The Matrix Unwinding Function, with an Application to Computing the Matrix Exponential

Aprahamian, Mary and Higham, Nicholas J.

2013

MIMS EPrint: **2013.21**

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

THE MATRIX UNWINDING FUNCTION, WITH AN APPLICATION TO COMPUTING THE MATRIX EXPONENTIAL*

MARY APRAHAMIAN† AND NICHOLAS J. HIGHAM†

Abstract. A new matrix function corresponding to the scalar unwinding number of Corless, Hare, and Jeffrey is introduced. This matrix unwinding function, $\mathcal{U}$, is shown to be a valuable tool for deriving identities involving the matrix logarithm and fractional matrix powers, revealing, for example, the precise relation between $\log A^\alpha$ and $\alpha \log A$. The unwinding function is also shown to be closely connected with the matrix sign function. An algorithm for computing the unwinding function based on the Schur–Parlett method with a special reordering is proposed. It is shown that matrix argument reduction using the function $\mathrm{mod}(A) = A - 2\pi i \mathcal{U}(A)$, which has eigenvalues with imaginary parts in the interval $(-\pi, \pi]$ and for which $\mathrm{e}^A = \mathrm{e}^{\mathrm{mod}(A)}$, can give significant computational savings in the evaluation of the exponential by scaling and squaring algorithms.

Key words. matrix unwinding function, unwinding number, matrix logarithm, matrix power, matrix exponential, argument reduction

AMS subject classifications. 65F30, 15A24

1. Introduction.

In previous work on the matrix logarithm [18, Chap. 11] one of us has made use of a scalar function called the unwinding number. In this work we define and investigate the corresponding primary matrix function. We show that the matrix unwinding function is just as useful as its scalar counterpart. First, it is a valuable tool for stating matrix identities involving the matrix logarithm and fractional matrix powers, because it elegantly prescribes the correction needed when identities that are generalized from the positive scalar case break down. For example, the unwinding function neatly captures the difference between $\log A^\alpha$ and $\alpha \log A$. In addition to this role as a theoretical tool, the unwinding function is also useful computationally. It enables us to preprocess a matrix so that its eigenvalues all have imaginary parts lying in the interval $(-\pi, \pi]$, while not changing the exponential of the matrix. We show that this matrix argument reduction can provide a large decrease in norm and can thereby produce significant computational savings when the scaling and squaring method is used to evaluate the matrix exponential.

We define the scalar unwinding number in the next section and recap some of its key properties. The matrix unwinding function $\mathcal{U}(A)$ is defined in section 3, where we deal carefully with a subtlety concerning the meaning of the derivative at points with imaginary parts an odd integer multiple of π. In section 3.1 basic properties of $\mathcal{U}(A)$ are derived. Bounds for the norm and the condition number of $\mathcal{U}(A)$ are given in section 3.2, where we also discuss estimation of the condition number. In section 3.3 we derive a number of matrix identities involving the functions $\log z$ and z^α. Connections with the matrix sign function are explored in section 3.4. In section 4 we give a Schur–Parlett algorithm for computing $\mathcal{U}(A)$ based on a reordering of the Schur form specific to the unwinding function and give some analysis connecting the conditioning of the Sylvester equations to the conditioning of $\mathcal{U}$. In section 5 we show

*Version of May 7, 2013. The work of both authors was supported by European Research Council Advanced Grant MATFUN (267526).

†School of Mathematics, The University of Manchester, Manchester, M13 9PL, England (mary.aprahamian@postgrad.manchester.ac.uk, nicholas.j.higham@manchester.ac.uk, http://www.ma.man.ac.uk/~higham). The work of the second author was also supported by Engineering and Physical Sciences Research Council grant EP/E050441/1 (CICADA: Centre for Interdisciplinary Computational and Dynamical Analysis).

via numerical experiments that the algorithm performs well in practice. In section 6 we explore the use of the unwinding function for argument reduction with the matrix exponential, showing that it can produce significant computational savings when used with the scaling and squaring method. Some final remarks are given in section 7.

2. The unwinding number. We first state our conventions for three key functions of a complex variable:

(i) arg is the principal argument: $-\pi < \arg z \le \pi$.
(ii) log is the principal logarithm: $-\pi < \operatorname{Im} \log z \le \pi$.
(iii) For $\alpha, z \in \mathbb{C}$ we define $z^\alpha = \mathrm{e}^{\alpha \log z}$. In particular, $z^{1/2}$ is the principal square root: $\operatorname{Re} z^{1/2} \ge 0$ and $(-1)^{1/2} = i$.

Motivation for these particular choices of where to close the branches is given by Kahan [26]. We will use repeatedly the key properties $\mathrm{e}^{\log z} = z$ and $\mathrm{e}^{z_1+z_2} = \mathrm{e}^{z_1}\mathrm{e}^{z_2}$. The open negative real axis will be denoted by $\mathbb{R}^-$.

The unwinding number of $z \in \mathbb{C}$ is defined by

$$\mathcal{U}(z) = \frac{z - \log \mathrm{e}^z}{2\pi i}. \tag{2.1}$$

The definition can be rewritten

$$z = \log \mathrm{e}^z + 2\pi i \mathcal{U}(z), \tag{2.2}$$

so that $2\pi i \mathcal{U}(z)$ is the discrepancy between $\log e^z$ and z.

The term "unwinding number", with a definition differing from ours only in sign, first appeared in Corless and Jeffrey [12] and Jeffrey, Hare, and Corless [25]. A definition with the same sign as (2.1), and an explanation of why this sign is preferred, is given by Bradford, Corless, Davenport, Jeffrey, and Watt [8]. Related definitions can be found in Apostol [4, Thm. 1.48], Aslaksen [5], Bradford [7], and Patton [35]. With the exception of [4], in all these references the interest in the unwinding number stems from its suitability for use in computer algebra.

The following lemma from [12], [25] gives a formula for the unwinding number that is easier to evaluate than (2.1).

LEMMA 2.1. *The unwinding number of $z \in \mathbb{C}$ can be expressed using the ceiling function as*

$$\mathcal{U}(z) = \left\lceil \frac{\operatorname{Im} z - \pi}{2\pi} \right\rceil. \tag{2.3}$$

Proof. Exponentiating both sides of (2.2) we have

$$\mathrm{e}^z = \mathrm{e}^{\log \mathrm{e}^z + 2\pi i \mathcal{U}(z)} = \mathrm{e}^z \mathrm{e}^{2\pi i \mathcal{U}(z)},$$

so $\mathrm{e}^{2\pi i \mathcal{U}(z)} = 1$ and hence $\mathcal{U}(z) \in \mathbb{Z}$. Taking imaginary parts in (2.2) gives $-\pi < \operatorname{Im} z - 2\pi \mathcal{U}(z) \le \pi$, which can be written

$$\frac{\operatorname{Im} z - \pi}{2\pi} \le \mathcal{U}(z) < \frac{\operatorname{Im} z + \pi}{2\pi}.$$

The result follows since $\mathcal{U}(z) \in \mathbb{Z}$. ∎

Thus $\mathcal{U}$ takes integer values and is constant for $\operatorname{Im} z$ on the intervals $((2k-1)\pi, (2k+1)\pi]$ for all integers k. It is therefore easy to characterize when $\mathcal{U}(z) = 0$, or equivalently, $\log \mathrm{e}^z = z$.

COROLLARY 2.2. *For $z \in \mathbb{C}$, $\mathcal{U}(z) = 0$ if and only if $\operatorname{Im} z \in (-\pi, \pi]$.*

We now consider some of the most useful properties of the unwinding number. In the formulae below it is implicitly understood that $z = 0$ is excluded from formulae involving $\log z$.

LEMMA 2.3. *For $z \in \mathbb{C}$,*

$$\mathcal{U}(\overline{z}) = \mathcal{U}(-z) = \begin{cases} -\mathcal{U}(z), & \operatorname{Im} z \neq (2k+1)\pi,\ k \in \mathbb{Z}, \\ -\mathcal{U}(z) - 1, & \textit{otherwise.} \end{cases}$$

Proof. Straightforward from Lemma 2.1. □

LEMMA 2.4. *For $z \in \mathbb{C}$ and $\alpha \in [-1, 1]$,*

$$\mathcal{U}(\alpha \log z) = \begin{cases} 0, & z \in \mathbb{C},\ \alpha \in (-1, 1] \text{ or } z \notin \mathbb{R}^-,\ \alpha = -1, \\ -1, & z \in \mathbb{R}^-,\ \alpha = -1. \end{cases}$$

Proof. Since $\operatorname{Im} \log z \in (-\pi, \pi]$, $\mathcal{U}(\alpha \log z) = 0$ for $\alpha \in (-1, 1]$ by Corollary 2.2. For $\alpha = -1$, $\mathcal{U}(-\log z) = 0$ unless $\operatorname{Im}(-\log z) = -\pi$, that is, $z \in \mathbb{R}^-$, in which case $\mathcal{U}(-\log z) = \mathcal{U}(-\pi i) = -1$. □

The next three results are some of the "useful theorems" that motivated the introduction of the unwinding number in [12]. They show that $\mathcal{U}$ provides the appropriate correction term in three important formulas involving the logarithm.

LEMMA 2.5. *For $z_1, z_2 \in \mathbb{C}$, $\log(z_1 z_2) = \log z_1 + \log z_2 - 2\pi i \mathcal{U}(\log z_1 + \log z_2)$.*

Proof. From (2.2) we have

$$\begin{aligned} \log z_1 + \log z_2 &= \log(\mathrm{e}^{\log z_1 + \log z_2}) + 2\pi i \mathcal{U}(\log z_1 + \log z_2) \\ &= \log(\mathrm{e}^{\log z_1} \mathrm{e}^{\log z_2}) + 2\pi i \mathcal{U}(\log z_1 + \log z_2) \\ &= \log(z_1 z_2) + 2\pi i \mathcal{U}(\log z_1 + \log z_2), \end{aligned}$$

as required. □

LEMMA 2.6. *For $\alpha, z \in \mathbb{C}$, $\log(z^\alpha) = \alpha \log z - 2\pi i \mathcal{U}(\alpha \log z)$.*

Proof. Immediate from the definitions of z^α and $\mathcal{U}(z)$. □

Lemmas 2.4 and 2.6 together give that for $\alpha \in (-1, 1]$ the identity $\log(z^\alpha) = \alpha \log z$ holds. Note that $\alpha = 1/2$ yields the important special case $\log(z^{1/2}) = \frac{1}{2} \log z$. Lemmas 2.4 and 2.6 also give $\log(z^{-1}) = -\log z$ for $z \notin \mathbb{R}^-$.

LEMMA 2.7. *For $z_1, z_2 \in \mathbb{C}$, $(z_1 z_2)^{1/2} = z_1^{1/2} z_2^{1/2} (-1)^{\mathcal{U}(\log z_1 + \log z_2)}$.*

Proof. Using Lemma 2.5 we have

$$\begin{aligned} (z_1 z_2)^{1/2} &= \exp\left(\tfrac{1}{2} \log(z_1 z_2)\right) \\ &= \exp\left(\tfrac{1}{2} \left(\log z_1 + \log z_2 - 2\pi i \mathcal{U}(\log z_1 + \log z_2)\right)\right) \\ &= z_1^{1/2} z_2^{1/2} \exp(-\pi i \mathcal{U}(\log z_1 + \log z_2)) \\ &= z_1^{1/2} z_2^{1/2} (-1)^{\mathcal{U}(\log z_1 + \log z_2)}. \end{aligned}$$

□

An important application of the unwinding number is in accurate evaluation of elements of functions of triangular matrices. It is well known that for $\lambda_1 \neq \lambda_2$ [18, sec. 4.6],

$$f\left(\begin{bmatrix} \lambda_1 & t_{12} \\ 0 & \lambda_2 \end{bmatrix}\right) = \begin{bmatrix} f(\lambda_1) & t_{12}\dfrac{f(\lambda_2)-f(\lambda_1)}{\lambda_2-\lambda_1} \\ 0 & f(\lambda_2) \end{bmatrix},$$

but in floating point arithmetic evaluation of the $(1,2)$ element from this formula can incur subtractive cancellation when λ_1 is close to λ_2. Consider the case where $f = \log$. Let $z = (\lambda_2-\lambda_1)/(\lambda_2+\lambda_1)$ and assume that a means for accurate evaluation of atanh is available. The following formula suggested by Higham [18, sec. 11.6.2] allows accurate evaluation when λ_1 and λ_2 are close but not equal:

$$f_{12} = t_{12}\frac{2\,\mathrm{atanh}(z) + 2\pi i\,\mathcal{U}(\log\lambda_2 - \log\lambda_1)}{\lambda_2-\lambda_1}.$$

This formula is used in `logm` in MATLAB. A similar formula is obtained by Higham and Lin [20, (5.6)] for $f(t) = t^p$, $p \in \mathbb{R}$ and used in [20] and [21].

3. The matrix unwinding function. We define the matrix unwinding function to be the matrix function corresponding to the unwinding number:

$$\mathcal{U}(A) = \frac{A - \log \mathrm{e}^A}{2\pi i}, \qquad A \in \mathbb{C}^{n\times n}. \tag{3.1}$$

To make this definition precise we need to clarify which matrix logarithm is being used. We cannot use the usual principal matrix logarithm, for which $\log(X)$ is defined only for X with no eigenvalues on $\mathbb{R}^-$ [18, Thm. 1.31]. Instead we take log to be the matrix function corresponding to the principal scalar logarithm defined at the start of Section 2. However, this is not sufficient to define $\log A$ for any nonsingular matrix. To see why, recall that for $A \in \mathbb{C}^{n\times n}$ with Jordan canonical form

$$Z^{-1}AZ = J = \mathrm{diag}(J_1, J_2, \ldots, J_p), \tag{3.2a}$$

$$J_k = J_k(\lambda_k) = \begin{bmatrix} \lambda_k & 1 & & \\ & \lambda_k & \ddots & \\ & & \ddots & 1 \\ & & & \lambda_k \end{bmatrix} \in \mathbb{C}^{m_k\times m_k}, \tag{3.2b}$$

$f(A)$ is defined as $f(A) = Zf(J)Z^{-1} = Z\,\mathrm{diag}(f(J_k))Z^{-1}$ [18, Def. 1.2], where

$$f(J_k) := \begin{bmatrix} f(\lambda_k) & f'(\lambda_k) & \ldots & \dfrac{f^{(m_k-1)}(\lambda_k)}{(m_k-1)!} \\ & f(\lambda_k) & \ddots & \vdots \\ & & \ddots & f'(\lambda_k) \\ & & & f(\lambda_k) \end{bmatrix}. \tag{3.3}$$

The principal logarithm log is discontinuous on $\mathbb{R}^-$ and so does not have any derivatives there. We will therefore define the first derivative for $z \in \mathbb{R}^-$ as the one-sided limit $\log'(z) = \lim_{h\to 0,\ \mathrm{Im}\, h\geq 0}[\log(z+h) - \log z]/h$, and so on for higher derivatives,

which are simply the usual derivatives evaluated on $\mathbb{R}^-$. Hence $\log A$ is now well defined.

Another way to define $\mathcal{U}(A)$ that is equivalent to (3.1) is by applying the Jordan form definition directly to the scalar unwinding number $\mathcal{U}(z)$, where the derivatives $\mathcal{U}'(z), \mathcal{U}''(z), \ldots$, are necessarily zero for $\operatorname{Im} z \neq (2j+1)\pi$, $j \in \mathbb{Z}$, and we define them to be zero for $\operatorname{Im} z = (2j+1)\pi$. That this definition is equivalent to (3.1) follows from the fact that the underlying scalar functions have the same values on the spectrum of A [18, Sec. 1.2.2]. It is immediate from (3.3) that $\mathcal{U}(J_k(\lambda_k)) = \mathcal{U}(\lambda_k)I$ for any Jordan block $J_k(\lambda_k)$. Hence, in terms of the Jordan form (3.2),

$$\mathcal{U}(A) = Z \operatorname{diag}(\mathcal{U}(\lambda_k) I_{m_k}) Z^{-1}, \tag{3.4}$$

so that $U(A)$ is diagonalizable and has integer eigenvalues. In particular, if all the eigenvalues of A have the same unwinding number u, then $\mathcal{U}(A) = uI$.

Note that $\mathcal{U}(z)$ is continuously differentiable as many times as we like in the open subset $\mathcal{D} = \{ z \in \mathbb{C} : \operatorname{Im} z \neq (2j+1)\pi \text{ for all } j \in \mathbb{Z} \}$ of $\mathbb{C}$. This implies, for example, that $\mathcal{U}$ is a continuous matrix function on the set of matrices $A \in \mathbb{C}^{n\times n}$ with spectrum in $\mathcal{D}$ [18, Thm. 1.19]. However, for most of our results we need just the following standard properties that hold for general matrix functions f [18, Chap. 1]:

(3.5a) $f(A)$ is a polynomial in A,

(3.5b) $A, B \in \mathbb{C}^{n\times n}$, $AB = BA \;\Rightarrow \begin{cases} f(A)f(B) = f(B)f(A), \\ f(A+B) \text{ commutes with } A \text{ and } B. \end{cases}$

3.1. Properties of the unwinding function. We now derive some properties of the matrix unwinding function, and in particular generalize some of the properties of the unwinding number given in Section 2.

THEOREM 3.1. *For $A \in \mathbb{C}^{n\times n}$, $\mathcal{U}(A) = 0$ if and only if the imaginary parts of all the eigenvalues of A lie in the interval $(-\pi, \pi]$.*

Proof. The result is immediate from (3.4) and Corollary 2.2. □

Note that $\mathcal{U}(A) = 0$ is equivalent to $\log \mathrm{e}^A = A$, and essentially the same conditions as in Theorem 3.1 for this equation to hold are proved in [18, Prob. 1.39] for the usual principal matrix logarithm without explicitly referring to the matrix unwinding function. The theorem implies that the spectral radius condition $\rho(A) < \pi$, or the stronger condition $\|A\| < \pi$ for some consistent matrix norm, are sufficient for $\log \mathrm{e}^A = A$ to hold. For several important classes of matrices the conditions of Theorem 3.1 are always satisfied: matrices with real eigenvalues (in particular, Hermitian matrices), and unitary, idempotent, or stochastic matrices (for all of which $|\lambda| \leq 1$ for every eigenvalue λ).

The next result, which gives a characterization of a class of matrix functions of which the matrix unwinding function is a special case, enables us to determine the behavior of $\mathcal{U}$ under conjugation and the form of $\mathcal{U}$ for real matrices. We denote by $\Lambda(A)$ the spectrum of A.

THEOREM 3.2. *Let f be analytic on an open subset $\Omega \subseteq \mathbb{C}$ such that for each connected component $\widetilde{\Omega}$ of Ω, $z \in \widetilde{\Omega}$ if and only if $-\overline{z} \in \widetilde{\Omega}$. Consider the corresponding matrix function f on its natural domain in $\mathbb{C}^{n\times n}$, the set $\mathcal{D} = \{ A \in \mathbb{C}^{n\times n} : \Lambda(A) \subseteq \Omega \}$. Then the following are equivalent:*

(a) $f(A^*) = -\overline{f(A)}^*$ *for all* $A \in \mathcal{D}$.

(b) $f(\overline{A}) = -\overline{f(A)}$ *for all* $A \in \mathcal{D}$.

(c) $f(\mathbb{R}^{n\times n}\cap\mathcal{D})\subseteq i\mathbb{R}^{n\times n}$.
(d) $f(\mathbb{R}\cap\Omega)\subseteq i\mathbb{R}$.

Proof. The proof is similar to that of [18, Thm. 1.18] and [22, Thm. 3.2] and so is omitted. ∎

Applying Theorem 3.2 to the matrix unwinding function we obtain the next result.

COROLLARY 3.3. *For $A\in\mathbb{C}^{n\times n}$ with no eigenvalues with imaginary parts of the form $(2k+1)\pi$, $k\in\mathbb{Z}$,*
(a) $\mathcal{U}(A^*)=-\mathcal{U}(A)^*$.
(b) $\mathcal{U}(\overline{A})=-\overline{\mathcal{U}(A)}$.
(c) $\mathcal{U}(A)$ *is pure imaginary if A is real.*

Proof. $\mathcal{U}$ is analytic on the open subset $\Omega=\mathbb{C}\setminus\{\,z:\operatorname{Im}z=(2k+1)\pi,\ k\in\mathbb{Z}\,\}$ of $\mathbb{C}$, which satisfies $z\in\widetilde{\Omega}$ if and only if $-\overline{z}\in\widetilde{\Omega}$ for each connected component $\widetilde{\Omega}$. Hence it suffices to show that any one of the statements in Theorem 3.2 holds. Indeed, for any $z\in\mathbb{R}$, $\mathcal{U}(z)=0\in i\mathbb{R}$, which is condition (d) of Theorem 3.2 ∎

We give two examples to illustrate the corollary. First,

$$A=\begin{bmatrix}4 & 16\\ -4 & 4\end{bmatrix},\quad \Lambda(A)=\{4\pm 8i\},\quad \mathcal{U}(A)=\begin{bmatrix}0 & -2i\\ 0.5i & 0\end{bmatrix},\quad \Lambda(\mathcal{U}(A))=\{\pm 1\}.$$

Second, a matrix due to Rutishauser, which is `gallery('toeppen',3)` in MATLAB (non-integers are shown here to three significant figures):

$$A=\begin{bmatrix}0 & 10 & 1\\ -10 & 0 & 10\\ 1 & -10 & 0\end{bmatrix},\quad \Lambda(A)=\{1,-0.500\pm 1.41i\},$$
$$\mathcal{U}(A)=i\begin{bmatrix}0.0354 & -1.42 & -0.0354\\ 1.42 & -0.0708 & -1.42\\ -0.0354 & 1.42 & 0.0354\end{bmatrix},\qquad \Lambda(\mathcal{U}(A))=\{-2,0,2\}.$$

In both cases, $\mathcal{U}(A)$ is pure imaginary and a further computation shows that $\mathcal{U}(A)^*=-\mathcal{U}(A^*)$.

We can give an explicit formula for the unwinding function of real, 2×2 matrices of the form that appear as diagonal blocks in the real Schur decomposition computed by LAPACK.

LEMMA 3.4. *For $A=\left[\begin{smallmatrix}a & b\\ c & a\end{smallmatrix}\right]\in\mathbb{R}^{2\times 2}$ with $bc<0$,*

$$\mathcal{U}(A)=\begin{cases}-i\dfrac{\mathcal{U}(i\mu)}{\mu}(A-aI), & \mu\neq(2k+1)\pi,\ k\in\mathbb{Z},\\ -\dfrac{i}{\mu}\big[(\mathcal{U}(i\mu)+\frac{1}{2})(A-aI)-\frac{1}{2}i\mu I\big], & \text{otherwise},\end{cases}\tag{3.6}$$

where $\mu=(-bc)^{1/2}$.

Proof. The eigenvalues of A are $\lambda=a+i\mu$ and $\overline{\lambda}$. Let $Z^{-1}AZ=\operatorname{diag}(\lambda,\overline{\lambda})=aI+i\mu K$, where $K=\left[\begin{smallmatrix}1 & 0\\ 0 & -1\end{smallmatrix}\right]$. Thus $A=aI+\mu W$, where $W=iZKZ^{-1}\in\mathbb{R}^{2\times 2}$. Hence, for $\mu\neq(2k+1)\pi$ with $k\in\mathbb{Z}$,

$$\begin{aligned}\mathcal{U}(A)&=Z\operatorname{diag}(\mathcal{U}(\lambda),\mathcal{U}(\overline{\lambda}))Z^{-1}\\ &=\mathcal{U}(\lambda)Z\operatorname{diag}(1,-1)Z^{-1}=\mathcal{U}(\lambda)ZKZ^{-1}\\ &=\frac{\mathcal{U}(\lambda)}{i}W=-i\mathcal{U}(\lambda)W\\ &=-i\mathcal{U}(\lambda)(A-aI)/\mu.\end{aligned}$$

If $\mu = (2k+1)\pi$ for some $k \in \mathbb{Z}$ then $\mathcal{U}(\overline{\lambda}) = -\mathcal{U}(\lambda) - 1$, by Lemma 2.3. Hence we need to add a correction term $-Z\,\mathrm{diag}(0,1)Z^{-1}$ to the formula above. This correction term can be written

$$-\frac{1}{2}Z(I-K)Z^{-1} = \frac{1}{2}(ZKZ^{-1} - I) = \frac{1}{2}\left(\frac{A-aI}{i\mu} - I\right) = -\frac{i}{2\mu}(A - aI - i\mu I). \qquad \square$$

The 2×2 example above illustrates the theorem.

LEMMA 3.5. *For $A \in \mathbb{C}^{n\times n}$, $\mathrm{e}^{2\pi i\,\mathcal{U}(A)} = I$.*

Proof. Straightforward from (3.4). $\square$

3.2. Norm and conditioning. We now obtain an upper bound for the norm of $\mathcal{U}(A)$ and a lower bound for its condition number. These will be useful in section 4 for understanding the behavior of an algorithm for computing $\mathcal{U}(A)$. The norm is any norm for which $\|\,\mathrm{diag}(d_i)\| = \max_i |d_i|$, $\rho(A)$ denotes the spectral radius, and $\kappa(A) = \|A\|\|A^{-1}\|$ is the condition number with respect to inversion.

LEMMA 3.6. *For $A \in \mathbb{C}^{n\times n}$ with Jordan canonical form $A = ZJZ^{-1}$,*

$$\|\,\mathcal{U}(A)\| \le \frac{\kappa(Z)(\rho(A)+\pi)}{2\pi}.$$

Proof. Using (3.4) we have $\|\,\mathcal{U}(A)\| \le \kappa(Z)\max_k |U(\lambda_k)|$. But

$$\max_k |U(\lambda_k)| = \max_k \left| \left\lceil \frac{\mathrm{Im}\,\lambda_k - \pi}{2\pi} \right\rceil \right| \le \left\lceil \frac{\rho(A)-\pi}{2\pi} \right\rceil \le \frac{\rho(A)+\pi}{2\pi}. \qquad \square$$

The (relative) condition number of the matrix unwinding function is defined by

$$\mathrm{cond}_{\mathcal{U}}(A) = \lim_{\epsilon\to 0} \sup_{\|E\|\le\epsilon\|A\|} \frac{\|\,\mathcal{U}(A+E) - \mathcal{U}(A)\|}{\epsilon\|\,\mathcal{U}(A)\|}.$$

We will obtain a lower bound for $\mathrm{cond}_{\mathcal{U}}(A)$. We can assume that $\rho(A) \ge \pi$, since $\rho(A) < \pi$ implies $\mathcal{U}(A) = 0$ by Theorem 3.1 and hence $\mathrm{cond}_{\mathcal{U}}(A) = 0$; then Lemma 3.6 gives $\|\,\mathcal{U}(A)\| \le \kappa(Z)2\rho(A)/(2\pi) \le \kappa(Z)\|A\|/\pi$. Hence, using [18, Thm. 3.14] we have

$$\begin{aligned} \mathrm{cond}_{\mathcal{U}}(A) &\ge \frac{\|A\|}{\|\,\mathcal{U}(A)\|} \max_{\lambda,\mu\in\Lambda(A)} \mathcal{U}[\lambda,\mu] \\ &\ge \frac{\pi}{\kappa(Z)} \max_{\lambda,\mu\in\Lambda(A)} \mathcal{U}[\lambda,\mu] \qquad (\rho(A)\ge\pi), \end{aligned} \tag{3.7}$$

where

$$\mathcal{U}[\lambda,\mu] = \begin{cases} \dfrac{\mathcal{U}(\lambda)-\mathcal{U}(\mu)}{\lambda-\mu}, & \lambda \ne \mu, \\ \mathcal{U}'(\lambda) = 0, & \lambda = \mu \end{cases}$$

is a divided difference and $A = ZJZ^{-1}$ is a Jordan canonical form. When $\mathrm{Im}\,\lambda$ and $\mathrm{Im}\,\mu$ lie close to but on opposite sides of $(2k+1)\pi$, for some k, then $\mathcal{U}[\lambda,\mu] = (\lambda-\mu)^{-1}$ and hence $\mathcal{U}[\lambda,\mu]$ is necessarily large if $\mathrm{Re}\,\lambda \approx \mathrm{Re}\,\mu$; in this case the lower bound for $\mathrm{cond}_{\mathcal{U}}(A)$ is large unless $\kappa(Z)$ is large,

We now turn to estimation of the condition number. By standard results [18, Sec. 3.1],

$$\operatorname{cond}_{\mathcal{U}}(A) = \frac{\|L_{\mathcal{U}}(A)\|\,\|A\|}{\|\mathcal{U}(A)\|},$$

where

$$\|L_{\mathcal{U}}(A)\| := \max_{Z\neq 0} \frac{\|L_{\mathcal{U}}(A,Z)\|}{\|Z\|}$$

and $L_{\mathcal{U}}(A,\cdot)$ is the Fréchet derivative of $\mathcal{U}$ at A. Moreover, since $L_{\mathcal{U}}$ is a linear operator,

$$L_{\mathcal{U}}(A,E)) = K_{\mathcal{U}}(A)\operatorname{vec}(E), \tag{3.8}$$

where $K_{\mathcal{U}}(A) \in \mathbb{C}^{n^2\times n^2}$ is the Kronecker form of the Fréchet derivative and vec is the operator that stacks the columns of a matrix on top of each other [18, Chap. 3]. Following [18, Alg. 3.22] we will approximate $\|L_{\mathcal{U}}(A)\|_1$ by $\|K_{\mathcal{U}}(A)\|_1$ and estimate the latter quantity using the block 1-norm estimation algorithm of Higham and Tisseur [23]. This algorithm requires the ability to evaluate matrix–vector products involving $K_{\mathcal{U}}(A)$ and $K_{\mathcal{U}}(A)^*$. A product $K_{\mathcal{U}}(A)y$ can be evaluated as the left-hand side of (3.8) with $\operatorname{vec}(E) = y$. This can be done using the formula

$$L_{\mathcal{U}}(A,E) = \frac{E - L_{\log}(\mathrm{e}^A, L_{\exp}(A,E))}{2\pi i}$$

obtained by applying the chain rule [18, Thm. 3.4] to (3.1), or by evaluating the unwinding function of a $2n\times 2n$ matrix [18, (3.13)] then extracting the upper right $n\times n$ block:

$$\mathcal{U}\left(\begin{bmatrix} A & E \\ 0 & A \end{bmatrix}\right) = \begin{bmatrix} \mathcal{U}(A) & L_{\mathcal{U}}(A,E) \\ 0 & \mathcal{U}(A) \end{bmatrix}. \tag{3.9}$$

Howe to evaluate a product $K_{\mathcal{U}}(A)^*y$ is not immediately obvious. We need to introduce the adjoint $L_f^\star$ of the Fréchet derivative L_f, which is defined by the condition

$$\langle L_f(A,G), H\rangle = \langle G, L_f^\star(A,H)\rangle \tag{3.10}$$

for all $G, H \in \mathbb{C}^{n\times n}$, where $\langle X, Y\rangle = \operatorname{trace}(Y^*X) = \operatorname{vec}(Y)^*\operatorname{vec}(X)$.

LEMMA 3.7. *Let f be $2n-1$ times continuously differentiable on an open subset $\mathcal{D}$ of $\mathbb{R}$ or $\mathbb{C}$ such that for each connected component $\tilde{\mathcal{D}}$ of $\mathcal{D}$, $z\in\tilde{\mathcal{D}}$ if and only if $-\bar{z}\in\tilde{\mathcal{D}}$. Suppose that $\bar{f}(A)^* = -\bar{f}(A^*)$ for all $A\in\mathbb{C}^{n\times n}$ with spectrum in $\mathcal{D}$, where $\bar{f}(z) := \overline{f(\bar{z})}$. Then*

$$L_f^\star(A,E) = L_{\overline{f}}(A^*,E) = -L_{\overline{f}}(A,E^*)^*. \tag{3.11}$$

Proof. The proof of the first equality is exactly the same as that of the corresponding equality in the analogous result [21, Lem. 6.2].

To prove the second equality we consider $g = \bar{f}$. By the definition of the Frèchet derivative, $L_g(A,E) = g(A+E) - g(A) + o(\|E\|)$. Taking the conjugate transpose, $L_g(A,E)^* = g(A+E)^* - g(A)^* + o(\|E\|) = -g(A^*+E^*) + g(A^*) + o(\|E\|) =$

$-L_g(A^*, E^*) + o(\|E\|)$. By the linearity of the Frèchet derivative we then have $L_g(A, E)^* = -L_g(A^*, E^*)$, which gives the second equality in (3.11). □

It is shown by Higham and Lin [21, Lem. 6.1] that $K_f(A)^* \operatorname{vec}(E) = \operatorname{vec}(L_f^\star(A, E))$ for any f. Combined with (3.11) this yields $K_f(A)^* \operatorname{vec}(E) = -\operatorname{vec}(L_{\overline{f}}(A, E^*)^*)$. For the unwinding function we have $\overline{f} = -f$ by Lemma 2.3, so $L_{\overline{f}} = -L_f$ and $K_{\mathcal{U}}(A)^* \operatorname{vec}(E) = \operatorname{vec}(L_{\mathcal{U}}(A, E^*)^*)$, and hence products with $K_{\mathcal{U}}(A)^*$ can be computed in exactly the same way as products with $K_{\mathcal{U}}(A)$.

3.3. Identities involving the logarithm and powers. We now use the matrix unwinding function to derive mathematical identities involving the matrix logarithm and fractional matrix powers.

For any nonsingular $A \in \mathbb{C}^{n\times n}$ and any $\alpha \in \mathbb{C}$ we define the principal matrix power

$$A^\alpha = \mathrm{e}^{\alpha \log A}, \tag{3.12}$$

where we recall that log denotes the principal matrix logarithm defined at the start of section 3. The following result is immediate from the definitions of $\mathcal{U}(A)$ and A^α.

LEMMA 3.8. *For nonsingular $A \in \mathbb{C}^{n\times n}$ and $\alpha \in \mathbb{C}$,*

$$\log A^\alpha = \alpha \log A - 2\pi i \mathcal{U}(\alpha \log A).$$

To establish when $\log A^\alpha = \alpha \log A$, we need to determine when $\mathcal{U}(\alpha \log A) = 0$. The following result describes a particularly useful context in which the latter condition holds.

COROLLARY 3.9. *For nonsingular $A \in \mathbb{C}^{n\times n}$, $\log A^\alpha = \alpha \log A$ for $\alpha \in (-1, 1]$ and for $\alpha = -1$ if A has no eigenvalues on $\mathbb{R}^-$.*

Proof. It is immediate from (3.4) and Lemma 2.4 that $\mathcal{U}(\alpha \log A) = 0$ under the given conditions, so the result follows by Lemma 3.8. □

Note the special cases $\alpha = -1$ and $\alpha = 1/2$. We have $\log A^{-1} = -\log A$ if A has no eigenvalues on $\mathbb{R}^-$ and $\log A^{1/2} = \frac{1}{2}\log A$ for all A. The latter identity can be used to write $2^k \log A^{1/2^k} = \log A$, for any $k \in \mathbb{Z}$, which underlies the inverse scaling and squaring algorithm for computing the matrix logarithm [2], [3], [10].

We next describe the result of powering successively by α and $1/\alpha$.

LEMMA 3.10. *For nonsingular A and $\alpha \in \mathbb{C}$,*

$$(A^\alpha)^{1/\alpha} = A\mathrm{e}^{-\frac{2}{\alpha}\pi i \mathcal{U}(\alpha \log A)}.$$

Proof. Using (3.12) and Lemma 3.8 we have

$$\begin{aligned}(A^\alpha)^{1/\alpha} &= \mathrm{e}^{\frac{1}{\alpha}\log A^\alpha} = \mathrm{e}^{\frac{1}{\alpha}(\alpha \log A - 2\pi i \mathcal{U}(\alpha \log A))} \\ &= A\mathrm{e}^{-\frac{2}{\alpha}\pi i U(\alpha \log A)}. \qquad \square\end{aligned}$$

We note the special case of Lemma 3.10 with $\alpha = 2$, which will be needed in the next subsection:

$$(A^2)^{1/2} = A\mathrm{e}^{-\pi i \mathcal{U}(2 \log A)}. \tag{3.13}$$

We proceed to study the relation between a logarithm of a matrix product and the logarithms of the matrices involved.

LEMMA 3.11. *Let $A, B \in \mathbb{C}^{n\times n}$ be nonsingular matrices such that $AB = BA$. Then*

$$\log(AB) = \log A + \log B - 2\pi i\,\mathcal{U}(\log A + \log B).$$

Proof. Recall that $\mathrm{e}^{(A+B)t} = \mathrm{e}^{At}\mathrm{e}^{Bt}$ for all t if and only if A and B commute, [18, Thm. 10.2]. Since A and B commute, so do $\log A$ and $\log B$. We therefore have

$$\begin{aligned}\log(AB) &= \log(\mathrm{e}^{\log A}\mathrm{e}^{\log B}) = \log(\mathrm{e}^{\log A+\log B})\\ &= \log A + \log B - 2\pi i\,\mathcal{U}(\log A + \log B),\end{aligned}$$

where we have used the definition (3.1) of the matrix unwinding function. ∎

Recall from [18, Cor. 1.41] that if A and B commute then for each eigenvalue μ_j of A there is an eigenvalue ν_j of B such that $\mu_j + \nu_j$ is an eigenvalue of $A + B$. We will call ν_j the eigenvalue corresponding to μ_j.

COROLLARY 3.12. *Let $A, B \in \mathbb{C}^{n\times n}$ be nonsingular matrices such that $AB = BA$. Then*

$$\log(AB) = \log A + \log B$$

if and only if $\arg\mu_j + \arg\nu_j \in (-\pi, \pi]$ *for every eigenvalue μ_j of A and the corresponding eigenvalue ν_j of B.*

Proof. Using Lemma 3.11, $\log(AB) = \log A + \log B$ if and only if $\mathcal{U}(\log A + \log B) = 0$. From Theorem 3.1 the latter equality holds if and only if the imaginary parts of the eigenvalues of $\log A + \log B$ lie in the interval $(-\pi, \pi]$, which is equivalent to $\arg\mu_j + \arg\nu_j \in (-\pi, \pi]$ for all j. ∎

Corollary 3.12 was proved by Higham [18, Thm. 11.3] directly from the definition of principal logarithm, and a variant of the result was obtained by Cheng, Higham, Kenney, and Laub [10, Lem. 2.1]; in both cases the additional assumption that A and B have no real negative eigenvalues was in force. The benefit of the matrix unwinding function is that it provides the correction term for the general case in Lemma 3.11.

The next result gives a relation between the power of a matrix product and the product of the powers.

THEOREM 3.13. *Let $A, B \in \mathbb{C}^{n\times n}$ be nonsingular matrices such that $AB = BA$. Then, for any $\alpha \in \mathbb{C}$,*

$$(AB)^\alpha = A^\alpha B^\alpha \mathrm{e}^{-2\pi\alpha i\,\mathcal{U}(\log A+\log B)}.$$

Proof. Applying (3.12), Lemma 3.11, and (3.5b) we have

$$\begin{aligned}(AB)^\alpha &= \mathrm{e}^{\alpha\log(AB)}\\ &= \mathrm{e}^{\alpha(\log A+\log B-2\pi i\,\mathcal{U}(\log A+\log B))}\\ &= A^\alpha B^\alpha \mathrm{e}^{-2\alpha\pi i\,\mathcal{U}(\log A+\log B)}.\end{aligned}$$ ∎

The next corollary characterizes when $(AB)^\alpha = A^\alpha B^\alpha$ holds in terms of the eigenvalues of A and B rather than $\log A$ and $\log B$.

COROLLARY 3.14. *Let $A, B \in \mathbb{C}^{n\times n}$ be nonsingular matrices such that $AB = BA$. Then $(AB)^\alpha = A^\alpha B^\alpha$ if and only if $\alpha\,\mathcal{U}(\log\mu_j + \log\nu_j) \in \mathbb{Z}$ for every eigenvalue μ_j of A and the corresponding eigenvalue ν_j of B.*

Proof. It follows from [18, Thm. 1.27] that all solutions of $\mathrm{e}^X = I$ are of the form $X = V\,\mathrm{diag}(2\pi i k_1, \dots, 2\pi i k_n)V^{-1}$, where V is an arbitrary nonsingular matrix and $k_j \in \mathbb{Z}$ for all j. Hence, given that $\mathcal{U}(\log A + \log B)$ is diagonalizable, $\exp(-2\alpha\pi i\,\mathcal{U}(\log A + \log B)) = I$ if and only if the eigenvalues of $2\alpha\pi i\,\mathcal{U}(\log A + \log B)$ are of the form $2\pi i k_j$, $k_j \in \mathbb{Z}$, which yields the result. ∎

Note that $\alpha\,\mathcal{U}(\log\mu_j + \log\nu_j) \in \mathbb{Z}$ holds when either $\alpha \in \mathbb{Z}$ or $\mathcal{U}(\log\mu_j + \log\nu_j) = 0$, since $\mathcal{U}(\log\mu_j + \log\nu_j) \in \{-1, 0, 1\}$.

An important case in which the condition of Corollary 3.14 holds for all α is when the eigenvalues of A and B have arguments in $(-\pi/2, \pi/2]$, for then $\mathrm{Im}(\log\mu_j + \log\nu_j) \in (-\pi, \pi]$ and so $\mathcal{U}(\log\mu_j + \log\nu_j) = 0$. As a special case we recover the result of [18, Prob. 1.35], which states that $(AB)^{1/2} = A^{1/2}B^{1/2}$ when A and B commute and both have eigenvalues lying in the open right half-plane.

The following result clarifies the relation between $(\mathrm{e}^A)^\alpha$ and $\mathrm{e}^{\alpha A}$, for $\alpha \in \mathbb{C}$.

THEOREM 3.15. *For $A \in \mathbb{C}^{n\times n}$ and $\alpha \in \mathbb{C}$, $(\mathrm{e}^A)^\alpha = \mathrm{e}^{\alpha A}\mathrm{e}^{-2\pi i\alpha\,\mathcal{U}(A)}$. Hence $(\mathrm{e}^A)^\alpha = \mathrm{e}^{\alpha A}$ if and only if $\alpha\,\mathcal{U}(\lambda) \in \mathbb{Z}$ for every eigenvalue λ of A.*

Proof. From the definitions (3.12) of matrix power and (3.1) of matrix unwinding function we have

$$(\mathrm{e}^A)^\alpha = \mathrm{e}^{\alpha\log\mathrm{e}^A} = \mathrm{e}^{\alpha(A - 2\pi i\,\mathcal{U}(A))} = \mathrm{e}^{\alpha A}\mathrm{e}^{-2\pi i\alpha\,\mathcal{U}(A)}.$$

The last part follows as in the proof of Corollary 3.14. ∎

When α is an integer, the correction term in Theorem 3.15 is the identity matrix, and after rescaling $A \leftarrow \alpha^{-1}A$ and setting $\alpha = 2^s$ we obtain the basis of the scaling and squaring method for computing the matrix exponential: $(\mathrm{e}^{A/2^s})^{2^s} = A$.

Note that Theorem 3.15 shows that it is not the case that $\mathrm{e}^A = (\mathrm{e}^{A/\alpha})^\alpha$ holds for all $\alpha \in \mathbb{C}$, as is incorrectly stated in [18, p. 241]!

3.4. Relation with the matrix sign function. We now explore some interesting connections between the matrix unwinding function and the matrix sign function. The scalar variants of some of these are given in [11, Table A.1].

Recall that the matrix sign function is defined only for $A \in \mathbb{C}^{n\times n}$ with no purely imaginary eigenvalues and is given by $\mathrm{sign}(A) = A(A^2)^{-1/2}$, as well as by various other equivalent formulas [18, Chap. 5], [27].

Taking the inverse of equation (3.13) we can write

$$\mathrm{sign}(A) = A(A^2)^{-1/2} = AA^{-1}\mathrm{e}^{\pi i\,\mathcal{U}(2\log A)} = \mathrm{e}^{\pi i\,\mathcal{U}(2\log A)}. \tag{3.14}$$

If the eigenvalues of A lie in the open right half-plane then the eigenvalues of $\log A$ have imaginary parts in the interval $(-\pi/2, \pi/2)$, hence $\mathcal{U}(2\log A) = 0$ and $\mathrm{sign}(A) = I$. Conversely, if the eigenvalues of A lie in the open left half-plane then the imaginary part of every eigenvalue λ of $\log A$ lies in $(-\pi, -\pi/2)$ or $(\pi/2, \pi]$ and hence $\mathcal{U}(2\lambda) = -1$ or 1, respectively, yielding $\mathrm{sign}(A) = -I$.

We note that the right-hand side of our formula (3.14) is defined for any nonsingular A. The formula gives a meaning to the sign function on the imaginary axis: for $y > 0$, $\mathrm{sign}(iy) = 1$ and $\mathrm{sign}(-iy) = -1$. Indeed, this conforms with the counter-clockwise continuity principle introduced by Kahan [26]. We will call

$\operatorname{sign}(A) := \mathrm{e}^{\pi i \mathcal{U}(2\log A)}$ the extended matrix sign function and we note that it is different to other extensions in the literature of the sign function to arbitrary nonsingular matrices [27, sec. I.D].

This result can be generalized for the matrix sector function [18, Sec. 2.14.3], [36], which for a given integer p and $A \in \mathbb{C}^{n\times n}$ with no eigenvalues with argument $(2k+1)\pi/p$, $k = 0 : p-1$, is defined as $\operatorname{sect}_p(A) = A(A^p)^{-1/p}$. From Lemma 3.10 we have

$$\operatorname{sect}_p(A) = \mathrm{e}^{\frac{2}{p}\pi i \mathcal{U}(p\log A)}.$$

Analogously to the relation $\operatorname{sign}(A) = A(A^2)^{-1/2}$, we have the following result involving the extended matrix sign function.

LEMMA 3.16. *For a nonsingular $A \in \mathbb{C}^{n\times n}$ with no eigenvalues with imaginary parts of the form $(2k+1)\pi$, for $k \in \mathbb{Z}$,*

$$\mathcal{U}(A) = \operatorname{sign}(A)\mathcal{U}\big((A^2)^{1/2}\big). \tag{3.15}$$

Proof. It suffices to prove the result for diagonalizable A, by [18, Thm. 1.20] (or simply because $\mathcal{U}(\cdot)$ and $\operatorname{sign}(\cdot)$ are diagonalizable), so the result reduces to the scalar case, $\mathcal{U}(z) = \operatorname{sign}(z)\mathcal{U}((z^2)^{1/2})$. If we now suppose z lies in the open left half-plane, or on $i\mathbb{R}^-$, $\operatorname{sign}(z) = -1$ and $(z^2)^{1/2} = -z$. Since for any $z \in \mathbb{C}$ such that $\operatorname{Im} z \neq (2k+1)\pi$ for all $k \in \mathbb{Z}$, $\mathcal{U}(-z) = -\mathcal{U}(z)$ by Lemma 2.3, the desired result follows. A similar argument applies for z in the open right half-plane or on $i\mathbb{R}^+$. ∎

We can use (3.15) to derive a formula for the matrix unwinding function of a particular block matrix.

THEOREM 3.17. *For nonsingular $A, B \in \mathbb{C}^{n\times n}$ such that $(AB)^{1/2}$ has no eigenvalues with imaginary parts of the form $(2k+1)\pi$, for $k \in \mathbb{Z}$,*

$$\mathcal{U}\left(\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A(BA)^{-1/2}\mathcal{U}((BA)^{1/2}) \\ B(AB)^{-1/2}\mathcal{U}((AB)^{1/2}) & 0 \end{bmatrix}.$$

Proof. The result is obtained by applying (3.15) to the matrix $C = \left[\begin{smallmatrix} 0 & A \\ B & 0 \end{smallmatrix}\right]$ then using

$$\mathcal{U}\big((C^2)^{1/2}\big) = \mathcal{U}\big(\operatorname{diag}((AB)^{1/2}, (BA)^{1/2}\big) = \operatorname{diag}\big(\mathcal{U}((AB)^{1/2}), \mathcal{U}((BA)^{1/2})\big)$$

and the following result of Higham, Mackey, Mackey, and Tisseur [22, Lem. 4.3], [18, Thm. 5.2] (which holds even when AB has eigenvalues on $\mathbb{R}^-$, given that we are using the extended matrix sign function):

$$\operatorname{sign}\left(\begin{bmatrix} 0 & A \\ B & 0 \end{bmatrix}\right) = \begin{bmatrix} 0 & A(BA)^{-1/2} \\ B(AB)^{-1/2} & 0 \end{bmatrix}. \qquad ∎$$

4. Algorithm. The matrix unwinding function can be computed directly via the definition (3.1), but this requires a matrix exponential and a matrix logarithm. Instead we will compute a Schur decomposition $A = QTQ^* \in \mathbb{C}^{n\times n}$, where Q is unitary and T is upper triangular, after which $\mathcal{U}(A) = Q\mathcal{U}(T)Q^*$. The problem is reduced to computing $\mathcal{U}(T)$, which we will do directly rather than via the exponential and logarithm.

The Parlett recurrence for computing a function of a triangular matrix T is not appropriate because it breaks down when T has repeated diagonal elements. The Schur–Parlett method [13], which is implemented in MATLAB function `funm`, reorders and blocks T so that the eigenvalues within a diagonal block are close while distinct diagonal blocks have well separated eigenvalues. We do not have a special way of evaluating the unwinding function of a triangular matrix with close eigenvalues, so we cannot use this general method.

Instead we adapt the Schur–Parlett method by putting eigenvalues having the same unwinding number in the same block. We use the algorithm of Bai and Demmel [6] (implemented in MATLAB function `ordschur`) to compute a unitary V such that $\widetilde{T} = V^*TV = (\widetilde{T}_{ij})$ is upper triangular with all the diagonal elements of $\widetilde{T}_{ii}$ having imaginary parts in the same interval $((2k_i-1)\pi, (2k_i+1)\pi]$, for some $k_i \in \mathbb{Z}$. The diagonal blocks of $F = \mathcal{U}(\widetilde{T})$ are therefore given by $F_{ii} = u_i I$ for all i, by (3.4). The off-diagonal blocks are obtained from the block Parlett recurrence, which is obtained by equating blocks in $F\widetilde{T} = \widetilde{T}F$:

$$\widetilde{T}_{ii}F_{ij} - F_{ij}\widetilde{T}_{jj} = (u_i - u_j)\widetilde{T}_{ij} + \sum_{k=i+1}^{j-1} (F_{ik}\widetilde{T}_{kj} - \widetilde{T}_{ik}F_{kj}), \qquad i < j. \tag{4.1}$$

These Sylvester equations are nonsingular, since $\widetilde{T}_{ii}$ and $\widetilde{T}_{jj}$ have no eigenvalue in common, and they can be solved a block column or a block superdiagonal at a time.

For notational simplicity, we express our algorithm at the scalar level, but it is mathematically equivalent to carrying out the blocking described above and using the block Parlett recurrence; blocking is preferred in practice as it allows the use of higher level BLAS. The following algorithm is similar to an algorithm of Ng [33, Sec. 7.4] for matrix argument reduction (see Section 6).

ALGORITHM 4.1. *Given $A \in \mathbb{C}^{n\times n}$ this algorithm computes the unwinding function $U = \mathcal{U}(A)$ using the Schur–Parlett method with a particular reordering and blocking.*

1 Compute the Schur decomposition $A = QTQ^*$ (Q unitary, T upper triangular).
2 If $\operatorname{Im} t_{ii} \in (-\pi, \pi]$ for all i, $U = 0$, quit, end
3 Assign t_{ii} to set $S_{\mathcal{U}(t_{ii})}$, $i = 1\colon n$, and use a unitary similarity transformation to reorder T so that all elements belonging to each set $S_{\mathcal{U}(t_{ii})}$ are contiguous, and update Q.
4 $f_{ii} = \mathcal{U}(t_{ii})$, $i = 1\colon n$
5 for $j = 2\colon n$
6 for $i = j-1\colon -1\colon 1$
7 if $f_{ii} = f_{jj}$
8 $f_{ij} = 0$
9 else
10 $f_{ij} = \left(t_{ij}(f_{ii} - f_{jj}) + \sum_{k=i+1}^{j-1} \left(f_{ik}t_{kj} - t_{ik}f_{kj} \right) \right) / \left(t_{ii} - t_{jj} \right)$
11 end
12 end
13 end
14 $U = QFQ^*$

Cost: $25n^3$ flops for the Schur decomposition plus the cost of the reordering, $n^3/3$ flops for F, and $3n^3$ flops to form U.

The cost of the reordering can be shown to be at most $10n^3 - 20n^2$ flops, but will usually be much less, because this bound assumes a worst case distribution of

diagonal entries of the Schur factor and block sizes.

Note that an alternative in Algorithm 4.1 is to reorder T so that $\operatorname{Im} t_{11} \leq \cdots \leq \operatorname{Im} t_{nn}$. However, this requires more swaps in general, so it is more expensive and introduces more rounding errors.

The condition number of the Sylvester equations (4.1) is proportional to the reciprocal of the separation of $\widetilde{T}_{ii}$ and $\widetilde{T}_{jj}$ [16, sec. 16.3], which is bounded below by the reciprocal of $\min\{\, |\lambda-\mu| : \lambda \in \Lambda(\widetilde{T}_{ii}),\ \mu \in \Lambda(\widetilde{T}_{jj}) \,\}$. Hence (4.1) can be ill conditioned, as there is no lower bound on the absolute value of the differences between eigenvalues of $\widetilde{T}_{ii}$ and $\widetilde{T}_{jj}$. A small eigenvalue difference occurs precisely when two consecutive blocks have eigenvalues λ_i and λ_j such that $\operatorname{Re}(\lambda_i-\lambda_j) < \epsilon$, and $\operatorname{Im}\lambda_i = (2k+1)\pi-\delta_1$, $\operatorname{Im}\lambda_j = (2k+1)\pi+\delta_2$, for some $k \in \mathbb{Z}$ and some small $\epsilon, \delta_1, \delta_2 \in \mathbb{R}$, which is equivalent to $\mathcal{U}[\lambda_i, \lambda_j]$ being large.

Recall from (3.7) that the condition number $\operatorname{cond}_{\mathcal{U}}$ of the matrix unwinding function satisfies $\operatorname{cond}_{\mathcal{U}}(A) \geq \pi \max_{\lambda,\mu\in\Lambda(A)} \mathcal{U}[\lambda,\mu]/\kappa(Z)$ for $\rho(A) \geq \pi$, and so $\operatorname{cond}_{\mathcal{U}}$ is necessarily large when $\max_{\lambda,\mu\in\Lambda(A)} \mathcal{U}[\lambda,\mu]$ is large if the matrix is close to normal. We can conclude that ill conditioning of the Sylvester equation may correspond to ill conditioning of the matrix unwinding function itself, which provides some indication that Algorithm 4.1 will perform in a numerically stable fashion in practice.

When A is real, $\mathcal{U}(A)$ is pure imaginary if A has no eigenvalues with imaginary parts an odd integer multiple of π, by Corollary 3.3. In this case we would like our algorithm to guarantee a pure imaginary result. We can compute a real Schur decomposition $A = QTQ^T$, where T is real and upper quasitriangular. The matrix unwinding function of any 2×2 diagonal blocks can be computed using (3.6). However, the block Parlett recurrence may break down due to two different diagonal blocks having the same eigenvalues, so this approach is not reliable and we will not consider it further. This inability to split complex conjugate eigenvalues affects the standard Schur–Parlett algorithm in the same way, preventing the derivation of a version tailored to real matrices [13], [18, sec. 9.4].

5. Numerical experiments. We investigate experimentally two algorithms for computing $\mathcal{U}(A)$:

- Algorithm 4.1.
- log-exp: evaluation of (3.1) using the scaling and squaring algorithm of Al-Mohy and Higham [1] for the exponential and the inverse scaling and squaring algorithm of Al-Mohy, Higham, and Relton [3] for the logarithm; the matrix A is reduced to Schur form (the real Schur form if A is real) before applying these algorithms.

All tests were done in MATLAB R2013a, for which the unit roundoff $u \approx 1.1 \times 10^{-16}$.

While log-exp is useful as a means for comparison, we note that it has two flaws that make it unsuitable as a general way to compute $\mathcal{U}(A)$. First, it is prone to overflow, since e^A can overflow when $\mathcal{U}(A)$ does not, as is clear from the scalar case; indeed, $\mathcal{U}(A)$ is very unlikely to overflow, in view of Lemma 3.6. Second, e^A can be singular, making the log computation fail. For example, for

$$A = \begin{bmatrix} 1 & 1 \\ 0 & -10^3 \end{bmatrix}, \quad fl(\mathrm{e}^A) = \begin{bmatrix} \mathrm{e} & fl(\mathrm{e}/(10^3+1)) \\ 0 & 0 \end{bmatrix},$$

since $fl(\mathrm{e}^{-10^3}) = 0$. Evaluation with log-exp fails since $fl(\mathrm{e}^A)$ is singular, yet $\mathcal{U}(A) = 0$, as is correctly computed by Algorithm 4.1.

To test the performance, we first constructed a set of 10×10 random matrices, Set 1, with eigenvalues λ_i that are odd integer multiples of πi perturbed by 10^{-6} times

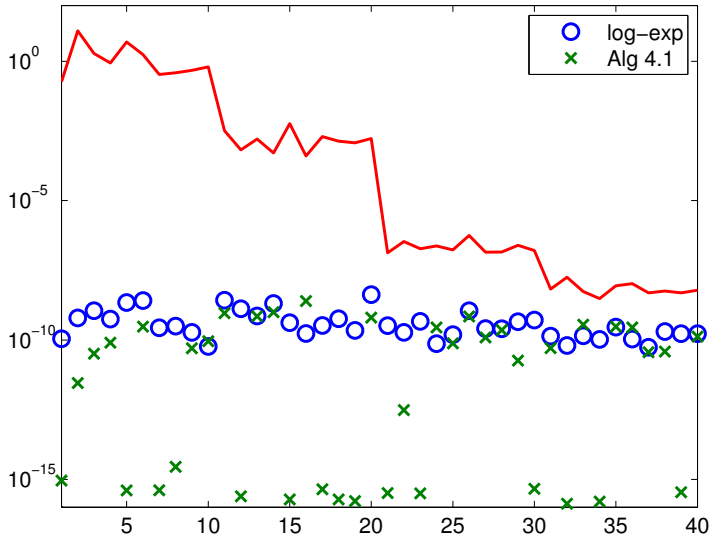

FIG. 5.1. *Relative errors for Set* 1. *The red line is an estimate of* $\mathrm{cond}_{\mathcal{U}}(A)u$.

complex numbers with $N(0,1)$ distributed real and imaginary parts. The matrices, 40 in total, are the upper triangular Schur factors of $A = XDX^{-1}$, where $D = \mathrm{diag}(\lambda_i)$ and X is random with 2-norm condition number 2, 10, 100, or 1000. Figure 5.1 shows the relative error $\|\mathcal{U}(A) - \widehat{U}\|_F / \|\mathcal{U}(A)\|_F$, where $\widehat{U}$ is the computed unwinding function and $\mathcal{U}(A)$ is the correctly rounded one; $\mathcal{U}(A)$ is obtained by evaluating (3.4) at 100 digit precision using the Symbolic Math Toolbox then rounding to double precision. The matrices are arranged according to decreasing value of $\kappa_2(X)$, with 10 matrices for each value. An estimate of $\mathrm{cond}_{\mathcal{U}}(A)u$ is shown in the figure, where $\mathrm{cond}_{\mathcal{U}}(A)$ is estimated as indicated in section 3.2, using Algorithm 4.1 with (3.9) to obtain the Fréchet derivatives. We see that both algorithms produce errors smaller than $\mathrm{cond}_{\mathcal{U}}(A)$ in every case, showing that they are performing in a forward stable fashion. Algorithm 4.1 produces errors substantially smaller than log-exp in many cases.

Our second test uses a set of 24 matrices, Set 2, drawn from the `gallery` function of MATLAB, the Matrix Computation Toolbox [15] and test problems provided with EigTool [37]. These matrices have been scaled to make them have nonzero unwinding functions or otherwise make them useful for test purposes. Figure 5.2 shows the relative errors for both algorithms, with the matrices arranged by decreasing estimated condition number $\mathrm{cond}_{\mathcal{U}}(A)$. Algorithm 4.1 performs in a forward stable way in every case, but log-exp is unstable on about half the matrices.

The conclusion from these tests is that Algorithm 4.1 is more accurate than log-exp and performs in a forward stable manner in the test sets.

6. Matrix argument reduction. A notion of matrix argument reduction was introduced by Ng in his PhD thesis [33] and pursued with a particular choice of "modulus" $(2\pi i)$ by McCurdy, Ng, and Parlett [29, Sec. 5.3]. In the latter paper, a matrix function mod is defined that is related to the matrix unwinding function by

$$\mathrm{mod}(A) = A - 2\pi i \mathcal{U}(A). \tag{6.1}$$

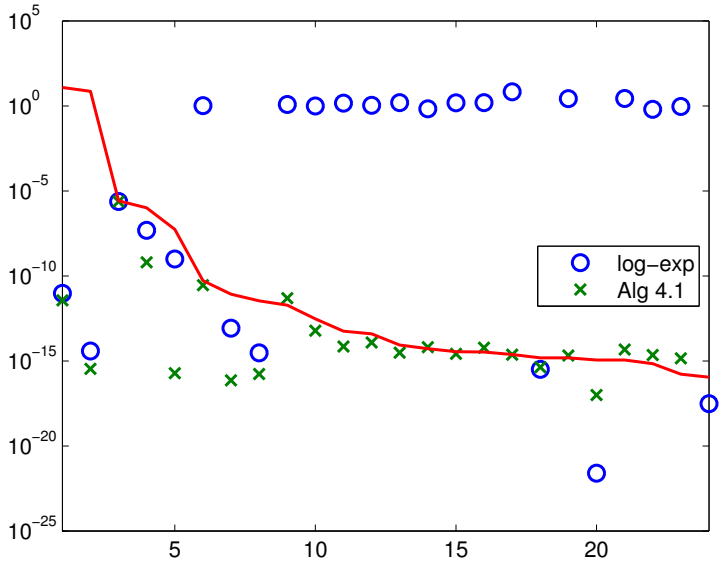

FIG. 5.2. *Relative errors for Set* 2. *The red line is an estimate of* $\mathrm{cond}_{\mathcal{U}}(A)u$.

The motivation for the use of mod was that for $A \in \mathbb{C}^{n\times n}$, $\mathrm{e}^A = \mathrm{e}^{\mathrm{mod}(A)}$ (by Lemma 3.5) and, because $\mathrm{mod}(A)$ has eigenvalues with bounded imaginary parts (indeed, in $(-\pi, \pi]$), the computation of $\mathrm{e}^{\mathrm{mod}(A)}$ may be easier than the computation of e^A. In [29] and [33] the authors do not explicitly identify the matrix unwinding function or obtain any of the properties given in section 3.

Many techniques are available for computing the matrix exponential, as explained in the classic paper by Moler and Van Loan [30], [31]. The MATLAB function `expm` uses the scaling and squaring algorithm of Higham [17], [19]. The algorithm is based on the relation $\mathrm{e}^A = (\mathrm{e}^{2^{-s}A})^{2^s}$ and the use of $[m, m]$ Padé approximants r_{mm} to the exponential; it approximates $\mathrm{e}^{2^{-s}A} \approx r_m(2^{-s}A)$ with a choice of s and m that depends on $\|A\|$. Opting for a larger than necessary value for s may lead to overscaling and possibly inaccurate results [18, sec. 10.3]. Overscaling can be avoided by using the sequence $\{\|A^k\|^{1/k}\}$ instead of $\|A\|$ when choosing the value of s, as shown by Al-Mohy and Higham [1], who derive an improved scaling and squaring algorithm, which we denote by `expm_new`. The cost of both algorithms is roughly $6 + s$ matrix multiplications and one solution of a multiple right-hand side system.

We combine `expm_new` with the matrix unwinding function in the following algorithm.

ALGORITHM 6.1. *Given* $A \in \mathbb{C}^{n\times n}$ *this algorithm computes* e^A *using the scaling and squaring method in conjunction with the matrix unwinding function.*

1 Compute the Schur decomposition $A = QTQ^*$ (Q unitary, T upper triangular).
2 Compute $U = T - 2\pi i\mathcal{U}(T)$ using Algorithm 4.1 and omitting line 14.
3 if $\|U\|_F > \|T\|_F$, $U = T$; end
4 Compute $X = e^U$ using `expm_new` from [1].
5 $X = QXQ^*$

Cost: $(30\frac{2}{3} + \theta + \frac{s_1}{3})n^3$, where θn^3 flops is the cost of the reordering in Algorithm 4.1 and s_1 is the scaling parameter used by `expm_new`.

Note that in line 3 we test whether the unwinding function increases the norm

of T, and if it does we work with T. The reason is that if $\mathcal{U}(T)$ exceeds T in norm then there is likely to be no benefit to the scaling and squaring method from using $\mathcal{U}(T)$ in place of T and, moreover, this is only likely to happen when $\mathcal{U}(T)$ is very ill conditioned, in which case the computed $\mathcal{U}(T)$ may be rather inaccurate.

It is instructive to compare the costs of computing e^A from Algorithm 6.1 and directly from `expm_new`. For `expm_new` applied directly to A the cost is $(14 + 2s_1)n^3$ flops, and if a Schur decomposition is computed and `expm_new` applied to the triangular factor the cost is $(30\frac{1}{3} + \frac{s_2}{3})n^3$; here, s_1 and s_2 are the respective scaling parameters chosen by `expm_new`. We see from these figures that for large s it is more efficient to use the Schur decomposition in the e^A computation, in which case denoting the cost of the reordering in Algorithm 4.1 by θn^3 flops, Algorithm 6.1 will be cheaper if its scaling parameter s satisfies $s < s_2 - 1 - 3\theta$. This inequality is unlikely to be satsified if θ achieves its maximum value of 20, but in the more typical case of $\theta = 1$ (say), the inequality is

$$s < s_2 - 4, \tag{6.2}$$

which is readily satisfied.

Many problems require the computation of a matrix exponential with an argument that has a very large norm due to the presence of eigenvalues with imaginary parts of large magnitude. In these applications it is often required to compute e^{At}, its norm [32], or a product $\mathrm{e}^{At}B$, for many values of t, not necessarily equally spaced. In these situations significant savings in cost are possible by computing with $\mathrm{mod}(A)$ instead of A.

Note that it is easy to show that

$$\mathrm{e}^{At} = \mathrm{e}^{\mathrm{mod}(A)t}\mathrm{e}^{2\pi i t\mathcal{U}(A)},$$

and $\mathrm{e}^{2\pi i t\mathcal{U}(A)}$ is the identity matrix precisely when $t\mathcal{U}(\lambda_i) \in \mathbb{Z}$ for every eigenvalue λ_i of A. So it is safe to compute e^{At} as $\mathrm{e}^{\mathrm{mod}(A)t}$ only when t is an integer.

Before describing two particular problems from applications we consider three matrices that demonstrate the reduction in computational cost that can be obtained by using argument reduction.

The first matrix is a 2×2 matrix with real entries and eigenvalues $1 \pm 500i$:

$$A = \begin{bmatrix} 1 & -500 \\ 500 & 1 \end{bmatrix}, \qquad \mathcal{U}(A) = \begin{bmatrix} 0 & 8i \\ -8i & 0 \end{bmatrix}. \tag{6.3}$$

Matrices like this often appear as diagonal blocks in a quasitriangular Schur form. The second matrix is the Tolosa matrix of dimension 1090 from Matrix Market [28]. It is a sparse matrix arising in the stability analysis of a model of an airplane in flight; it has many eigenvalues with large imaginary part, as shown in Figure 6.1. Our last matrix is the block upper triangular matrix

$$\begin{bmatrix} 0 & 30 & 1 & 1 & 1 & 1 \\ -100 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & -6 & 1 & 1 \\ 0 & 0 & 500 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 200 \\ 0 & 0 & 0 & 0 & -15 & 0 \end{bmatrix} \tag{6.4}$$

from [34, p. 7, Ex I], which has two triple eigenvalues $\pm 10\sqrt{30}i$. We compute the exponential e^{At} of each of the three matrices for $t = 1$ and $t = 100$, first using

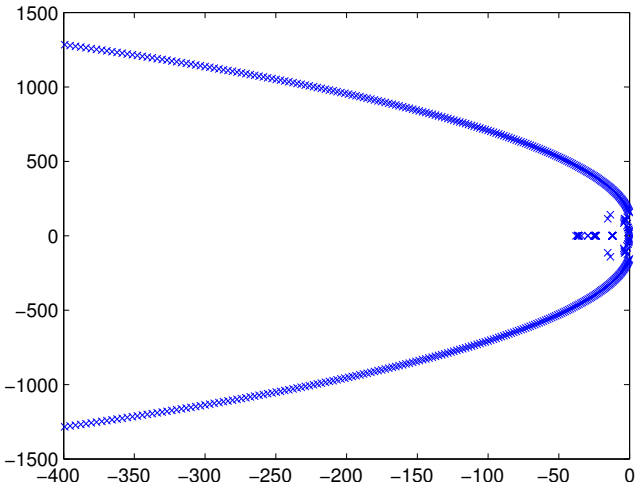

FIG. 6.1. *Spectrum of Tolosa matrix of dimension* 1090.

TABLE 6.1
Scaling parameter s in scaling and squaring method for evaluating e^{At}.

	`expm`		`expm_new`	
	A	mod(A)	A	mod(A)
Matrix (6.3), $t = 1$	7	0	7	0
Matrix (6.3), $t = 100$	14	5	14	5
Tolosa matrix, $t = 1$	16	10	8	6
Tolosa matrix, $t = 100$	22	14	15	12
Matrix (6.4), $t = 1$	7	3	4	0
Matrix (6.4), $t = 100$	14	9	11	2

MATLAB function `expm` and `expm_new` from [1], and then by Algorithm 6.1 using `expm` and `expm_new` on line 4. Table 6.1 shows the value of s used in the scaling and squaring method in each case. We see reductions in the value of s for mod(At) in every case, the amount of reduction varying with A and t, but being as much as 9, and with (6.2) being satisfied in over half of the cases. For `expm_new` the values of s are smaller than for `expm`, since `expm_new` gathers and exploits information about the nonnormality of the matrices, but argument reduction still leads to a decrease in s, which is especially notable for (6.4) and (6.3), for which no scaling is needed when $t = 1$. The number of swaps required by the Tolosa matrix is about 250, which yields a value of θ about 0.0084.

6.1. Example 1. Open quantum systems. Problems in open quantum systems study the the interaction of a closed quantum system with elements external to it, that is, the environment. The Markovian quantum master equation can be written as

$$\frac{d}{dt}\rho(t) = \mathcal{L}\rho(t), \quad \rho(0) = \rho_0. \tag{6.5}$$

Here, $\mathcal{L}$ stands for the Lindbladian super-operator, which is an $n \times n$ skew-Hermitian matrix, with three nonzero diagonals and bandwidth $2n^{1/2}+3$, perturbed by terms in-

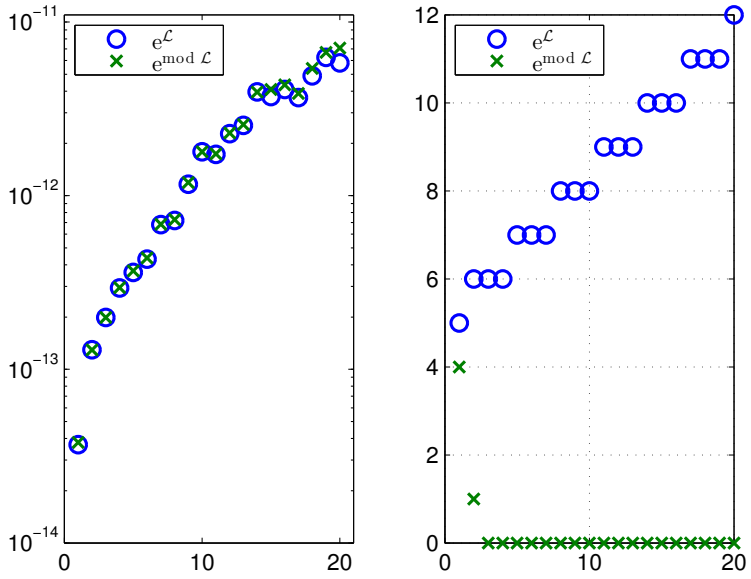

FIG. 6.2. *Relative errors (left) and scaling factors s (right) for computing $e^{\mathcal{L}}$ and $e^{\mathrm{mod}\,\mathcal{L}}$.*

duced from interaction of the system with the environment, for example, from damping. These matrices are characterized by having eigenvalues with large imaginary parts and relatively small real parts [9].

The exact solution to equation (6.5) is $\rho(t) = e^{\mathcal{L}t}\rho_0$ and hence we consider computing the exponential of the Lindbladian. For the example of the quantum damped harmonic oscillator, Figure 6.2 shows the relative errors in computing $e^{\mathcal{L}}$ using `expm_new` and $e^{\mathrm{mod}\,\mathcal{L}}$ using Algorithm 6.1 and a comparison of the scaling factors s the two approaches require. Twenty 100×100 matrices with different parameters are used; their eigenvalues have real parts of order 1 or less and imaginary parts up to order 10^3. We observe that $e^{\mathrm{mod}\,\mathcal{L}}$ is computed with very similar levels of accuracy to $e^{\mathcal{L}}$, but requires a much smaller scaling factor and in many cases no scaling at all. The replacement of U by T in line 3 of Algorithm 6.1 was not carried out for any of these matrices.

6.2. Example 2. Convection–diffusion problems. We consider the convection (advection)–diffusion problem

$$u_t + cu_x = du_{xx}, \tag{6.6}$$

where c and $d > 0$ are constants [24, sec. 3.4]. Spatial discretization using second order central differences yields $u_t = Au$ with a tridiagonal A, so again the solution is given in terms of the matrix exponential.

When the system is dominated by the convection term, i.e., $d \ll |c|$, the matrix A has eigenvalues with small real and large imaginary parts.

We constructed a set of 20 discretization matrices of dimension 100, arranged such that the convection coefficient is increasing and the diffusion coefficient is decreasing: $c = (1.6)^k$ and $d = 0.2(0.5)^k$ for $k = 1\colon 20$. Figure 6.3 shows the relative errors in computing e^A by `expm_new` and $e^{\mathrm{mod}\,A}$ by Algorithm 6.1 and a comparison of the scaling factors the two methods employ. We see that $e^{\mathrm{mod}\,A}$ is computed with the

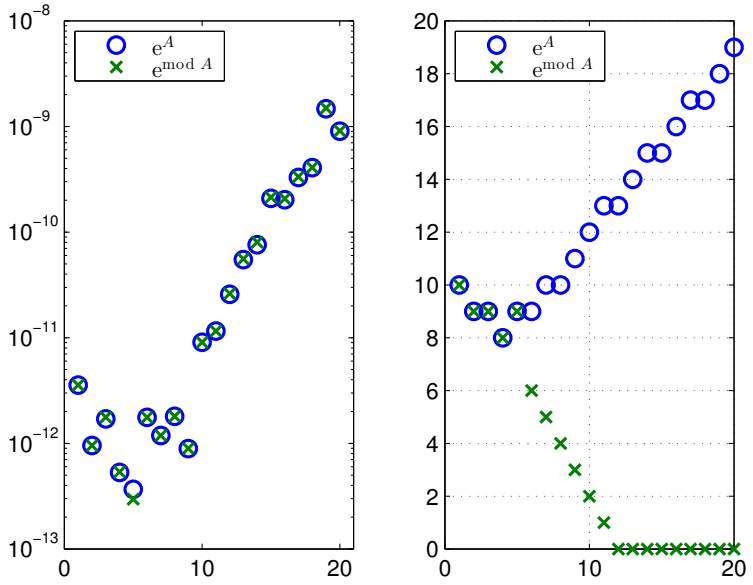

FIG. 6.3. *Relative errors (left) and scaling factors s (right) for computing* e^A *and* $e^{\mathrm{mod}\,A}$.

same accuracy as e^A. For the first five matrices the eigenvalues have imaginary parts smaller or not much larger than the real parts and about the same amount of scaling is required to compute e^A and $e^{\mathrm{mod}\,A}$; thereafter the imaginary parts of the eigenvalues dominate and the use of the matrix unwinding function results in smaller scaling parameters, and no scaling is required for matrices 12 onwards. The replacement of U by T in line 3 of Algorithm 6.1 was used for matrices 3–5.

Finally, we note that for both of these examples inequality (6.2) is satisfied in most cases, with $\theta < 1$.

7. Concluding remarks. The matrix unwinding function is a new primary matrix function that is both an elegant theoretical tool for working with multivalued matrix functions and also a computational tool for matrix argument reduction. Although we have concentrated here on equations involving the logarithm and fractional matrix powers, the matrix unwinding function also plays a useful role in working with inverse trigonometric matrix functions, and this is the subject of our ongoing investigations.

Matrix argument reduction for the exponential is an important ingredient in an algorithm of Güttel and Nakatsukasa, where it helps to avoid a difficult case that arises when the eigenvalues have large imaginary parts [14]. We are also investigating the use of matrix argument reduction in the computation of other matrix functions such as trigonometric functions.

[1] 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.

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

[3] Awad H. Al-Mohy, Nicholas J. Higham, and Samuel D. Relton. Computing the Fréchet derivative of the matrix logarithm and estimating the condition number. MIMS EPrint 2012.72, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, July 2012. 17 pp. Revised December 2012.

[4] Tom M. Apostol. *Mathematical Analysis*. Second edition, Addison-Wesley, Reading, MA, USA, 1974. xvii+492 pp.

[5] Helmer Aslaksen. Multiple-valued complex functions and computer algebra. *SIGSAM Bulletin*, 30(2):12–20, 1996.

[6] Zhaojun Bai and James W. Demmel. On swapping diagonal blocks in real Schur form. *Linear Algebra Appl.*, 186:73–95, 1993.

[7] Russell Bradford. Algebraic simplification of multiple-valued functions. In *Design and Implementation of Symbolic Computation Systems*, John Fitch, editor, volume 721 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, 1993, pages 13–21.

[8] Russell Bradford, Robert M. Corless, James H. Davenport, David J. Jeffrey, and Stephen M. Watt. Reasoning about the elementary functions of complex analysis. *Annals of Mathematics and Artificial Intelligence*, 36:303–318, 2002.

[9] Heinz-Peter Breuer and Francesco Petruccione. *The Theory of Open Quantum Systems*. 2002. xxi+625 pp. ISBN 0-19-852063-8.

[10] Sheung Hun Cheng, Nicholas J. Higham, Charles S. Kenney, and Alan J. Laub. Approximating the logarithm of a matrix to specified accuracy. *SIAM J. Matrix Anal. Appl.*, 22(4):1112–1125, 2001.

[11] Robert M. Corless. *Essential Maple 7: An Introduction for Scientific Programmers*. Springer-Verlag, New York, 2002. xv+282 pp. ISBN 0-387-95352-3.

[12] Robert M. Corless and David J. Jeffrey. The unwinding number. *ACM SIGSAM Bulletin*, 30 (2):28–35, 1996.

[13] Philip I. Davies and Nicholas J. Higham. A Schur–Parlett algorithm for computing matrix functions. *SIAM J. Matrix Anal. Appl.*, 25(2):464–485, 2003.

[14] Stefan Güttel and Yuji Nakatsukasa. Scaled and squared subdiagonal Padé approximation for the matrix exponential exploiting conditioning. In preparation.

[15] Nicholas J. Higham. The Matrix Computation Toolbox. `http://www.ma.man.ac.uk/~higham/mctoolbox`.

[16] Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms*. Second edition, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. xxx+680 pp. ISBN 0-89871-521-0.

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

[18] 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.

[19] Nicholas J. Higham. The scaling and squaring method for the matrix exponential revisited. *SIAM Rev.*, 51(4):747–764, 2009.

[20] Nicholas J. Higham and Lijing Lin. A Schur–Padé algorithm for fractional powers of a matrix. *SIAM J. Matrix Anal. Appl.*, 32(3):1056–1078, 2011.

[21] Nicholas J. Higham and Lijing Lin. An improved Schur–Padé algorithm for fractional powers of a matrix and their Fréchet derivatives. MIMS EPrint 2013.1, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, January 2013. 20 pp. Revised May 2013.

[22] Nicholas J. Higham, D. Steven Mackey, Niloufer Mackey, and Françoise Tisseur. Functions preserving matrix groups and iterations for the matrix square root. *SIAM J. Matrix Anal. Appl.*, 26(3):849–877, 2005.

[23] 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.

[24] Willem Hundsdorfer and Jan Verwer. *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*. Springer-Verlag, Berlin, 2003. x+471 pp. ISBN 3-540-03440-4.

[25] David J. Jeffrey, D. E. G. Hare, and Robert M. Corless. Unwinding the branches of the Lambert W function. *Math. Scientist*, 21:1–7, 1996.

[26] W. Kahan. Branch cuts for complex elementary functions or much ado about nothing's sign bit. In *The State of the Art in Numerical Analysis*, A. Iserles and M. J. D. Powell, editors, Oxford University Press, 1987, pages 165–211.

[27] Charles S. Kenney and Alan J. Laub. The matrix sign function. *IEEE Trans. Automat. Control*, 40(8):1330–1348, 1995.

[28] Matrix Market. `http://math.nist.gov/MatrixMarket/`.

[29] A. McCurdy, K. C. Ng, and B. N. Parlett. Accurate computation of divided differences of the exponential function. *Math. Comp.*, 43(168):501–528, 1984.

[30] Cleve B. Moler and Charles F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix. *SIAM Rev.*, 20(4):801–836, 1978.

[31] Cleve B. Moler and Charles F. Van Loan. Nineteen dubious ways to compute the exponential of a matrix, twenty-five years later. *SIAM Rev.*, 45(1):3–49, 2003.

[32] Yuri M. Nechepurenko and Miloud Sadkane. A low-rank approximation for computing the matrix exponential norm. *SIAM J. Matrix Anal. Appl.*, 32(2):349–363, 2011.

[33] Kwok Choi Ng. Contributions to the computation of the matrix exponential. Technical Report PAM-212, Center for Pure and Applied Mathematics, University of California, Berkeley, February 1984. 72 pp. PhD thesis.

[34] Beresford N. Parlett and Kwok Choi Ng. Development of an accurate algorithm for $\exp(Bt)$. Technical Report PAM-294, Center for Pure and Applied Mathematics, University of California, Berkeley, August 1985. 23 pp. Fortran program listings are given in an appendix with the same report number printed separately.

[35] Charles M. Patton. A representation of branch-cut information. *SIGSAM Bulletin*, 30(2): 21–24, 1996.

[36] L. S. Shieh, Y. T. Tsay, and C. T. Wang. Matrix sector functions and their applications to system theory. *IEE Proc.*, 131(5):171–181, 1984.

[37] Thomas G. Wright. Eigtool, 2002. `http://www.comlab.ox.ac.uk/pseudospectra/eigtool/`.