SIAM J. MATRIX ANAL. APPL.
Vol. 33, No. 2, pp. 460–479

BACKWARD STABILITY OF ITERATIONS FOR COMPUTING THE POLAR DECOMPOSITION*

YUJI NAKATSUKASA† AND NICHOLAS J. HIGHAM†

Abstract. Among the many iterations available for computing the polar decomposition the most practically useful are the scaled Newton iteration and the recently proposed dynamically weighted Halley iteration. Effective ways to scale these and other iterations are known, but their numerical stability is much less well understood. In this work we show that a general iteration $X_{k+1} = f(X_k)$ for computing the unitary polar factor is backward stable under two conditions. The first condition requires that the iteration is implemented in a mixed backward–forward stable manner and the second requires that the mapping f does not significantly decrease the size of any singular value relative to the largest singular value. Using this result we show that the dynamically weighted Halley iteration is backward stable when it is implemented using Householder QR factorization with column pivoting and either row pivoting or row sorting. We also prove the backward stability of the scaled Newton iteration under the assumption that matrix inverses are computed in a mixed backward–forward stable fashion; our proof is much shorter than a previous one of Kiełbasiński and Ziętak. We also use our analysis to explain the instability of the inverse Newton iteration and to show that the Newton–Schulz iteration is only conditionally stable. This work shows that by carefully blending perturbation analysis with rounding error analysis it is possible to produce a general result that can prove the backward stability or predict or explain the instability (as the case may be) of a wide range of practically interesting iterations for the polar decomposition.

Key words. polar decomposition, Newton iteration, inverse Newton iteration, Newton–Schulz iteration, dynamically weighted Halley iteration, QR factorization, row pivoting, row sorting, column pivoting, backward error analysis, rounding error analysis, numerical stability

AMS subject classifications. 15A23, 65F30, 65G50

DOI. 10.1137/110857544

1. Introduction.

Any matrix $A \in \mathbb{C}^{s\times n}$ with $s \geq n$ has a polar decomposition $A = UH$, where $U \in \mathbb{C}^{s\times n}$ has orthonormal columns and H is Hermitian positive semidefinite [14, Chap. 8]. The matrix H is unique, and U is unique when A has full rank. Interest in the polar decomposition has principally stemmed from two key properties of U: it is the nearest matrix with orthonormal columns to A in any unitarily invariant norm [7], [14, Thm. 8.7] and it solves the orthogonal Procrustes problem $\min\{\, \|B-CQ\|_F : Q^*Q = I \,\}$, where $B, C \in \mathbb{C}^{s\times n}$, when $A = C^*B$ [8], [14, Thm. 8.6]. We are interested in the polar decomposition for another reason: it can function as a kernel in computing the symmetric eigenvalue decomposition and the singular value decomposition (SVD). In [21] we use the QR-based dynamically weighted Halley (QDWH) algorithm of Nakatsukasa, Bai, and Gygi [20], which computes U and H via a dynamically weighted Halley iteration, to construct new algorithms for the symmetric eigenvalue decomposition and the SVD that have optimal communication costs and have flop counts within a small factor of those for the best existing algorithms.

*Received by the editors December 1, 2011; accepted for publication (in revised form) February 22, 2012; published electronically June 5, 2012.
http://www.siam.org/journals/simax/33-2/85754.html

†School of Mathematics, The University of Manchester, Manchester, M13 9PL, UK (yuji.nakatsukasa@manchester.ac.uk, http://www.ma.man.ac.uk/∼yuji, higham@ma.man.ac.uk, http://www.ma.man.ac.uk/∼higham). The work of the first author was supported by Engineering and Physical Sciences Research Council grant EP/I005293/1. The work of the second author was supported by Engineering and Physical Sciences Research Council grants EP/I006702/1 and EP/E050441/1 (CICADA: Centre for Interdisciplinary Computational and Dynamical Analysis) and European Research Council Advanced Grant MATFUN (267526).

The optimal communication costs stem from the fact that QWDH requires just QR factorizations and matrix multiplications, both of which can be implemented in a communication-optimal manner [1], and in particular does not require any form of pivoting.

The backward stability of the algorithms of [21] rests on the backward stability of the computation of the polar decomposition. The question therefore arises of whether the QDWH algorithm is backward stable. The algorithm performed in a backward stable manner in the numerical experiments of [20], but no theoretical analysis has been carried out. The same question has previously been asked of the scaled Newton iteration for computing the the polar factors and has a long history. The Newton iteration with scaling was introduced by Higham [11], who found that it performed numerically stably in all his tests. Seventeen years later, Kiełbasiński and Ziętak [18] gave a long and complicated analysis proving backward stability of the scaled Newton iteration under the assumption that matrix inverses are computed in a mixed backward–forward stable way. Byers and Xu [4] subsequently obtained an alternative proof using much simpler arguments, but some incompleteness of the analysis has been pointed out in [19].

In this work we prove backward stability of a general iteration for computing the polar decomposition, under two assumptions. The first is that each iterate is obtained from the previous one in a mixed backward–forward stable way in floating point arithmetic. The second assumption is that no singular value of an iterate significantly decreases relative to the largest singular value from one iteration to the next, which is a condition on the iteration function. Our analysis makes no direct reference to acceleration parameters or implementation details of the iteration. It can therefore be applied to a wide variety of iterations.

We use our analysis to prove backward stability of the QDWH algorithm under the assumption that column pivoting and either row pivoting or row sorting are used in the QR factorization. The backward error bound we derive involves a growth factor that can be exponentially large in n but is known to be small in practice. We also show that the algorithm can be unstable without pivoting, although such instability appears to be rare. In addition, we

- prove that the scaled Newton iteration is backward stable—our proof is much shorter and less laborious than the previous one of Kiełbasiński and Ziętak [18];
- give insight into why the scaled inverse Newton iteration is not backward stable;
- show that the (scaled) Newton–Schulz iteration is backward stable if the starting matrix has 2-norm safely less than $\sqrt{3}$, but can be unstable if the norm is close to $\sqrt{3}$ (which is the boundary of the region of convergence).

The organization of the paper is as follows. In the next section we give a precise definition of what we mean by backward stability of an algorithm for computing the polar decomposition. In section 3 we summarize the QDWH algorithm and its properties. Section 4 contains our backward error analysis, which is used in section 5 to prove the backward stability of the QDWH algorithm with pivoting. Section 5.3 gives some numerical experiments that illustrate our analysis. In section 6 the analysis is applied to the Newton and inverse Newton iterations and to the Newton–Schulz iteration.

2. Backward stability of the polar decomposition. We denote by ϵ a matrix or scalar such that $\|\epsilon\| \leq f(n)u$ for some modest function f depending only on n

(such as a low degree polynomial) and some fixed norm, where $u > 0$ is a small, fixed parameter. In this section and in section 5 onwards we take u to be the unit roundoff, but the results in section 4 are independent of floating point arithmetic and so for these u can be regarded as an arbitrary parameter. We are not interested in tracking the constant $f(n)$, so this notation suppresses it and we will freely write $2\epsilon = \epsilon$, $n\epsilon = \epsilon$, and so on. We will also freely drop higher order terms $f_k(n)\epsilon^k$, $k \geq 2$. All terms related to the matrix of interest, such as $\|A\|$ and $\kappa(A)$, will be kept strictly separate from ϵ.

We define what we mean by backward stability of an algorithm for computing the polar decomposition. Let $\widehat{U}$ and $\widehat{H}$ denote the computed unitary and Hermitian polar factors of $A \in \mathbb{C}^{s\times n}$, and assume that $\widehat{H}$ is Hermitian (if it is not then we can replace it by the nearest Hermitian matrix, $(\widehat{H} + \widehat{H}^*)/2$ [7], [14, Thm. 8.7]). We say the algorithm is backward stable if

$$\widehat{U}\widehat{H} = A + \Delta A, \qquad \|\Delta A\| = \epsilon\|A\|, \tag{2.1a}$$

$$\widehat{H} = H + \Delta H, \qquad \|\Delta H\| = \epsilon\|H\|, \tag{2.1b}$$

$$\widehat{U} = U + \Delta U, \qquad \|\Delta U\| = \epsilon\|U\|, \tag{2.1c}$$

where H is Hermitian positive semidefinite and U is unitary. The conditions (2.1b) and (2.1c) allow for the fact that we cannot expect the computed $\widehat{U}$ to be exactly unitary or the computed $\widehat{H}$ to be positive semidefinite when $\widehat{U}$ and $\widehat{H}$ are represented explicitly as a single matrix.[1]

The condition (2.1a) is expressed in terms of $\widehat{U}$ and $\widehat{H}$ rather than U and H as in [14, (8.32)], as this is the most convenient form for our analysis. However, given (2.1b) and (2.1c) it is easy to show that (2.1a) is equivalent to $UH = A + \Delta A$ with $\|\Delta A\| = \epsilon\|A\|$, which permits the interpretation that the computed polar factors $\widehat{U}$ and $\widehat{H}$ are close to the exact polar factors of a matrix close to A. This is mixed backward–forward stability, and as it is the strongest form of stability we can expect it is reasonable to refer to it as backward stability (as is done by Byers and Xu [4]).

As noted in [14, p. 209], the algorithm that computes the polar decomposition by first computing the SVD $A = P\Sigma Q^*$ and then forming $U = PQ^*$ and $H = Q\Sigma Q^*$ is backward stable, but this is an expensive approach [14, Prob. 8.24], [21].

Note that H in (2.1b) is an arbitrary Hermitian positive semidefinite matrix and not necessarily the Hermitian polar factor of A. However, it is easy to show that (2.1) implies that (2.1b) holds with H the Hermitian polar factor of A, using the fact that the Hermitian polar factor is well conditioned: the Hermitian polar factor of $A + \Delta A$ differs from that of A by at most $\sqrt{2}\|\Delta A\|_F$ in the Frobenius norm [2, p. 215], [14, p. 200].

Finally, we note that another definition of stability, formulated in [14, sect. 4.9.4], is applicable to iterations for computing the unitary polar factor. This definition requires that, close to the limit, an error in one iterate has a bounded effect on later iterates. It turns out that a wide class of iterations, including all those considered in this paper, are stable in this asymptotic sense [14, Thm. 8.19]. A contribution of this paper is to show how a global analysis, incorporating errors incurred throughout the iteration, can be developed that is powerful enough to prove that some iterations are backward stable and to correctly predict that some others are not.

[1] $\widehat{U}$ and $\widehat{H}$ could alternatively be represented in product form, for example, $\widehat{U}$ as a product of Givens rotations and Householder matrices. In this case we could require $\widehat{U}$ to be unitary and $\widehat{H}$ to be positive semidefinite.

3. QDWH for the polar decomposition. This section reviews the QDWH algorithm for computing the polar decomposition of $A \in \mathbb{C}^{s\times n}$ $(s \geq n)$ proposed in [20]. The algorithm is mathematically expressed as

$$X_{k+1} = X_k(a_k I + b_k X_k^* X_k)(I + c_k X_k^* X_k)^{-1}, \qquad X_0 = A/\alpha, \tag{3.1}$$

where $\alpha > 0$ is an estimate[2] of $\|A\|_2$, a safe choice of which is $\|A\|_F$. The iteration (3.1) can be regarded as a generalized version of Halley's iteration $X_{k+1} = X_k(3I + X_k^* X_k)(I + 3X_k^* X_k)^{-1}$, which is a member of the Padé family of iterations [14, sect. 8.5]. The parameters a_k, b_k, c_k are dynamically chosen to accelerate convergence. They are computed by

$$a_k = h(\ell_k), \quad b_k = (a_k - 1)^2/4, \quad c_k = a_k + b_k - 1, \tag{3.2}$$

where

$$h(\ell) = \sqrt{1+\gamma} + \frac{1}{2}\sqrt{8 - 4\gamma + \frac{8(2-\ell^2)}{\ell^2\sqrt{1+\gamma}}}, \quad \gamma = \sqrt[3]{\frac{4(1-\ell^2)}{\ell^4}}. \tag{3.3}$$

Here, ℓ_k is a lower bound for the smallest singular value of X_k. Fortunately, once $\ell_0 \leq \sigma_{\min}(X_0)$ is obtained (for example, via a condition number estimator), effective and sharp bounds can be obtained at no cost from the recurrence

$$\ell_k = \ell_{k-1}(a_{k-1} + b_{k-1}\ell_{k-1}^2)/(1 + c_{k-1}\ell_{k-1}^2), \qquad k \geq 1. \tag{3.4}$$

With such parameters, the iteration (3.1) needs at most six iterations for convergence to the unitary polar factor U of A with the tolerance $u = 2^{-53} \simeq 1.1 \times 10^{-16}$ (the unit roundoff for IEEE double precision arithmetic) for any matrix A with $\kappa_2(A) \leq u^{-1}$.

Iteration (3.1) can be implemented in an inverse-free form by using a QR factorization:

$$X_0 = A/\alpha, \tag{3.5a}$$

$$\begin{bmatrix} \sqrt{c_k}X_k \\ I \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \end{bmatrix} R, \quad X_{k+1} = \frac{b_k}{c_k}X_k + \frac{1}{\sqrt{c_k}}\left(a_k - \frac{b_k}{c_k}\right) Q_1 Q_2^*, \qquad k \geq 1. \tag{3.5b}$$

The main costs of a QDWH iteration (3.5) are one QR factorization of an $(s+n)\times n$ matrix and a matrix multiplication, both of which can be done in a communication-minimal manner [1].

Intuitively, one would expect that for (3.5) to be numerically stable the coefficients of X_k and $Q_1 Q_2^*$ should be of order 1, in order to minimize the possibility of subtractive cancellation. In fact, this is the case. From (3.2),

$$c_k = a_k + (a_k - 1)^2/4 - 1 = \frac{1}{4}((a_k - 1)^2 + 4(a_k - 1)) = \frac{1}{4}(a_k - 1)(a_k + 3),$$

and hence $b_k/c_k = (a_k - 1)/(a_k + 3)$. Since we have $a_k \geq 3$ [20] (and see (5.6b)),

$$\frac{1}{3} \leq \frac{b_k}{c_k} \leq 1. \tag{3.6}$$

[2]We note that while [20] assumes $\alpha \geq \|A\|_2$ one can easily prove convergence of QDWH for any $\alpha > 0$. The speed of QDWH is not severely affected if α is an estimate of $\|A\|_2$ such that $\beta\|A\|_2 \leq \alpha \leq f(n)\|A\|_2$ for $\beta \approx 0.5$ and some function $f(n)$ that grows moderately with n.

Furthermore,

$$\frac{1}{\sqrt{c_k}}\left(a_k - \frac{b_k}{c_k}\right) = \frac{2}{\sqrt{(a_k-1)(a_k+3)}}\left(a_k - \frac{a_k-1}{a_k+3}\right) = \frac{2(a_k+1)^2}{(a_k+3)\sqrt{(a_k-1)(a_k+3)}}.$$

Using $a_k \geq 3$ again, we obtain

$$1.53 \leq \frac{1}{\sqrt{c_k}}\left(a_k - \frac{b_k}{c_k}\right) \leq 2. \tag{3.7}$$

The inequalities (3.6) and (3.7) will be needed in section 5.

Once the iteration (3.5) has been terminated, yielding a computed unitary polar factor $\widehat{U}$, we compute $\widehat{H}$ by [14, sect. 8.8]

$$\widehat{H} = \frac{1}{2}(\widehat{U}^*A + (\widehat{U}^*A)^*). \tag{3.8}$$

The numerical experiments with QWDH in [20] demonstrate excellent backward stability. In section 5 we prove that QWDH is backward stable when the QR factorizations in (3.5) are carried out with row sorting (or pivoting) and column pivoting. In the next section we give the more general backward error analysis that will be needed.

4. Backward error analysis. Let $A \in \mathbb{C}^{s\times n}$ with $s \geq n$ have the SVD $A = P\Sigma Q^*$, where $P \in \mathbb{C}^{s\times n}$ and $Q \in \mathbb{C}^{n\times n}$ have orthonormal columns and $\Sigma = \mathrm{diag}(\sigma_i) \in \mathbb{R}^{n\times n}$ is diagonal. For an arbitrary function $f : [0,\infty) \to [0,\infty)$ with $f(x) = 0$ only if $x = 0$, we define $f(A) = Pf(\Sigma)Q^*$, where $f(\Sigma) = \mathrm{diag}(f(\sigma_i))$. It is easy to show that $f(A)$ does not depend on the particular choice of the SVD of A. Note that this definition of $f(A)$ is nonstandard and differs from the more usual definition that can be phrased in terms of the eigensystem [14].

The unitary polar factors of $f(X)$ and X are identical for all X, as f preserves the unitary SVD factors. We begin with a lemma that gives a sufficient condition for the unitary polar factor of a computed approximation to $f(X)$ to provide a backward stable polar decomposition of X.

LEMMA 4.1. *Let $X \in \mathbb{C}^{s\times n}$ with $s \geq n$. Let $\widehat{Y}$ be a computed approximation to $Y = f(X)$ obtained in a mixed backward–forward stable manner, so that there is an $\widetilde{X} \in \mathbb{C}^{s\times n}$ such that*

$$\widehat{Y} = f(\widetilde{X}) + \epsilon\|\widehat{Y}\|_2, \qquad \widetilde{X} = X + \epsilon\|X\|_2. \tag{4.1}$$

Let the singular values of $\widetilde{X}$ be $M := \sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r =: m > \sigma_{r+1} = \cdots = \sigma_n = 0$. Suppose that the function $f(x)$ satisfies

$$\frac{f(x)}{\max_{m\leq x\leq M} f(x)} \geq \frac{x}{dM}, \qquad x \in [m, M], \tag{4.2}$$

where d is a positive constant that necessarily satisfies[3] $d \geq 1$. If $\widetilde{X}$ is rank deficient, suppose further that $f(0) = 0$. Then any unitary polar factor U of $\widehat{Y}$ and $\widehat{H} =$

[3] By taking $x = M$ in (4.2).

$\frac{1}{2}((U^*X) + (U^*X)^*)$ *satisfy*

$$U\widehat{H} = X + d\epsilon\|X\|_2, \tag{4.3}$$

$$\widehat{H} = H + d\epsilon\|H\|_2, \tag{4.4}$$

where H is the Hermitian polar factor of X.

Proof. We first consider the square case, $s = n$. For $\widetilde{X}$ in (4.1), let $\widetilde{X} = \widetilde{P}\widetilde{\Sigma}\widetilde{Q}^*$ be an SVD, where $\widetilde{\Sigma} = \mathrm{diag}(\sigma_i)$, so that $f(\widetilde{X}) = \widetilde{P}f(\widetilde{\Sigma})\widetilde{Q}^*$. Also let $\widetilde{H} = \widetilde{Q}\widetilde{\Sigma}\widetilde{Q}^*$, which is the Hermitian polar factor of $\widetilde{X}$.

The Hermitian polar factor is very well conditioned, as we noted in section 2. Hence by $f(\widetilde{X}) = \widehat{Y} + \epsilon\|\widehat{Y}\|_2$ we must have $H_{\widehat{Y}} = f(\widetilde{H}) + \epsilon\|\widehat{Y}\|_2$, where $H_{\widehat{Y}}$ is the Hermitian polar factor of $\widehat{Y}$ and $f(\widetilde{H}) = \widetilde{Q}f(\widetilde{\Sigma})\widetilde{Q}^*$ is the Hermitian polar factor of $f(\widetilde{X})$. Hence we have

$$\widehat{Y} = UH_{\widehat{Y}} = Uf(\widetilde{H}) + \epsilon\|\widehat{Y}\|_2 = U\widetilde{Q}f(\widetilde{\Sigma})\widetilde{Q}^* + \epsilon\|\widehat{Y}\|_2.$$

Since also, from (4.1), $\widehat{Y} = \widetilde{P}f(\widetilde{\Sigma})\widetilde{Q}^* + \epsilon\|\widehat{Y}\|_2$, and $\|\widehat{Y}\|_2 = \|f(\widetilde{\Sigma})\|_2 + \epsilon\|\widehat{Y}\|_2 = (1+\epsilon)\|f(\widetilde{\Sigma})\|_2$, we obtain

$$\widetilde{P}f(\widetilde{\Sigma})\widetilde{Q}^* = U\widetilde{Q}f(\widetilde{\Sigma})\widetilde{Q}^* + \epsilon\|f(\widetilde{\Sigma})\|_2. \tag{4.5}$$

Now, with a superscript "+" denoting the pseudoinverse, we right-multiply (4.5) by $\widetilde{Q}f(\widetilde{\Sigma})^+\widetilde{\Sigma}\widetilde{Q}^* = \widetilde{Q}\mathrm{diag}(\sigma_1/f(\sigma_1), \ldots, \sigma_r/f(\sigma_r), 0, \ldots, 0)\widetilde{Q}^*$ to obtain

$$\widetilde{X} = \widetilde{P}\widetilde{\Sigma}\widetilde{Q}^* = U\widetilde{Q}\widetilde{\Sigma}\widetilde{Q}^* + d\epsilon\|\widetilde{X}\|_2, \tag{4.6}$$

where for the last term we have used the bound

$$\begin{aligned}
\|\widetilde{Q}f(\widetilde{\Sigma})^+\widetilde{\Sigma}\widetilde{Q}^*\|_2 &= \max_{i\le r}\frac{\sigma_i}{f(\sigma_i)} \\
&\le \max_{m\le x\le M}\frac{x}{f(x)} \qquad (\text{because } \sigma_i \in [m, M]) \\
&\le \frac{dM}{\max_{m\le x\le M} f(x)} \qquad (\text{by (4.2)}) \\
&\le \frac{dM}{\max_i f(\sigma_i)} = \frac{d\|\widetilde{X}\|_2}{\|f(\widetilde{\Sigma})\|_2}.
\end{aligned}$$

Left-multiplying (4.6) by U^* and using $\widetilde{X} = X + \epsilon\|X\|_2$, by (4.1), we obtain

$$U^*X = \widetilde{H} + d\epsilon\|X\|_2. \tag{4.7}$$

Therefore the residual of U and $\widehat{H}$ as approximate polar factors of X is

$$\begin{aligned}
X - U\widehat{H} &= X - U\cdot\frac{1}{2}\big(U^*X + (U^*X)^*\big) \\
&= \frac{1}{2}U\big(U^*X - (U^*X)^*\big) \\
&= \frac{1}{2}U\big(\widetilde{H} + d\epsilon\|X\|_2 - (\widetilde{H} + d\epsilon\|X\|_2)^*\big) \\
&= d\epsilon\|X\|_2.
\end{aligned}$$

This proves (4.3).

To prove (4.4), first note that since $\widetilde{H}$ is the (well conditioned) Hermitian polar factor of $\widetilde{X} = X + \epsilon\|X\|_2$, we must have $H = \widetilde{H} + \epsilon\|X\|_2$. Combining this with (4.7) and $\widehat{H} = \frac{1}{2}((U^*X) + (U^*X)^*)$ we conclude that

$$\begin{aligned}\widehat{H} - H &= \frac{1}{2}\big(U^*X + (U^*X)^*\big) - H \\ &= \frac{1}{2}\big(\widetilde{H} + d\epsilon\|X\|_2 + (\widetilde{H} + d\epsilon\|X\|_2)^*\big) - (\widetilde{H} + \epsilon\|X\|_2) \\ &= d\epsilon\|X\|_2 = d\epsilon\|H\|_2.\end{aligned}$$

Finally, consider the rectangular case, $s > n$. In this case we append $s-n$ columns of zeros to X, $\widetilde{X}$, and $\widehat{Y}$, making them $s \times s$, and apply the argument above to $s \times s$ matrices. If $A = UH$ is a polar decomposition then $\left[\begin{smallmatrix} A & 0 \end{smallmatrix}\right] = \left[\begin{smallmatrix} U & U_2 \end{smallmatrix}\right] \left[\begin{smallmatrix} H & 0 \\ 0 & 0 \end{smallmatrix}\right]$ is a polar decomposition, where U_2 is any matrix such that $\left[\begin{smallmatrix} U & U_2 \end{smallmatrix}\right] \in \mathbb{C}^{s\times s}$ is unitary. From this it follows that the expanded matrices H, $\widetilde{H}$, $f(H)$, $f(\widetilde{H})$ take the form

$$\begin{bmatrix} \times & 0_{n,s-n} \\ 0_{s-n,n} & 0_{s-n,s-n} \end{bmatrix},$$

while $\widehat{H} = \frac{1}{2}((U^*X) + (U^*X)^*)$ has the form

$$\begin{bmatrix} \times & E^* \\ E & 0_{s-n,s-n} \end{bmatrix}.$$

The conditions (4.1) and (4.2) can be seen to hold for the expanded matrices, and therefore (4.3) and (4.4) hold. Since (4.4) holds for the expanded $\widehat{H}$ it also holds for the original $n \times n$ $\widehat{H}$, and this same equation implies $\|E\|_2 = \epsilon\|H\|_2$, which ensures that (4.3) holds for the original matrices. ∎

Recalling that σ_i denotes the ith singular value of $\widetilde{X}$, we note that (4.2) implies

$$\frac{f(\sigma_i)}{\|f(\widetilde{X})\|_2} \geq \frac{1}{d}\left(\frac{\sigma_i}{\|\widetilde{X}\|_2}\right). \tag{4.8}$$

Thus if d is not too large then no singular value can significantly reduce in size relative to the largest singular value under the mapping f. In particular, any singular value can be mapped to a large value close to $\|f(\widetilde{X})\|_2$, but no large singular value $\sigma_i \approx \|\widetilde{X}\|_2$ can be mapped to a value much less than $\|f(\widetilde{X})\|_2$. But if $f(\sigma_i)/\|f(\widetilde{X})\|_2 \ll \sigma_i/\|\widetilde{X}\|_2$ for some i then $d \gg 1$ by (4.8) and the backward error terms in (4.3) and (4.4) are then large.

A simple example illustrates the effect of "unstable mappings" of singular values—those that require a large value of d. Let $\mu = 10^{-10}$ and define A (representing $\widetilde{X}$ in Lemma 4.1) by $A = P\mathrm{diag}(1, \mu^{1/2}, \mu)Q^T$, where P and Q are orthogonal matrices generated randomly using `gallery('qmult')` in MATLAB; A has the polar factors $U = PQ^T$ and $H = U^TA$. Now let $f(x)$ be a function for which $f(1) = \mu$, $f(\mu^{1/2}) = \mu$, and $f(\mu) = 1$, which is an unstable mapping because (4.2) holds only for $d \geq 1/\mu = 10^{10}$. In MATLAB we compute the SVD $f(A) + \theta E = \widetilde{P}\widetilde{\Sigma}\widetilde{Q}^T$, where E is a random matrix from the normal (0,1) distribution scaled so that $\|E\|_2 = 1$, and then form $\widetilde{U} = \widetilde{P}\widetilde{Q}^T$ and $\widetilde{H} = \frac{1}{2}(\widetilde{U}^TA + (\widetilde{U}^TA)^T)$. The θE term represents the forward error in Lemma 4.1. We take two values of θ: $\theta = 10^4u \approx 1 \times 10^{-12}$, which corresponds to a forward perturbation substantially larger than the rounding level, and $\theta = 0$, which

corresponds to backward and forward perturbations at the rounding level. We find that $\|A-\widetilde{U}\widetilde{H}\|_2/\|A\|_2$ is approximately equal to $\|H-\widetilde{H}\|_2/\|H\|_2$ and takes the values 7×10^{-8} for $\theta = 0$ and 5×10^{-6} for $\theta = 10^4 u$; both values are much larger than the underlying perturbation and reveal a large backward error for the factorization and hence instability of the mapping. Now consider another function such that $f(1) = 1$, $f(\mu^{1/2}) = 10^{-1}$, and $f(\mu) = 0.01$, for which we can take $d = 1$ in (4.2) (since the proof shows that it suffices to take the maximum in (4.2) over the singular values). When we ran the same process for 10^4 randomly generated P, Q, and E we found that $\max(\|A-\widetilde{U}\widetilde{H}\|_2/\|A\|_2, \|H-\widetilde{H}\|_2/\|H\|_2)$ was no larger than 3×10^{-15} for $\theta = 0$ and 9×10^{-13} for $\theta = 10^4 u$. Now the backward errors are no larger than the underlying perturbations, demonstrating the stability of the mapping.

Such an unstable mapping of singular values does not happen in the QDWH or scaled Newton iterations, which we will show in sections 5 and 6.1 to be backward stable, but does happen in the inverse Newton iteration, as we will show in section 6.2. Lemma 4.1 shows that if $\frac{f(x)}{\max_{m\le x\le M} f(x)}$ lies above (or not much below) x/M then the mapping is stable. To illustrate the idea, we show in Figures 4.1–4.4 plots of $\frac{f(x)}{\max_{m\le x\le M} f(x)} = \frac{f(x)}{\|f(x)\|_\infty}$ and x/M for these three methods and the Newton–Schulz iteration. The plots show the case when $\kappa_2(X) = 20$ and optimal scaling/weighting parameters are used. Observe that $f(x)/\|f(x)\|_\infty$ lies above x/M in the interval $[m, M]$ in Figures 4.1 and 4.2 (indicating that (4.2) holds with $d = 1$), but not in Figures 4.3 and 4.4, which represent unstable mappings.

We note that in the rank-deficient case we will usually have $m = \epsilon\|X\|_2$. In this case, if $f(0) = 0$ and $f(x)$ is continuous at $x = 0$, then d in (4.2) essentially satisfies $\frac{f(x)}{\max_{0\le x\le M} f(x)} \ge \frac{x}{dM}$ in the interval $[0, M]$, so d does not depend sensitively on the exact value of ϵ. This argument holds for all the known iterations applicable for rank-deficient matrices, but not for the scaled Newton iteration applicable only for square nonsingular matrices, for which $f(x)$ is not continuous at $x = 0$.

Now we state the main result of this section.

THEOREM 4.2. *Let the nonzero matrix $A \in \mathbb{C}^{s\times n}$ with $s \ge n$. Consider an iteration*

$$X_{k+1} = f_k(X_k), \qquad X_0 = A/\alpha, \tag{4.9}$$

for computing a unitary polar factor U_A of A, with $\alpha > 0$, and denote the computed iterates by $\widehat{X}_k$, where $\widehat{X}_0 = X_0$. Suppose that, for some integer ℓ, $\widehat{X}_\ell^*\widehat{X}_\ell = I + \epsilon$, and let $\widehat{U} = \widehat{X}_\ell$ and $\widehat{H} = \frac{1}{2}(\widehat{U}^*A + (\widehat{U}^*A)^*)$. Suppose, furthermore, that for $k = 0:\ell-1$ the following two conditions hold:

(a) *$\widehat{X}_{k+1}$ is computed from $\widehat{X}_k$ in a mixed backward–forward stable manner, so that there is an $\widetilde{X}_k \in \mathbb{C}^{n\times n}$ such that*

$$\widehat{X}_{k+1} = f_k(\widetilde{X}_k) + \epsilon\|\widehat{X}_{k+1}\|_2, \qquad \widetilde{X}_k = \widehat{X}_k + \epsilon\|\widehat{X}_k\|_2; \tag{4.10}$$

(b) *the function f_k satisfies*

$$\frac{f_k(x)}{\max_{m_k\le x\le M_k} f_k(x)} \ge \frac{x}{dM_k}, \qquad x \in [m_k, M_k], \tag{4.11}$$

where m_k is the smallest positive singular value of $\widetilde{X}_k$, $M_k = \sigma_{\max}(\widetilde{X}_k)$, and $d \ge 1$. If $\widetilde{X}_k$ is rank deficient suppose also that $f_k(0) = 0$.

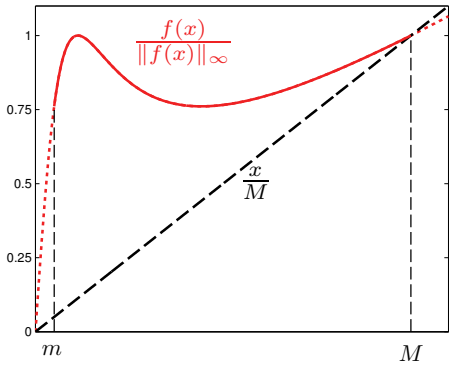

FIG. 4.1. *QDWH iteration,* $f(x) = x\frac{a+bx^2}{1+cx^2}$*: a stable mapping because* (4.2) *holds with* $d = 1$.

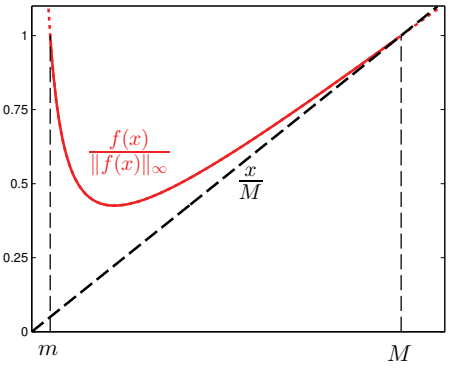

FIG. 4.2. *Scaled Newton iteration,* $f(x) = \frac{1}{2}(\mu x + (\mu x)^{-1})$*: a stable mapping because* (4.2) *holds with* $d = 1$.

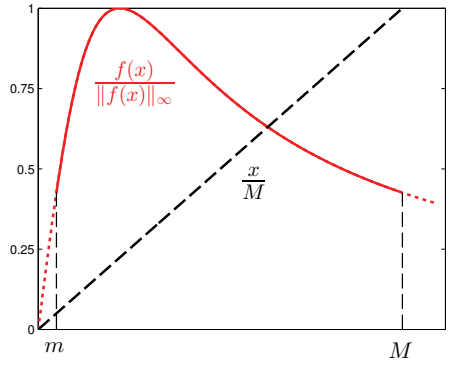

FIG. 4.3. *Inverse Newton iteration,* $f(x) = 2\mu x(1+\mu^2 x^2)^{-1}$*: an unstable mapping.*

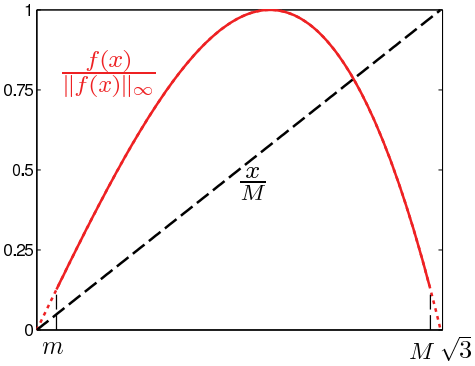

FIG. 4.4. *Newton–Schulz iteration,* $f(x) = \frac{1}{2}x(3-x^2)$*: an unstable mapping if* $M \approx \sqrt{3}$.

Then

$$\widehat{U}\widehat{H} = A + d\epsilon\|A\|_2, \tag{4.12}$$

$$\widehat{H} = H + d\epsilon\|H\|_2, \tag{4.13}$$

where H *is the Hermitian polar factor of* A*. Furthermore, if* A *has full rank then*

$$\widehat{U} = U_A + d\epsilon\kappa_2(A). \tag{4.14}$$

Proof. To simplify the notation we will assume, without loss of generality, that $\alpha = 1$. Let $\widehat{X}_j$ have a polar decomposition $\widehat{X}_j = U_j H_j$ for $j = 1, 2, \ldots$. We first show that for any given k, any unitary polar factor U_k of $\widehat{X}_k$ satisfies

$$\widehat{X}_j - U_k\widehat{H}_j = d\epsilon\|\widehat{X}_j\|_2, \quad H_j - \widehat{H}_j = d\epsilon\|H_j\|_2, \qquad j = 0\colon k, \tag{4.15}$$

where $\widehat{H}_j = \frac{1}{2}((U_k^*\widehat{X}_j) + (U_k^*\widehat{X}_j)^*)$; in other words, an exact unitary polar factor of $\widehat{X}_k$ serves as an approximate unitary polar factor of all the previous computed iterates.

We prove (4.15) by induction. For $j = k$, (4.15) is trivial, because $\widehat{H}_k = H_k$ and so (4.15) holds with $\epsilon = 0$. For $j = k-1$, (4.15) follows immediately from Lemma 4.1 (substituting $\widehat{X}_{k-1}$ for X and $\widehat{X}_k$ for $\widehat{Y}$) and the assumptions (4.10) and (4.11).

Next we consider $j = k-2$. Since (4.15) holds for $j = k-1$ and $\|H_{k-1}\|_2 = \|\widehat{X}_{k-1}\|_2$ we have

$$\widehat{X}_{k-1} = U_k\widehat{H}_{k-1} + d\epsilon\|\widehat{X}_{k-1}\|_2 = U_kH_{k-1} + d\epsilon\|\widehat{X}_{k-1}\|_2. \tag{4.16}$$

By the assumption (4.10) for the case $k := k-2$ we also have

$$\widehat{X}_{k-1} = f_{k-2}(\widetilde{X}_{k-2}) + \epsilon\|\widehat{X}_{k-1}\|_2, \tag{4.17}$$

and by equating (4.16) and (4.17) we obtain

$$U_kH_{k-1} = f_{k-2}(\widetilde{X}_{k-2}) + d\epsilon\|U_kH_{k-1}\|_2.$$

This equation is of the form (4.1) with $\widehat{Y} = U_kH_{k-1}$ (a polar decomposition) and $X = \widehat{X}_{k-2}$. Hence we can invoke Lemma 4.1 to conclude that (4.15) is satisfied for $j = k-2$. By repeating the same argument we can prove (4.15) for $j = k-3, k-4, \ldots, 0$.

Setting $k = \ell$ and $j = 0$ in (4.15), and using $\widehat{X}_0 = X_0 = A$, yields

$$A - U\widehat{H}_0 = d\epsilon\|A\|_2, \qquad H - \widehat{H}_0 = d\epsilon\|H\|_2, \tag{4.18}$$

where U is a unitary polar factor of $\widehat{X}_\ell = \widehat{U}$, $\widehat{H}_0 = \frac{1}{2}(U^*A + (U^*A)^*)$, and H is the Hermitian polar factor of A.

There are two differences between (4.18) and (4.12): in the latter we have $\widehat{U}$ instead of U and $\widehat{H} = \frac{1}{2}(\widehat{U}^*A + (\widehat{U}^*A)^*)$ instead of $\widehat{H}_0$. The differences are reconciled by using the fact that $\widehat{U}^*\widehat{U} = I + \epsilon$ implies $\widehat{U} = U + \epsilon$, by [14, Lem. 8.17]. Thus

$$\begin{aligned} A - \widehat{U}\widehat{H} &= A - U\widehat{H} + (U - \widehat{U})\widehat{H} \\ &= A - U\widehat{H}_0 + U(\widehat{H}_0 - \widehat{H}) + (U - \widehat{U})\widehat{H} \\ &= d\epsilon\|A\|_2 + \epsilon\|A\|_2 + \epsilon\|A\|_2 = d\epsilon\|A\|_2, \end{aligned}$$

which is (4.12).

Finally, using $\widehat{U} = U + \epsilon$ again,

$$\begin{aligned} \widehat{H} - H &= \frac{1}{2}(\widehat{U}^*A + (\widehat{U}^*A)^*) - H \\ &= \frac{1}{2}\left(\widehat{U}^*A + (\widehat{U}^*A)^*) - (U^*A + (U^*A)^*\right) + \frac{1}{2}(U^*A + (U^*A)^*) - H \\ &= \epsilon\|A\|_2 + \widehat{H}_0 - H \\ &= \epsilon\|A\|_2 + d\epsilon\|H\|_2 = d\epsilon\|H\|_2, \end{aligned}$$

where we have used the second equation in (4.18). This proves (4.13).

It remains to prove (4.14). Since (4.12), (4.13) and $\widehat{U} = U + \epsilon$ mean $A + d\epsilon\|A\|_2 = UH$ and $A = U_AH$ are both polar decompositions, it follows from standard perturbation theory [14, Thm. 8.9] that for full rank A, $\|U - U_A\|_F \le d\epsilon\kappa_2(A)$, and hence $\widehat{U} = U + \epsilon = U_A + d\epsilon\kappa_2(A)$. ∎

To summarize, Theorem 4.2 shows that the conditions (4.10) and (4.11), with d of order 1, are sufficient for the iteration (4.9) to produce a backward stable computed polar decomposition, assuming that the computed iterates converge to a matrix with numerically orthonormal columns. Note that the theorem gives (2.1b) in the apparently stronger—but equivalent, as noted in section 2—form with H the Hermitian polar factor of A.

Theorem 4.2 does not require that A has full rank. A difficulty is that the known iterations applicable to rank-deficient matrices (including QDWH, the inverse Newton iteration, and Newton–Schulz, but not the scaled Newton iteration) preserve the rank of the iterates in exact arithmetic (hence X_∞ is the partial isometry in the canonical polar decomposition [14, Chap. 8]), so the condition $\widehat{X}_\ell^* \widehat{X}_\ell = I + \epsilon$ will not hold. However, as noted in [14, p. 205], rounding errors usually perturb the zero singular values (this means $\widetilde{X}_k$ or $\widehat{X}_{k+1}$ has more positive singular values than $\widehat{X}_k$), which eventually converge to 1. Hence in practice we usually have $\widehat{X}_\ell^* \widehat{X}_\ell = I + \epsilon$ for large enough ℓ, even if A is rank-deficient.

Finally, it is worth emphasizing that our analysis exploits the key facts that (a) the Hermitian polar factor H is very well conditioned and (b) if $X^T X$ is close to I then X is close to its unitary polar factor. Crucially, we did not use perturbation bounds for the unitary polar factor, whose condition number is inversely proportional to the one or two smallest singular values of A [14, Thm. 8.9].

5. Numerical stability of QDWH with row and column pivoting. In this section we use Theorem 4.2 to prove that the QDWH algorithm is backward stable provided that the QR factorizations are computed by Householder transformations with column pivoting and either row pivoting or row sorting.

In view of Theorem 4.2, it suffices to prove that the two conditions (4.10) and (4.11) are satisfied throughout the QDWH iterations. We treat these separately in the following two subsections.

5.1. Mixed backward–forward stability of a QDWH iteration. The goal of this subsection is to prove that the mixed backward–forward stability condition (4.10) is satisfied in QDWH. For simplicity we will assume that $\alpha = \|A\|_2$ in (3.5), which implies, using $f(1) = 1$ and $0 \le f(x) \le 1$ on $[0, 1]$ (as shown in [20, sect. 3]), that $\|X_k\|_2 \equiv 1$. This assumption is not fundamental, as the argument below holds with slight modifications as long as $\|A\|_2/f_1(n) \le \alpha \le f_2(n)\|A\|_2$ for $f_1(n)$ and $f_2(n)$ of modest size such that our convention $f_i(n)\epsilon = \epsilon$ holds. In practice, even if $\|X_k\|_2 = 1$, we will have $\|\widehat{X}_k\|_2 = 1 + \epsilon$, but this again does not affect the argument.

If we express a general QDWH iteration step as $Y = f(X)$, where $f(x) = x(a + bx^2)/(1 + cx^2)$, then we need to show that the computed $\widehat{Y}$ satisfies (4.10). Thus, since $\|X\|_2 = \|Y\|_2 = 1$, our goal is to show that

$$\widehat{Y} = f(\widetilde{X}) + \epsilon, \quad \text{where} \quad \widetilde{X} = X + \epsilon. \tag{5.1}$$

We need the following result that describes the row-wise stability of Householder QR factorization with column pivoting and either row pivoting (analogous to partial pivoting for Gaussian elimination) or row sorting (which initially orders the rows by decreasing order of ∞-norm); see [13, sect. 19.4] for details of these pivoting strategies.

THEOREM 5.1 (see [5], [13, sect. 19.4]). *Let $\widehat{Q} \in \mathbb{R}^{s\times n}$ and $\widehat{R} \in \mathbb{R}^{n\times n}$ be the computed QR factors of $A \in \mathbb{R}^{s\times n}$ $(s \ge n)$ obtained from Householder QR factorization with column pivoting and row pivoting or row sorting. Then there exists a $Q \in \mathbb{R}^{s\times n}$ with orthonormal columns such that $(A + \Delta A)\Pi = Q\widehat{R}$, where Π is a permutation matrix and $\|\Delta A(i,:)\|_2 \le c_{s,n}\rho_i u\|A(i,:)\|_2$ for all i, where $c_{s,n}$ is a polynomial in s and n and ρ_i is a row-wise growth factor.*

The growth factors ρ_i in Theorem 5.1, whose precise definition is given in [5], [13, sect. 19.4], are bounded by $\sqrt{s}(1 + \sqrt{2})^{n-1}$, and while this bound is approximately attainable they are usually small in practice [13, sect. 19.4]. The practical implication of the theorem is therefore that row pivoting or sorting together with column pivoting

ensures a small row-wise backward error for Householder QR factorization, and this is what we will need for our analysis. In what follows we will assume ρ_i to be of moderate size, and hence write $\rho_i\epsilon = \epsilon$.

Now we use this theorem to prove (5.1). If the QR factorization of $\left[\begin{smallmatrix}\sqrt{c}X\\ I\end{smallmatrix}\right]$ in (3.5b) is computed by Householder transformations with column pivoting and row pivoting or row sorting then by Theorem 5.1 the computed upper triangular $\widehat{R}$ satisfies

$$\begin{bmatrix}\sqrt{c}(X+\epsilon)\\ I+\epsilon_1\end{bmatrix}\Pi = \begin{bmatrix}Q_1\\ Q_2\end{bmatrix}\widehat{R}, \tag{5.2}$$

for some Q_1 and Q_2 such that $Q = [Q_1^T\ Q_2^T]^T$ has orthonormal columns (recall that $\|X\|_2 = 1$). Our convention is that a subscripted ϵ is an instance of ϵ that takes a fixed value in all appearances. We rewrite (5.2) as

$$\begin{bmatrix}\sqrt{c}\widetilde{X}\\ I+\epsilon_1\end{bmatrix} := \begin{bmatrix}\sqrt{c}(X+\epsilon)\\ I+\epsilon_1\end{bmatrix} = \begin{bmatrix}Q_1\\ Q_2\end{bmatrix}\widehat{R}\Pi^*. \tag{5.3}$$

The proof of Theorem 5.1 shows that Q is equal to the first n columns of $(P_nP_{n-1}\cdots P_1)^T$, where P_i is a Householder matrix defined in terms of the computed quantities from the ith stage of the factorization. The computed version $\widehat{Q}$ is precisely the first n columns of $fl(P_1P_2\cdots P_n)$ and it is easy to show [13, p. 360] that

$$Q = \widehat{Q} + \epsilon = \begin{bmatrix}\widehat{Q}_1+\epsilon_{11}\\ \widehat{Q}_2+\epsilon_{21}\end{bmatrix}. \tag{5.4}$$

Now we note the general result that if $B = [B_1^T\ B_2^T]^T$ has full column rank and QR factorization $B = \left[\begin{smallmatrix}V_1\\ V_2\end{smallmatrix}\right]G$ then $BG^{-1} = \left[\begin{smallmatrix}V_1\\ V_2\end{smallmatrix}\right]$ and hence $V_1V_2^* = B_1G^{-1}G^{-*}B_2^* = B_1(G^*G)^{-1}B_2^* = B_1(B^*B)^{-1}B_2^*$. Using this fact with (5.3) and (5.4) we obtain

$$(\widehat{Q}_1+\epsilon_{11})(\widehat{Q}_2+\epsilon_{21})^* = \sqrt{c}\widetilde{X}\big((I+\epsilon_1)^*(I+\epsilon_1) + c\widetilde{X}^*\widetilde{X}\big)^{-1}(I+\epsilon_1)^*,$$

and hence

$$\widehat{Q}_1\widehat{Q}_2^* = \sqrt{c}\widetilde{X}\big((I+\epsilon_1)^*(I+\epsilon_1) + c\widetilde{X}^*\widetilde{X}\big)^{-1}(I+\epsilon_1)^* - \widehat{Q}_1\epsilon_{21}^* - \epsilon_{11}\widehat{Q}_2^* - \epsilon_{11}\epsilon_{21}^*.$$

Now $-\widehat{Q}_1\epsilon_{21}^* - \epsilon_{11}\widehat{Q}_2^* - \epsilon_{11}\epsilon_{21}^* = \epsilon$, and the rounding errors in forming $fl(\widehat{Q}_1\widehat{Q}_2^*)$ can also be represented by ϵ, so

$$fl(\widehat{Q}_1\widehat{Q}_2^*) = \sqrt{c}\widetilde{X}\big((I+\epsilon_1)^*(I+\epsilon_1) + c\widetilde{X}^*\widetilde{X}\big)^{-1}(I+\epsilon_1)^* + \epsilon. \tag{5.5}$$

Therefore the floating point evaluation of (3.5b) yields

$$\begin{aligned}\widehat{Y} &= fl\left(\frac{b}{c}X + \frac{1}{\sqrt{c}}\left(a-\frac{b}{c}\right)\widehat{Q}_1\widehat{Q}_2^*\right)\\ &= \frac{b}{c}X + \left(a-\frac{b}{c}\right)\widetilde{X}\big((I+\epsilon_1)^*(I+\epsilon_1) + c\widetilde{X}^*\widetilde{X}\big)^{-1}(I+\epsilon_1)^* + \epsilon,\end{aligned}$$

where the last term ϵ includes the last term in (5.5) and the rounding error caused by performing the addition, and we have used the fact that both terms in the addition are of order 1, which follows from (3.6) and (3.7).

Now

$$\begin{aligned}\left((I+\epsilon_1)^*(I+\epsilon_1)+c\widetilde{X}^*\widetilde{X}\right)^{-1} &= \left(I+c\widetilde{X}^*\widetilde{X}+(\epsilon_1^*+\epsilon_1+\epsilon_1^*\epsilon_1)\right)^{-1}\\ &= (I+c\widetilde{X}^*\widetilde{X})^{-1}+\epsilon,\end{aligned}$$

since the singular values of $I+c\widetilde{X}^*\widetilde{X}$ are all larger than 1. Therefore we obtain

$$\widehat{Y}=\frac{b}{c}X+\left(a-\frac{b}{c}\right)\widetilde{X}(I+c\widetilde{X}^*\widetilde{X})^{-1}(I+\epsilon_1)^*+\epsilon.$$

Since the norms of the first two terms are $O(1)$, and $X=\widetilde{X}+\epsilon$, we conclude that

$$\begin{aligned}\widehat{Y} &= \frac{b}{c}\widetilde{X}+\left(a-\frac{b}{c}\right)\widetilde{X}(I+c\widetilde{X}^*\widetilde{X})^{-1}+\epsilon\\ &= \widetilde{X}(aI+b\widetilde{X}^*\widetilde{X})(I+c\widetilde{X}^*\widetilde{X})^{-1}+\epsilon,\end{aligned}$$

which is (5.1).

5.2. The requirement on f. We now show that the iteration function $f_k(x)=x(a_k+b_kx^2)/(1+c_kx^2)$ for the general kth iteration satisfies (4.11). To ease the notation we will drop the subscripts k in this subsection when this can be done without loss of clarity.

We first note the following properties of f and the parameters a,b,c,ℓ:

(5.6a) $\quad b=(a-1)^2/4, \quad c=a+b-1,$

(5.6b) $\quad 3\le a\le\dfrac{2+\ell}{\ell},$

(5.6c) $\quad x\le f(x)\le 1=f(1) \quad \text{for } 0<x<1,$

(5.6d) $\quad f'(x)\ge 0 \quad \text{for } x\ge 1,$

(5.6e) $\quad g(x)=\dfrac{f(x)}{x}$ satisfies $g(0)=a$, $g(1)=1$, and $g'(x)<0$ for $x>0$,

(5.6f) $\quad 0\le f'(1)=(a-3)^2/(a+1)^2<1$ for $a\ge 3$.

These properties are obtained as follows. Property (5.6a) is a restatement of (3.2). Property (5.6b) is from [20, Rem. 1]. The upper bound in (5.6c) follows from the derivation of the parameters in [20]. To obtain the lower bound in (5.6c) we compute, using (5.6a) and (5.6b),

$$\begin{aligned}f(x)-x &= \frac{ax+bx^3}{1+cx^2}-x=\frac{(a-1)x+(b-c)x^3}{1+cx^2}\\ &= \frac{(a-1)x+(1-a)x^3}{1+cx^2}=\frac{(a-1)x(1-x^2)}{1+cx^2}\ge 0 \quad \text{for } x\in(0,1).\end{aligned}$$

Property (5.6d) follows from (5.6f) together with the fact that f is a $(3,2)$-degree odd rational function, which can have at most one local maximum and one local minimum in $(0,\infty)$, both of which lie in $(0,1)$ as shown in the appendix of [20]. To prove (5.6e) we note that $g'(x)=(2x(b-ac))/(1+cx^2)^2$ and $b-ac=(-a^3-a^2+a+1)/4<0$ since $a\ge 3$.

Figure 5.1 plots the functions $y=f(x)$ and $y=x$ for the case $\ell=0.1$ (where $\ell\equiv\ell_k$ is the lower bound for the smallest singular value of X from the recurrence

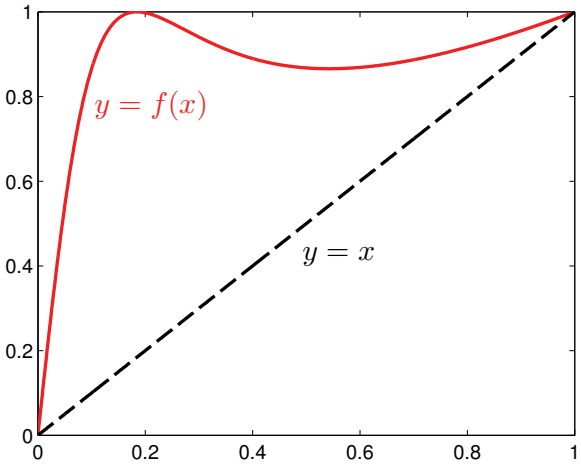

FIG. 5.1. *Plot of $f(x)$ for the QDWH iteration for $\ell = 0.1$.*

(3.4)); the plot corresponds to $m = \ell$ and $M = 1$ in (4.11) and shows the typical behavior of $f(x)$.

Identifying (4.10) with (5.1), we now consider (4.11). We consider separately two cases: $\|\widetilde{X}\|_2 \geq 1$ and $\|\widetilde{X}\|_2 < 1$. When $\|\widetilde{X}\|_2 \geq 1$ we have $M = \|\widetilde{X}\|_2$ and by (5.6c) and (5.6d), (4.11) is satisfied if

$$\frac{f(x)}{x} \geq \frac{f(M)}{dM}, \qquad x \in [0, M].$$

This holds with $d = 1$ because by (5.6e), $f(x)/x$ is decreasing on $(0, \infty)$. Hence (4.11) is satisfied in this case.

Next, when $\|\widetilde{X}\|_2 < 1$ we have $M = \|\widetilde{X}\|_2 < 1$. From (5.6c), with $d = 1/M$, we have

$$\frac{f(x)}{x} \geq 1 = \frac{1}{dM} \geq \frac{\max_{m \leq x \leq M} f(x)}{dM}, \qquad x \in [0, M],$$

so (4.11) holds. By (5.6c) and (5.6d), in exact arithmetic $\|X\|_2 < 1$ only when $\|A\|_2/\alpha < 1$, in which case $\|X\|_2 \geq \|A\|_2/\alpha$. Thus $d_1 = 1/\|X\|_2$ is of modest size unless α is a severe overestimate of $\|A\|_2$. But since we can always take $\alpha = \|A\|_F \leq \sqrt{n}\|A\|_2$, we will have $d_1 \leq \sqrt{n}$. The same bound for d_1 holds to within ϵ for the computed $\widehat{X}$.

We conclude that in all cases (4.11) is satisfied with a modest d.

Note that the above analysis does not involve ℓ_0, suggesting that the estimate of $\sigma_{\min}(X_0)$ has no effect on the backward stability of QDWH. Of course, it does play a fundamental role in the speed of convergence.

The results of this subsection and the previous one combined with Theorem 4.2 prove that QDWH, implemented using Householder QR factorization with column pivoting and either row sorting or row pivoting, is backward stable.

5.3. Numerical experiments. To obtain insight into the analysis above we carried out an experiment using a set of 105 $n \times n$ matrices with $n = 10, 50, 100, 250$ drawn from the MATLAB `gallery` function and the Matrix Computation Toolbox [9], and including `gallery('randsvd')` matrices with 2-norm condition numbers $10^3, 10^6, 10^9, 10^{12}, 10^{15}$, and 5 different singular value distributions. This set excludes matrices with 2-norm condition numbers exceeding $u^{-1}/2$, since the QDWH

TABLE 5.1

Results for QDWH using no pivoting, column pivoting, and row sorting and column pivoting for 105 matrices. Each pair of numbers comprises the maximum values of $\|A - \widehat{U}\widehat{H}\|_F/\|A\|_F$, $\|\widehat{U}^*\widehat{U} - I\|_F/\sqrt{n}$ over all matrices of the particular dimension.

Pivoting	$n = 10$	$n = 50$	$n = 100$	$n = 250$
None	4.5e-15, 8.3e-16	2.3e-15, 8.7e-16	2.7e-15, 1.1e-15	8.3e-15, 1.7e-15
Col	1.2e-15, 1.2e-15	1.2e-15, 1.2e-15	1.9e-15, 1.7e-15	4.0e-15, 3.9e-15
Row & col	1.2e-15, 8.9e-16	1.2e-15, 1.1e-15	1.8e-15, 1.6e-15	3.5e-15, 3.5e-15

algorithm was originally designed for matrices of full rank (as are all the other iterations we discuss). We applied QDWH using three forms of pivoting in the QR factorization: no pivoting, column pivoting, and row sorting and column pivoting. Table 5.1 reports the relative residuals for the computed polar decomposition and a measure of the orthonormality of $\widehat{U}$. In addition we also computed the quantity $-\min\{\,\lambda_{\min}(\widehat{H}), 0\,\}/\|A\|_F$, which measures the closeness of $\widehat{H}$ to positive semidefiniteness; it had maximum value 6.1×10^{-17}. We see that the QDWH algorithm performs in a backward stable manner with row sorting and column pivoting, as expected. The algorithm also performs stably with column pivoting and with no pivoting. Indeed the algorithm as proposed in [20] uses QR factorization without pivoting and was observed there to perform in a backward stable way.

One reason that QWDH performs so well without pivoting can be seen from the structure of the matrix $\left[\begin{smallmatrix} \sqrt{c}X \\ I \end{smallmatrix}\right]$ in (3.5). From (5.6a) and (5.6b) we have $c \geq 3$ and c can be arbitrarily large. Therefore the matrix whose QR factorization we compute is already row-sorted in the block sense. Experiments show that swapping the order of the two blocks so that $\left[\begin{smallmatrix} I \\ \sqrt{c}X \end{smallmatrix}\right]$ is factorized makes the algorithm without pivoting unstable (and column pivoting does not help), even though it is mathematically equivalent to (3.5).

Despite the unexpectedly good stability of QDWH without pivoting, using the direct search optimization techniques described in [12], [13, Chap. 26] it is possible to generate matrices for which the algorithm is unstable: for example, we found a 6×6 matrix with $\kappa_2(A) \approx 10^{10}$ such that $\|A - \widehat{U}\widehat{H}\|_F/\|A\|_F = 9 \times 10^{-12}$.

The use of column pivoting is undesirable for high-performance computing because of its high communication costs. A possible remedy for instability when no pivoting is used is to generate a random unitary matrix W, form $B = AW$, compute the polar decomposition $B = UH$, and then recover the unitary polar factor from the polar decomposition $A = UW^* \cdot WHW^*$. This cures the instability in all the cases we have tried.

6. Backward stability of other polar decomposition iterations. Theorem 4.2 is sufficiently general that it can be used to investigate the backward stability of a wide range of polar decomposition iterations. In this section we use the theorem to show that the scaled Newton iteration is backward stable provided that matrix inverses are computed in a mixed backward–forward stable manner. This condition is not always satisfied if the inverses are computed in the usual way, by Gaussian elimination with partial pivoting, but it is if the inverses are computed by bidiagonalization, as shown by Byers and Xu [4].

We also give an explanation of why the scaled inverse Newton iteration [20], which is mathematically equivalent to the scaled Newton iteration, fails to be backward stable. Finally, we show that the Newton–Schulz iteration is stable away from, but not very close to, the boundary of its region of convergence.

6.1. The scaled Newton iteration is backward stable. The scaled Newton iteration is a well known and very effective method for computing the unitary polar factor of a nonsingular matrix $A \in \mathbb{C}^{n\times n}$ [11], [14]. The iteration has the form

$$X_{k+1} = \frac{1}{2}\left(\mu_k X_k + \mu_k^{-1} X_k^{-*}\right), \quad X_0 = A, \tag{6.1}$$

where $\mu_k > 0$ is a scaling factor. The optimal scaling $\mu_k^{\mathrm{opt}} = \left(\sigma_1(X_k)\sigma_n(X_k)\right)^{-1/2}$ minimizes a natural measure of error on each iteration and ensures convergence in at most n iterations [14, sect. 8.6], [16], but is too expensive to compute. Practical alternatives include the $(1,\infty)$-norm [11] and Frobenius-norm [6], [10] scalings

$$\mu_k^{1,\infty} = \left(\frac{\|X_k^{-1}\|_1 \|X_k^{-1}\|_\infty}{\|X_k\|_1 \|X_k\|_\infty}\right)^{1/4}, \tag{6.2}$$

$$\mu_k^{F} = \left(\frac{\|X_k^{-1}\|_F}{\|X_k\|_F}\right)^{1/2}, \tag{6.3}$$

and the suboptimal scaling in [4].

We will establish backward stability of the scaled Newton iteration for all the scaling strategies above by using Theorem 4.2. We need to show that the conditions (4.10) and (4.11) are both satisfied.

Proving (4.10). We assume that the inverses are computed by a mixed backward–forward stable algorithm, so that the computed $\widehat{Z} = fl(X^{-1})$ satisfies $\widehat{Z} = (X + \epsilon\|X\|_2)^{-1} + \epsilon\|\widehat{Z}\|_2 =: \widetilde{X}^{-1} + \epsilon\|\widehat{Z}\|_2$. Therefore the computed approximation $\widehat{Y}$ to $Y = \frac{1}{2}\left(\mu X + \mu^{-1}X^{-*}\right)$ satisfies

$$\begin{aligned}
\widehat{Y} &= \frac{1}{2}\big(\mu X + \mu^{-1}\widehat{Z}^*\big) + \epsilon \max\{\|\mu X\|_2, \|\mu^{-1}\widehat{Z}\|_2\} \\
&= \frac{1}{2}\big(\mu X + \mu^{-1}(\widetilde{X}^{-*} + \epsilon\|\widehat{Z}\|_2)\big) + \epsilon \max\{\|\mu X\|_2, \|\mu^{-1}\widehat{Z}\|_2\} \\
&= \frac{1}{2}\big(\mu(\widetilde{X} + \epsilon\|X\|_2) + \mu^{-1}\widetilde{X}^{-*}\big) + \epsilon \max\{\|\mu X\|_2, \|\mu^{-1}\widehat{Z}\|_2\} \\
&= \frac{1}{2}\big(\mu\widetilde{X} + \mu^{-1}\widetilde{X}^{-*}\big) + \epsilon \max\{\|\mu X\|_2, \|\mu^{-1}\widehat{Z}\|_2\}.
\end{aligned}$$

This implies (on considering the SVD of $\widetilde{X}$) that

$$\begin{aligned}
\|\widehat{Y}\|_2 &\geq \frac{1}{2}\max\{\|\mu\widetilde{X}\|_2, \mu^{-1}\|\widetilde{X}^{-1}\|_2\} + \epsilon \max\{\|\mu X\|_2, \|\mu^{-1}\widehat{Z}\|_2\} \\
&\approx \frac{1}{2}\max\{\|\mu X\|_2, \|\mu^{-1}\widehat{Z}\|_2\},
\end{aligned}$$

so we conclude that $\widehat{Y} = \frac{1}{2}\big(\mu\widetilde{X} + \mu^{-1}\widetilde{X}^{-*}\big) + \epsilon\|\widehat{Y}\|_2$. Hence the iteration (6.1) is evaluated in a mixed backward–forward stable manner.

Proving (4.11). The condition (4.11) for the scaled Newton iteration is $g(\mu x)/\max_{m\leq x\leq M} g(\mu x) \geq x/dM$ on $[m, M]$, where $g(x) = \frac{1}{2}(x + x^{-1})$, $m = \sigma_{\min}(\widetilde{X})$, and $M = \sigma_{\max}(\widetilde{X})$. Note that $\max_{m\leq x\leq M} g(x) = \max(g(\mu m), g(\mu M))$, because on a closed positive interval the function $g(\mu x)$ takes its maximum only at the endpoints. Hence we need to show

$$\frac{g(\mu x)}{\max(g(\mu m), g(\mu M))} \geq \frac{x}{dM} \quad \text{on} \quad [m, M], \tag{6.4}$$

for a modest constant $d \geq 1$. We distinguish two cases: $g(\mu m) \leq g(\mu M)$, which happens when $\mu \geq \mu_{\text{opt}}$, where $\mu_{\text{opt}} = (mM)^{-1/2}$ is the optimal scaling parameter which satisfies $\|\mu_{\text{opt}} X\|_2 = \|(\mu_{\text{opt}} X)^{-1}\|_2$, and $g(\mu m) > g(\mu M)$.

When $g(\mu m) \leq g(\mu M)$, the condition (6.4) becomes $\frac{g(\mu x)}{g(\mu M)} \geq \frac{x}{dM}$ on $[m, M]$, which can be rewritten as

$$\frac{g(\mu x)}{\mu x} \geq \frac{1}{d} \cdot \frac{g(\mu M)}{\mu M}, \qquad x \in [m, M].$$

This inequality holds with $d = 1$, because $\frac{g(\mu x)}{\mu x} = \frac{1}{2}(1 + (\mu x)^{-2})$ is a decreasing function on $(0, M]$ and equality holds when $x = M$.

In the second case (when $g(\mu m) > g(\mu M)$), similar arguments show that (6.4) is equivalent to the condition $\frac{g(\mu x)}{\mu x} \geq \frac{g(\mu m)}{d\mu M}$ on $[m, M]$. The function $\frac{g(\mu x)}{\mu x}$ takes its minimum on $[m, M]$ at $x = M$, at which

$$\frac{g(\mu M)}{\mu M} = \left(\frac{g(\mu M)}{g(\mu m)}\right) \frac{g(\mu m)}{\mu M},$$

so $\frac{g(\mu x)}{\mu x} \geq \frac{g(\mu m)}{d\mu M}$ holds with $d = \frac{g(\mu m)}{g(\mu M)}$. Hence backward stability can be lost only if $g(\mu M) \ll g(\mu m)$, which happens when $1/(\mu m) \gg \mu M$, or $\mu \ll (mM)^{-1/2} = \mu_{\text{opt}}$. We note that this danger of choosing μ too small was pointed out in [17]. Fortunately, for all the practical scaling strategies mentioned above, namely the $(1, \infty)$-norm (6.2) and Frobenius-norm (6.3) scalings and the suboptimal scaling in [4], $\mu \ll \mu_{\text{opt}}$ cannot occur. Indeed $\widetilde{X} = X + \epsilon\|X\|$ and the first two scalings differ from μ_{opt} by a factor at most $n^{1/4}$. The suboptimal scaling takes $\mu = (\widetilde{\sigma}_{\max}\widetilde{\sigma}_{\min})^{-1/2}$, where $\widetilde{\sigma}_{\max} \geq \sigma_{\max}$ and $\widetilde{\sigma}_{\min} \leq \sigma_{\min}$ are bounds for the extremal singular values (which are computed via scalar iterations after the first iteration). Since in practice we always have $\widetilde{\sigma}_{\max} \leq n^{1/2}\sigma_{\max}$, it follows that we always have $\mu \geq \mu_{\text{opt}}$.

Thus (4.11) holds in all cases. Since we have shown that the conditions (4.10) and (4.11) are both satisfied, we conclude that the scaled Newton iteration is backward stable under the assumption that the matrix inverses are computed in a mixed backward–forward stable way.

As noted in section 1, Kiełbasiński and Ziętak [19] point out some incompleteness of the analysis of Byers and Xu [4]. The main observation in [19] is that $\|U_k - U\|_2$ can be arbitrarily larger than ϵ when $\kappa_2(A) \gg 1$, where U_k and U are the unitary polar factors of $\widehat{X}_k$ and A, respectively, as in the proof of Theorem 4.2. The analysis in [4] uses $\|U_k - U\|_2 = \epsilon$, which may not hold. Our (more general) analysis in section 4 overcomes the issue because it does not refer to $\|U_k - U\|_2$; it shows that U_k yields a backward stable polar decomposition of A, even though U_k might be very different from U.

6.2. The inverse Newton iteration is not backward stable. Byers and Xu [3] and Nakatsukasa, Bai, and Gygi [20] observe that a QR-based implementation of the inverse Newton iteration, called QSNV in [20], is not backward stable. We can explain this instability by showing that d in (4.11) must be large when $\kappa_2(A) \gg 1$.

The iteration function for the scaled inverse Newton iteration is $f(x) = 2\mu x(1 + \mu^2 x^2)^{-1}$ (the inverse of the iteration function for the Newton iteration). For simplicity, suppose that the optimal scaling factor $\mu = (\sigma_{\min}(X)\sigma_{\max}(X))^{-1/2}$ is used. The condition (4.11) at $x = \sigma_{\max}(X)$ becomes $\max_{\sigma_{\min} \leq x \leq \sigma_{\max}} f(x)/f(\sigma_{\max}) \leq d$. Since $\max_{\sigma_{\min} \leq x \leq \sigma_{\max}} f(x) = f(1/\mu) = 1$, we need $d \geq 1/f(\sigma_{\max}) \approx \kappa_2(X)^{1/2}/2$, and this lower bound is large for ill conditioned matrices.

TABLE 6.1
Residual $\|A-\widehat{U}\widehat{H}\|_F/\|A\|_F$ and numbers of iteration for varying δ with initial matrices $X_0 = A$ and $X_0 = A/\|A\|_2$, for $A = P\mathrm{diag}(1, \sqrt{3}-\delta, \sqrt{3}-\delta)Q^*$ with orthogonal P and Q.

δ		0.5	10^{-2}	10^{-5}	10^{-10}	10^{-15}
$X_0 = A$	res	2.5e-16	5.8e-16	8.7e-13	2.1e-7	1.5e-3
	iter	6	15	32	61	90
$X_0 = \frac{A}{\|A\|_2}$	res	2.8e-16	2.5e-16	3.7e-16	1.7e-16	2.6e-16
	iter	6	7	7	7	7

Our analysis suggests that the instability is a fundamental feature of the inverse Newton iteration, independent of any particular implementation. Indeed numerical experiments show that even if row sorting and column pivoting is used in the Householder QR-based implementation of the inverse Newton iteration (note that pivoting was not considered in [3], [20]), instability is still present.

6.3. The Newton–Schulz iteration is conditionally backward stable. The Newton–Schulz iteration

$$X_{k+1} = \frac{1}{2}X_k(3I - X_k^*X_k) =: f(X_k), \qquad X_0 = A, \tag{6.5}$$

converges quadratically to the unitary polar factor for full rank $A \in \mathbb{C}^{s\times n}$ ($s \geq n$) with $\|A\|_2 < \sqrt{3}$ [14, sect. 8.3]. To make the iteration converge for a general full rank A we can simply change X_0 to $X_0 = A/\alpha$, where $\alpha > \|A\|_2/\sqrt{3}$. Higham and Schreiber [15] propose an algorithm that starts with the scaled Newton iteration and switches to the Newton–Schulz iteration once fast convergence of the latter is ensured. The Newton–Schulz is also of practical interest as a tool to improve the numerical orthogonality of the computed unitary polar factor. For example, in the experiments in Table 5.1, running one Newton–Schulz iteration on the converged $\widehat{X}_\ell$ as a postprocessing step yielded a polar decomposition with $\|\widehat{U}^*\widehat{U} - I\|_F/\sqrt{n} \leq 5.5 \times 10^{-16}$ in every case (it also typically improves the backward errors, but to a lesser extent).

We note first that Newton–Schulz converges slowly when $\|X_0\|_2 = \sqrt{3} - \delta$ with $0 < \delta \ll 1$ or $\kappa_2(X_0) \gg 1$. In the former case, X_1 has a small singular value $f(\|X_0\|_2) \approx 3\delta$, which needs many iterations to converge to 1. When $\kappa_2(X_0) \gg 1$, X_0 must have a small singular value (given that $\|X_0\|_2 < \sqrt{3}$), so again many iterations are needed.

Consider 3×3 matrices of the form $A = P\mathrm{diag}(1, \sqrt{3}-\delta, \sqrt{3}-\delta)Q^*$, where P and Q are random orthogonal matrices. They are very well conditioned, with $\kappa_2(A) \leq \sqrt{3}$ for $\delta \leq 0.5$. Table 6.1 shows relative residuals for two choices of starting matrix: $X_0 = A$ and $X_0 = A/\|A\|_2$. With $X_0 = A/\|A\|_2$ we observe small backward error and convergence within seven iterations. However, with $X_0 = A$ we see a large number of iterations and instability. The Newton–Schulz iteration is clearly unstable close to the boundary of its region of convergence.

We can explain this behavior using Theorem 4.2. Specifically, we prove that the Newton–Schulz iteration is backward stable if $\|X_0\|_2$ is safely less than $\sqrt{3}$, but can be unstable if $\|X_0\|_2 = \sqrt{3} - \delta$ for $0 < \delta \ll 1$.

First we show the mixed backward–forward stability condition (4.10) always holds for any $\|X_0\|_2 < \sqrt{3}$. We consider the computed approximation $\widehat{Y}$ to $Y = \frac{1}{2}X(3I - X^*X)$, where $\|X\|_2 < \sqrt{3}$ (this bound holds in all iterations; in fact $\|X_k\|_2 \leq 1$ for $k \geq 1$). If we compute $\widehat{Y}$ by first forming XX^*X, then performing the subtraction

(other ways of computing $\widehat{Y}$ yield similar results), we have

$$\widehat{Y} = \frac{3}{2}X - \frac{1}{2}(XX^*X + \epsilon_1) + \epsilon_2, \tag{6.6}$$

where ϵ_1 represents the forward error in forming XX^*X and ϵ_2 is the error from the subtraction. Since $\|X\|_2 < \sqrt{3}$ and $\|XX^*X\|_2 = \|X\|_2^3 < 3\sqrt{3}$ we can write $\epsilon_1 = \epsilon\|X\|_2$ and $\epsilon_2 = \epsilon\|X\|_2$, so overall $\widehat{Y} = \frac{1}{2}X(3I - X^*X) + \epsilon\|X\|_2 = \frac{1}{2}X(3I - X^*X) + \epsilon\|\widehat{Y}\|_2$, that is, Y is computed with a small forward error. Hence (4.10) is satisfied.

We next investigate the condition (4.11) on the mapping function $f(x) = \frac{1}{2}x(3 - x^2)$. The condition for $k = 0$ is $f(x)/x \geq \max_{m\leq x\leq M} f(x)/(dM)$ for $x \in [m, M]$, where $m = \sigma_{\min}(\widetilde{X}_0)$ and $M = \sigma_{\max}(\widetilde{X}_0)$. Since $f(x)/x$ is a decreasing function on $[0, \sqrt{3}]$ the condition becomes $f(M)/M \geq \max_{m\leq x\leq M} f(x)/(dM)$, that is, $d \geq \max_{m\leq x\leq M} f(x)/f(M)$. Since $M = \sigma_{\max}(X_0) + \epsilon$ and f is increasing on $[0, 1]$, we have $d = 1$ for $\|X_0\|_2 < 1$. If $\|X_0\|_2 \geq 1$ then d is certainly of modest size if $\|X_0\|_2$ is not too close to $\sqrt{3}$. If $\|X_0\|_2 = \sqrt{3} - \delta$ with $0 < \delta \ll 1$ then $d \approx 1/(3\delta) \gg 1$. For the matrix $X_0 = A$ in Table 6.1, $u/(3\delta)$ is a reasonable estimate for the backward errors in the first row of the table. Our conclusion, from Theorem 4.2, is that Newton–Schulz is stable if $\|X_0\|_2$ is safely less than $\sqrt{3}$, but that for $\|X_0\|_2 \approx \sqrt{3}$ it can be unstable. Note that the instability arises only on the first iteration, because $\|X_k\|_2 \leq 1$ for $k \geq 1$.

It is natural to consider using scaling $X_k \leftarrow \gamma_k X_k$ during the Newton–Schulz iteration, just as in the scaled Newton iteration. The optimal scaling factor γ_k satisfies $\|\gamma_k X_k\|_2 = \sqrt{3}\kappa_2(X_k)/\sqrt{1 + \kappa_2(X_k) + \kappa_2(X_k)^2}$, because it maximizes $\sigma_{\min}(X_{k+1})$ and minimizes $\kappa_2(X_{k+1})$, which we can verify through a detailed analysis of $f(x)$. Unfortunately, using this optimal scaling results in an unstable iteration, because such a γ_k yields $\|\gamma_k X_k\|_2 \approx \sqrt{3}$ if X_k is ill conditioned.

Our conclusion is that in order for Newton–Schulz to combine stability with fast convergence we need X_0 to have norm safely less than $\sqrt{3}$ and to be not too ill conditioned (which is the case for its usage in [15]).

Acknowledgment. We thank the referees for their helpful comments and suggestions.

REFERENCES

[1] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz, *Minimizing communication in numerical linear algebra*, SIAM J. Matrix Anal. Appl., 32 (2011), pp. 866–901.

[2] R. Bhatia, *Matrix Analysis*. Springer-Verlag, New York, 1997.

[3] R. Byers and H. Xu, *An Inverse Free Method for the Polar Decomposition*, unpublished manuscript, 2001.

[4] R. Byers and H. Xu, *A new scaling for Newton's iteration for the polar decomposition and its backward stability*, SIAM J. Matrix Anal. Appl., 30 (2008), pp. 822–843.

[5] A. J. Cox and N. J. Higham, *Stability of Householder QR factorization for weighted least squares problems*, in Numerical Analysis 1997, in Proceedings of the 17th Dundee Biennial Conference, D. F. Griffiths, D. J. Higham, and G. A. Watson, eds., Pitman Res. Notes Math. Ser. 380, Longman, Harlow, UK, 1998, pp. 57–73.

[6] A. A. Dubrulle, *An optimum iteration for the matrix polar decomposition*, Electron. Trans. Numer. Anal., 8 (1999), pp. 21–25.

[7] K. Fan and A. J. Hoffman, *Some metric inequalities in the space of matrices*, Proc. Amer. Math. Soc., 6 (1955), pp. 111–116.

[8] B. F. Green, *The orthogonal approximation of an oblique structure in factor analysis*, Psychometrika, 17 (1952), pp. 429–440.

[9] N. J. Higham, *The Matrix Computation Toolbox*, http://www.ma.man.ac.uk/∼higham/mctoolbox.
[10] N. J. Higham, *Nearness Problems in Numerical Linear Algebra*, Ph.D. thesis, University of Manchester, Manchester, England, 1985.
[11] N. J. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 1160–1174.
[12] N. J. Higham, *Optimization by direct search in matrix computations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 317–333.
[13] N. J. Higham, *Accuracy and Stability of Numerical Algorithms*, 2nd ed., SIAM, Philadelphia, 2002.
[14] N. J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, Philadelphia, 2008.
[15] N. J. Higham and R. S. Schreiber, *Fast polar decomposition of an arbitrary matrix*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 648–655.
[16] C. S. Kenney and A. J. Laub, *On scaling Newton's method for polar decomposition and the matrix sign function*, SIAM J. Matrix Anal. Appl., 13 (1992), pp. 688–706.
[17] A. Kiełbasiński, P. Zieliński, and K. Ziętak, *Numerical Experiments with Higham's Scaled Method for Polar Decomposition*, Technical Report I18/2006/P-013, Institute of Mathematics and Computer Science, Wroclaw University of Technology, Wroclaw, Poland, 2006.
[18] A. Kiełbasiński and K. Ziętak, *Numerical behaviour of Higham's scaled method for polar decomposition*, Numer. Algorithms, 32 (2003), pp. 105–140.
[19] A. Kiełbasiński and K. Ziętak, *Note on "A new scaling for Newton's iteration for the polar decomposition and its backward stability" by R. Byers and H. Xu*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 1538–1539.
[20] Y. Nakatsukasa, Z. Bai, and F. Gygi, *Optimizing Halley's iteration for computing the matrix polar decomposition*, SIAM J. Matrix Anal. Appl., 31 (2010), pp. 2700–2720.
[21] Y. Nakatsukasa and N. J. Higham, *Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD*, MIMS EPrint, Manchester Institute for Mathematical Sciences, The University of Manchester, UK, 2012, in preparation.