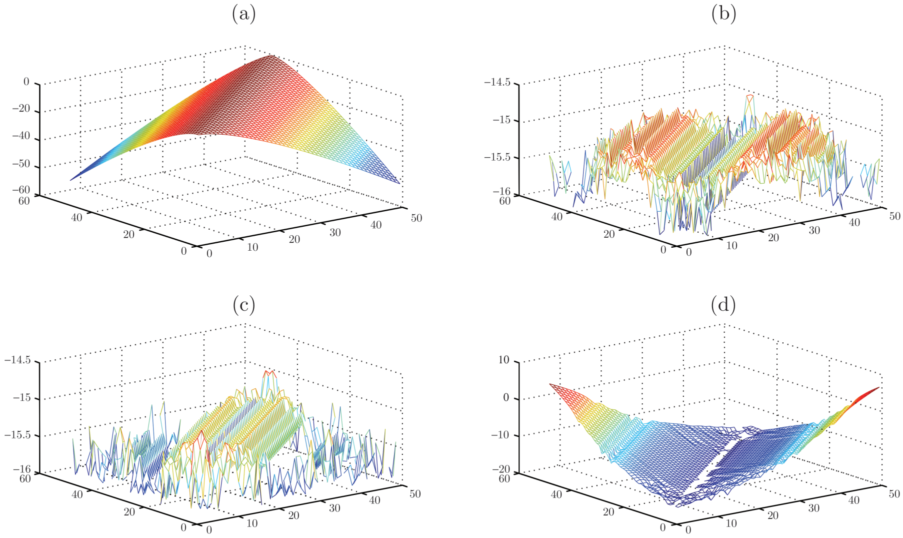

FIGURE 1. (a) $E = \exp(-T_n)$ (exact); (b) `abs(E − E1)./E` (Taylor); (c) `abs(E − E2)./E` (Polynomial); (d) `abs(E − E3)./E` (MATLAB's `expm`)

We see that the entries of $\exp(T_n)$ decays by several orders of magnitude away from the main diagonal and correspondingly the accuracy of `expm` deteriorates as entries become very small. The Taylor series method and the Polynomial method are, however, unaffected by the scaling of the entries and maintain the entrywise relative error in the order of the machine precision.

We have also tested the programs for different values of n. In Table 1, we list the largest entrywise relative error for $n = 30, 35, 40, 45, 50$. As n increases, the smallest entry of E decreases. The entrywise relative error increases for `expm` while the other methods are unaffected.

TABLE 1. Maximal entrywise relative errors of E_1 (Taylor), E_2 (Polynomial), E_3 (MATLAB's `expm`)

n	30	35	40	45	50
err_1	$1.2e-15$	$1.4e-15$	$1.4e-15$	$1.4e-15$	$1.4e-15$
err_2	$1.6e-15$	$1.6e-15$	$1.5e-15$	$1.6e-15$	$1.6e-15$
err_3	$8.9e-8$	$2.1e-4$	$6.9e-1$	$4.7e+3$	$2.6e+6$

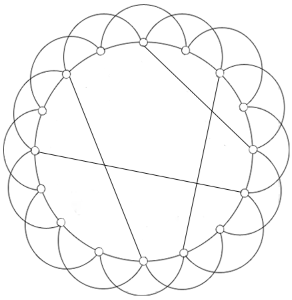

FIGURE 2. A small-world network of 16 nodes with $k = 2$ and four shortcuts

Example 2. Consider the negative of the discretization of 2-dimensional Laplacian operator, i.e., the matrix $-T_{m\times n} := -T_m \otimes I_n - I_m \otimes T_n$ where $\otimes$ denote the Kronecker product; see [9, p. 268]. Then it is easy to check that $\exp(-T_{m\times n}) = \exp(-T_m) \otimes \exp(-T_n)$ We compute $\exp(-T_{m\times n})$ through computing $\exp(-T_m)$ and $\exp(-T_n)$ using MATLAB's Symbolic Toolbox with 100-digit arithmetic and we consider the result E exact for comparisons. We then compute $\exp(-T_{m\times n})$ by the Taylor series method, the Polynomial method, and MATLAB's `expm` and compare them with E. Again, denote the three computed results by E_1, E_2 and E_3, and the maximal entrywise relative errors err_1, err_2 and err_3, respectively. In Table 2, we present the results for $T_{25\times25}$, $T_{25\times30}$, $T_{25\times35}$, $T_{25\times40}$, $T_{30\times30}$. Again, the Taylor series method and the polynomial method produce solutions with high entrywise relative accuracy, while `expm` may not compute extremely small entries with any relative accuracy.

TABLE 2. Maximal entrywise relative errors of E_1 (Taylor), E_2 (Polynomial), E_3 (MATLAB's `expm`)

$m \times n$	25×25	25×30	25×35	25×40	30×30
err_1	$3.9e-15$	$4.1e-15$	$4.0e-15$	$3.8e-15$	$3.9e-15$
err_2	$2.7e-14$	$2.7e-14$	$2.7e-14$	$2.6e-14$	$2.7e-14$
err_3	$8.6e-5$	$2.0e-2$	2.7	$1.5e+2$	2.7

Example 3. We consider a Watts-Strogatz model of 'small-world' networks [27] as generated by the function `smallw.m` in MATLAB toolbox CONTEST available at: http://www.maths.strath.ac.uk/research/groups/numerical-analysis/contest. The syntax `smallw(n,k,p)` returns a network of n nodes, beginning with a k-nearest ring (nodes i and j are connected if and only if $|i-j| \leq k$ or $|n-|i-j|| \leq k$), and then with a fixed probability p each node being given an extra link—a short cut—connecting it to a node chosen uniformly at random across the network; see Figure 2. The following tested network is produced by `smallw(200,2,0.03)`. It is a 2-nearest ring network of 200 nodes, with four shortcuts connecting the pairs of

nodes (16,30), (74,85), (90,128) and (138,147). The resulting adjacency matrix A is a five-diagonal symmetric matrix of order 200, with 14 extra off-diagonal nonzeros. Since we do not have $\exp(A)$ exactly, we use the solutions generated by the Taylor series method as the "accurate" solutions, based on which we measure the entrywise relative errors of the solutions produced by the polynomial method and `expm`. The entries of $\exp(A)$ vary greatly in magnitudes, from 4.48e-51 to 9.15.

The relative error for each entry of $\exp(A)$ produced by the polynomial method is in the order of $1.0e-14$; however, the relative errors for over 20% of the entries of $\exp(A)$ produced by `expm` are bigger than 1. We also compare the algorithms in computing the *betweenness* according to (1.3) for all the nodes, with $\exp(A)$ and $\exp(A - E_r)$ computed by the Taylor series method, the polynomial method and `expm` respectively. The betweenness lies in the interval $[0.0139, 0.3721]$ for this network. The relative errors of the betweenness computed by the polynomial method are less than 1.0e-13 for all nodes. However, the relative errors of the betweenness computed by `expm` are over 0.01 for nearly 90% of the nodes and over 0.1 for 10% of the nodes.

We see that even for this fairly small size network, the entries of the exponential vary so significantly that smaller entries as computed by `expm` have poor or no relative accuracy. As a result, certain measures of network properties cannot be computed with any accuracy by the traditional algorithms. The new algorithms developed in this paper do produce accurate results and would be necessary for this type of problem.

6. Concluding remarks

We have shown how the Taylor series method and the polynomial method can be carefully implemented to compute the exponential of an essentially non-negative matrix entrywise to high relative accuracy. The resulting algorithms have entrywise forward relative errors comparable to those already made in rounding the matrix. Numerical examples demonstrate the entrywise relative accuracy achieved by these algorithms.

When the matrices are sparse, the polynomial method has a computational complexity of $O(n^3)$ which is comparable to traditional methods such as the Padé approximation. In general, however, rational approximations such as the Padé approximation are more efficient than a polynomial approximation method. However, as discussed in the introduction, there appear to be some unsurmountable difficulties associated with the rational approximations in terms of accurate inversion as well as determining the degree of rational approximations to ensure entrywise relative accuracy, which may be unbounded anyway. It appears that the potential extra computational complexity associated with the Taylor series method and the polynomial approximation methods may be a necessary cost to achieve the high entrywise relative accuracy.

Acknowledgment

We would like to thank an anonymous referee for pointing out references [6, 13].

References

1. A. H. AI-Mohy and N. J. Higham, *A new scaling and squaring algorithm for the matrix exponential,* SIAM J. Matrix Anal. Appl. **31** (2009), 970–989. MR2538661 (2010i:15016)
2. A. S. Alfa, J. Xue and Q. Ye, *Accurate computation of the smallest eigenvalue of a diagonally dominant M-matrix,* Math. Comp. **71** (2002), 217–236. MR1862996 (2002h:65054)
3. M. Arioli, B. Codenotti and C. Fassino, *The Padé method for computing the matrix expoential,* Linear Algebra Appl. **240** (1996), 111–130. MR1387290 (98f:65016)
4. M. Benzi and P. Boito, *Quadrature rule-based bounds for functions of adjacency matrices,* Linear Algebra Appl. **433** (2010), 637–652. MR2653828 (2011d:65106)
5. M. Benzi and G. H. Golub, *Bounds for the entries of matrix functions with applications to preconditioning,* BIT **39** (1999), 417–438. MR1708693 (2000h:65047)
6. B. Codenotti and C. Fassino, *Error analysis of two algorithms for the computation of the matrix exponential,* Calcolo **29** (1992), 1–31. MR1219619 (94c:65057)
7. L. Dieci and A. Papini, *Padé approximation for the exponential of a block triangular matrix,* Linear Algebra Appl. **308** (2000), 183–202. MR1751139 (2001d:65057)
8. L. Deng and J. Xue, *Accurate computation of exponentials of triangular essentially non-negative matrices,* Journal of Fudan University (Natural Science) **50** (2011), 78–86. MR2841355
9. J. Demmel, *Applied Numerical Linear Algebra,* SIAM, Philadelphia, 1997. MR1463942 (98m:65001)
10. E. Estrada and J. A. Rodríguez-Velázqez, *Subgraph centrality in complex networks,* Phys. Rev. E **71** (2005), article 056103. MR2171832 (2006d:94103)
11. E. Estrada, N. Hatano and J. A. Rodríguez-Velázqez, *Communicability in complex networks,* Phys. Rev. E **77** (2008), article 036111. MR2495430 (2010i:91171)
12. E. Estrada, D. J. Higham and N, Hatano, *Communicability betweenness in complex networks,* Phys. A **388** (2009), 764–774.
13. C. Fassino, *Computation of Matrix Function,* PhD thesis, University of Pisa, Dottorato in Informatica, 1993.
14. W. Grassmann, *Transient solutions in Markovian queueing systems,* Comput. Opns. Res. **4** (1977), 47–56.
15. D. Gross and D. R. Miller, *The randomization technique as modeling tool and solution procedure for transient Markov processes,* Operations Research **32** (1984), 343–361, 1984. MR747747 (86b:60125)
16. N. J. Higham, *The scaling and squaring method for the matrix exponential revisited,* SIAM J. Matrix Anal. Appl. **26** (2005), 1179–1193. MR2178217 (2006g:65066)
17. N. J. Higham, *Functions of Matrices, Theory and computation,* SIAM, Philadelphia, 2008. MR2396439 (2009b:15001)
18. C. B. Moler and C. Van Loan, *Nineteen dubious ways to compute the exponential of a matrix,* SIAM Rev. **20** (1978), 807–836. MR508383 (80c:15004)
19. C. B. Moler and C. Van Loan. *Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later,* SIAM Rev. **45** (2003), 3–49. MR1981253 (2004d:15012)
20. I. Najfeld and T. F. Havel, *Derivatives of the matrix exponential and their computation,* Adv. Appl. Math. **16** (1995), 321–375. MR1342832 (96f:65013)
21. B. N. Parlett and K. C. Ng, *Development of an accurate algorithm for* exp*(Bt),* Tech. Report PAM-294, Center for Pure and Applied Mathematics, University of California, Berkeley, CA, 1985.
22. A. V. Ramesh and K. S. Trivedi, *On the Sensitivity of Transient Solution of Markov Models,* Proc. 1993 ACM SIGMETRICS Conference, Santa Clara, CA, May 1993.
23. R. B. Sidje, *Expokit: A software package for computing matrix exponentials,* ACM Trans. Math. Soft. **24** (1998), 130–156.
24. C. Van Loan, *The sensitivity of the matrix exponential,* SIAM J. Numer. Anal. **14** (1977), 971–981. MR0468137 (57:7975)
25. R. S. Varga, *Matrix Iterative Analysis,* Springer, 2000. MR1753713 (2001g:65002)
26. R. C. Ward, *Numerical computation of the matrix exponential with accuracy estimate,* SIAM J. Numer Anal. **14** (1977), 600–610. MR0445806 (56:4140)
27. D. J. Watts and S. H. Strogatz, *Collective dynamics of 'small-world' networks,* Nature **393** (1998), 440–442.

28. J. Xue and Q. Ye, *Entrywise relative perturbation bounds for exponentials of essentially nonnegative matrices,* Numer. Math. **110** (2008), 393–403. MR2430985 (2009e:65073)

School of Mathematical Science, Fudan University, Shanghai, 200433, China
E-mail address: xuej@fudan.edu.cn

Department of Mathematics, University of Kentucky, Lexington, Kentucky 40506-0027
E-mail address: qye@ms.uky.edu