An Efficient Bound for the Condition Number of the Matrix Exponential

Al-Mohy, Awad H.

2015

MIMS EPrint: **2015.5**

Manchester Institute for Mathematical Sciences
School of Mathematics

The University of Manchester

Reports available from: http://eprints.maths.manchester.ac.uk/
And by contacting: The MIMS Secretary
School of Mathematics
The University of Manchester
Manchester, M13 9PL, UK

ISSN 1749-9097

AN EFFICIENT BOUND FOR THE CONDITION NUMBER OF THE MATRIX EXPONENTIAL *

AWAD H. AL-MOHY†

Abstract.
A new bound for the condition number of the matrix exponential is presented. Using the bound, we propose an efficient approximation to the condition number, denoted by $\kappa_g(s, X)$, that *avoids* the computation of the Fréchet derivative of the matrix exponential that underlies condition number estimation in the existing algorithms. We exploit the identity $e^X = (e^{X/2^s})^{2^s}$ for a nonnegative integer s with the properties of the Fréchet derivative operator to obtain the bound. Our cost analysis reveals that considerable computational savings are possible since estimating the condition number by the existing algorithms requires several invocation of the Fréchet derivative of the matrix exponential whose single invocation costs as twice as the cost of the matrix exponential itself. The bound and hence $\kappa_g(s, X)$ only involve Fréchet derivative of a monomial of degree 2^s, which can be computed exactly in $2s$ matrix multiplications. We propose two versions of the scaling and squaring algorithm that implement $\kappa_g(s, X)$. Our numerical experiments show that $\kappa_g(s, X)$ captures the behavior of the condition number and moreover outperforms the condition number in the estimation of relative forward errors for a wide range of problems.

Key words. condition number, matrix exponential, scaling and squaring method, Fréchet derivative, squaring phase, Padé approximation, backward error analysis, MATLAB, error estimate, `expm`

AMS subject classifications. 65F30, 65F60

1. Introduction.

The matrix exponential plays a seminal role in the solution of linear systems of ordinary differential equations, making it the most studied matrix function after the matrix inversion. Thus solution quality and the sensitivity of the problem are important to understand. The source of error can be from input data (e.g. inaccurate measurement) or can result from accumulation of rounding errors, so having a numerical algorithm for a problem, it is important to know how accurate the computed solution is and how small perturbation in input data can effect outputs. *Condition numbers* of matrix functions measure the first order sensitivity. Several authors have worked toward understanding the sensitivity of the matrix exponential. Van loan [15] and Kågström [11] derive bounds and perturbation bounds for the problem. Though some of these bounds are cheap to compute, some of them are rather pessimistic. So the condition number is used as a reliable measure to the sensitivity.

Given a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ [7, sect. 1.2], its condition number at a matrix X is defined as [7, sect. 3.1]

$$\operatorname{cond}(f, X) := \lim_{\epsilon\to 0} \sup_{\|E\|\le\epsilon\|X\|} \frac{\|f(X+E) - f(X)\|}{\epsilon\|f(X)\|}, \tag{1.1}$$

where the norm is any matrix norm.

The Fréchet derivative of a matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ at a point $X \in \mathbb{C}^{n\times n}$ is a linear operator

$$\begin{array}{ccc} \mathbb{C}^{n\times n} & \xrightarrow{L_f(X)} & \mathbb{C}^{n\times n} \\ E & \longmapsto & L_f(X, E) \end{array}$$

* Version of January 13, 2015.

†Department of Mathematics, King Khalid University, PO Box 9004 Abha, Saudi Arabia (ahalmohy@kku.edu.sa, http://www.ma.man.ac.uk/~almohy).

such that

$$f(X+E) - f(X) - L_f(X,E) = o(\|E\|) \tag{1.2}$$

for all $E \in \mathbb{C}^{n\times n}$. $L_f(X,E)$ is the value of the Fréchet derivative of f at X in the direction E. If such an operator exists, f is said to be Fréchet differentiable. The norm of $L_f(X)$ is defined as

$$\|L_f(X)\| := \max_{Z\neq 0} \frac{\|L_f(X,Z)\|}{\|Z\|}. \tag{1.3}$$

Since $L_f(X,E)$ is linear in E, we have the important representation

$$\operatorname{vec}(L_f(X,E)) = K_f(X)\operatorname{vec}(E), \tag{1.4}$$

where $K_f(X) \in \mathbb{C}^{n^2\times n^2}$ is called the *Kronecker form* of the Fréchet derivative and vec is the operator that stacks the columns of a matrix on top of each other [7, sect. 3.2].

The condition number of f at X can be expressed in terms of the Fréchet derivative.

THEOREM 1.1 ([7, Thm. 3.1]). *Suppose that the matrix function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$ is Fréchet differentiable at $X \in \mathbb{C}^{n\times n}$. Then*

$$\operatorname{cond}(f,X) = \frac{\|L_f(X)\|\,\|X\|}{\|f(X)\|}. \qquad \square$$

In view of the definition of the condition number (1.1), if we have an algorithm that produces an approximation $\widehat{Y}$ to $f(X)$ and if $\Delta X \in \mathbb{C}^{n\times n}$ satisfies $\widehat{Y} = f(X+\Delta X)$ as a backward error then the forward error $\Delta f := \widehat{Y} - f(X)$ satisfies the rule of thumb

$$\frac{\|\Delta f\|}{\|f(X)\|} \lesssim \operatorname{cond}(f,X)\frac{\|\Delta X\|}{\|X\|}. \tag{1.5}$$

The scaling and squaring method is the most popular method for computing the matrix exponential and has drawn the attention of researchers for decades. A successful scaling and squaring algorithm is that of Higham [6], which forms the base of the MATLAB function `expm`. Recently Al-Mohy and Higham [2] add several improvements to the algorithm such as the special treatment for triangular matrices and the way the scaling parameter s is liberally chosen so that overscaling problem is avoided and less computational cost is attained. Recent catalogue by Higham and Deadman surveys the software implementing these algorithms [8].

Al-Mohy and Higham [1] propose an algorithm for simultaneously computing the matrix exponential and its Fréchet derivative and then extend that algorithm to an algorithm that computes the matrix exponential and estimates its condition number. The later computes the Fréchet derivative of the matrix exponential severals time in various direction to maximize the norm in (1.3).

In this paper we derive a new bound for the condition number of e^X free from the norm of the Fréchet derivative of e^X and show that significant computational cost can be saved. The paper is organized as follows. In section 2 we derive the new bound for the condition number of e^X, which involves a condition number of a monomial and, in particular, we show that the condition number of a monomial is equal to its degree for normal matrices with respect to the 2-norm. In section 3 we review a standard algorithm used to exactly compute condition numbers of

matrix function if their Fréchet derivatives are available. We then give a numerical experiment to demonstrate the sharpness of the bound. Section 4 presents heuristic estimator to the condition number of e^X. We show by intensive numerical experiments that this estimator is a reliable approximation to the condition number. In section 5 we review a practical way to estimate the condition number implementing the block 1-norm estimator that the existing algorithms use. In section 6 we give two versions of the scaling and squaring algorithm in which the new bound is used to estimate the condition number and support the algorithms bu numerical experiments. In section 7 we present cost analysis and give percentages of potential computational savings. Finally we draw some concluding remarks in section 8.

2. New bound for the condition number of the matrix exponential. In this section we derive a new bound for the condition number of the matrix exponential based on the identity $e^X = \left(e^{2^{-s}X}\right)^{2^s}$, where s is a nonnegative integer. The scaling and squaring method exploits this identity where the exponential of the scaled matrix $e^{2^{-s}X}$ is accurately approximated by Padé approximants, $r_m(2^{-s}X)$, for suitably chosen s and m and then recovers the original exponential by repeatedly squaring $r_m(2^{-s}X)$ for s times.

The weakness of the scaling and squaring method lies in its squaring phase where error begins to propagate. Thus, it is reasonable to link the condition number of the matrix exponential to the condition number of the squaring phase, which can be viewed as a monomial of degree 2^s.

The following lemma gives a bound for the norm of the Fréchet derivative of the matrix exponential at the point X by a product of the norm of the Fréchet derivative of the monomial x^{2^s} at the point $e^{2^{-s}X}$ and the norm of the Fréchet derivative of the matrix exponential at the point $2^{-s}X$.

LEMMA 2.1. *For any nonnegative integer s and any matrix norm we have*

$$\|L_{\exp}(X)\| \le 2^{-s}\|L_{x^{2^s}}(e^{2^{-s}X})\|\|L_{\exp}(2^{-s}X)\|. \tag{2.1}$$

Proof. Let $g(X) = X^{2^s}$, $h(X) = e^X$, and $z(X) = 2^{-s}X$. Then we have $e^X = (g \circ h \circ z)(X)$. Fréchet differentiating the side of this equation at X in an arbitrary direction E using the chain rule [7, Thm. 3.4] we obtain

$$\begin{aligned} L_{\exp}(X,E) &= L_g(h\circ z(X), L_h(z(X), L_z(X,E))) \\ &= L_g(e^{2^{-s}X}, L_{\exp}(2^{-s}X, 2^{-s}E)) \\ &= 2^{-s}L_g(e^{2^{-s}X}, L_{\exp}(2^{-s}X, E)) \end{aligned}$$

By the definition of the Fréchet derivative norm in (1.3), $\|L_g(e^{2^{-s}X})\|$ satisfies the inequality

$$\|L_g(e^{2^{-s}X}, Z)\| \le \|L_g(e^{2^{-s}X})\|\|Z\|$$

for any matrix $Z \in \mathbb{C}^{n\times n}$. In particular, take $Z = L_{\exp}(2^{-s}X, E)$. Thus

$$\|L_{\exp}(X,E)\| \le 2^{-s}\|L_g(e^{2^{-s}X})\|\|L_{\exp}(2^{-s}X, E)\|.$$

Dividing through by $\|E\|$ and maximizing over all nonzero E, the result follows immediately again by (1.3). ∎

From (1.4) we have

$$\|L_f(X,E)\|_F = \|K_f(X)\,\mathrm{vec}(E)\|_2, \tag{2.2}$$

and on dividing by $\|E\|_F = \|\,\mathrm{vec}\,(E)\|_2$ and maximizing over all nonzero E, it follows that

$$\|L_f(X)\|_F = \|K_f(X)\|_2. \tag{2.3}$$

This leads to the following important theorem that relates the condition number of the matrix exponential to the condition number of the monomial $g(X) = X^{2^s}$.

THEOREM 2.2. *For any nonnegative integer s let $g(X) = X^{2^s}$ then we have*

$$\mathrm{cond}(\exp, X) \le \frac{\|2^{-s}X\|\, e^{\|2^{-s}X\|}}{\|e^{2^{-s}X}\|}\,\mathrm{cond}(g, e^{2^{-s}X}), \tag{2.4}$$

where the norm is any matrix norm.

Proof. By using (2.1) and the relation $\|L_{\exp}(2^{-s}X)\| \le e^{\|2^{-s}X\|}$ [7, Lem. 10.15] we have

$$\begin{aligned}
\mathrm{cond}(\exp, X) &= \frac{\|X\|\|L_{\exp}(X)\|}{\|e^X\|} \\
&\le \frac{\|2^{-s}X\|}{\|e^X\|}\|L_g(e^{2^{-s}X})\|\|L_{\exp}(2^{-s}X)\| \\
&\le \|2^{-s}X\|e^{\|2^{-s}X\|}\,\frac{\|L_g(e^{2^{-s}X})\|}{\|\left(e^{2^{-s}X}\right)^{2^s}\|} \\
&= \frac{\|2^{-s}X\|e^{\|2^{-s}X\|}}{\|e^{2^{-s}X}\|}\left(\frac{\|e^{2^{-s}X}\|\|L_g(e^{2^{-s}X})\|}{\|\left(e^{2^{-s}X}\right)^{2^s}\|}\right) \\
&= \frac{\|2^{-s}X\|e^{\|2^{-s}X\|}}{\|e^{2^{-s}X}\|}\left(\frac{\|e^{2^{-s}X}\|\|L_g(e^{2^{-s}X})\|}{\|g(e^{2^{-s}X})\|})\right) \\
&= \frac{\|2^{-s}X\|\, e^{\|2^{-s}X\|}}{\|e^{2^{-s}X}\|}\,\mathrm{cond}(g, e^{2^{-s}X}). \qquad \square
\end{aligned}$$

If the matrix X is normal and the norm is specified to the 2-norm, Van Loan [15, Corollary 2] shows that $\mathrm{cond}(\exp, X) = \|X\|_2$. We have analogous result for the condition number of a monomial.

LEMMA 2.3. *Let $f(X) = X^\nu$, where ν is a positive integer. Then in the 2-norm we have*

$$\mathrm{cond}(f, X) = \nu$$

whenever X is normal.

Proof. By [1, Thm. 3.2(3.5)] we have $\|L_f(X)\|_2 \le \nu\|X\|_2^{\nu-1}$. Since X is normal we have $\|X\|_2^\nu = \|X^\nu\|_2$ for any positive integer ν. Thus $\|L_f(X)\|_2 \le \nu\|X^{\nu-1}\|_2$. On the other hand we have $\|L_f(X)\|_2 \ge \|L_f(X, I)\|_2 = \nu\|X^{\nu-1}\|_2$ by the definition (1.3). Hence $\|L_f(X)\|_2 = \nu\|X^{\nu-1}\|_2$ and

$$\mathrm{cond}(f, X) = \frac{\|X\|_2(\nu\|X\|_2^{\nu-1})}{\|X^\nu\|_2} = \frac{\nu\|X\|_2^\nu}{\|X^\nu\|_2} = \nu. \qquad \square$$

As a result from this lemma we conclude in the 2-norm that $\mathrm{cond}(g, e^{2^{-s}X}) = 2^s$ if X is normal. Note that the matrix $e^{2^{-s}X}$ is normal if and only if X is normal, following immediately from the fact that X is normal if and only if it is unitarily diagonalizable.

The sharpness of the bound (2.4) depends upon the choice of s. If $s = 0$ and the $\|X\|$ is large, the bound is very pessimistic. Thus to obtain a sharp bound s needs to be chosen so that $\|2^{-s}X\|$ is reasonable small. We will see below how to choose the parameter s and how to implement the bound to estimate the condition number of e^X problem.

We recall the following backward error result from [2].

THEOREM 2.4. *Let $X \in \mathbb{C}^{n\times n}$ and $r_m(x) =: p_m(x)/q_m(x)$ denote the $[m/m]$ Padé approximant to e^x. If s is a nonnegative integer such that the spectral radii $\rho(e^{2^{-s}X} r_m(2^{-s}X) - I) < 1$ and $\rho(2^{-s}X) \le \min\{\, |t| : q_m(t) = 0 \,\}$ then there exists a matrix $\Delta X \in \mathbb{C}^{n\times n}$ such that*

$$r_m(2^{-s}X)^{2^s} = e^{X+\Delta X}$$

and

$$\frac{\|\Delta X\|}{\|X\|} \le \sum_{k=m}^{\infty} |c_{m,k}|\, \alpha_p(2^{-s}X)^{2k}, \tag{2.5}$$

where $c_{m,k}$'s are the coefficients of the power series expansion of $\log(e^{-x} r_m(x))$, the principle logarithm, and

$$\alpha_p(2^{-s}X) = 2^{-s} \max\bigl(\|X^{2p}\|^{1/(2p)}, \|X^{2p+2}\|^{1/(2p+2)}\bigr) \tag{2.6}$$

and the integer $p \ge 1$ is such that $m \ge p(p-1)$. □

Let $\theta_m = \{\, \theta : \sum_{k=m}^{\infty} |c_{m,k}|\, \theta^{2k} \le u \,\}$, where $u = 2^{-53} \approx 1.11 \times 10^{-16}$,the unit roundoff in the IEEE double precision arithmetic. Thus if s is chosen so that $\alpha_p(2^{-s}X) \le \theta_m$, then $r_m(2^{-s}X)^{2^s}$ approximates e^X with backward error $\|\Delta X\|/\|X\| \le u$ in exact arithmetic. Higham [6] uses a variant of the bound (2.5) with $\alpha_p(2^{-s}X)$ being replaced by $\|2^{-s}X\|$ and evaluates θ_m symbolically in high precision. By careful cost analysis he chooses $m \in \{\, 3, 5, 7, 9, 13 \,\}$. The values of θ_m are tabulated in [6, Table 2.3] and [2, Table 3.1], where θ_{13} is reduced to 4.25 for stability reasoning. Al-Mohy and Higham [2] use the bound (2.5) and base the selection of the scaling parameter s upon a minimal value of $\alpha_p(2^{-s}X)$ over all $p \ge 1$ satisfying the constraint $m \ge p(p-1)$. Since $\rho(X) \le \alpha_p(X) \le \|X\|$ and $\alpha_p(X) \ll \|X\|$ is possible for nonnormal matrices, the algorithm of Al-Mohy and Higham is more efficient and a phenomenon known as overscaling is resolved.

3. Condition number of the squaring phase. The condition number of the squaring phase requires the Fréchet derivative of the matrix function $g(X) = X^{2^s}$ at the point $e^{2^{-s}X}$. The recurrence

$$L_{k+1} = e^{2^{k-s}X} L_k + L_k e^{2^{k-s}X}, \quad L_0 = E, \quad k = 0 : s-1 \tag{3.1}$$

yields $L_s = L_g(e^{2^{-s}X}, E)$, which follows from applying the product rule of the Fréchet derivative on $x^{2^{k+1}} = x^{2^k} x^{2^k}$ at the point $e^{2^{-s}X}$ in direction E. Now for suitably chosen s and m, the following algorithm computes e^X and $L_g(e^{2^{-s}X}, E)$.

ALGORITHM 3.1 (basic version). *This algorithm computes $Y \approx e^X$ and $L \approx L_g(e^{2^{-s}X}, E)$. The selection of s and m is based on the algorithm of Higham* [6, Alg. 2.3] *except we set $\theta_{13} = 4.25$.*

```
1  Y = r_m(2^{-s}X), L = E
2  for k = 1: s
3        L ← YL + LY
4        Y ← Y^2
5  end
```

Recall from the proof of Theorem 2.2 that

$$\operatorname{cond}(g, e^{2^{-s}X}) = \frac{\|e^{2^{-s}X}\| \|L_g(e^{2^{-s}X})\|}{\|e^X\|}, \tag{3.2}$$

so the key component to compute or estimate this condition number is to evaluate $\|L_g(e^{2^{-s}X})\|$. If the norm is specified to the Frobenius norm, the relation (2.3) enables us to compute the condition number exactly. Having a method for $L_f(X, E)$, we can compute the jth column of $K_f(X)$ explicitly via $K_f(X)e_j = \operatorname{vec}(L_f(X, \operatorname{unvec}(e_j)))$, where $\{e_j : j = 1\colon n^2\}$ is the standard basis for $\mathbb{C}^{n^2}$ and the operator unvec reverses the action of vec. The following algorithm computes the condition number exactly.

ALGORITHM 3.2 ([7, Alg. 3.17]). *Given a function f and its Fréchet derivative and $X \in \mathbb{C}^{n\times n}$, this algorithm computes* $\operatorname{cond}(f, X)$ *in the Frobenius norm.*

```
1  for j = 1 : n^2
2       Compute Y = L_f(X, unvec(e_j))
3       K(:, j) = vec(Y)
4  end
5  cond(f, X) = ‖K‖_2 ‖X‖_F / ‖f(X)‖_F
```

Cost: $O(n^5)$ flops if $f(X)$ and $L_f(X, E)$ cost $O(n^3)$ flops.

This algorithm is very expensive for large n, so the condition number needs to be estimated rather than computed exactly as we mention below, but for now we use this algorithm to demonstrate the sharpness of the bound (2.4).

REMARK 3.3. *It is worth mentioning that the condition number is independent of the numerical method. We can use any algorithm to compute $e^{2^{-s}X}$ and any method to evaluate $L_g(e^{2^{-s}X}, E)$ such as finite difference approximation* [12] *or complex step approximation* [3] *if X and E are real. However, the scaling and squaring method makes the s powers of $e^{2^{-s}X}$ available, so they can be reused as in Algorithm* 3.1. *Notice that the sharpness of the bound* (2.4) *depends upon the parameter s.*

3.1. Numerical experiment I. In this section we test the sharpness of the bound in (2.4) for $\operatorname{cond}(\exp, X)$ in Frobenius norm. We took 83 test matrices, including some from MATLAB `gallery` function, some from the Matrix Computation Toolbox [4], and test matrices from the e^X literature. We use Algorithm 3.2 to exactly compute the condition number of the matrix exponential and the condition number of the squaring phase, using Algorithm 3.1 to compute $L_g(e^{2^{-s}X}, E)$ and [1, Alg. 6.4] to compute $L_{\exp}(X, E)$. The scaling parameter s is selected by a MATLAB function `expm_mod`, which is based on Higham algorithm [6, Alg. 2.3] with reduction of θ_{13} from 5.4 to 4.25 as suggested by Al-Mohy and Higham [2, Table 3.1] for the accuracy of the Padé approximants evaluation. The top part of Figure 4.1 displays the relative errors $e_j := \|e^X - \widehat{Y}\|_F / \|e^X\|_F$, where $\widehat{Y}$ is the computed exponentials by `expm_mod`, the corresponding condition number $\operatorname{cond}(\exp, X)$, and the bound (2.4); both quantities are multiplied by u to roughly estimate the relative forward errors in light of (1.5).

The bottom part of the figure shows overestimate ratios of the bound (2.4) over the condition numbers. Clearly the worst overestimating ratio is about 45.36 while the average is 10.17 and about 67% of the cases lies below the average.

4. Estimating the condition number. One way to make the bound in (2.4) even sharper is to increase the value of s so that $\|2^{-s}X\|$ becomes smaller and so the value of $e^{\|2^{-s}X\|}$. When we modify `expm_mod` in Experiment I in 3.1 to select s so that $\|2^{-s}X\|_F \le \theta$ for the values $\theta = 3$, $\theta = 2$, and $\theta = 1$, respectively, the overestimate ratios become 13.22, 6.74, and 2.50, respectively. However this approach is unwelcome because increasing s effects the accuracy and the cost of the scaling and squaring algorithm. Alternately, we heuristically neglect the term $e^{\|2^{-s}X\|}/\|e^{2^{-s}X}\|$ from the bound (2.4) and consider the formula

$$\kappa_g(s,X) = \begin{cases} \|X\|, & s = 0, \\ \mathrm{cond}(g, e^{2^{-s}X}), & s > 0. \end{cases} \tag{4.1}$$

as an approximation to the condition number of the matrix exponential. To justify this choice, notice that $\kappa_g(0,X) = \|X\|$ is the lower bound of $\mathrm{cond}(\exp, X)$ by [7, Lem. 10.15]. When s is large, it is the squaring phase that is responsible for the loss of accuracy since the scaling and squaring algorithm guarantees that the computation of the exponential of the scaled matrix is accurate, $O(u)$. In addition, $e^{\|2^{-s}X\|}$ is not the sharpest bound we hope for $\|L_{\exp}(2^{-s}X)\|$.

The next numerical experiment shows that our heuristic formula captures the behavior of $\mathrm{cond}(\exp, X)$.

4.1. Numerical experiment II. We repeat Experiment I in 3.1 by using $\kappa_g(s,X)$ in place of the bound (2.4). Figure 4.2 displays the relative errors $\|e^X - \widehat{Y}\|_F/\|e^X\|_F$, $\kappa_g(s,X)u$, and $\mathrm{cond}(\exp, X)u$. It is obvious that $\kappa_g(s,X)$ nicely captures the behavior of $\mathrm{cond}(\exp, X)$, and the worst underestimate and overestimate ratios are 0.24 and 2.93, respectively, that is,

$$0.24 \le \frac{\kappa_g(s,X)}{\mathrm{cond}(\exp, X)} \le 2.93.$$

In meanwhile Figure 4.3 presents a performance profile for $\kappa_g(s,X)u$ and $\mathrm{cond}(\exp, X)u$ as error estimates to the relative forward errors, e_j. That is, for the jth test matrix we compute the relative quantities $|e_j - \kappa_g(s,X)u|/e_j$ and $|e_j - \mathrm{cond}(\exp, X)u|/e_j$ with e_j being set to 10^{-18} if $e_j = 0$ to avoid division by zero. The performance profile indicates that $\kappa_g(s,X)u$ better estimates the relative forward errors than $\mathrm{cond}(\exp, X)u$.

4.2. Numerical experiment III. In this experiment we use an optimization function to find extreme points for which $\kappa_g(s,X)$ over- and underestimates $\mathrm{cond}(\exp, X)$. We implement the multi-directional search method of Dennis and Torczon [13, 14] used by Higham [5] in the context of matrix computations where he examined the reliability of matrix condition number estimators. A MATLAB code of the multi-directional search method is available in [4] under the name `mdsmax`. We run the maximizer `mdsmax` to seek the maximal value of the ratio $\kappa_g(s,X)/\mathrm{cond}(\exp, X)$ and its reciprocal. For each ratio we run the function `mdsmax` 83 times using the 83 matrices that we used the preceding experiments as an initial value. We found in the worst case that

$$0.23 \le \frac{\kappa_g(s,X)}{\mathrm{cond}(\exp, X)} \le 3.35.$$

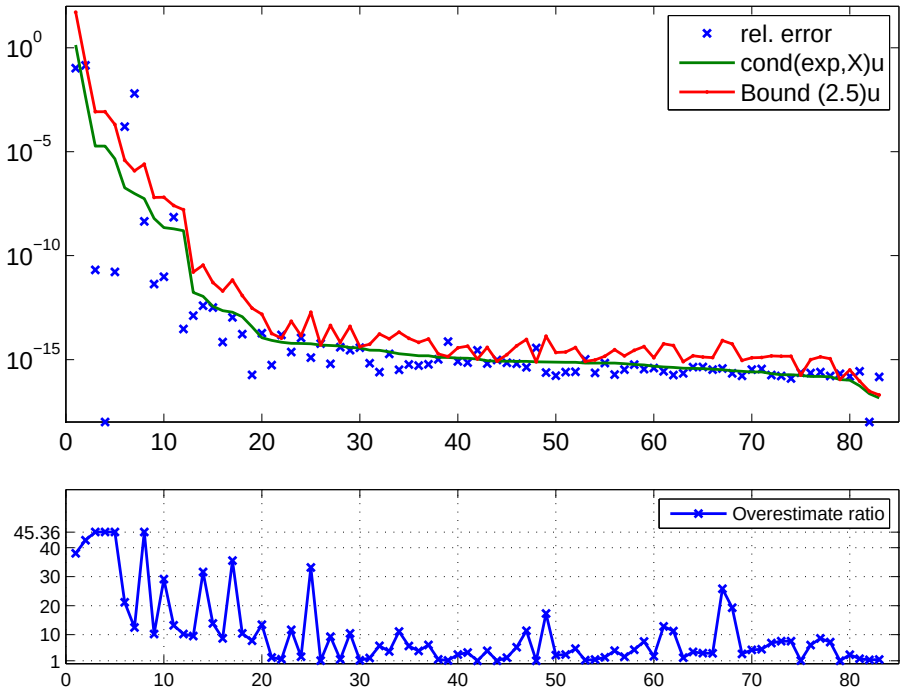

FIG. 4.1. *Experiment I in* 3.1. *Normwise relative errors in computing* e^X, $\mathrm{cond}(\exp, X)u$, *and the bound* (2.4) *multiplied by* u *(top). Overestimate ratio for each case (bottom).*

We have noticed that the exponential of the matrices, where extreme points attained, is very ill-conditioned.

5. Practical norm estimation of the Fréchet derivative. Our numerical experiments above demonstrate that the condition number of the monomial $g(X) = X^{2^s}$ at the point $e^{2^{-s}X}$ can serve as an approximant to the condition number of the matrix exponential since $\kappa_g(s, X) = \mathrm{cond}(g, e^{2^{-s}X})$ for $s > 0$. As we mention before, the key component to compute or estimate $\mathrm{cond}(g, e^{2^{-s}X})$ is $\|L_g(e^{2^{-s}X})\|$. If we base the condition number on the Frobenius norm, Algorithm 3.2, for a general matrix function f, forms the matrix $K_f(X)$ and then evaluates its 2-norm (recall that $\|L_f(X)\|_F = \|K_f(X)\|_2$), which is all we need to compute the condition number of the matrix function. However forming $K_f(X)$ is impractical since it costs $O(n^5)$ flops. To estimate $\|K_f(X)\|_2$, the power method [12], [7, sect. 3.4] requires only the action of the matrices $K_f(X)$ and $K_f(X)^*$ on some vectors z and w, respectively, as follows.

$$K_f(X)z = \mathrm{vec}\big(L_f(X, \mathrm{unvec}(z))\big), \quad K_f(X)^*w = \mathrm{vec}\big(L_f^\star(X, \mathrm{unvec}(w))\big). \tag{5.1}$$

Here, $L_f^\star(X) = L_{\overline{f}}(X^*)$, the adjoint operator of $L_f(X)$ with respect to the inner product $\langle X, Y\rangle = \mathrm{trace}(Y^*X)$ on $\mathbb{C}^{n\times n}$, and $\overline{f}(z) = \overline{f(\overline{z})}$ [9, Lem. 6.2]. If $f : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$, $\overline{f} = f$ and hence $L_f^\star(X) = L_f(X^*)$. In this case, for any $E \in \mathbb{C}^{n\times n}$, we have

$$L_f^*(X, E) = L_f(X^*, E) = L_f(X, E^*)^*. \tag{5.2}$$

However, the power method lacks convergence tests, and because of its linear convergence rate the number of iteration required is unpredictable. Instead, the condition

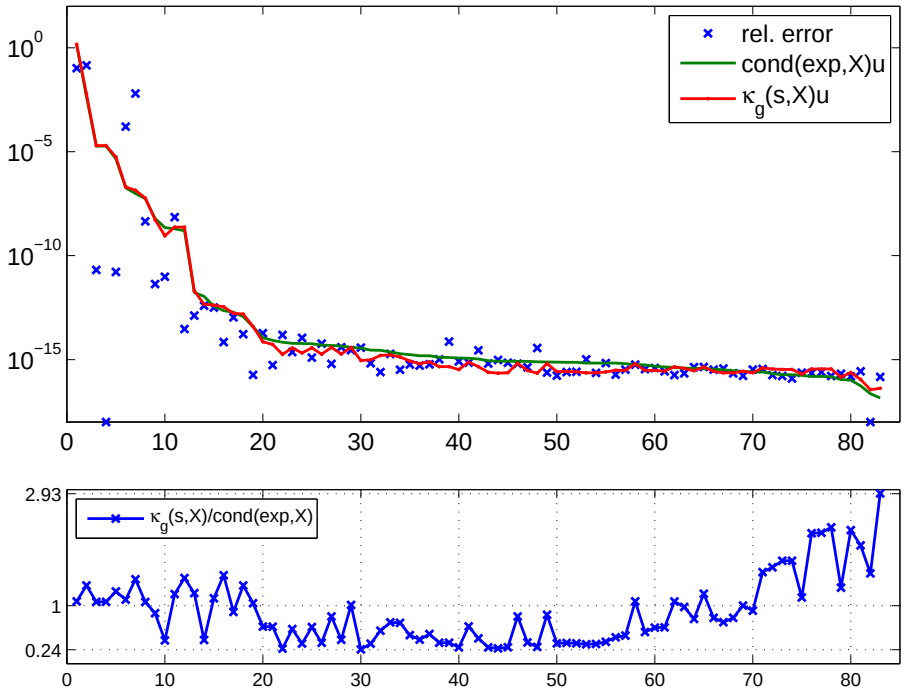

FIG. 4.2. *Experiment II in* 4.1. *Normwise relative errors in* e^X, $\kappa_g(s,X)u$, *and* $\mathrm{cond}(\exp,X)u$ *(top). Values of* $\kappa_g(s,X)/\mathrm{cond}(\exp,X)$ *for each case (bottom).*

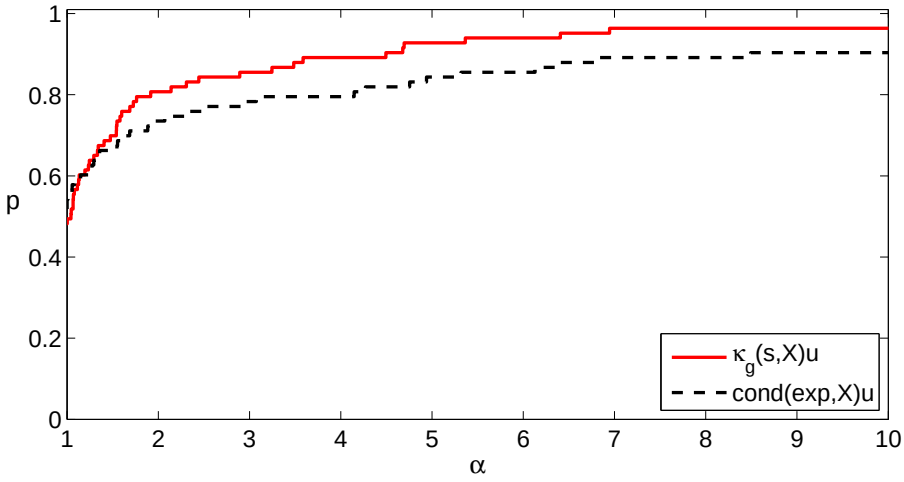

FIG. 4.3. *The performance profile for Experiment II in* 4.1.

number is based on the 1-norm. The block 1-norm estimation algorithm of Higham and Tisseur [10, Alg. 2.4], which forms the basis of MATLAB function `normest1`, has a "built-in" starting matrix, convergence test, and a more predictable number of iterations. Although there is no analogue to the relation (2.2) for the 1-norm, the next lemma shows the relation between $\|K_f(X)\|_1$ and $\|L_f(X)\|_1$.

LEMMA 5.1 ([7, Lemma 3.18]). *For $X \in \mathbb{C}^{n\times n}$ and any Fréchet differentiable function $f : \mathbb{C}^{n\times n} \to \mathbb{C}^{n\times n}$,*

$$\frac{\|L_f(X)\|_1}{n} \le \|K_f(X)\|_1 \le n\|L_f(X)\|_1. \qquad (5.3)$$ □

We can apply [10, Alg. 2.4] implicitly on $\|K_f(X)\|_1$ using (5.1) to estimate the 1-norm of the Fréchet derivative.

ALGORITHM 5.2 (block 1-norm estimator for the Fréchet derivative). *Given a matrix $X \in \mathbb{C}^{n\times n}$ this algorithm uses a block 1-norm estimator to produce an estimate η of $\|L_f(X)\|_1$, given the ability to compute $L_f(X, E)$ and $L_f^\star(X, E)$ for any E. More precisely, $\eta \le \|K_f(X)\|_1$, where $\|K_f(X)\|_1$ satisfies* (5.3).

1 Apply Algorithm 2.4 from Higham and Tisseur [10] with parameter $t = 2$ to the Kronecker matrix representation $K_f(X)$ of $L_f(X)$, making use of the relations in (5.1).

6. Algorithms. In this section we give two versions of the scaling and squaring algorithm that compute e^X alongside $\kappa_g(s, X)$. The first is an update to the algorithm of Al-Mohy and Higham [1, Alg. 7.4], and the second is an extension to the algorithm of Al-Mohy and Higham [2, Alg. 6.1].

6.1. Algorithm I. The algorithm of Al-Mohy and Higham [1, Alg. 7.4] returns e^X and an estimate to its condition number using the Fréchet derivative $L_{\exp}(X, E)$. Next algorithm estimates the condition number using only the Fréchet derivative of the squaring phase.

ALGORITHM 6.1. *Given a matrix $X \in \mathbb{C}^{n\times n}$ this algorithm computes $Y_0 = e^X$ and estimates $\gamma \approx \kappa_g(s, X)$ as a condition number of the problem. The algorithm uses the parameters θ_m tabulated in* [2, Table 3.1].

1 $s = 0$
2 for $m = [3\ 5\ 7\ 9]$
3 if $\|X\|_1 \le \theta_m$
4 Evaluate $Y_s = r_m(X)$, $\gamma = \|X\|_1$, goto 20
5 end
6 end
7 $s = \lceil \log_2(\|X\|_1/\theta_{13}) \rceil$
8 Evaluate $Y_s = r_m(2^{-s}X)$, % store Y_s.
9 for $i = s\colon -1\colon 1$
10 $Y_{i-1} = Y_i^2$, % store Y_{i-1}.
11 end
12 Use Algorithm 5.2 to produce an estimate $\eta \approx \|L_g(Y_s)\|_1$.
13 $\cdots\cdots$ To compute $L = L_g(Y_s, E)$ for a given E:
14 $L = E$
15 for $i = s\colon -1\colon 1$
16 $L \leftarrow Y_i L + L Y_i$
17 end
18 $\cdots\cdots$ To compute $L_g^*(Y_s, E)$ for a given E:
19 Execute lines 14–17 with E replaced by E^* then take L^* at the end.
20 $\gamma = \|Y_s\|_1 \eta / \|Y_0\|_1$

Cost: $\pi_m + 17s$ matrix multiplications plus one solve of multiple right-hand side linear system to form $r_m(2^{-s}X)$. The cost analysis is given in section 7.

6.2. Algorithm II. Recently, Al-Mohy and Higham improved the scaling and squaring algorithm upon `expm` using two key ideas [2, Alg. 6.1]. They tackled the overscaling problem that causes loss of accuracy in floating point arithmetic due to choosing a larger than necessary scaling parameter s by `expm`. First, they based the selection of s on members of the sequence $\{\, \alpha_p(X) \,\}$ instead of $\|X\|$. Second, they treated triangular matrices in a special way. Their algorithm computes and updates diagonal and superdiagonal elements in the squaring phase using exact formulas. As a result, their algorithm is no slower than `expm` and delivers better accuracy.

We extend the algorithm of Al-Mohy and Higham [2, Alg. 6.1] to produce an estimate to $\kappa_g(s, X)$ by incorporating the lines 12–20 of Algorithm 6.1 inside it.

ALGORITHM 6.2. *Given a matrix $X \in \mathbb{C}^{n\times n}$ this algorithm computes $Y_0 = e^X$ and estimates $\gamma \approx \kappa_g(s, X)$ as an error estimate for the problem. The algorithm uses the parameters θ_m tabulated in* [2, Table 3.1]. *The functions* `normest` *and* `ell` *are defined in* [2, Alg. 5.1].

1 $s = 0$
2 $X_2 = X^2$
3 $d_6 = \texttt{normest}(X_2, 3)^{1/6}$, $\alpha_2 = \max\big(\texttt{normest}(X_2, 2)^{1/4}, d_6\big)$
4 if $\alpha_2 \le \theta_3$ and $\texttt{ell}(X, 3) = 0$
5 Evaluate $Y_s = r_3(X)$, $\gamma = \|X\|_1$
6 quit
7 end
8 $X_4 = X_2^2$, $d_4 = \|X_4\|_1^{1/4}$
9 $\alpha_2 = \max\big(d_4, d_6\big)$
10 if $\alpha_2 \le \theta_5$ and $\texttt{ell}(X, 5) = 0$
11 Evaluate $Y_s = r_5(X)$, $\gamma = \|X\|_1$
12 quit
13 end
14 $X_6 = X_2X_4$, $d_6 = \|X_6\|_1^{1/6}$
15 $d_8 = \texttt{normest}(X_4, 2)^{1/8}$, $\alpha_3 = \max(d_6, d_8)$
16 for $m = [7, 9]$
17 if $\alpha_3 \le \theta_m$ and $\texttt{ell}(X, m) = 0$
18 Evaluate $Y_s = r_m(X)$, $\gamma = \|X\|_1$
19 quit
20 end
21 end
22 $\alpha_4 = \max\big(d_8, \texttt{normest}(X_4, X_6)^{1/10}\big)$
23 $\eta = \min(\alpha_3, \alpha_4)$
24 $s = \max\big(\lceil \log_2(\eta/\theta_{13}) \rceil, 0\big)$
25 $s = s + \texttt{ell}(2^{-s}X, 13)$
26 $X \leftarrow 2^{-s}X$, $X_2 \leftarrow 2^{-2s}X_2$, $X_4 \leftarrow 2^{-4s}X_4$, $X_6 \leftarrow 2^{-6s}X_6$, % to form r_{13}.
27 Evaluate $Y_s = r_{13}(X)$, % store Y_s
28 if X is triangular
29 Invoke [2, Code Fragment 2.1], % store the powers of Y_s therein.
30 else
31 for $i = s\colon -1\colon 1$
32 $Y_{i-1} = Y_i^2$, % store Y_{i-1}
33 end
34 end
35 To estimate η, execute lines 12–20 of Algorithm 6.1.

Cost: $\pi_m + 17s$ matrix multiplications plus one solve of multiple right-hand side linear system to form $r_m(2^{-s}X)$. The cost analysis is given in section 7.

7. Computational cost analysis. In this section we show how significant the reduction in computational cost is when considering $\kappa_g(s, X)$ as an approximant to the condition number of the matrix exponential. The algorithm of Al-Mohy and Higham [1, Alg. 7.4] that computes e^X and estimates $\mathrm{cond}(\exp, X)$ in the 1-norm typically costs $17(\pi_m + s) + 8$ matrix multiplications and *two* solves of multiple right-hand side linear systems [1, sect. 7], where π_m is the number of matrix multiplications required to form the Padé approximant $r_m(2^{-s}X)$, [6, Table 2.2]. The details are as follows

1. $\pi_m + s$ matrix multiplications plus one solve of multiple right-hand side linear system are needed to compute $r_m(2^{-s}X)^{2^s} \approx e^X$.
2. Computing $L_{r_m}(X, E) \approx L_{\exp}(X, E)$ requires $2\pi_m + 1 + 2s$ matrix multiplications plus one solve of multiple right-hand side linear system, utilizing the powers of X and the s powers of $r_m(2^{-s}X)$ used in item 1, [1, sect. 6].
3. The block 1-norm estimator of Higham and Tisseur [10, Alg. 2.4] typically requires 8 invocations of $L_{\exp}(X, E)$ to estimate $\|K_{\exp}(X)\|_1$ [1, sect. 7].

However, computing e^X and estimating $\kappa_g(s, X)$ require $\pi_m + 17s$ matrix multiplications plus one solve of multiple right-hand side linear system. The details are as above with items 2 and 3 updated as follows

2. Computing $L_g(e^{2^{-s}X}, E)$ requires $2s$ matrix multiplications, assuming the s powers of $r_m(2^{-s}X)$ are available from 1.
3. The block 1-norm estimator of Higham and Tisseur [10, Alg. 2.4] typically requires 8 invocations of $L_g(e^{2^{-s}X}, E)$ to estimate $\|K_g(e^{2^{-s}X})\|_1$.

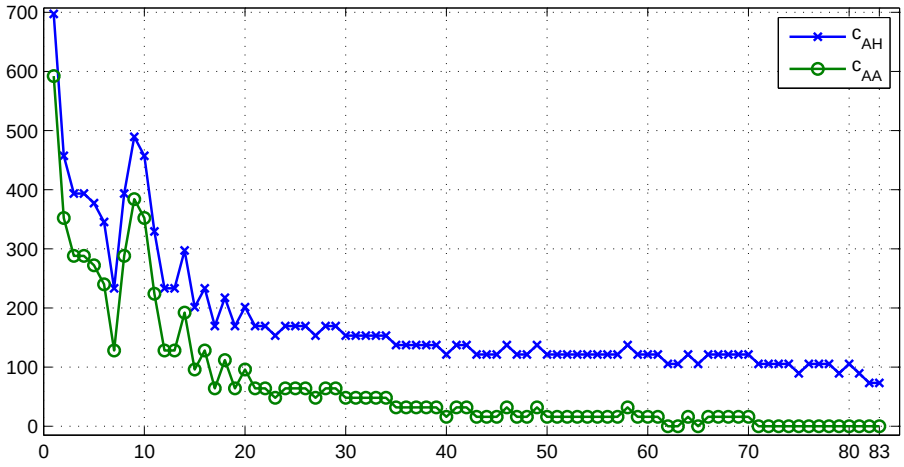

FIG. 7.1. *The cost in matrix multiplication to compute* $\mathrm{cond}(\exp, X)$ *and* $\kappa_g(s, X)$.

Thus the *total saving* is $16\pi_m + 8$ matrix multiplications plus one solve of multiple right-hand side linear system. Neglecting the cost of the matrix exponential, it is obvious that estimating $\mathrm{cond}(\exp, X)$ by the algorithm of Al-Mohy and Higham costs about $c_{AH} := 16(\pi_m + s) + 8 + 4/3$, where the term $4/3$ accounts (in matrix product) for the cost of the solution of the multiple right-hand side linear system [7, Table C.1] whereas $\kappa_g(s, X)$ costs only $c_{AA} := 16s$. Thus the percentage of $O(n^3)$ flops we save

by using $\kappa_g(s, X)$ is

$$\frac{100(c_{AH} - c_{AA})}{c_{AH}} = \frac{100(12\pi_m + 7)}{12(\pi_m + s) + 7},$$

so when $s = 0$ we save 100% of the cost since $\kappa_g(0, X) = \|X\|_1$, and the percentage decreases as s increases. As an example, for a matrix X with $\|X\|_1 = 10^8$, Algorithm 6.1 below will select $s = 25$ and $m = 13$. Since $\pi_{13} = 6$ [6, Table 2.2], Algorithm 6.1 can save about 20.8% of the computational cost comparing to the algorithm of Al-Mohy and Higham. Figure 7.1 displays the number of matrix multiplications required to compute $\mathrm{cond}(\exp, X)$ and $\kappa_g(s, X)$ for every test matrix X described in our numerical experiments.

7.1. Numerical experiment IV. In Experiment II in 4.1, the Frobenius norms of $L_{\exp}(X)$ and $L_g(e^{2^{-s}X})$ are computed exactly using Algorithm 3.2. However, the practical algorithms (the algorithm of Al-Mohy and Higham [1, Alg. 7.4] and Algorithm 6.1) implement the 1-norm estimator to estimate the 1-norm of these operators. Here we repeat the experiment using Algorithm 6.1 to estimate $\kappa_g(s, X)$ and the algorithm of Al-Mohy and Higham to estimate $\mathrm{cond}(\exp, X)$ in the 1-norm for every test matrix X used in all above experiments. Figure 7.2 shows that the 1-norm estimation of $\kappa_g(s, X)$ perfectly captures the behavior of the 1-norm estimation of $\mathrm{cond}(\exp, X)$. In addition, the performance profile in Figure 7.3 displays that $\kappa_g(s, X)$ outperforms $\mathrm{cond}(\exp, X)$ in terms of error estimation of the relative forward errors.

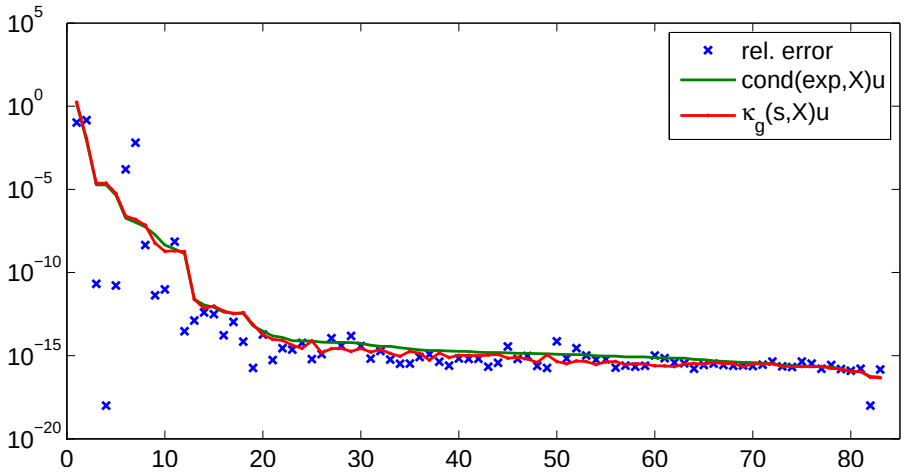

FIG. 7.2. *Experiment IV in* 7.1*: Normwise relative errors in* e^X*,* $\kappa_g(s, X)u$*, and* $\mathrm{cond}(\exp, X)u$*. The evaluation of* e^X *and the estimation of* $\kappa_g(s, X)$ *are given by Algorithm* 6.1*.*

We then repeat this experiment using Algorithm 6.2 in place of Algorithm 6.1) to compute the exponential of the test matrices and estimate $\kappa_g(s, X)$. Figure 7.4 displays the result (the data is sorted in descending order based on $\kappa_g(s, X)$) . Notice that the curve of $\mathrm{cond}(\exp, X)u$ has several spikes and for these cases $\kappa_g(s, X)u$ gives the better estimates of the relative forward errors, so $\kappa_g(s, X)$ cannot be regarded as an approximant to the condition number. Yet, the performance profile in Figure 7.5 demonstrates that $\kappa_g(s, X)$ outperforms the condition number as an error estimator of the relative forward errors. Recall that Algorithm 6.2 selects the scaling parameter

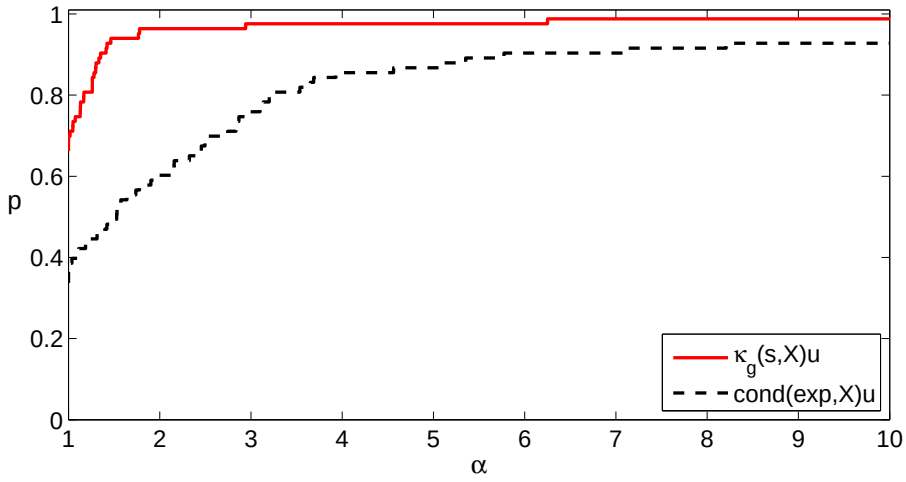

FIG. 7.3. *Experiment IV in* 7.1*: Performance profile for the data in Figure* 7.2.

s using the sequence $\{\, \alpha_p(2^{-s}X) \,\}$ instead of $\|X\|$, so for some nonnormal matrices it is possible that $\alpha_p(2^{-s}X) \ll \|X\|$. Thus, $\|2^{-s}X\|$ can be significantly large and hence the bound in (2.4) becomes arbitrary huge and no longer sharp because of the term $e^{\|2^{-s}X\|}$. However, $\kappa_g(s, X)$ is not affected by $\|2^{-s}X\|$ but by the error propagation in the squaring phase.

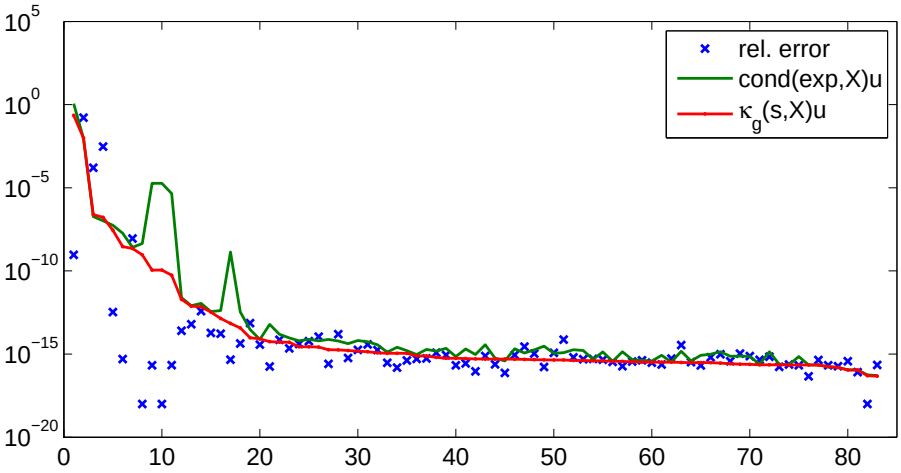

FIG. 7.4. *Experiment IV in* 7.1*: Normwise relative errors in* e^X, $\kappa_g(s, X)u$, *and* $\mathrm{cond}(\exp, X)u$. *The evaluation of* e^X *and the estimation of* $\kappa_g(s, X)$ *are given by Algorithm* 6.2.

8. Conclusion. In his book, Higham [7, Prob. 10.16] gives an open question about the stability of the scaling and squaring algorithm for e^X. He wonders whether the rounding errors in the squaring phase can be related to the condition number of e^X problem. We have reached a milestone in answering this question. We relate the condition number of e^X to the condition number of the squaring phase and our numerical experiments reveal that the condition number of the matrix exponential is

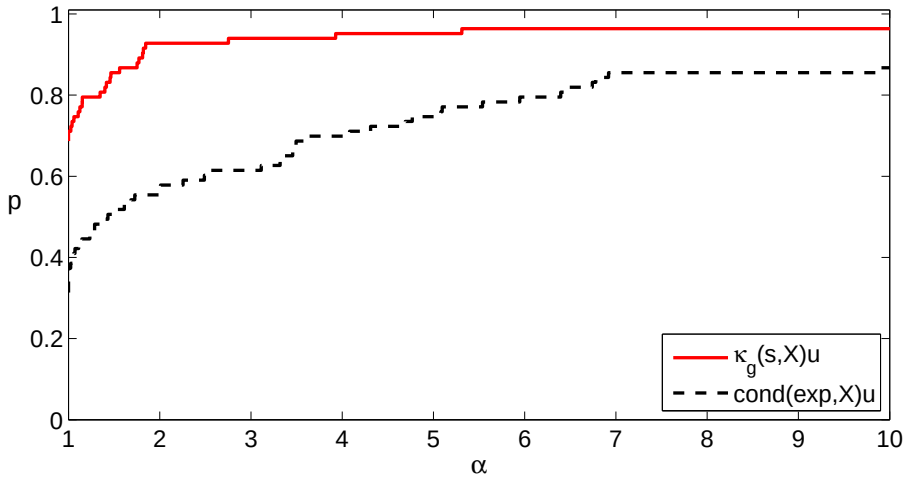

FIG. 7.5. *Experiment IV in* 7.1*: Performance profile for the data in Figure* 7.4

almost the condition number of the squaring phase. However, rigorous rounding error analysis remains unavailable. The great advantage of this relation is that we obtain an efficient and reliable error estimator for the scaling and squaring algorithm.

REFERENCES

[1] Awad H. Al-Mohy and Nicholas J. Higham. Computing the Fréchet derivative of the matrix exponential, with an application to condition number estimation. *SIAM J. Matrix Anal. Appl.*, 30(4):1639–1657, 2009.

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

[3] Awad H. Al-Mohy and Nicholas J. Higham. The complex step approximation to the Fréchet derivative of a matrix function. *Numer. Algorithms*, 53(1):133–148, 2010.

[4] Nicholas J. Higham. The Matrix Computation Toolbox. `http://www.maths.manchester.ac.uk/~higham/mctoolbox`.

[5] Nicholas J. Higham. Optimization by direct search in matrix computations. *SIAM J. Matrix Anal. Appl.*, 14(2):317–333, 1993.

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

[7] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2008. xx+425 pp. ISBN 978-0-898716-46-7.

[8] Nicholas J. Higham and Edvin Deadman. A catalogue of software for matrix functions. Version 1.0. MIMS EPrint 2014.8, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, February 2014. 19 pp.

[9] Nicholas J. Higham and Lijing Lin. An improved Schur–Padé algorithm for fractional powers of a matrix and their Fréchet derivatives. *SIAM J. Matrix Anal. Appl.*, 34(3):1341–1360, 2013.

[10] Nicholas J. Higham and Françoise Tisseur. A block algorithm for matrix 1-norm estimation, with an application to 1-norm pseudospectra. *SIAM J. Matrix Anal. Appl.*, 21(4):1185–1201, 2000.

[11] Bo Kågström. Bounds and perturbation bounds for the matrix exponential. *BIT*, 17:39–57, 1977.

[12] Charles S. Kenney and Alan J. Laub. Condition estimates for matrix functions. *SIAM J. Matrix Anal. Appl.*, 10(2):191–209, 1989.

[13] Virginia J. Torczon. *Multi-Directional Search: A Direct Search Algorithm for Parallel Machines*. PhD thesis, Rice University, Houston, TX, USA, May 1989. vii+85 pp.

[14] Virginia J. Torczon. On the convergence of the multidirectional search algorithm. *SIAM J. Optim.*, 1(1):123–145, 1991.

[15] Charles F. Van Loan. The sensitivity of the matrix exponential. *SIAM J. Numer. Anal.*, 14 (6):971–981, 1977.