ITERATIONS FOR THE UNITARY SIGN DECOMPOSITION AND THE UNITARY EIGENDECOMPOSITION

EVAN S. GAWLIK*

Abstract. We construct fast, structure-preserving iterations for computing the sign decomposition of a unitary matrix A with no eigenvalues equal to $\pm i$. This decomposition factorizes A as the product of an involutory matrix $S = \operatorname{sign}(A) = A(A^2)^{-1/2}$ times a matrix $N = (A^2)^{1/2}$ with spectrum contained in the open right half of the complex plane. Our iterations rely on a recently discovered formula for the best (in the minimax sense) unimodular rational approximant of the scalar function $\operatorname{sign}(z) = z/\sqrt{z^2}$ on subsets of the unit circle. When A has eigenvalues near $\pm i$, the iterations converge significantly faster than Padé iterations. Numerical evidence indicates that the iterations are backward stable, with backward errors often smaller than those obtained with direct methods. This contrasts with other iterations like the scaled Newton iteration, which suffers from numerical instabilities if A has eigenvalues near $\pm i$. As an application, we use our iterations to construct a stable spectral divide-and-conquer algorithm for the unitary eigendecomposition.

Key words. Matrix sign function, matrix iteration, structure-preserving, unitary eigendecomposition, Zolotarev, minimax, Padé iteration, Newton iteration

AMS subject classifications. 65F60, 65F15, 41A20, 30E10, 41A50, 15A23

1. Introduction.

Every matrix $A \in \mathbb{C}^{n\times n}$ with no purely imaginary eigenvalues can be written uniquely as a product

$$A = SN,$$

where $S \in \mathbb{C}^{n\times n}$ is involutory ($S^2 = I$), $N \in \mathbb{C}^{n\times n}$ has spectrum in the open right half of the complex plane, and S commutes with N. This is the celebrated matrix sign decomposition [11], whose applications are widespread [3, 17]. In terms of the principal square root $(\cdot)^{1/2}$, we have $S = A(A^2)^{-1/2} =: \operatorname{sign}(A)$ and $N = (A^2)^{1/2}$.

When A is unitary, so too are S and N. It follows that $S = S^{-1} = S^*$, so we may write, for any unitary A with $\Lambda(A) \cap i\mathbb{R} = \emptyset$,

$$A = SN, \quad S^2 = I,\ S = S^*,\ N^2 = A^2,\ N^*N = I,\ \Lambda(N) \subset \mathbb{C}_+, \tag{1.1}$$

where $\Lambda(N)$ denotes the spectrum of N and $\mathbb{C}_+ = \{z \in \mathbb{C} \mid \operatorname{Re}(z) > 0\}$. We refer to this decomposition as the *unitary sign decomposition.*

We say that an algorithm for computing the decomposition (1.1) is backward stable if it computes matrices $\widehat{S}$ and $\widehat{N}$ with the property that the quantities

$$\|A - \widehat{S}\widehat{N}\|,\ \|\widehat{S}^2 - I\|,\ \|\widehat{S} - \widehat{S}^*\|,\ \|\widehat{N}^*\widehat{N} - I\|,\ \|\widehat{N}^2 - A^2\|,\ \max\{0, -\min_{\lambda\in\Lambda(\widehat{N})} \operatorname{Re}\lambda\} \tag{1.2}$$

are each a small multiple of the unit roundoff u ($= 2^{-53}$ in double-precision arithmetic).[1] Here, $\|\cdot\|$ denotes the 2-norm.

The goal of this paper is to design backward stable iterations for computing the decomposition (1.1). To illustrate why this is challenging, let us point out the pitfalls of naive approaches. A widely used iteration for computing the sign of a general matrix $A \in \mathbb{C}^{n\times n}$ is the Newton iteration [21] [12, Section 5.3]

$$X_{k+1} = \frac{1}{2}(X_k + X_k^{-1}), \quad X_0 = A. \tag{1.3}$$

*Department of Mathematics, University of Hawaiʻi at Mānoa (egawlik@hawaii.edu)

Funding: The author was supported by NSF grants DMS-1703719 and DMS-2012427.

[1] Note that this property implies $\|\widehat{N}\widehat{S} - \widehat{S}\widehat{N}\|$ is small as well; see Lemma 3.6.

If A is unitary, then the first iteration is simply

$$X_1 = \frac{1}{2}(A + A^*). \tag{1.4}$$

In floating point arithmetic, this calculation is susceptible to catastrophic cancellation if A has eigenvalues near $\pm i$. Indeed, if we carry out (1.4) followed by (1.3) for $k = 1, 2, \dots$ on the 100×100 unitary matrix `A = gallery('orthog',100,3)` from the MATLAB matrix gallery, then the iteration diverges. Scaling the iterates with standard scaling heuristics [15] leads to convergence, but the computed sign of A satisfies $\|\widehat{S} - \widehat{S}^*\| > 0.1$ in typical experiments. This happens because A has several eigenvalues lying near $\pm i$.

The above algorithm can be reinterpreted in a different way: It is computing the unitary factor in the polar decomposition of $(A + A^*)/2$. Indeed, the Newton iteration $X_{k+1} = \frac{1}{2}(X_k + X_k^{-*})$ for the polar decomposition [12, Section 8.3] coincides with (1.3) on Hermitian matrices. This suggests another family of potential algorithms: compute the polar decomposition of $(A + A^*)/2$ via iterative methods or other means. However, numerical experiments confirm that such algorithms are similarly inaccurate on matrices with eigenvalues near $\pm i$. This unstable behavior is also shared by the superdiagonal Padé iterations for the matrix sign function [14], all of which map eigenvalues $\lambda \approx \pm i$ of A to a small real number (or the inverse thereof) in the first iteration.

One way to overcome these difficulties is to adopt structure-preserving iterations. Here, we say that an iteration $X_{k+1} = g_k(X_k)$ for the unitary sign decomposition is structure-preserving if the iterates X_k are unitary for every k. Examples include the diagonal family of Padé iterations [13], whose lowest-order member is the iteration

$$X_{k+1} = X_k(3I + X_k^2)(I + 3X_k^2)^{-1}, \quad X_0 = A. \tag{1.5}$$

By keeping X_k unitary, a structure-preserving iteration ensures that the eigenvalues of X_k remain on the unit circle, ostensibly skirting the dangers of catastrophic cancellation. We observe numerically that, if implemented in a clever way (described in Section 3), the diagonal Padé iterations are backward stable. However, they can take excessively long to converge on matrices with eigenvalues near $\pm i$. For example, when `A = gallery('orthog',100,3)`, the iteration (1.5) takes 34 iterations to converge.

We construct in this paper a family of structure-preserving iterations for the unitary sign decomposition that converge more rapidly—sometimes dramatically more so—than the diagonal Padé iterations. Numerical evidence indicates that these iterations are backward stable, with backward errors often smaller than those obtained with direct methods.

The key ingredient that we use to construct our iterations is a recently discovered formula for the best (in the minimax sense) unimodular rational approximant of the scalar function $\operatorname{sign}(z) = z/\sqrt{z^2}$ on subsets of the unit circle [7]. Remarkably, it can be shown that composing two such approximants yields a best approximant of higher degree [7], laying the foundations for an iteration. When applied to matrices, the iteration produces a sequence of unitary matrices $X_0 = A$, X_1, X_2, $\dots$ that converges rapidly to $S = \operatorname{sign}(A)$, often significantly faster than the corresponding diagonal Padé iteration. When `A = gallery('orthog',100,3)`, for example, the lowest-order iteration converges within 6 iterations, which is about 6 times faster than the corresponding diagonal Padé iteration (1.5).

Prior work. Matrix iterations constructed from rational minimax approximants have attracted growing interest in recent years. Early examples include the optimal scaling heuristic proposed by Byers and Xu [2] for the Newton iteration for the polar decomposition, as well as an analogous scaling heuristic for the matrix square root proposed by Wachspress [22] and Beckermann [1]. Nakatsukasa, Bai, and Gygi [18] designed an optimal scaling heuristic for the Halley iteration for the polar decomposition, and their strategy was generalized to higher order by Nakatsukasa and Freund [19]. The latter work elucidated the link between these scaling heuristics and the seminal work of Zolotarev [23] on rational minimax approximation. The iterations derived in [19] have a variety of applications, including algorithms for the symmetric eigendecomposition, singular value decomposition, polar decomposition, and CS decomposition [19, 8].

All of the aforementioned algorithms rely crucially on the following fact: if two rational minimax approximants of the scalar function $\operatorname{sign}(x)$ on suitable real intervals are composed with one another, then their composition is a best approximant of higher degree [19]. A related composition law for rational minimax approximants of $\sqrt{z}$ has been used to construct iterations for the matrix square root [4]. These iterations were generalized to the matrix pth root in [5] and used to derive approximation theoretic results in [6]. An even more recent advancement—a composition law for rational minimax approximants of $\operatorname{sign}(z)$ on subsets of the unit circle [7]—is what inspired the present paper.

Connections to other iterations. The iterations we derive in this paper are intimately connected to several existing iterations for the matrix sign function and the polar decomposition. When applied to a unitary matrix A, our iterations produce a sequence of unitary matrices whose Hermitian part coincides with the sequence of matrices generated by Nakatsuka and Freund's iterations [19] for the polar decomposition of $(A + A^*)/2$. A special case of this result is a connection between our lowest-order iteration for $\operatorname{sign}(A)$ and the optimally scaled Halley iteration for the polar decomposition of $(A + A^*)/2$ [18]. It is important to note that these equivalences hold only in exact arithmetic. In floating-point arithmetic, our iterations behave very differently from the aforementioned algorithms.

There is also a link between our iterations and the diagonal Padé iterations. Roughly speaking, our iterations are designed using rational minimax approximants of $\operatorname{sign}(z)$ on two circular arcs containing ± 1. If these arcs are each shrunk to a point, then the diagonal Padé iterations are recovered. This helps to explain the slow convergence of the diagonal Padé iterations on unitary matrices with eigenvalues near $\pm i$: The iterations need to approximate $\operatorname{sign}(z)$ near $z = \pm i$, but they use rational functions that are designed to approximate $\operatorname{sign}(z)$ near $z = \pm 1$.

Unitary eigendecomposition. Our emphasis on handling eigenvalues near $\pm i$ is not merely pedantic. It is precisely the sort of situation that one often encounters if the unitary sign decomposition is used as part of a spectral divide-and-conquer algorithm for the unitary eigendecomposition.

Indeed, consider a unitary matrix $A \in \mathbb{C}^{m \times m}$ with eigendecomposition $A = V \Lambda V^*$. The matrix $(I + \operatorname{sign}(A))/2$ is a spectral projector onto the invariant subspace $\mathcal{V}_+$ of A associated with eigenvalues having positive real part. A spectral divide-and-conquer algorithm uses this projector to find orthonormal bases $U_1 \in \mathbb{C}^{m \times m_1}$, $U_2 \in \mathbb{C}^{m \times m_2}$, $m_1 + m_2 = m$, for $\mathcal{V}_+$ and its orthogonal complement. Then $\begin{pmatrix} U_1 & U_2 \end{pmatrix}^* A \begin{pmatrix} U_1 & U_2 \end{pmatrix}$ is block diagonal, so recursion can be used to determine V and Λ. At each step, scalar multiplication by complex numbers with unit modulus can be used to rotate the spectrum so that it is distributed approximately evenly between the left and right half-planes. If A has a cluster of nearby eigenvalues, then it is reasonable to expect this process to center the cluster near $\pm i$ at some step. This is precisely what we observe in practice, and the ability to compute the unitary sign decomposition quickly and accurately in the presence of eigenvalues near $\pm i$ becomes paramount.

Organization. This paper is organized as follows. We begin in Section 2 by studying rational minimax approximants of $\operatorname{sign}(z)$ on the unit circle. This material is largely drawn from [7], but we add some additional results and insights to relate these approximants to Padé approximants. Next, we use these approximants to construct matrix iterations for the unitary sign decomposition in Section 3. We illustrate their utility by constructing a spectral divide-and-conquer algorithm for the unitary eigendecomposition in Section 4. We conclude with numerical examples in Section 5.

2. Rational Approximation of the Sign Function on the Unit Circle.

In this section, we study rational approximants of the scalar function $\operatorname{sign}(z) = z/\sqrt{z^2}$ on the set

$$\mathbb{S}_\Theta = \{z \in \mathbb{C} \mid |z| = 1, \arg z \notin (\Theta, \pi - \Theta) \cup (-\pi + \Theta, -\Theta)\},$$

where $\Theta \in (0, \pi/2)$. Since our ultimate interest is in constructing structure-preserving iterations for the unitary sign decomposition, we focus on rational functions r satisfying $|r(z)| = 1$ for $|z| = 1$. We call such rational functions unimodular. Unimodular rational functions have the property that $r(A)$ is unitary for any unitary matrix A.

The problem of determining the best (in the minimax sense) unimodular rational approximant of $\operatorname{sign}(z)$ on $\mathbb{S}_\Theta$ has recently been solved in [7]. To describe the solution, let us introduce some notation. We use $\operatorname{sn}(\cdot, \ell)$, $\operatorname{cn}(\cdot, \ell)$, and $\operatorname{dn}(\cdot, \ell)$ to denote Jacobi's elliptic functions with modulus ℓ, and we use $\ell' = \sqrt{1-\ell^2}$ to denote the modulus complementary to ℓ. We denote the complete elliptic integral of the first kind by $K(\ell) = \int_0^{\pi/2} (1 - \ell^2 \sin^2 \theta)^{-1/2}\, d\theta$. We say that a rational function $r(z) = p(z)/q(z)$ has type (m, n) if p and q are polynomials of degree at most m and n, respectively.

Theorem 2.1. *Let $\Theta \in (0, \pi/2)$ and $n \in \mathbb{N}_0$. Among all rational functions r of type $(2n+1, 2n+1)$ that satisfy $|r(z)| = 1$ for $|z| = 1$, the ones which minimize*

$$\max_{z \in \mathbb{S}_\Theta} \left| \arg\left(\frac{r(z)}{\operatorname{sign}(z)} \right) \right|$$

are

$$r(z) = r_{2n+1}(z; \Theta) = z \prod_{j=1}^{n} \frac{z^2 + a_j}{1 + a_j z^2}$$

and its reciprocal, where

$$a_j = a_j(\Theta) = \left(\frac{\ell \operatorname{sn}(v_j, \ell') + \operatorname{dn}(v_j, \ell')}{\operatorname{cn}(v_j, \ell')} \right)^{2(-1)^{j+n}},$$

$v_j = \frac{2j-1}{2n+1} K(\ell')$, $\ell = \cos\Theta$, *and* $\ell' = \sqrt{1-\ell^2} = \sin\Theta$.

Proof. See [7, Theorem 2.1 and Remark 2.2]. □

Remark 2.2. For simplicity, we have chosen to focus only on best unimodular rational approximants of $\operatorname{sign}(z)$ on $\mathbb{S}_\Theta$ of type $(2n+1, 2n+1)$ in this paper. Best approximants of type $(2n, 2n)$ can also be written down; see [7] for details.

The rational function $r_{2n+1}(z;\Theta)$ has the following remarkable behavior under composition.

THEOREM 2.3. *Let $\Theta \in (0,\pi/2)$, $m,n \in \mathbb{N}_0$, and $\widetilde{\Theta} = \left|\arg(r_{2n+1}(e^{i\Theta};\Theta))\right|$. Then*

$$r_{2m+1}(r_{2n+1}(z;\Theta);\widetilde{\Theta}) = r_{(2m+1)(2n+1)}(z;\Theta).$$

Proof. See [7, Theorem 3.3 and Remark 3.6]. □

We also have the following error estimate.

THEOREM 2.4. *Let $\Theta \in (0,\pi/2)$ and $n \in \mathbb{N}_0$. We have*

$$\max_{z\in\mathbb{S}_\Theta} \left|\arg\left(\frac{r_{2n+1}(z;\Theta)}{\operatorname{sign}(z)}\right)\right| \le 4\rho^{-(2n+1)},$$

where

$$\rho = \rho(\Theta) = \exp\left(\frac{\pi K(\cos\Theta)}{2K(\sin\Theta)}\right). \tag{2.1}$$

Proof. See [7, Theorem 3.2], and note that their definition of ρ differs from ours by a factor of 2 in the exponent. □

Remark 2.5. Theorems 2.3 and 2.4 continue to hold when $\Theta = 0$ if we adopt the convention that $\rho(0) = \infty$, $\mathbb{S}_0 = \{-1,1\}$, and $r_{2n+1}(z;0) = z\prod_{j=1}^{n}\frac{z^2+a_j(0)}{1+a_j(0)z^2}$. We elaborate on this fact below.

2.1. Connections with Other Rational Approximants. The rational function $r_{2n+1}(z;\Theta)$ is closely connected to several other well-known rational approximants of $\operatorname{sign}(z)$.

PROPOSITION 2.6. *As $\Theta \to 0$, $r_{2n+1}(z;\Theta)$ converges coefficientwise to $zp_n(z^2)$, where $p_n(z)$ is the type-(n,n) Padé approximant of $z^{-1/2}$ at $z = 1$.*

Proof. This is a consequence of [7, Proposition 3.9], where it is shown that $\sqrt{z}/r_{2n+1}(\sqrt{z};\Theta)$ converges coefficientwise to $1/p_n(z)$ as $\Theta \to 0$. □

In the notation of Remark 2.5, the above proposition states that

$$r_{2n+1}(z;0) = zp_n(z^2).$$

This rational function has been studied extensively in [14, 16, 9] [12, Theorem 5.9]. It satisfies [12, Theorem 5.9]

$$zp_n(z^2) = \tanh((2n+1)\operatorname{arctanh} z) \tag{2.2}$$

It also has the following properties. Both $p_n(z)$ and $zp_n(z^2)$ are unimodular [13]; that is, for any $n \in \mathbb{N}_0$,

$$|zp_n(z^2)| = |p_n(z)| = 1, \text{ if } |z| = 1.$$

Under composition, we have [12, Theorem 5.9)(c)]

$$r_{2m+1}(r_{2n+1}(z;0);0) = r_{(2m+1)(2n+1)}(z;0) \tag{2.3}$$

for any $m,n \in \mathbb{N}_0$. Finally, $r_{2n+1}(1;0) = -r_{2n+1}(-1;0) = 1$ for all $n \in \mathbb{N}_0$. These last two facts justify Remark 2.5.

The rational functions $r_{2n+1}(z; 0)$, $n \in \mathbb{N}_0$, have been used in [14] to construct iterations for computing the matrix sign function. The iterations constitute the diagonal family of Padé iterations. The first few diagonal Padé approximants of $z^{-1/2}$ at $z = 1$ are

$$p_0(z) = 1, \; p_1(z) = \frac{3+z}{1+3z}, \; p_2(z) = \frac{5+10z+z^2}{1+10z+5z^2}, \; p_3(z) = \frac{7+35z+21z^2+z^3}{1+21z+35z^2+7z^3}.$$

More generally, Padé iterations can be constructed from rational functions of the form $zp_{m,n}(z^2)$, where $p_{m,n}(z)$ is the type-(m, n) Padé approximant of $z^{-1/2}$ at $z = 1$. However, when $m \neq n$, the Padé iterations are not structure-preserving, as $|p_{m,n}(z)| \not\equiv 1$ for $|z| = 1$ and $m \neq n$.

We now turn our attention back to the rational function $r_{2n+1}(z, \Theta)$ with positive Θ. Interestingly, this function is intimately connected to the solution of another rational approximation problem: approximating $\operatorname{sign}(x)$ on the union of real intervals $[-1, -\ell] \cup [\ell, 1]$.

THEOREM 2.7. *Let $\Theta \in [0, \pi/2)$ and $n \in \mathbb{N}_0$. For $z \in \mathbb{C}$ with $|z| = 1$, we have*

$$\operatorname{Re} r_{2n+1}(z; \Theta) = \widehat{R}_{2n+1}(\operatorname{Re} z; \cos\Theta), \tag{2.4}$$

where

$$\widehat{R}_m(x; \ell) = \begin{cases} \frac{R_m(x;\ell)}{\max_{y\in[\ell,1]} R_m(y;\ell)} & \text{if } \ell \in (0,1), \\ xp_n(x^2), & \text{if } \ell = 1, \end{cases}$$

and

$$R_m(\cdot; \ell) = \operatorname*{arg\,min}_{R \in \mathcal{R}_{m,m}} \max_{x \in [-1,-\ell]\cup[\ell,1]} |R(x) - \operatorname{sign}(x)|.$$

Proof. This identity is proven for $\Theta \in (0, \pi/2)$ in [7, Theorem 2.4]. To see that it also holds when $\Theta = 0$, we must show that if $|z| = 1$ and $x = \operatorname{Re} z = \frac{1}{2}(z + 1/z)$, then

$$\frac{1}{2}\left(\tanh((2n+1)\operatorname{arctanh} z) + \frac{1}{\tanh((2n+1)\operatorname{arctanh} z)}\right) = \tanh((2n+1)\operatorname{arctanh} x).$$

Since $\frac{1+x}{1-x} = -\left(\frac{1+z}{1-z}\right)^2$, we have $\operatorname{arctanh} x = \frac{1}{2}\log\left(\frac{1+x}{1-x}\right) = \log\left(i\frac{1+z}{1-z}\right)$. Thus,

$$\tanh((2n+1)\operatorname{arctanh} x) = \tanh\left((2n+1)\log\left(i\frac{1+z}{1-z}\right)\right). \tag{2.5}$$

On the other hand, the identity $\tanh(2y) = \frac{2\tanh y}{1+\tanh^2 y}$ shows that

$$\begin{aligned} &\frac{1}{2}\left(\tanh((2n+1)\operatorname{arctanh} z) + \frac{1}{\tanh((2n+1)\operatorname{arctanh} z)})\right) \\ &\quad = \coth((4n+2)\operatorname{arctanh} z) \\ &\quad = \coth\left((2n+1)\log\left(\frac{1+z}{1-z}\right)\right). \end{aligned} \tag{2.6}$$

Since $(2n+1)\log\left(\frac{1+z}{1-z}\right)$ differs from $(2n+1)\log\left(i\frac{1+z}{1-z}\right)$ by an odd multiple of $\frac{\pi i}{2}$, it follows that (2.5) and (2.6) are equal. ∎

Written another way, the lemma above states that

$$\frac{1}{2}\left(r_{2n+1}(z;\Theta)+\frac{1}{r_{2n+1}(z;\Theta)}\right)=\widehat{R}_{2n+1}\left(\frac{z+1/z}{2};\cos\Theta\right) \tag{2.7}$$

for all z with $|z|=1$. In particular,

$$\frac{1}{2}\left(zp_n(z^2)+\frac{1}{zp_n(z^2)}\right)=\left(\frac{z+1/z}{2}\right)p_n\left(\left(\frac{z+1/z}{2}\right)^2\right).$$

Since these equalities hold on the unit circle, they hold on all of $\mathbb{C}$.

By combining (2.3), (2.4), and Theorem 2.3, one sees that the function $\widehat{R}_{2n+1}(x;\ell)$ satisfies

$$\widehat{R}_{2m+1}(\widehat{R}_{2n+1}(x,\ell),\widetilde{\ell})=\widehat{R}_{(2m+1)(2n+1)}(x,\ell),\qquad \text{if } \widetilde{\ell}=\widehat{R}_{2n+1}(\ell,\ell) \tag{2.8}$$

for all $m,n\in\mathbb{N}_0$ and all $\ell\in(0,1]$. This equality was derived in [19] for $\ell\in(0,1)$ by counting extrema of $\widehat{R}_{2m+1}(\widehat{R}_{2n+1}(x,\ell),\widetilde{\ell})-\operatorname{sign}(x)$. It can be leveraged to construct iterations for the matrix sign function, and such iterations are particularly well-suited for computing the sign of a Hermitian matrix B (which coincides with the unitary factor in the polar decomposition of B); see (3.5-3.6) below.

3. Algorithm.

3.1. Matrix Iteration.

Theorem 2.3 suggests the following iteration for computing the sign of a unitary matrix A with spectrum contained in $\mathbb{S}_\Theta$, $\Theta\in[0,\pi/2)$:

$$X_{k+1}=r_{2n+1}(X_k;\Theta_k),\qquad X_0=A, \tag{3.1}$$

$$\Theta_{k+1}=|\arg r_{2n+1}(e^{i\Theta_k};\Theta_k)|,\qquad \Theta_0=\Theta. \tag{3.2}$$

Below we summarize the properties of the iteration (3.1-3.2).

Proposition 3.1. *The iteration (3.1-3.2) is structure-preserving. That is, if A is unitary, then X_k is unitary for every $k\geq 0$.*

Proof. Since $|r_{2n+1}(z;\Theta_k)|=1$ for every scalar z with unit modulus, $r_{2n+1}(X;\Theta_k)$ is unitary for every unitary matrix X. □

Theorem 3.2. *Let A be a unitary matrix with spectrum contained in $\mathbb{S}_\Theta$ for some $\Theta\in(0,\pi/2)$. For any $n\in\mathbb{N}$, the iteration (3.1-3.2) converges to $\operatorname{sign}(A)$ with order of convergence $2n+1$. In fact,*

$$\|\log(X_k\operatorname{sign}(A)^{-1})\|\leq 4\rho^{-(2n+1)^k}, \tag{3.3}$$

for every $k\geq 0$, where ρ is given by (2.1).

Proof. By Theorem 2.3, we have

$$X_k=r_{(2n+1)^k}(A;\Theta)$$

for every $k\geq 0$. Thus, every eigenvalue of $X_k\operatorname{sign}(A)^{-1}$ is of the form $r_{(2n+1)^k}(\lambda;\Theta)/\operatorname{sign}(\lambda)$

for some eigenvalue λ of A. By Theorem 2.4,

$$\begin{aligned}\|\log(X_k \operatorname{sign}(A)^{-1})\| &= \max_{\lambda\in\Lambda(X_k \operatorname{sign}(A)^{-1})} |\arg\lambda| \\ &= \max_{\lambda\in\Lambda(A)} \left|\arg\left(\frac{r_{(2n+1)^k}(\lambda;\Theta)}{\operatorname{sign}(\lambda)}\right)\right| \\ &\le \max_{z\in\mathbb{S}_\Theta} \left|\arg\left(\frac{r_{(2n+1)^k}(z;\Theta)}{\operatorname{sign}(z)}\right)\right| \\ &\le 4\rho^{-(2n+1)^k}. \qquad \square\end{aligned}$$

3.2. Connections with Other Iterations. There is an intimate connection between the iteration (3.1-3.2) and several existing iterations for the matrix sign function. First, Proposition 2.6 implies that (3.1-3.2) reduces to the diagonal Padé iteration when we set $\Theta = 0$:

$$X_{k+1} = X_k p_n(X_k^2), \quad X_0 = A. \tag{3.4}$$

Second, there is a link between the iteration (3.1-3.2) and the iteration

$$Y_{k+1} = \widehat{R}_{2n+1}(Y_k;\ell_k), \qquad Y_0 = B, \tag{3.5}$$

$$\ell_{k+1} = \widehat{R}_{2n+1}(\ell_k;\ell_k), \qquad \ell_0 = \ell, \tag{3.6}$$

which was introduced in [19] to compute the sign of a Hermitian matrix B with spectrum contained in $[-1,-\ell]\cup[\ell,1]$. Note that (3.5-3.6) reduces to

$$Y_{k+1} = Y_k p_n(Y_k^2), \qquad Y_0 = B \tag{3.7}$$

when we set $\ell = 1$ and ignore the spectrum of B. This is the same iteration as (3.4), but with a starting matrix labelled B rather than A.

Proposition 3.3. *Let A be a unitary matrix with no eigenvalues equal to $\pm i$. Let $n\in\mathbb{N}$ and $\Theta\in[0,\pi/2)$. If $B = (A+A^*)/2$ and $\ell = \cos\Theta$, then the iterations (3.1-3.2) and (3.5-3.6) generate sequences satisfying*

$$Y_k = \frac{1}{2}(X_k + X_k^*), \text{ and } \ell_k = \cos\Theta_k$$

for every $k\ge 0$.

Proof. It follows from Theorem 2.3 that in the iteration (3.1-3.2), we have

$$X_k = r_{(2n+1)^k}(A;\Theta), \quad \Theta_k = |\arg r_{(2n+1)^k}(e^{i\Theta};\Theta)|,$$

for each $k\ge 0$. On the other hand, the composition law (2.8) implies that in the iteration (3.5-3.6), we have

$$Y_k = \widehat{R}_{(2n+1)^k}(B;\ell), \quad \ell_k = \widehat{R}_{(2n+1)^k}(\ell;\ell),$$

for each $k\ge 0$. Thus, by (2.7),

$$\begin{aligned}\frac{1}{2}(X_k + X_k^*) &= \frac{1}{2}(X_k + X_k^{-1}) \\ &= \frac{1}{2}\left(r_{(2n+1)^k}(A;\Theta) + r_{(2n+1)^k}(A;\Theta)^{-1}\right) \\ &= \widehat{R}_{(2n+1)^k}((A+A^{-1})/2;\cos\Theta) \\ &= \widehat{R}_{(2n+1)^k}(B;\ell) \\ &= Y_k.\end{aligned}$$

Also, by Theorem 2.7,

$$\cos\Theta_k = \operatorname{Re} e^{i\Theta_k} = \operatorname{Re} r_{(2n+1)^k}(e^{i\Theta};\Theta) = \widehat{R}_{(2n+1)^k}(\operatorname{Re} e^{i\Theta};\cos\Theta) = \widehat{R}_{(2n+1)^k}(\ell;\ell) = \ell_k.$$

□

In the case that $\Theta = 0$, the above result implies a connection between the diagonal Padé iterations (3.4) and (3.7).

Corollary 3.4. *Let A be a unitary matrix with no eigenvalues equal to $\pm i$, and let $n \in \mathbb{N}$. If $B = (A + A^*)/2$, then the diagonal Padé iterations (3.4) and (3.7) generate sequences satisfying*

$$Y_k = \frac{1}{2}(X_k + X_k^*)$$

for every $k \geq 0$.

3.3. Implementation. To implement the kth step of the iteration (3.1-3.2), one must compute products of unitary matrices of the form

$$V_j = (X_k^2 + a_j I)(I + a_j X_k^2)^{-1} = (X_k + a_j X_k^*)(X_k^* + a_j X_k)^{-1}, \quad j = 1, 2, \dots, n, \tag{3.8}$$

where X_k is unitary. The following lemma describes a method for computing (3.8) that is guaranteed to produce a matrix that is unitary to machine precision.

Lemma 3.5. *Let $B \in \mathbb{C}^{m\times m}$ be a nonsingular normal matrix. Let $Q_1R_1 = B$ and $Q_2R_2 = B^*$ be the QR factorizations of B and B^*, respectively. Then*

$$BB^{-*} = Q_1Q_2^*.$$

Proof. Since R_1 is the Cholesky factor of B^*B and R_2 is the Cholesky factor of $BB^* = B^*B$, we have $R_1 = R_2$. Hence, $BB^{-*} = Q_1R_1R_2^{-1}Q_2^* = Q_1Q_2^*$. □

Once (3.8) has been computed for each j, one must decide in what order to multiply the matrices $V_1, V_2, \dots, V_n$, and X_k. Our numerical experience suggests that this decision has a strong influence on the backward stability of the algorithm. We find that the choice

$$X_{k+1} = \frac{1}{2}(X_kV_1V_2\cdots V_n + V_nV_{n-1}\cdots V_1X_k) \tag{3.9}$$

is preferable to, for instance, $X_{k+1} = X_kV_1V_2\cdots V_n$ or $X_{k+1} = V_nV_{n-1}\cdots V_1X_k$. This choice appears to guarantee that $\|X_kA - AX_k\| = O(u)$ for each k, which is essential for backward stability; see Lemma 3.7 for details. A proof that $\|X_kA - AX_k\| = O(u)$ when (3.9) is used remains an open problem.

Termination. We must also decide how to terminate the iteration. Here we suggest terminating slightly early and applying two post-processing steps—symmetrization followed by one step of the Newton-Schulz iteration [12, Equation 8.20] for the polar decomposition—to ensure that the computed matrix $\widehat{S} \approx \operatorname{sign}(A)$ is Hermitian and unitary to machine precision. These post-processing steps have the following effect. Let $\{\sigma_j \cos\theta_j + i\sin\theta_j\}_{j=1}^m$ be the eigenvalues of X_k, where $\sigma_j \in \{-1, 1\}$ and $|\theta_j| < \pi/2$ for each j. Then

$$Y = \frac{1}{2}(X_k + X_k^*) \tag{3.10}$$

has eigenvalues $\{\sigma_j \cos\theta_j\}_{j=1}^m$, and

$$Z = \frac{1}{2}Y(3I - Y^*Y) = \frac{1}{2}Y(3I - Y^2) \tag{3.11}$$

has eigenvalues $\{\frac{1}{2}\sigma_j \cos\theta_j(3-\cos^2\theta_j)\}_{j=1}^m$. For small θ_j, we have

$$\frac{1}{2}\sigma_j \cos\theta_j(3-\cos^2\theta_j) = \sigma_j\left(1-\frac{3}{8}\theta_j^4\right)+O(\theta_j^6).$$

This number will lie within a tolerance δ of ±1 if

$$\theta_j \lesssim \left(\frac{8\delta}{3}\right)^{1/4}. \tag{3.12}$$

The above calculations suggest the following termination criterion. Since the eigenvalues of $X_k - X_k^*$ are $\{2i\sin\theta_j\}_{j=1}^m \approx \{2i\theta_j\}_{j=1}^m$, we terminate the iteration and carry out the post-processing steps (3.10-3.11) as soon as

$$\|X_k - X_k^*\| \leq 2\left(\frac{8\delta}{3}\right)^{1/4}.$$

Note that since the Frobenius norm $\|\cdot\|_F$ is an upper bound for the 2-norm $\|\cdot\|$, we may safely replace $\|X_k - X_k^*\|$ by $\|X_k - X_k^*\|_F$ in the criterion above. If desired, a second symmetrization can be performed after the Newton-Schulz step. This has virtually no effect on the eigenvalues' distance to ±1, but it may be desirable if an exactly Hermitian matrix is sought.

Spectral angle. Let us also mention how to determine Θ so that $\Lambda(A) \subset \mathbb{S}_\Theta$. We hereafter refer to the smallest such Θ as the *spectral angle* of A, denoted $\Theta(A)$. A simple heuristic is to estimate the eigenvalues λ_+ and λ_- of A that lie closest to i and $-i$, respectively. Then one can set

$$\Theta = \max\{\pi/2 - |\arg(i\lambda_-)|, |\arg(i\lambda_+)| - \pi/2\}.$$

In practice, it is not necessary to determine the spectral angle of A precisely. Our experience suggests that underestimates and overestimates of Θ can be used without significant harm, unless Θ is very close to $\pi/2$.

Spectral angles close to $\pi/2$. There are a few delicate numerical issues that arise when the spectral angle of A is close to $\pi/2$. First, as noted in [19, Section 4.3], the built-in MATLAB functions `ellipj` and `ellipke` cannot be used to reliably compute $\mathrm{sn}(\cdot,\ell')$, $\mathrm{cn}(\cdot,\ell')$, $\mathrm{dn}(\cdot,\ell')$, and $K(\ell')$ when $\Theta = \arccos\ell$ is close to $\pi/2$. Instead, the code described in [19, Section 4.3] is preferred. In addition, the lowest-order iteration ($n=1$) appears to be more reliable than the higher-order iterations when $\Theta > \pi/2 - u^{1/2}$, so we advocate using the lowest-order iteration until Θ_k falls below $\pi/2 - u^{1/2}$ (recall that $u = 2^{-53}$ denotes the unit roundoff). Typically this takes two or fewer iterations, after which one can switch to a higher-order iteration if desired.

To implement the lowest-order iteration ($n=1$) when $\Theta > \pi/2 - u^{1/2}$, we have found the following heuristic to be useful for ensuring rapid convergence. If, at the kth iteration, Θ_k lies above $\pi/2 - u^{1/2}$, we compute Θ_{k+1} as $\Theta_{k+1} = \Theta(X_{k+1})$ (the spectral angle of X_{k+1}) rather than via (3.2). This tends to speed up the iteration. To improve stability, we have also found it prudent to replace Θ_k by $\pi/2 - 10u$ if $\Theta_k > \pi/2 - 10u$.

A summary of our proposed algorithm for computing the unitary sign decomposition is presented in Algorithm 3.1.

3.4. Backward Stability. We now discuss how some of the choices made above are inspired by backward stability considerations.

Algorithm 3.1 Order-$(2n+1)$ iteration for the unitary sign decomposition
Inputs: Unitary matrix $A \in \mathbb{C}^{m \times m}$, tolerance $\delta > 0$, degree $n \in \mathbb{N}$
Outputs: Matrices $S, N \in \mathbb{C}^{m \times m}$ satisfying (1.1)

```
$\Theta_0 = \min\{\Theta(A), \pi/2 - 10u\}$
$X_0 = A$, $n_0 = n$, $k = 0$
while $\|X_k - X_k^*\|_F > 2(8\delta/3)^{1/4}$ do
  if $\Theta_k > \pi/2 - u^{1/2}$ then $n = 1$ else $n = n_0$ end if
  $Y = X_k$, $Z = X_k$
  for $j = 1$ to $n$ do
    $Q_1 R_1 = X_k + a_j(\Theta_k) X_k^*$ (QR factorization)
    $Q_2 R_2 = X_k^* + a_j(\Theta_k) X_k$ (QR factorization)
    $Y = Y Q_1 Q_2^*$
    $Z = Q_1 Q_2^* Z$
  end for
  $X_{k+1} = \frac{1}{2}(Y + Z)$
  if $\Theta_k > \pi/2 - u^{1/2}$ then
    $\Theta_{k+1} = \min\{\Theta(X_{k+1}), \pi/2 - 10u\}$
  else
    $\Theta_{k+1} = |\arg r_{2n+1}(e^{i\Theta_k}; \Theta_k)|$
  end if
  $k = k + 1$
end while
$S = (X_k + X_k^*)/2$
$S = S(3I - S^2)/2$
$S = (S + S^*)/2$
$N = SA$
return $S$, $N$
```

We first address a remark that was made in the footnote of this paper's introduction concerning the list of backward errors (1.2). At first glance, this list may appear to be incomplete because the norm of $\widehat{N}\widehat{S} - \widehat{S}\widehat{N}$ is absent. The following lemma shows that if $\widehat{S}$ and $\widehat{N}$ are well-conditioned matrices and $\|\widehat{N}^2 - A^2\|$, $\|A - \widehat{S}\widehat{N}\|$, and $\|\widehat{S}^2 - I\|$ are small, then $\|\widehat{N}\widehat{S} - \widehat{S}\widehat{N}\|$ is automatically small as well.

Lemma 3.6. *Let $A \in \mathbb{C}^{m \times m}$ be a unitary matrix. For any invertible matrices $\widehat{S}, \widehat{N} \in \mathbb{C}^{m \times m}$, we have*

$$\|\widehat{N}\widehat{S} - \widehat{S}\widehat{N}\| \leq \left(\|\widehat{N}^2 - A^2\| + (1 + \|\widehat{S}\|\|\widehat{N}\|)\|A - \widehat{S}\widehat{N}\| + \|\widehat{N}\|^2\|\widehat{S}^2 - I\| \right) \|\widehat{N}^{-1}\|\|\widehat{S}^{-1}\|.$$

Proof. This follows from the identity

$$(\widehat{N}\widehat{S} - \widehat{S}\widehat{N})\widehat{S}\widehat{N} = \widehat{N}^2 - A^2 + A(A - \widehat{S}\widehat{N}) + (A - \widehat{S}\widehat{N})\widehat{S}\widehat{N} + \widehat{N}(\widehat{S}^2 - I)\widehat{N}. \qquad \square$$

The next lemma shows that in order to achieve backward stability, it is prudent to compute a Hermitian matrix $\widehat{S}$ such that $\|\widehat{S}^2 - I\|$ and $\|A\widehat{S} - \widehat{S}A\|$ are small, and then set $\widehat{N} = \widehat{S}A$. This highlights the importance of ensuring the smallness of $\|AX_k - X_k A\|$ in Algorithm 3.1.

Lemma 3.7. *Let $A \in \mathbb{C}^{m \times m}$ be a unitary matrix, let $\widehat{S}$ be an invertible Hermitian* *matrix, and let* $\widehat{N} = \widehat{S}A$. *Then*

$$\|\widehat{N}^*\widehat{N} - I\| \leq \|\widehat{S}^2 - I\|, \tag{3.13}$$

$$\|A - \widehat{S}\widehat{N}\| \leq \|\widehat{S}^2 - I\|, \tag{3.14}$$

$$\|\widehat{N}^2 - A^2\| \leq \|\widehat{S}\|\|A\widehat{S} - \widehat{S}A\| + \|\widehat{S}^2 - I\|, \tag{3.15}$$

$$\|\widehat{N}\widehat{S} - \widehat{S}\widehat{N}\| \leq \|\widehat{S}\|\|A\widehat{S} - \widehat{S}A\|. \tag{3.16}$$

Proof. Since $A^*A = I$, $\widehat{N} = \widehat{S}A$, and $\widehat{S} = \widehat{S}^*$, we have

$$\widehat{N}^*\widehat{N} - I = A^*\widehat{S}^2A - I = A^*(\widehat{S}^2 - I)A.$$

Taking the norm of both sides proves (3.13). Similarly, the equalities

$$\begin{aligned} A - \widehat{S}\widehat{N} &= (I - \widehat{S}^2)A, \\ \widehat{N}^2 - A^2 &= \widehat{S}A\widehat{S}A - A^2 = \widehat{S}(A\widehat{S} - \widehat{S}A)A + (\widehat{S}^2 - I)A^2, \\ \widehat{N}\widehat{S} - \widehat{S}\widehat{N} &= \widehat{S}(A\widehat{S} - \widehat{S}A) \end{aligned}$$

yield (3.14-3.16). □

4. A Spectral Divide-and-Conquer Algorithm for the Unitary Eigendecomposition. The iteration we have proposed for computing the unitary sign decomposition can be used to construct a spectral divide-and-conquer algorithm for the unitary eigendecomposition, following [20, 19]. The idea is as follows. Given a unitary matrix $A \in \mathbb{C}^{m\times m}$, we scale A by a complex number $e^{i\phi}$ so that roughly half (say, m_1) of the eigenvalues of $e^{i\phi}A$ lie in the right half of the complex plane, and roughly half (say, m_2) lie in the left half of complex plane. We then compute $S = \mathrm{sign}(e^{i\phi}A)$ using Algorithm 3.1. The matrix $P = (I+S)/2$ is a spectral projector onto the invariant subspace $\mathcal{V}_+$ of $e^{i\phi}A$ associated with the eigenvalues of $e^{i\phi}A$ having positive real part. Using subspace iteration, we can compute orthonormal bases $U_1 \in \mathbb{C}^{m\times m_1}$ and $U_2 \in \mathbb{C}^{m\times m_2}$ (where $m_1 + m_2 = m$) for $\mathcal{V}_+$ and its orthogonal complement. Then

$$\begin{pmatrix} U_1^* \\ U_2^* \end{pmatrix} A \begin{pmatrix} U_1 & U_2 \end{pmatrix} = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$$

is block diagonal, so we can recurse to find eigendecompositions $A_1 = V_1\Lambda_1V_1^*$ and $A_2 = V_2\Lambda_2V_2^*$. The eigendecomposition of A is then $A = V\Lambda V^*$, where

$$V = \begin{pmatrix} U_1V_1 & U_2V_2 \end{pmatrix}$$

and

$$\Lambda = \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix}.$$

Since every eigenvalue of P is either 0 and 1, subspace iteration with P typically converges in one iteration, or, in rare cases, two. To choose the scalar $e^{i\phi}$, a simple heuristic is to compute the median μ of the arguments of the diagonal entries of A and set $\phi = \pi/2 - \mu$. When A is nearly diagonal, this has the effect of centering the eigenvalues around i.

A summary of the algorithm just described is presented in Algorithm 4.1.

5. Numerical Examples. In this section, we study the iteration (3.1-3.2) numerically, and we test Algorithms 3.1 and 4.1 on a collection of unitary matrices.

Algorithm 4.1 Divide-and-conquer algorithm for the unitary eigendecomposition
Inputs: Unitary matrix $A \in \mathbb{C}^{m \times m}$
Outputs: Matrices $V, \Lambda \in \mathbb{C}^{m \times m}$ satisfying $V \Lambda V^* = A$, $V^* V = I$, and Λ diagonal

```
$\phi = \frac{\pi}{2} - \text{median}\{\arg A_{11}, \dots, \arg A_{mm}\}$
$S = \text{sign}(e^{i\phi} A)$
$P = (I + S)/2$
Use subspace iteration to compute orthonormal bases $U_1 \in \mathbb{C}^{m \times m_1}$ and $U_2 \in \mathbb{C}^{m \times m_2}$ for the 0- and 1-eigenspaces of $P$.
$A_1 = U_1^* A U_1,\ A_2 = U_2^* A U_2$
Recurse to find eigendecompositions $V_1 \Lambda_1 V_1^* = A_1$ and $V_2 \Lambda_2 V_2^* = A_2$.
$V = \begin{pmatrix} U_1 V_1 & U_2 V_2 \end{pmatrix}$
$\Lambda = \begin{pmatrix} \Lambda_1 & 0 \\ 0 & \Lambda_2 \end{pmatrix}$
return $V$, $\Lambda$
```

	$\frac{\pi}{2} - \Theta$										
n	1.5	1	0.5	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}	10^{-14}	10^{-16}
1	1	2	2	3	4	4	5	5	5	5	5
2	1	2	2	3	3	3	3	3	3	4	4
3	1	1	2	2	2	3	3	3	3	3	3
4	1	1	1	2	2	2	3	3	3	3	3
5	1	1	1	2	2	2	2	2	3	3	3
6	1	1	1	2	2	2	2	2	2	2	2
7	1	1	1	2	2	2	2	2	2	2	2
8	1	1	1	2	2	2	2	2	2	2	2

Table 5.1
Smallest integer k for which $4\rho(\Theta)^{-(2n+1)^k} \le (8\delta/3)^{1/4}$, where $\delta = 10^{-16}$, for various values of n and Θ.

5.1. Scalar Iteration. To understand how rapidly the iteration (3.1-3.2) can be expected to converge, let us study the upper bound (3.3). Table 5.1 reports the smallest integer k for which $4\rho(\Theta)^{-(2n+1)^k}$ falls below the number $(8\delta/3)^{1/4}$ appearing in the convergence criterion (3.12). Here, we took $\delta = 10^{-16}$ and considered various choices of n and Θ. The integer k so computed provides an estimate for the number of iterations one can expect (3.1-3.2) to take to converge to $\text{sign}(A)$ if A has spectrum contained in $\mathbb{S}_\Theta$.

For comparison, we computed the number of iterations needed for the scalar Padé iteration

$$z_{k+1} = r_{2n+1}(z_k; 0) = z_k p_n(z_k^2)$$

to converge to $\text{sign}\, z_0$, starting from $z_0 = e^{i\Theta}$. The results, reported in Table 5.2, show that the Padé iterations take significantly longer to converge if Θ is close to $\pi/2$. This suggests the matrix Padé iteration (3.4) will require a large number of iterations to converge to $\text{sign}(A)$ if the spectral angle $\Theta(A)$ is close to $\pi/2$.

5.2. Matrix Iteration. To test Algorithm 3.1, we computed the sign decomposition of four unitary matrices:

1. A matrix sampled randomly from the Haar measure on the $m \times m$ unitary group.

						$\frac{\pi}{2}-\Theta$					
n	1.5	1	0.5	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}	10^{-14}	10^{-16}
1	1	2	3	7	11	15	19	24	28	32	37
2	1	2	2	5	8	10	13	16	19	22	25
3	1	2	2	4	6	9	11	13	16	18	21
4	1	1	2	4	6	8	10	12	14	16	19
5	1	1	2	3	5	7	9	11	13	15	17
6	1	1	2	3	5	7	9	10	12	14	16
7	1	1	2	3	5	6	8	10	12	13	15
8	1	1	2	3	5	6	8	9	11	13	14

Table 5.2

Smallest integer k for which $|r_{(2n+1)^k}(e^{i\Theta};0)-1| \le (8\delta/3)^{1/4}$, where $\delta = 10^{-16}$, for various values of n and Θ.

2. `A = gallery('orthog',m,3)`. This is the m-point discrete Fourier transform matrix with entries $A_{jk} = e^{2\pi i(j-1)(k-1)/m}/\sqrt{m}$. Its eigenvalues are $1, -1, i, -i$. The spectrum of the floating point representation of A therefore includes $O(u)$-perturbations of $\pm i$, posing a challenge to numerical algorithms for the unitary sign decomposition.
3. `A = circshift(eye(m),1)`. This is a permutation matrix with eigenvalues $e^{2\pi ij/m}$, $m = 1, 2, \dots, m$. For even m, the spectrum of A includes $\pm i$. The same is true of the floating point representation of A, since the entries of A are integers.
4. `A = gallery('orthog',m,-2)` (with columns normalized). The entries of A (prior to normalizing columns) are $A_{jk} = \cos((k-1/2)(j-1)\pi/m)$. The spectrum of A is clustered near ± 1, making its sign decomposition somewhat easy to compute iteratively.

In our numerical experiment, we used $m = 100$. The computed spectral angles for the matrices above were $\pi/2 - \Theta(A) = 0.026$, 4.4×10^{-16}, 0, and 0.95, respectively.

On each of the matrices above, we compared 10 algorithms:

- Algorithm 3.1 with $n = 1, 4, 8$.
- The diagonal Padé iteration (3.4) with $n = 1, 4, 8$. We implemented this by running Algorithm 3.1 with line 1 replaced by $\Theta_0 = 0$.
- Three algorithms that compute the unitary factor S in the polar decomposition of $B = (A + A^*)/2$. The first uses the Newton iteration with $1,\infty$-norm scaling, as described in [12, Section 8.6] and implemented in [10]. The second uses the Zolo-pd algorithm from [19]. The third computes S as $S = UV^*$, where $B = U\Sigma V^*$ is the SVD of B. In all three cases, we applied post-processing to S ($S = (S + S^*)/2$, followed by $S = S(3I - S^2)/2$, followed by $S = (S + S^*)/2$) and set $N = SA$.
- A direct method: computing the eigendecomposition $A = V\Lambda V^*$ of A and setting $S = V \operatorname{sign}(\Lambda) V^*$. We computed the eigendecomposition by using the MATLAB command `schur(A,'complex')` and setting the off-diagonal entries of the triangular factor to zero. We applied post-processing to S ($S = S(3I - S^2)/2$ followed by $S = (S + S^*)/2$) and set $N = SA$.

The results of the tests are reported in Table 5.3. All of the algorithms under consideration performed in a backward stable way on the first and fourth matrices. On the second and third matrices (gallery('orthog',m,3) and circshift(eye(m),1)), only the direct method and the structure-preserving iterations (Algorithm 3.1 and the Padé iteration (3.4)) exhibited backward stability. Among the structure-preserving iterations, Algorithm 3.1 consistently converged more quickly than the Padé iteration (3.4) for each degree n. The reduction in iteration count was particularly noticeable for gallery('orthog',m,3) and circshift(eye(m),1).

5.3. Unitary eigendecomposition.

Next, we tested our spectral divide-and-conquer algorithm 4.1 on the same four matrices. We implemented line 2 of Algorithm 4.1 in nine different ways, namely, by using the nine indirect methods considered in the previous experiment. We compared the results with the following direct method: [V,Lambda]=schur(A,'complex'); Lambda = diag(diag(Lambda)). The results are reported in Table 5.4.

All of the algorithms under consideration performed in a backward stable way on the first, second, and fourth matrices. On the third matrix circshift(eye(m),1), the algorithms that used Zolo-pd and the SVD did not. Curiously, the algorithm that used the Newton iteration succeeded, but this is an anomaly. Changing circshift(eye(m),1) to circshift(eye(m),1)+eps*randn(m) leads to a backward error $\|A-\widehat{V}\widehat{\Lambda}\widehat{V}^*\|$ close to 0.1 for the Newton-based algorithm, and it has a negligible effect on the other algorithms' backward errors.

6. Conclusion.

This paper constructed structure-preserving iterations for computing the unitary sign decomposition using rational minimax approximants of the scalar function sign(z) on the unit circle. Relative to other structure-preserving iterations, they converge significantly faster, and relative to non-structure-preserving iterations, they exhibit much better numerical stability. We used our iterations to construct a spectral divide-and-conquer algorithm for the unitary eigendecomposition.

REFERENCES

[1] B. Beckermann, *Optimally scaled Newton iterations for the matrix square root*, Advances in Matrix Functions and Matrix Equations workshop, Manchester, UK, 2013.

[2] R. Byers and H. Xu, *A new scaling for Newton's iteration for the polar decomposition and its backward stability*, SIAM Journal on Matrix Analysis and Applications, 30 (2008), pp. 822–843.

[3] E. D. Denman and A. N. Beavers Jr, *The matrix sign function and computations in systems*, Applied Mathematics and Computation, 2 (1976), pp. 63–94.

[4] E. S. Gawlik, *Zolotarev iterations for the matrix square root*, SIAM Journal on Matrix Analysis and Applications, 40 (2019), pp. 696–719.

[5] E. S. Gawlik, *Rational minimax iterations for computing the matrix pth root*, Constructive Approximation (to appear), (2020).

[6] E. S. Gawlik and Y. Nakatsukasa, *Approximating the pth root by composite rational functions*, arXiv preprint arXiv:1906.11326, (2019).

[7] E. S. Gawlik and Y. Nakatsukasa, *Zolotarev's fifth and sixth problems*, arXiv preprint arXiv:2011.10877, (2020).

[8] E. S. Gawlik, Y. Nakatsukasa, and B. D. Sutton, *A backward stable algorithm for computing the CS decomposition via the polar decomposition*, SIAM Journal on Matrix Analysis and Applications, 39 (2018), pp. 1448–1469.

[9] O. Gomilko, F. Greco, and K. Ziętak, *A Padé family of iterations for the matrix sign function and related problems*, Numerical Linear Algebra with Applications, 19 (2012), pp. 585–605.

[10] N. J. Higham, *The matrix computation toolbox*. http://www.ma.man.ac.uk/~higham/mctoolbox.

[11] N. J. Higham, *The matrix sign decomposition and its relation to the polar decomposition*, Linear Algebra and its Applications, 212 (1994), pp. 3–20.

[12] N. J. Higham, *Functions of matrices: Theory and computation*, SIAM, 2008.

[13] N. J. Higham, D. S. Mackey, N. Mackey, and F. Tisseur, *Computing the polar decomposition and the matrix sign decomposition in matrix groups*, SIAM Journal on Matrix Analysis and Applications, 25 (2004), pp. 1178–1192.

[14] C. Kenney and A. J. Laub, *Rational iterative methods for the matrix sign function*, SIAM Journal on Matrix Analysis and Applications, 12 (1991), pp. 273–291.

[15] C. Kenney and A. J. Laub, *On scaling Newton's method for polar decomposition and the matrix sign function*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 688–706.

[16] C. S. Kenney and A. J. Laub, *A hyperbolic tangent identity and the geometry of Padé sign function iterations*, Numerical Algorithms, 7 (1994), pp. 111–128.

[17] C. S. Kenney and A. J. Laub, *The matrix sign function*, IEEE Transactions on Automatic Control, 40 (1995), pp. 1330–1348.

[18] Y. Nakatsukasa, Z. Bai, and F. Gygi, *Optimizing Halley's iteration for computing the matrix polar decomposition*, SIAM Journal on Matrix Analysis and Applications, 31 (2010), pp. 2700–2720.

[19] Y. Nakatsukasa and R. W. Freund, *Computing fundamental matrix decompositions accurately via the matrix sign function in two iterations: The power of Zolotarev's functions*, SIAM Review, 58 (2016), pp. 461–493.

[20] Y. Nakatsukasa and N. J. Higham, *Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD*, SIAM Journal on Scientific Computing, 35 (2013), pp. A1325–A1349.

[21] J. D. Roberts, *Linear model reduction and solution of the algebraic Riccati equation by use of the sign function*, International Journal of Control, 32 (1980), pp. 677–687.

[22] E. L. Wachspress, *Positive definite square root of a positive definite square matrix*, Unpublished, (1962).

[23] E. I. Zolotarev, *Applications of elliptic functions to problems of functions deviating least and most from zero*, Zapiski Imperatorskoj Akademii Nauk po Fiziko-Matematiceskomu Otdeleniju, 30 (1877), pp. 1–59.

Algorithm	k	$\|A-\widehat{S}\widehat{N}\|$	$\|\widehat{S}^2-I\|$	$\|\widehat{N}^*\widehat{N}-I\|$	$\|\widehat{N}^2-A^2\|$	$\mu(\widehat{N})$
Alg. 3.1 ($n=1$)	3	1.3e−15	1.1e−15	1.8e−15	2.4e−15	0.0e+0
Alg. 3.1 ($n=4$)	2	1.2e−15	9.4e−16	1.9e−15	3.9e−15	0.0e+0
Alg. 3.1 ($n=8$)	2	1.2e−15	1.0e−15	1.8e−15	4.9e−15	0.0e+0
Padé ($n=1$)	6	1.2e−15	9.4e−16	1.8e−15	2.7e−15	0.0e+0
Padé ($n=4$)	3	1.2e−15	9.4e−16	2.1e−15	4.8e−15	0.0e+0
Padé ($n=8$)	3	1.2e−15	9.7e−16	1.7e−15	6.2e−15	0.0e+0
Polar (Newton)	7	1.2e−15	1.0e−15	1.7e−15	2.8e−14	0.0e+0
Polar (Zolo-pd)	2	1.0e−15	6.4e−16	1.6e−15	2.5e−15	0.0e+0
Polar (SVD)	0	1.2e−15	9.5e−16	1.7e−15	7.4e−15	0.0e+0
Direct	0	1.2e−15	1.1e−15	1.8e−15	1.1e−14	0.0e+0
Alg. 3.1 ($n=1$)	6	1.2e−15	9.8e−16	2.3e−15	3.3e−15	0.0e+0
Alg. 3.1 ($n=4$)	4	1.2e−15	1.0e−15	2.3e−15	1.1e−14	2.1e−15
Alg. 3.1 ($n=8$)	4	1.2e−15	9.8e−16	1.8e−15	7.6e−15	1.0e−15
Padé ($n=1$)	34	1.3e−15	1.3e−15	1.8e−15	9.1e−15	0.0e+0
Padé ($n=4$)	17	1.3e−15	1.2e−15	2.0e−15	6.3e−14	1.6e−16
Padé ($n=8$)	14	1.3e−15	1.1e−15	2.3e−15	7.7e−14	3.7e−15
Polar (Newton)	8	1.2e−15	9.2e−16	1.7e−15	3.4e−1	1.7e−1
Polar (Zolo-pd)	2	1.2e−15	6.5e−16	2.5e−15	2.0e−1	1.0e−1
Polar (SVD)	0	1.8e−2	1.8e−2	1.8e−2	3.6e−1	1.8e−1
Direct	0	1.2e−15	1.1e−15	1.8e−15	8.5e−15	0.0e+0
Alg. 3.1 ($n=1$)	6	1.2e−15	9.6e−16	1.1e−15	4.4e−15	0.0e+0
Alg. 3.1 ($n=4$)	4	1.3e−15	8.7e−16	1.2e−15	6.4e−15	0.0e+0
Alg. 3.1 ($n=8$)	4	1.1e−15	9.4e−16	1.0e−15	5.5e−15	0.0e+0
Padé ($n=1$)	37	4.1e−15	4.1e−15	4.1e−15	8.0e−15	5.4e−16
Padé ($n=4$)	19	1.6e−15	1.6e−15	1.6e−15	5.0e−14	5.6e−16
Padé ($n=8$)	14	1.8e−15	1.8e−15	1.8e−15	1.1e−13	7.2e−16
Polar (Newton)	7	7.1e−16	6.4e−16	6.8e−16	2.0e+0	1.0e+0
Polar (Zolo-pd)	2	7.0e−6	7.0e−6	7.0e−6	2.0e+0	1.0e+0
Polar (SVD)	0	2.3e−15	1.6e−15	2.1e−15	2.0e+0	1.0e+0
Direct	0	1.0e−15	1.0e−15	1.0e−15	1.1e−14	0.0e+0
Alg. 3.1 ($n=1$)	2	1.5e−15	1.2e−15	2.0e−15	2.5e−15	0.0e+0
Alg. 3.1 ($n=4$)	1	1.3e−15	1.2e−15	1.9e−15	3.0e−15	0.0e+0
Alg. 3.1 ($n=8$)	1	1.3e−15	9.6e−16	2.1e−15	3.8e−15	0.0e+0
Padé ($n=1$)	3	1.5e−15	1.0e−15	2.2e−15	2.5e−15	0.0e+0
Padé ($n=4$)	2	1.3e−15	1.0e−15	2.2e−15	3.1e−15	0.0e+0
Padé ($n=8$)	1	1.3e−15	1.0e−15	2.0e−15	3.8e−15	0.0e+0
Polar (Newton)	4	1.2e−15	8.4e−16	2.2e−15	3.8e−15	0.0e+0
Polar (Zolo-pd)	1	1.2e−15	6.4e−16	2.0e−15	2.3e−15	0.0e+0
Polar (SVD)	0	1.2e−15	1.0e−15	2.1e−15	8.0e−15	0.0e+0
Direct	0	1.3e−15	1.0e−15	2.0e−15	9.5e−15	0.0e+0

Table 5.3

Performance of algorithms for computing the unitary sign decomposition of the matrices 1-4. The table reports the iteration count k and backward errors $\|A-\widehat{S}\widehat{N}\|$, $\|\widehat{S}^2-I\|$, $\|\widehat{N}^*\widehat{N}-I\|$, $\|\widehat{N}^2-A^2\|$, $\mu(\widehat{N})=\max\{0,-\min_{\lambda\in\Lambda(\widehat{N})}\operatorname{Re}\lambda\}$ for each algorithm.

Algorithm	$\|A - \widehat{V}\widehat{\Lambda}\widehat{V}^*\|$	$\|\widehat{V}^*\widehat{V} - I\|$	Algorithm	$\|A - \widehat{V}\widehat{\Lambda}\widehat{V}^*\|$	$\|\widehat{V}^*\widehat{V} - I\|$
Alg. 3.1 ($n = 1$)	$4.1e{-}15$	$3.3e{-}15$	Alg. 3.1 ($n = 1$)	$4.8e{-}15$	$4.2e{-}15$
Alg. 3.1 ($n = 4$)	$5.0e{-}15$	$3.8e{-}15$	Alg. 3.1 ($n = 4$)	$5.2e{-}15$	$3.6e{-}15$
Alg. 3.1 ($n = 8$)	$4.8e{-}15$	$3.2e{-}15$	Alg. 3.1 ($n = 8$)	$4.9e{-}15$	$3.3e{-}15$
Padé ($n = 1$)	$5.2e{-}15$	$3.9e{-}15$	Padé ($n = 1$)	$5.7e{-}15$	$4.7e{-}15$
Padé ($n = 4$)	$5.1e{-}15$	$3.4e{-}15$	Padé ($n = 4$)	$2.3e{-}14$	$3.7e{-}15$
Padé ($n = 8$)	$5.7e{-}15$	$3.8e{-}15$	Padé ($n = 8$)	$5.0e{-}14$	$3.3e{-}15$
Polar (Newton)	$1.3e{-}14$	$3.2e{-}15$	Polar (Newton)	$4.8e{-}15$	$4.1e{-}15$
Polar (Zolo-pd)	$5.2e{-}15$	$3.5e{-}15$	Polar (Zolo-pd)	$5.5e{-}1$	$3.3e{-}15$
Polar (SVD)	$4.4e{-}15$	$3.3e{-}15$	Polar (SVD)	$4.8e{-}1$	$4.1e{-}15$
Direct	$1.5e{-}14$	$1.2e{-}14$	Direct	$2.1e{-}14$	$1.7e{-}14$
Alg. 3.1 ($n = 1$)	$6.0e{-}15$	$2.9e{-}15$	Alg. 3.1 ($n = 1$)	$4.6e{-}15$	$3.3e{-}15$
Alg. 3.1 ($n = 4$)	$5.9e{-}15$	$2.8e{-}15$	Alg. 3.1 ($n = 4$)	$4.7e{-}15$	$3.8e{-}15$
Alg. 3.1 ($n = 8$)	$6.3e{-}15$	$2.8e{-}15$	Alg. 3.1 ($n = 8$)	$4.9e{-}15$	$3.5e{-}15$
Padé ($n = 1$)	$5.9e{-}15$	$3.3e{-}15$	Padé ($n = 1$)	$4.8e{-}15$	$4.0e{-}15$
Padé ($n = 4$)	$6.2e{-}15$	$3.1e{-}15$	Padé ($n = 4$)	$5.0e{-}15$	$3.6e{-}15$
Padé ($n = 8$)	$6.6e{-}15$	$2.7e{-}15$	Padé ($n = 8$)	$5.8e{-}15$	$3.5e{-}15$
Polar (Newton)	$1.3e{-}14$	$2.8e{-}15$	Polar (Newton)	$9.2e{-}15$	$3.4e{-}15$
Polar (Zolo-pd)	$6.3e{-}15$	$2.6e{-}15$	Polar (Zolo-pd)	$5.5e{-}15$	$3.6e{-}15$
Polar (SVD)	$8.5e{-}15$	$2.6e{-}15$	Polar (SVD)	$6.2e{-}15$	$3.5e{-}15$
Direct	$1.7e{-}14$	$1.1e{-}14$	Direct	$1.3e{-}14$	$8.7e{-}15$

Table 5.4

Performance of algorithms for computing the unitary eigendecomposition of the matrices 1-2 (left) and (3-4) (right). With the exception of the entries labeled "Direct", the entries reported in column 1 refer to the algorithms for the unitary sign decomposition used in line 2 of Algorithm 4.1.