Algorithms for the computation of the matrix logarithm based on the double exponential formula

Fuminori Tatsuoka†, Tomohiro Sogabe‡, Yuto Miyatake‡, Shao-Liang Zhang§

January 24, 2019

Abstract

We consider the computation of the matrix logarithm by using numerical quadrature. The efficiency of numerical quadrature depends on the integrand and the choice of quadrature formula. The Gauss–Legendre quadrature has been conventionally employed; however, the convergence could be slow for ill-conditioned matrices. This effect may stem from the rapid change of the integrand values. To avoid such situations, we focus on the double exponential formula, which has been developed to address integrands with endpoint singularity. In order to utilize the double exponential formula, we must determine a suitable finite integration interval, which provides the required accuracy and efficiency. In this paper, we present a method for selecting a suitable finite interval based on an error analysis as well as two algorithms, and one of these algorithms addresses error control.

1 Introduction

A logarithm of \( A \in \mathbb{R}^{n \times n} \) is defined as any matrix \( X \) such that

\[
\exp(X) = A,
\]

where \( \exp(X) := I + X + \frac{1}{2!}X^2 + \frac{1}{3!}X^3 + \cdots \) \([7, p. 269]\). If all eigenvalues of \( A \) lie in the set \( \mathbb{C}\setminus(-\infty, 0] \), there is a unique logarithm of \( A \) all of whose eigenvalues lie in the strip \( \{z \in \mathbb{C} : -\pi < \text{Im}(z) < \pi \} \) \([7, \text{ Thm. 1.31}]\). This logarithm is called the principal logarithm of \( A \), and denoted by \( \log(A) \). Throughout this paper, we assume that all eigenvalues of \( A \) lie in the set \( \mathbb{C}\setminus(-\infty, 0] \), and we consider the principal logarithm of \( A \).

The matrix logarithm is utilized in many fields of research, such as quantum mechanics \([13]\), quantum chemistry \([6]\), biomolecular dynamics \([8]\), buckling simulation \([11]\), and deep learning \([9]\). The computational methods include the inverse scaling and squaring (ISS) algorithm \([1]\), an algorithm based on the arithmetic-geometric mean (AGM) iteration \([3]\), and numerical quadrature. In this paper, we focus on numerical quadrature, which employs the following integral representation (see e.g. \([2, \text{ Thm. 11.1}]\)):

\[
\log(A) = (A - I) \int_0^1 [t(A - I) + I]^{-1} \, dt. \quad (1)
\]
We use the numerical quadrature method for two reasons. First, numerical quadrature can make use of the sparseness of $A$ if $A$ is sparse; i.e., $\log(A)$ can be computed column by column without computing and storing dense matrices. Conversely, the ISS algorithm and the algorithm based on AGM iteration include the computation of the matrix square root, which means that the calculation involves dense matrices even if $A$ is sparse. The second reason is that numerical quadrature is potentially more favorable for parallel computers because of independent computation of the integrand on each abscissa.

Because the integrand in (1) includes matrix inversion, the computational cost of numerical quadrature depends on the number of evaluations of the integrand. Although numerical quadrature is suitable for parallelizing, the quadrature formula should be selected carefully to reduce the computational cost and save computational resources.

The method conventionally used to compute (1) is the Gauss–Legendre (GL) quadrature. If the spectral radius of $A - I$ is smaller than 1, the GL quadrature, which can be regarded as a rational approximation of $\log(A)$ coincides with the Padé approximation of $\log(A)$ around $I$ [5 Thm. 4.3]. Therefore, it is natural to use the GL quadrature to reduce the number of abscissas when $A$ is close to $I$. However, the convergence of the GL quadrature becomes slow when $A$ is not close to $I$. For example, the convergence in our experiments became slow when $A$ was ill-conditioned which may be explained by rapid changes in the integrand value when it is closer to the endpoint of the interval.

In this paper, we consider the double exponential (DE) formula [12], which can be used to compute integrals in scenarios in which the GL quadrature does not perform well. However, when using the DE formula, a finite interval needs to be selected because the integrand in (1) is transformed into a corresponding function on the infinite interval. If the finite interval is too narrow, the accuracy of the computational result becomes low, but if it is too wide, the convergence of the DE formula becomes slow.

By performing an error analysis, we provide a method of selecting the appropriate finite interval, as well as two algorithms for the computation of $\log(A)$ based on the $m$-point DE formula.

The remainder of this paper is organized as follows: in Section 2, we present an error analysis and propose two algorithms; in Section 3, we show the results of numerical experiments; in Section 4, we conclude the study.

**Notation:** Unless otherwise stated, $\| \cdot \|$ denotes a consistent matrix norm, e.g., the $p$-norm or the Frobenius norm, and $\| \cdot \|_2$ and $\| \cdot \|_F$ denote the 2-norm and the Frobenius norm, respectively. The inverse functions of sinh and tanh are referred to as arcsinh and artanh, respectively.

## 2 Algorithms for the computation of $\log(A)$ based on the DE formula

In this section we propose a method of selecting a finite interval for the DE formula by estimating the interval truncation error and present two algorithms. Before considering the truncation error, let us apply the following transformations to (1). By applying $u = 2t - 1$ substitution to (1), we obtain

$$\log(A) = (A - I) \int_{-1}^{1} [(1 + u)(A - I) + 2I]^{-1} \, du. \quad (2)$$

Then, by applying the DE transformation $u = \tanh(\sinh(x))$, it follows that

$$\log(A) = (A - I) \int_{-\infty}^{\infty} F_{DE}(x) \, dx, \quad (3)$$

where

$$F_{DE}(x) := \cosh(x) \sech^2(\sinh(x)) \left[ (1 + \tanh(\sinh(x)))(A - I) + 2I \right]^{-1}. \quad (4)$$

---

1. By multiplying the integral representation of $\log(A)$ [1] from both left and right by the $i$-th vector of the standard basis of $\mathbb{R}^n$, $e_i$, the integrand becomes $[(A - I) + I]^{-1} e_i$. Then, the $i$-th column of $\log(A)$ can be computed using sparse solvers of linear systems, without the direct computation of $\log(A)$ for sparse $A$.

2. A matrix norm is consistent if $\|AB\| \leq \|A\|\|B\|$ for all $A \in \mathbb{C}^{m \times n}$ and $B \in \mathbb{C}^{n \times p}$. See, e.g., [7] p. 327.
Subsection 2.1 shows the estimation of the upper bound for the error between the integral in (3) and the same integral defined in the finite interval \([l, r]\),

\[
\left\| \log(A) - (A - I) \int_l^r F_{DE}(x) \, dx \right\|.
\]  

(4)

In subsection 2.2, we propose a method of setting the interval \([l, r]\) so that the relative truncation error is small than or approximately equal to the given tolerance \(\epsilon\). Our algorithms are described in subsection 2.3.

2.1 Estimation of the error from the interval truncation

The error that stems from the interval truncation (4) can be rewritten as

\[
\log(A) - (A - I) \int_l^r F_{DE}(x) \, dx = (A - I) \left[ \int_{-\infty}^l F_{DE}(x) \, dx + \int_r^\infty F_{DE}(x) \, dx \right].
\]  

(5)

Using the triangle inequality for the right-hand side (RHS) of (5), it holds that

\[
\left\| (A - I) \left[ \int_{-\infty}^l F_{DE}(x) \, dx + \int_r^\infty F_{DE}(x) \, dx \right] \right\| \leq \left\| (A - I) \int_{-\infty}^l F_{DE}(x) \, dx \right\| + \left\| (A - I) \int_r^\infty F_{DE}(x) \, dx \right\|.
\]  

(6)

By estimating the RHS of (6), we obtain an upper bound for (4).

Initially, we focus on the first term which is on the RHS of (6). To avoid cumbersome notation, instead of \(F_{DE}\), which including hyperbolic functions, we consider the integrand of (1). By applying the transformation \(t = \frac{\tanh(\sinh(x)) + 1}{2}\), we have

\[
\left\| (A - I) \int_{-\infty}^l F_{DE}(x) \, dx \right\| = \left\| (A - I) \int_0^a F(t) \, dt \right\|,
\]  

(7)

where,

\[
F(t) := \left[ t(A - I) + I \right]^{-1}, \quad a := \frac{\tanh(\sinh(l)) + 1}{2}.
\]

The following lemma shows an upper bound for the RHS of (7), if \(a\) is small enough to warrant the use of the Neumann series expansion of \(F(t)\).

**Lemma 2.1.** Suppose that \(A \neq I\). Then, for \(a \in \left(0, 1/(2\|A - I\|)\right]\), we have

\[
\left\| (A - I) \int_0^a F(t) \, dt \right\| \leq \frac{3a}{2}\|A - I\|.
\]  

(8)

**Proof.** For all \(t \in [0, a]\) where \(a \in \left(0, 1/(2\|A - I\|)\right]\), it holds that

\[
t \leq \frac{1}{2\|A - I\|}.
\]

Hence,

\[
\|t(I - A)\| \leq \frac{1}{2} (< 1).
\]
By applying the Neumann series expansion to $F(t)$ we get the following:

$$F(t) = [t(A - I) + I]^{-1}$$
$$= [I - t(I - A)]^{-1}$$
$$= \sum_{k=0}^{\infty} t^k (I - A)^k.$$

Therefore, the integral of $F$ can be rewritten as follows:

$$(A - I) \int_{0}^{a} F(t) \, dt = (A - I) \int_{0}^{a} \left[ \sum_{k=0}^{\infty} t^k (I - A)^k \right] \, dt$$
$$= (A - I) \sum_{k=0}^{\infty} (I - A)^k \left[ \int_{0}^{a} t^k \, dt \right]$$
$$= (A - I) \sum_{k=0}^{\infty} \frac{a^{k+1}}{k+1} (I - A)^k$$
$$= a(A - I) + a(A - I) \sum_{k=1}^{\infty} \left[ \frac{a^k}{k+1} (I - A)^k \right].$$

By using triangle inequality and consistency of the norm we get the following:

$$\left\| (A - I) \int_{0}^{a} F(t) \, dt \right\| \leq a \| A - I \| + a \| A - I \| \sum_{k=1}^{\infty} \left[ \frac{1}{k+1} \| a^k (A - I)^k \| \right]$$
$$\leq a \| A - I \| + a \| A - I \| \sum_{k=1}^{\infty} \left[ \frac{1}{k+1} \| a(A - I) \| \| (A - I)^k \| \right]$$
$$\leq a \| A - I \| + a \| A - I \| \sum_{k=1}^{\infty} \left[ \frac{1}{k} \left( \frac{1}{2} \right)^k \right]$$
$$= a \| A - I \| + \frac{a}{2} \| A - I \|$$
$$= \frac{3a}{2} \| A - I \|.$$  

\[ \Box \]

The calculation to estimate the second term on the RHS of (6) is similar to that of the first term. By applying the transformation $t = [\tanh(\sinh(x)) + 1]/2$, we get the following:

$$\left\| (A - I) \int_{r}^{\infty} F_{DE}(x) \, dx \right\| = \left\| (A - I) \int_{b}^{1} F(t) \, dt \right\|,$$

where $b = [\tanh(\sinh(r)) + 1]/2$.

The following lemma shows an upper bound of (10), if $b$ is close enough to 1 to warrant the use of the Neumann series expansion of $F(t)$.

**Lemma 2.2.** For $b \in [2\|A^{-1}\|/(2\|A^{-1}\| + 1), 1)$, we have

$$\left\| (A - I) \int_{b}^{1} F(t) \, dt \right\| \leq \left( -\log(b) + \frac{1 - b}{2b} \right) \| A - I \| \| A^{-1} \|.$$  

\[ \text{To calculate the second line from the first line, term-by-term integration is applied to (9), as the series converge uniformly. This uniform convergence is shown by applying the Weierstrass M-test for each element of the integrand by using Jordan decomposition.} \]
Proof. The outline of this proof is similar to that of Lemma 2.1. For all \( t \in [b, 1] \) where \( b \in \left[ 2\|A^{-1}\|/(2\|A^{-1}\| + 1), 1 \right) \), it holds that

\[
t \geq \frac{2\|A^{-1}\|}{2\|A^{-1}\| + 1}.
\]

Hence,

\[
\left\| \frac{t - 1}{t} A^{-1} \right\| \leq \frac{1}{2} (< 1).
\]

By applying the Neumann series expansion to \( F(t) \), we get

\[
F(t) = \left[ t(A - I) + I \right]^{-1}
\]

\[
= \frac{1}{t} A^{-1} \left[ I - \frac{t - 1}{t} A^{-1} \right]^{-1}
\]

\[
= \frac{1}{t} A^{-1} \sum_{k=0}^{\infty} \left( \frac{t - 1}{t} \right)^k A^{-k}.
\]

Therefore, the integral of \( F \) can be rewritten as follows:

\[
(A - I) \int_b^1 F(t) \, dt = (A - I) \int_b^1 \left[ \frac{1}{t} A^{-1} \sum_{k=0}^{\infty} \left( \frac{t - 1}{t} \right)^k A^{-k} \right] \, dt
\]

\[
= (A - I) A^{-1} \sum_{k=0}^{\infty} \left[ A^{-k} \int_b^1 \frac{1}{t} \left( \frac{t - 1}{t} \right)^k \, dt \right]
\]

\[
= (A - I) A^{-1} \left[ \log(b) I + \sum_{k=1}^{\infty} A^{-k} \int_b^1 \frac{1}{t} \left( \frac{t - 1}{t} \right)^k \, dt \right].
\]

By using the triangle inequality and consistency of the norm, it follows

\[
\left\| (A - I) \int_b^1 F(t) \, dt \right\| \leq \| A - I \| \| A^{-1} \| \left\{ | - \log(b) | + \sum_{k=1}^{\infty} \left\| A^{-1} \right\| k \int_b^1 \frac{1}{t} \left| \frac{t - 1}{t} \right|^k \, dt \right\} \right\}.
\]

(12)

In (12), it holds that \( | - \log(b) | = - \log(b) \) because \( b \in \left[ 2\|A^{-1}\|/(2\|A^{-1}\| + 1), 1 \right) \), and \( |(t - 1)/t| = (1 - t)/t \) because \( t \in [b, 1] \). For the second term in the bracket on the RHS of (12), we have:

\[
\sum_{k=1}^{\infty} \left\| A^{-1} \right\| k \int_b^1 \frac{1}{t} \left( \frac{1 - t}{t} \right)^k \, dt \leq \sum_{k=1}^{\infty} \left\| A^{-1} \right\| k \int_b^1 \frac{1}{b} \left( \frac{1 - t}{b} \right)^k \, dt
\]

\[
= \sum_{k=1}^{\infty} \left\| A^{-1} \right\| k \frac{1}{k + 1} \left( 1 - b \right)^{k + 1}
\]

\[
= \frac{1 - b}{b} \sum_{k=1}^{\infty} \left[ \frac{1}{k + 1} \left\| A^{-1} \right\| \right]^k
\]

\[
\leq \frac{1 - b}{b} \sum_{k=1}^{\infty} \left[ \frac{1}{2} \left( \frac{1}{2} \right) \right]^k
\]

\[
= \frac{1 - b}{2b}.
\]

(13)

By substituting (13) in (12), we obtain the inequality (11).
In the final part of this subsection, we estimate the upper bound of (4).

**Proposition 2.1.** Suppose that \( A \neq I \). For a given \([l, r]\), let \( a = \tanh(\sinh(l) + 1)/2 \) and \( b = \tanh(\sinh(r) + 1)/2 \). Then, if \( a \leq 1/(2\|A - I\|) \) and \( b \geq 2\|A^{-1}\|/(2\|A^{-1}\| + 1) \), it holds that

\[
    \left\| \log(A) - (A - I) \int_l^r F_{DE}(x) \, dx \right\| \leq \frac{3}{2} a \|A - I\| + \left( -\log(b) + \frac{1 - b}{2b} \right) \|A - I\| \|A^{-1}\|. \tag{14}
\]

**Proof.** By combining (5) and (6), and as well as substituting (8) and (11) in (6), we get (14). \( \square \)

### 2.2 Setting the integral interval

To develop algorithms for computing \( \log(A) \) based on the DE formula, we need to determine the appropriate finite integration interval, \([l, r]\) in advance. The finite interval should be ideally set such that the relative error is guaranteed to be smaller than or equal to the given tolerance, \( \epsilon > 0 \), i.e.,

\[
    \frac{\left\| \log(A) - (A - I) \int_l^r F_{DE}(x) \, dx \right\|}{\| \log(A) \|} \leq \epsilon.
\]

To accomplish this, a lower bound of \( \| \log(A) \| \) must be estimated. The following lemma shows the lower bound in terms of the spectral radius of \( A \).

**Lemma 2.3.** Let \( \rho(\cdot) \) be the spectral radius, i.e., the largest absolute value of eigenvalues. Then, the following two inequations hold:

\[
    \| \log(A) \| \geq |\log(\rho(A))|, \tag{15}
\]

\[
    \| \log(A) \| \geq |\log(\rho(A^{-1}))|. \tag{16}
\]

**Proof.** First, we show (15). Using consistency of the norm, we have

\[
    \| \log(A) \| \geq \rho(\log(A)). \tag{17}
\]

Let \( \lambda_i \) \((i = 1, \ldots, n)\) be the eigenvalues of \( A \). It follows that \( \rho(\log(A)) = \max_i |\log(\lambda_i)| \). For \( z \in \mathbb{C} \setminus (-\infty, 0] \), we see that

\[
    |\log(z)| = |\log(|z|) + i \arg(z)| = \sqrt{[\log(|z|)]^2 + [\arg(z)]^2} \geq |\log(|z|)|.
\]

Let the largest absolute eigenvalue be \( \lambda_{\text{max}} \); then it follows:

\[
    \rho(\log(A)) = \max_i |\log(\lambda_i)| \geq |\log(\lambda_{\text{max}})| \geq |\log(|\lambda_{\text{max}}|)|. \tag{18}
\]

From (17) and (18), we get \( \| \log(A) \| \geq |\log(|\lambda_{\text{max}}|)| \); by substituting \( |\lambda_{\text{max}}| = \rho(A) \), we obtain (15).

Next, we show (16). For the smallest absolute eigenvalue, \( \lambda_{\text{min}} \), it is true that

\[
    \| \log(A) \| \geq |\log(|\lambda_{\text{min}}|)|.
\]

Since

\[
    |\log(|\lambda_{\text{min}}|)| = |\log(\rho(A^{-1}))| = |\log(\rho(A^{-1}))|,
\]

we have (16). \( \square \)

In the following proposition, we show how to set the finite interval such that the relative truncation error in 2-norm is smaller than or equal to the given tolerance, \( \epsilon > 0 \).
Proposition 2.2. Suppose that \( A \neq I \). Let \( \theta \) be a lower bound of \( \| \log(A) \|_2 \), the tolerance \( \epsilon > 0 \) satisfy

\[
\epsilon < \frac{3 \| A - I \|_2 \| A^{-1} \|_2}{\theta (1 + \| A^{-1} \|_2)},
\]

and \( s \) be a real positive solution of the equation

\[
\frac{1}{\theta} (-\log(s) + \frac{1 - s}{2s}) \| A - I \|_2 \| A^{-1} \|_2 = \frac{\epsilon}{2}.
\]

Define \( l \) and \( r \) as

\[
l := \text{arsinh}(\text{artanh}(2a - 1)), \quad r := \text{arsinh}(\text{artanh}(2b - 1)),
\]

where

\[
a := \min \left\{ \frac{\theta \epsilon}{3 \| A - I \|_2}, \frac{1}{2 \| A - I \|_2} \right\}, \quad b := \max \left\{ s, \frac{2 \| A^{-1} \|_2}{2 \| A^{-1} \|_2 + 1} \right\}.
\]

Then, it holds that

\[
\frac{\| \log(A) - (A - I) \int_0^r \text{DE}(x) \, dx \|_2}{\| \log(A) \|_2} \leq \epsilon.
\]

Proof. From the definition of \( a \) and \( b \), it is true that \( a \leq 1/(2 \| A - I \|_2) \) and \( b \geq 2 \| A^{-1} \|_2/(2 \| A^{-1} \|_2 + 1) \). In addition, since

\[
b - a \geq \frac{2 \| A^{-1} \|_2}{2 \| A^{-1} \|_2 + 1} - \frac{\theta \epsilon}{3 \| A - I \|_2} > \frac{2 \| A^{-1} \|_2}{2 \| A^{-1} \|_2 + 1} - \frac{\| A^{-1} \|_2}{1 + \| A^{-1} \|_2}
\]

\[
= \frac{2 \| A^{-1} \|_2 (1 + \| A^{-1} \|_2) - (2 \| A^{-1} \|_2 + 1) \| A^{-1} \|_2}{(2 \| A^{-1} \|_2 + 1)(1 + \| A^{-1} \|_2)}
\]

\[
= \frac{\| A^{-1} \|_2}{(2 \| A^{-1} \|_2 + 1)(1 + \| A^{-1} \|_2)} > 0,
\]

it follows that \( a < b \), and \( l < r \). Therefore, we can choose a finite interval \([l, r]\) that satisfies the assumptions of Proposition 2.1. By dividing the inequality (14) with \( \| \log(A) \|_2 \geq \theta \), it follows:

\[
\frac{\| \log(A) - (A - I) \int_0^r \text{DE}(x) \, dx \|_2}{\| \log(A) \|_2} \leq \frac{3a}{2\theta} \| A - I \|_2 + \frac{1}{\theta} \left( -\log(b) + \frac{1 - b}{2b} \right) \| A - I \|_2 \| A^{-1} \|_2.
\]

From the definition of \( a \) and \( b \), it holds that \( a \leq \theta \epsilon/(3 \| A - I \|_2) \) and \( b \geq s \). Therefore,

\[
\frac{3a}{2\theta} \| A - I \|_2 \leq \frac{\epsilon}{2}, \quad \frac{1}{\theta} \left( -\log(b) + \frac{1 - b}{2b} \right) \| A - I \|_2 \| A^{-1} \|_2 \leq \frac{\epsilon}{2}.
\]

We obtain (21) by substituting (23) in (22).

\[
\square
\]

2.3 Algorithms

In this subsection we establish two algorithms based on subsection 2.1 and 2.2. One of the algorithms is designed to compute \( \log(A) \) using the \( m \)-point DE formula on a finite interval with an interval truncation error smaller than or approximately equal to the given tolerance, \( \epsilon > 0 \). The other algorithm is an error control algorithm:
designed to compute \( \log(A) \) by automatically adding abscissas until the error is smaller than or approximately equal to a given tolerance \( \zeta > 0 \).

If the tolerance \( \epsilon \) given in Proposition 2.2 is sufficiently small, a linear approximation to the nonlinear equation (19) can be used to determine an appropriate interval. We describe our calculation in detail below.

Suppose that \( \epsilon \) is sufficiently small and the solution \( s \) of (19) is approximately equal to 1. Then, because \( 3(1-s)/2 \) is the first-order Taylor approximation of \(- \log(s) + (1-s)/2s \) around \( s = 1 \), the solution \( s \) can be approximated by using the solution \( \tilde{s} \) of the following equation:

\[
\frac{1}{\theta} \frac{3(1 - \tilde{s})}{2} \|A - I\|_2 \|A^{-1}\|_2 = \frac{\epsilon}{2}. \tag{24}
\]

The solution of (24) is given by

\[
\tilde{s} = 1 - \frac{\theta \epsilon}{3\|A - I\|_2 \|A^{-1}\|_2}.
\]

Under the assumptions of Proposition 2.2, by choosing \( \theta \) and setting \( \tilde{b} \) as

\[
\tilde{b} = \max \left\{ \tilde{s}, \frac{2\|A^{-1}\|_2}{2\|A^{-1}\|_2 + 1} \right\}
\]

instead of (20), and setting \( \tilde{r} = \text{arsinh}(\text{artanh}(2\tilde{b} - 1)) \), the interval truncation will be smaller than or approximately equal to \( \epsilon \):

\[
\frac{\| \log(A) - (A - I) \int_{\tilde{r}}^h F_{DE}(x) \text{d}x \|_2}{\| \log(A)\|_2} \leq \epsilon. \tag{25}
\]

The summary of the first algorithm which computes \( \log(A) \) using the \( m \)-point DE formula based on the interval truncation error (25) is as shown in Algorithm 1.

When \( \epsilon \) is sufficiently small, an accurate computation of \( \| I - A \|_2 \), \( \| A^{-1} \|_2 \) and \( \rho(A) \) at Step 2 of Algorithm 1 may not be required because the errors that stem from these values have little effect on the accuracy of \( \log(A) \); see Appendix A for more detail. At Step 3, a lower bound of \( \| \log(A)\|_2 \) is computed based on (15). By setting \( \theta = \max \{ \| \log(\rho(A)) \|, \| \log(\rho(A^{-1})) \| \} \), a tighter lower bound can be obtained. In particular, when \( A \) is positive definite, \( \| \log(\rho(A^{-1})) \| \) can be obtained without additional computation because \( \rho(A^{-1}) = \| A^{-1} \|_2 \) is already computed in Step 2.

Once an approximate finite interval is obtained, the accuracy of the DE formula can be improved with the following procedure. Let \( m \) be the number of abscissas, \( h = (r - l)/(m + 1) \) be the mesh size, and \( T(h) \) be the trapezoidal rule for the mesh size \( h \):

\[
T(h) := \frac{h}{2} (F_{DE}(l) + F_{DE}(r)) + h \sum_{i=1}^{m-2} F_{DE}(l + ih).
\]

Then, \( T(h/2) \) can be computed using \( T(h) \):

\[
T \left( \frac{h}{2} \right) = \frac{1}{2} T(h) + \frac{h}{2} \sum_{i=1}^{m-1} F_{DE} \left( l + (2i - 1)\frac{h}{2} \right).
\]

In addition, we can apply the following estimation of the trapezoidal error for a sufficiently small value of \( h \) using [2] Eq. (4.3):

\[
\left\| \int_{l}^{r} F_{DE}(x) - T \left( \frac{h}{2} \right) \right\| \approx \frac{1}{3} \left\| T(h) - T \left( \frac{h}{2} \right) \right\|. \tag{26}
\]

Our error control algorithm which is based on (26) is presented as Algorithm 2.
Algorithm 1 Computation of $\log(A)$ based on the DE formula.

**Input:** $A \in \mathbb{R}^{n \times n}, m \in \mathbb{N}, \epsilon > 0$ for (25).

**Output:** $X \approx \log(A)$

1. Set $F_{DE}(x) = \cosh(x)\text{sech}^2((\sinh(x))(1 + \tanh(\sinh(x)))(A - I) + 2I)^{-1}$
2. Compute $\|A - I\|_2, \|A^{-1}\|_2, \rho(A)$
3. $\theta = |\log(\rho(A))|$
4. $\epsilon_{\max} = \frac{3\|A - I\|_2\|A^{-1}\|_2}{\theta}$
5. if $\epsilon \geq \epsilon_{\max}$ then
   6. $\epsilon \leftarrow \epsilon_{\max}/2$
7. end if
8. $a = \min \left\{ \frac{\theta\epsilon}{3\|A - I\|_2}, \frac{1}{2\|A - I\|_2} \right\}$
9. $b = \max \left\{ 1 - \frac{3\|A - I\|_2\|A^{-1}\|_2}{\theta}, \frac{2\|A^{-1}\|_2 + 1}{2\|A^{-1}\|_2} \right\}$
10. $l = \text{arsinh}(\text{artanh}(2a - 1))$
11. $r = \text{arsinh}(\text{artanh}(2b - 1))$
12. $h = (l - r)/(m - 1)$
13. $T = \frac{h}{2}(F_{DE}(l) + F_{DE}(r)) + h\sum_{i=1}^{m-2}F_{DE}(l + ih)$
14. $X = (A - I)T$

Algorithm 2 Computation of $\log(A)$ based on the DE formula with error control.

**Input:** $A \in \mathbb{R}^{n \times n}, m_0 \in \mathbb{N}, \epsilon > 0$ for (25), $\zeta > 0$ for trapezoidal truncation error.

**Output:** $X \approx \log(A)$

1. Set $F_{DE}(x) = \cosh(x)\text{sech}^2((\sinh(x))(1 + \tanh(\sinh(x)))(A - I) + 2I)^{-1}$
2. Computing $l, r, \theta$ using Algorithm 1 from step 2 to step 11
3. $h_0 = (r - l)/(m_0 - 1)$
4. $T_0 = \frac{h_0}{2}F_{DE}(l) + \frac{h_0}{2}F_{DE}(r) + h_0\sum_{i=1}^{m-2}F_{DE}(l + ih)$
5. for $k = 0, 1, 2, \ldots$ until convergence do
   6. $h_{k+1} = h_k/2$
   7. $T_{k+1} = \frac{h_{k+1}}{2}T_k + h_{k+1}\sum_{i=1}^{m-k-1}F_{DE}(l + (2i - 1)h_{k+1})$
   8. $m_{k+1} = 2m_k - 1$
   9. if $\frac{1}{\theta}\|T_{k+1} - T_k\|/\theta \leq \zeta$ then
      10. $T = T_{k+1}$
      11. break
   end if
12. end for
13. $X = (A - I)T$
3 Numerical experiments

The numerical experiments were carried out using Julia 1.0 on a Core-i7 (3.4GHz) CPU with 16GB RAM. We used the IEEE double precision arithmetic. We computed abscissas and weights in the GL quadrature with QuadGK [10].

3.1 Experiment 1: Checking the convergence

In this experiment, we checked the convergence of \( \log(A) \) computed by the GL quadrature and the DE formula. Our test matrices are presented in Table 1. We generated the first three matrices in Table 1 by using the following procedure:

1. We generated an orthogonal matrix \( Q \) by QR decomposition of a random 50 \times 50 matrix.

2. We generated a diagonal matrix whose diagonal elements were geometric sequence: \( \{d_i\}_{i=1,...,50} \) where 
   \[
   d_1 = \kappa^{1/2} \quad \text{and} \quad d_{50} = \kappa^{1/2} \quad \text{for} \quad \kappa = 10^1.
   \]

3. \( A = QDQ^T \).

4. We repeated Step 2 and Step 3 by setting \( \kappa \) as \( 10^4 \) and \( 10^7 \).

The experimental procedure is as follows:

1. We scaled the test matrices as \( \tilde{A} = (10/\rho(A))A \) because some matrices had values that were too large to use in computation.

2. We prepared the “exact” \( \log(\tilde{A}) \) with the arbitrary precision arithmetic and rounded the result to double precision. We implemented the ISS algorithm [7, Alg. 11.10] with the BigFloat type of Julia.

3. We computed \( \log(\tilde{A}) \) using Algorithm 1 where \( \epsilon = 2^{-53} \approx 1.1 \times 10^{-16} \). If the test matrix was symmetric positive definite, we set \( \theta = \max\{|\log(\rho(\tilde{A}))|, |\log(\rho(\tilde{A}^{-1}))|\} \) as stated in subsection 2.3. We computed \( \rho(\tilde{A}) \) using the eigvals function of Julia, which computes all eigenvalues of \( \tilde{A} \). Similarly, we computed \( \|I - \tilde{A}\|_2 \) and \( \|\tilde{A}^{-1}\|_2 \) using the svdvals function, which computes all singular values of \( \tilde{A} \).

4. We computed \( \log(\tilde{A}) \) by applying the GL quadrature to (2).

Figure 1 shows the convergence histories of the DE formula and the GL quadrature for each matrix. Several observations can be made:

\[ \text{Table 1: Test matrices. The condition number of } A \text{ is presented as } \kappa_2(A) = \|A\|_2\|A^{-1}\|_2. \]

| Matrix | Size  | \( \kappa_2(A) \) | Structure |
|--------|-------|-------------------|-----------|
| SPD 1  | 50    | \( 1.0 \times 10^1 \) | SPD       |
| SPD 2  | 50    | \( 1.0 \times 10^4 \) | SPD       |
| SPD 3  | 50    | \( 1.0 \times 10^7 \) | SPD       |
| parter [14] | 10   | \( 2.4 \times 10^9 \) | Unsymmetric |
| frank [18] | 10   | \( 2.9 \times 10^3 \) | Unsymmetric |
| vand [14] | 10   | \( 3.1 \times 10^{12} \) | Unsymmetric |
| bcsstk02 [4] | 66 | \( 4.3 \times 10^3 \) | SPD       |
| bcsstk03 [4] | 112 | \( 6.8 \times 10^6 \) | SPD       |
| ck104 [4] | 104  | \( 5.5 \times 10^3 \) | Unsymmetric |

\[ \text{4 We present some numerical results, for which } \rho(\tilde{A}), \|\tilde{A} - I\|_2, \text{ and } \|\tilde{A}^{-1}\|_2 \text{ are computed with row accuracy, as shown in Appendix A.} \]
Figure 1: Convergence histories of the DE formula (Algorithm [1]) and the GL quadrature. The vertical axes show the relative error, $\| \log(\tilde{A}) - X \|_F / \| \log(\tilde{A}) \|_F$, and the horizontal axes show the number of abscissas, $m$. 
• The accuracy of the DE formula is almost the same as that of the GL quadrature, and the accuracies of the DE formula and the GL quadrature depend on the condition number of test matrices.

• For the well-conditioned matrices, such as SPD 1 and parter matrix, the GL quadrature converged faster than the DE formula. Conversely, for the ill-conditioned matrices, such as SPD 3 and vand matrix, the DE formula converged faster than the GL quadrature.

The above observations suggest that algorithm 1 selects an appropriate interval and the DE formula is suitable for ill-conditioned matrices.

3.2 Experiment 2: Checking Algorithm 2

In this experiment, we check the performance of algorithm 2 by using the same matrices that were used in experiment 1 (see subsection 3.1). We compared algorithm 2 with algorithm 3, which is based on the GL quadrature in Appendix B.

We conducted the experiment using the following procedure:

1. We computed \( \log(\tilde{A}) \) by using algorithm 2. We set \( m_0 = 16 \) and \( \zeta = \epsilon \in \{10^{-8}, 10^{-11}\} \). In Step 2 of algorithm 2 in which algorithm 1 was implemented, we computed the spectral radius and the 2-norm of matrices using `eigvals` and `svdvals` functions, as is done in experiment 1.

2. We computed \( \log(\tilde{A}) \) by using algorithm 3. We set \( m_0 = 16 \) and \( \zeta \in \{10^{-8}, 10^{-11}\} \). In Step 2 of algorithm 3, the lower bound \( \theta \) was computed using (15) and (16) in the same way as was done for the DE formula. If the number of integrand evaluations was more than 2000, we stopped the computation.

| \( \zeta \)   | Algorithm 2 (DE) | Algorithm 3 (GL) |
|-------------|-----------------|-----------------|
|             | 10\(^{-8}\)  | 10\(^{-11}\)  | 10\(^{-8}\)  | 10\(^{-11}\)  |
| SPD 1       | 61 (2.2 \(\times\) 10\(^{-9}\)) | 61 (2.7 \(\times\) 10\(^{-12}\)) | 48 (4.6 \(\times\) 10\(^{-16}\)) | 112 (5.7 \(\times\) 10\(^{-16}\)) |
| SPD 2       | 121 (6.7 \(\times\) 10\(^{-10}\)) | 241 (6.4 \(\times\) 10\(^{-13}\)) | 1008 (1.8 \(\times\) 10\(^{-15}\)) | 1008 (1.8 \(\times\) 10\(^{-15}\)) |
| SPD 3       | 241 (3.0 \(\times\) 10\(^{-10}\)) | 481 (4.9 \(\times\) 10\(^{-13}\)) | (–)           | (–)           |
| parter      | 61 (2.6 \(\times\) 10\(^{-9}\)) | 121 (2.3 \(\times\) 10\(^{-12}\)) | 112 (3.3 \(\times\) 10\(^{-16}\)) | 112 (3.3 \(\times\) 10\(^{-16}\)) |
| frank       | 481 (1.0 \(\times\) 10\(^{-12}\)) | 1921 (2.1 \(\times\) 10\(^{-13}\)) | 496 (1.5 \(\times\) 10\(^{-11}\)) | (–)           |
| vand        | 1921 (1.3 \(\times\) 10\(^{-7}\)) | 1921 (1.3 \(\times\) 10\(^{-7}\)) | (–)           | (–)           |
| bcsstk02    | 121 (2.8 \(\times\) 10\(^{-9}\)) | 121 (3.1 \(\times\) 10\(^{-12}\)) | 496 (1.7 \(\times\) 10\(^{-15}\)) | 1008 (1.0 \(\times\) 10\(^{-15}\)) |
| bcsstk03    | 241 (1.4 \(\times\) 10\(^{-9}\)) | 241 (1.5 \(\times\) 10\(^{-12}\)) | (–)           | (–)           |
| ck104       | 121 (6.7 \(\times\) 10\(^{-10}\)) | 121 (7.4 \(\times\) 10\(^{-13}\)) | 496 (2.2 \(\times\) 10\(^{-15}\)) | 496 (2.2 \(\times\) 10\(^{-15}\)) |

The number of integrand evaluations and the corresponding relative error when the two algorithms stopped are shown in Table 2. Several observations can be made:

• Algorithm 2 successfully computed the logarithm with the desired accuracy within 2000 integrand evaluations for all test matrices, except vand matrix, whereas algorithm 3 could not succeed for three matrices. For vand matrix, the stopping criterion \( \zeta \) may be too strict because the convergence history of vand matrix (as shown in Figure 1) hardly reach the value of 10\(^{-8}\). Our future studies will focus on the method of selecting suitable stopping criteria.
• Even if the condition number of $A$ is small as is the case for SPD 1 and parter matrix, the number of integrand evaluations of algorithm 2 could be smaller than that of algorithm 3 because algorithm 2 can reuse all previous results when improving accuracy.

These observations show that algorithm 2 can be a practical choice for the computation of the matrix logarithm by numerical quadrature.

4 Conclusion

In this paper, we focused on the DE formula as a new choice for the numerical quadrature formula of $\log(A)$. When using the DE formula, an appropriate finite interval needs to be selected. We first presented an upper bound of the interval truncation error for the given finite interval, after which we demonstrated a procedure to select a finite interval with an interval truncation error smaller than or approximately equal to the given tolerance. We also proposed two algorithms. The first algorithm is designed to compute $\log(A)$ by using the $m$-point DE formula and the finite interval is selected based on the above error estimate. The second algorithm is designed to compute $\log(A)$ by automatically adding abscissas until the trapezoidal error is smaller than or approximately equal to the given tolerance.

We carried out two numerical experiments. The first experiment suggested that the finite interval selected by our algorithm was appropriate, and showed that the DE formula converged faster than the GL quadrature for ill-conditioned matrices. The second experiment demonstrated that the proposed error control algorithm worked well when provided with appropriate stopping criteria.

Our future work will focus on three problems. The first one is analyses of the convergence rate for the DE formula and the GL formula, the second one is a method of selecting appropriate stopping criteria, and the third one is the verification of the practical performance of the presented algorithms, when applied to large sparse matrices from currently researched problems.

References

[1] A. H. Al-Mohy and N. J. Higham. Improved inverse scaling and squaring algorithms for the matrix logarithm. *SIAM J. Sci. Comput.*, 34(4):C153–C169, 2012.

[2] J. R. Cardoso. Computation of the matrix $p$th root and its Fréchet derivative by integrals. *Electron. Trans. Numer. Anal.*, 39:414–436, 2012.

[3] J. R. Cardoso and R. Ralha. Matrix arithmetic-geometric mean and the computation of the logarithm. *SIAM J. Matrix Anal. Appl.*, 37(2):719–743, 2016.

[4] T. A. Davis and Y. Hu. The University of Florida sparse matrix collection. *ACM Trans. Math. Software*, 38(1):Art. 1, 25, 2011.

[5] L. Dieci, B. Morini, and A. Papini. Computational techniques for real logarithms of matrices. *SIAM J. Matrix Anal. Appl.*, 17(3):570–593, 1996.

[6] A. Heßelmann and A. Görling. Random phase approximation correlation energies with exact Kohn–Sham exchange. *Mol. Phys.*, 108(3-4):359–372, 2010.

[7] N. J. Higham. *Functions of Matrices: Theory and Computation*. SIAM, Philadelphia, 2008.

[8] I. Horenko and Ch. Schütte. Likelihood-based estimation of multidimensional Langevin models and its application to biomolecular dynamics. *Multiscale Model. Sim.*, 7(2):731–773, 2008.
In this section we consider the effect of the errors in $\rho(A)$, $\|A - I\|_2$, and $\|A^{-1}\|_2$ at Step 2 of Algorithm 1. Assume that $\epsilon$ in Algorithm 1 is sufficiently small, and that $a$ and $b$ in Step 9 are chosen as

$$a = \frac{\theta \epsilon}{3\|A - I\|_2}, \quad b = 1 - \frac{\theta \epsilon}{3\|A - I\|_2 \|A^{-1}\|_2},$$

where $\theta$ is a lower bound of $\|\log(A)\|_2$.

Let $\Delta_1$ and $\Delta_2$ be the relative errors of $\theta/\|A - I\|_2$ and $1/\|A^{-1}\|_2$ respectively. Then, the computational result of $a$ is equal to

$$\epsilon = \frac{\theta}{3\|A - I\|_2} (1 + \Delta_1) = \frac{\theta}{3\|A - I\|_2} \epsilon (1 + \Delta_1),$$

and that of $b$ is equal to

$$1 - \frac{\epsilon}{3\|A - I\|_2} (1 + \Delta_1) \frac{1}{\|A^{-1}\|_2} (1 + \Delta_2) = 1 - \frac{\theta}{3\|A - I\|_2 \|A^{-1}\|_2} \epsilon (1 + \Delta_1 + \Delta_2 + \Delta_1 \Delta_2).$$

Therefore, the upper bound of the truncation error $\epsilon$ in (25) computed by considering the relative errors $\Delta_1$, $\Delta_2$ is almost equal to the upper bound of the truncation error when the tolerance is set as $\epsilon (1 + \Delta_1 + \Delta_2 + \Delta_1 \Delta_2)$. For example, when $\Delta_1, \Delta_2 = 10^{-2}$, $\epsilon (1 + \Delta_1 + \Delta_2 + \Delta_1 \Delta_2) \approx 1.02\epsilon$, which means that the upper bound of the truncation error changes by approximately 2%. If $\epsilon$ is sufficiently small, the effect of these errors will be negligible.

Here we present some numerical examples. The convergence histories of the DE formula are shown in Figure 2. Each graph shows two histories: one is obtained by using the eigvals and svdvals functions that are used to compute $\rho(A)$, $\|A - I\|_2$, and $\|A^{-1}\|_2$; the other obtained using the 1% perturbed values. The figure shows that the behaviors of the histories are almost equal, and the effects of the errors did not appear.

In conclusion, the parameters used at Step 2 of Algorithm 1 can be computed roughly.

## B Error control algorithm based on the GL quadrature

In this section we show an error control algorithm based on the GL quadrature for (2), using a technique from [2].

[9] Z. Huang and L. Van Gool. A Riemannian network for SPD matrix learning. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[10] M. Piibeleht and A. Arslan. QuadGK.jl. https://github.com/JuliaMath/QuadGK.jl.

[11] T. Schenk, I. M. Richardson, M. Kraska, and S. Ohnimus. Modeling buckling distortion of DP600 overlap joints due to gas metal arc welding and the influence of the mesh density. Comp. Mater. Sci., 46(4):977–986, 2009.

[12] H. Takahasi and M. Mori. Double exponential formulas for numerical integration. Publ. Res. Inst. Math. Sci., 9:721–741, 1974.

[13] C. K. Zachos. A classical bound on quantum entropy. J. Phys. A.: Math. Theor., 40(21):F407, 2007.

[14] W. Zhang and N. J. Higham. Matrix Depot: an extensible test matrix collection for Julia. PeerJ Comput. Sci., 2:e58, 2016.
Figure 2: Convergence histories of the DE formula (Algorithm 1) obtained using the exact estimations of $\|A - I\|_2$, $\|A^{-1}\|_2$, $\rho(A)$ (using eigvals and svdvals) and the perturbed one. The vertical axes show the relative error, $\|\log(\hat{A}) - X\|_F / \|\log(\hat{A})\|_F$, and the horizontal axes show the number of nodes, $m$.

Let $G(m) := \sum_{i=1}^{m} w_i F_{\text{GL}}(u_i)$ be the results of the $m$-point GL quadrature, where $F_{\text{GL}}(u_i) = [(1 + u)(A - I) + 2I]^{-1}$. Using [2, Eq. (4.6)], the following error estimate can be applied:

$$\left\| G(2m) - \int_{-1}^{1} F_{\text{GL}}(t) dt \right\| \leq \|G(m) - G(2m)\|.$$ (27)

Based on (27), we present the algorithm designed to compute $\log(A)$ by automatically adding abscissas until the error is smaller than the given tolerance as Algorithm 3.

**Algorithm 3** Computation $\log(A)$ based on Gauss–Legendre quadrature

**Input:** $A \in \mathbb{R}^{n \times n}$, The number of initial nodes $m_0 \geq 2$, $\zeta > 0$ for a tolerance of the truncation error.

**Output:** $X \approx \log(A)$

1. Set $F_{\text{GL}}(u) := [(1 + u)(A - I) + 2I]^{-1}$
2. Compute $\theta$, a lower bound of $\|\log(A)\|_2$.
3. Compute nodes $u_i$ and weights $w_i$ of $m_0$-point Gauss–Legendre quadrature ($i = 1, \ldots, m_0$)
4. $G_0 = \sum_{i=1}^{m_0} w_i F_{\text{GL}}(u_i)$
5. for $k = 0, 1, 2, \ldots$ until convergence do
6. $m_{k+1} = 2m_k$
7. Compute nodes $u_i$ and weights $w_i$ of $m_{k+1}$-point Gauss–Legendre quadrature ($i = 1, \ldots, m_{k+1}$)
8. $G_{k+1} = \sum_{i=1}^{m_{k+1}} w_i F_{\text{GL}}(u_i)$
9. if $\|G_{k+1} - G_k\| / \theta \leq \zeta$ then
10. $G = G_{k+1}$
11. break
12. end if
13. end for
14. $X = (A - I)G$