HEWLETT PACKARD

Frobenius Iteration for the Matrix Polar Decomposition

Augustin A. Dubrulle
Computer Research Center

HPL-94-117
December, 1994

matrix computations,
polar decomposition,
singular values,
Newton's method,
arithmetic-harmonic-mean iteration

Higham's iterative computation of the matrix polar decomposition uses an acceleration parameter derived from economical approximations of the l_2 norm and nearly optimum for that norm. It is shown here that the iteration based on a parameter optimum for the Frobenius norm converges as fast as the l_2 iteration and lends itself to more efficient implementation and easier iteration control. The description of a practical algorithm is included for illustration.

FROBENIUS ITERATION FOR THE MATRIX POLAR DECOMPOSITION

AUGUSTIN A. DUBRULLE
Hewlett-Packard Laboratories
1501 Page Mill Road
Palo Alto, CA 94304
dubrulle@hpl.hp.com

December 1994

ABSTRACT

Higham's iterative computation of the matrix polar decomposition uses an acceleration parameter derived from economical approximations of the ℓ_2 norm and nearly optimum for that norm. It is shown here that the iteration based on a parameter optimum for the Frobenius norm converges as fast as the ℓ_2 iteration and lends itself to more efficient implementation and easier iteration control. The description of a practical algorithm is included for illustration.

1 Introduction

In [3], Higham describes a simple and effective algorithm for the computation of the orthogonal factor of the matrix polar decomposition. Although not as efficient in general as the approach based on the singular-value decomposition [2], this algorithm is useful in the not uncommon cases where the matrix of the problem is not far from orthogonality. It is based on a matrix version of Newton's iteration with acceleration that minimizes the largest singular value of a matrix and maps it into the largest singular value of its iterate. The convergence of this "ℓ_2 iteration" is monotonic, a property especially desirable in the context of software development. Unfortunately, the determination of the optimum acceleration parameter requires evaluations of matrix ℓ_2 norms, which is not a simple problem. A few ways to approximate these norms prove to be adequate in practice, but without guarantee of monotonicity.

In [5], Kenney and Laub report satisfactory results from experiments with a rough approximation to the ℓ_2 acceleration parameter obtained by substituting Frobenius norms for ℓ_2 norms. It is shown here that there are good reasons for these results to be satisfactory, namely that this Frobenius iteration (1) is optimum in its own right, (2) is monotonic for the associated norm, and (3) converges as fast as the ℓ_2 iteration. Since the Frobenius norm is easily computed, monotonicity can be exploited for software implementation. The material is organized as follows. Section 2 summarizes previous work and basic properties of Newton's iteration with acceleration. Section 3 provides an analysis of the Frobenius iteration, including a proof that it converges as fast as its ℓ_2 counterpart. Finally, an efficient algorithm for software development is discussed in Section 4.

The numerical experiments for this paper were conducted on a Hewlett-Packard Vectra VL2 personal computer with MATLAB (Student Version) and APLi386 from Iverson Software Inc.

2 Background

The iterative computation of the orthogonal polar factor of a nonsingular matrix $\boldsymbol{A} \in \mathcal{R}^{n\times n}$ by Newton's iteration [3] is defined by

$$\boldsymbol{X}^{(0)} = \boldsymbol{A}, \qquad \boldsymbol{X}^{(k+1)} = \frac{1}{2}\left(\boldsymbol{X}^{(k)} + \boldsymbol{X}^{(k)-T}\right). \tag{2.1}$$

$\boldsymbol{X}^{(k)}$ converges to the unique orthogonal matrix $\boldsymbol{X}$ such that

$$\boldsymbol{A} = \boldsymbol{X}\boldsymbol{M}, \qquad \boldsymbol{X}^T = \boldsymbol{X}^{-1}, \qquad \boldsymbol{M} = \boldsymbol{M}^T, \quad \boldsymbol{w}^T\boldsymbol{M}\boldsymbol{w} > 0 \quad \forall \boldsymbol{w} \neq \boldsymbol{0}.$$

The iteration transforms the singular values of $\boldsymbol{X}^{(k)}$ according to

$$\sigma_j^{(k+1)} = \frac{1}{2}\left(\sigma_j^{(k)} + \frac{1}{\sigma_j^{(k)}}\right), \qquad i = 1, \ldots, n, \tag{2.2}$$

and leaves the singular vectors invariant. From equation (2.2), it is easy to show that convergence of the singular values to unity is quadratic, monotonic, and with preservation of order for $k \geq 1$:

$$1 \leq \sigma_j^{(k+1)} \leq \sigma_j^{(k)} \qquad \forall j \qquad k \geq 1,$$

$$\sigma_j^{(k+1)} - 1 = \frac{1}{2\sigma_j^{(k)}}\left(\sigma_j^{(k)} - 1\right)^2,$$

$$\sigma_i^{(k)} \leq \sigma_j^{(k)} \quad \Rightarrow \quad \sigma_i^{(k+1)} \leq \sigma_j^{(k+1)}.$$

Iteration (2.2) is the arithmetic-harmonic-mean algorithm applied to reciprocal initial values [1]. The distance between two successive matrix iterates is directly related to departure from orthogonality:

$$\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)} = \frac{1}{2}\left(\boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k)-T}\right).$$

Factoring the right-hand side to extract $\boldsymbol{X}^{(k)}$ and taking norms, we get

$$\frac{\| \boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k+1)} \|}{\| \boldsymbol{X}^{(k)} \|} \leq \frac{1}{2} \| \boldsymbol{I} - (\boldsymbol{X}^{(k)T} \boldsymbol{X}^{(k)})^{-1} \|.$$

Kenney and Laub [5] show that

$$\| \boldsymbol{X}^{(k)} - \boldsymbol{X} \|_2 \leq \| \boldsymbol{X}^{(k)} - \boldsymbol{X}^{(k)-T} \|_2,$$

where $\boldsymbol{X}$ is the limit of $\boldsymbol{X}^{(k)}$. This inequality allows for the computation of bounds on the distance of the iterate to the solution.

To accelerate convergence, scaling can be used at each iteration,

$$\boldsymbol{X}^{(k+1)} = \frac{1}{2} \left[\gamma^{(k)} \boldsymbol{X}^{(k)} + (\gamma^{(k)} \boldsymbol{X}^{(k)})^{-T} \right], \qquad \gamma_k > 0,$$

and the equation for the singular values becomes:

$$\sigma_j^{(k+1)} = \frac{1}{2} \left(\gamma^{(k)} \sigma_j^{(k)} + \frac{1}{\gamma^{(k)} \sigma_j^{(k)}} \right), \qquad i = 1, \ldots, n.$$

The salient properties of this modified iteration are summarized below.

The singular values converge to unity from above after the first iteration,

$$\sigma_j^{(k)} \geq 1 \qquad \forall\, k \geq 1,$$

and those satisfying the inequality

$$\sigma_j^{(k)} \geq \frac{1}{\sqrt{\gamma^{(k)}}} \geq 1 \tag{2.3}$$

enjoy accelerated convergence:

$$\frac{1}{2\gamma^{(k)} \sigma_j^{(k)}} \left(\gamma^{(k)} \sigma_j^{(k)} - 1 \right)^2 \leq \frac{1}{2\sigma_j^{(k)}} \left(\sigma_j^{(k)} - 1 \right)^2.$$

Finally, the order of a pair of singular values $\{\sigma_i^{(k)}, \sigma_j^{(k)}\}$ is preserved by the iteration under the following condition:

$$\sigma_i^{(k)} \sigma_j^{(k)} \geq \frac{1}{\gamma^{(k)2}} \quad \wedge \quad \sigma_i^{(k)} \geq \sigma_j^{(k)} \qquad \Rightarrow \qquad \sigma_i^{(k+1)} \geq \sigma_j^{(k+1)}.$$

Using this property for

$$\sigma_{max}^{(k)} = \| \boldsymbol{X}^{(k)} \|_2, \qquad \sigma_{min}^{(k)} = \| \boldsymbol{X}^{(k)-T} \|_2^{-1},$$

we get the condition for the iteration to map the largest singular value of $\boldsymbol{X}^{(k)}$ into that of $\boldsymbol{X}^{(k+1)}$:

$$\gamma^{(k)} \geq \frac{1}{\sqrt{\sigma_{max}^{(k)}\,\sigma_{min}^{(k)}}}.$$

From inequality (2.3), such a value of $\gamma^{(k)}$ also produces accelerated convergence for the singular values not smaller than $\sqrt[4]{\sigma_{max}^{(k)}\,\sigma_{min}^{(k)}}$. In particular, the choice

$$\gamma_2^{(k)} = \frac{1}{\sqrt{\sigma_{max}^{(k)}\,\sigma_{min}^{(k)}}}, \tag{2.4}$$

which minimizes the ℓ_2 bound

$$\| \boldsymbol{X}^{(k+1)} \|_2 \leq \frac{1}{2}\left(\gamma^{(k)} \| \boldsymbol{X}^{(k)} \|_2 + \frac{1}{\gamma^{(k)}} \| \boldsymbol{X}^{(k)-T} \|_2\right),$$

results in monotonic quadratic convergence for the ℓ_2 norm of the iterate. This "ℓ_2 iteration" conflates the smallest and largest singular values of $\boldsymbol{X}^{(k)}$ into the largest singular value of $\boldsymbol{X}^{(k+1)}$,

$$\sigma_{max}^{(k+1)} = \frac{1}{2}\left(\sqrt{\frac{\sigma_{max}^{(k)}}{\sigma_{min}^{(k)}}} + \sqrt{\frac{\sigma_{min}^{(k)}}{\sigma_{max}^{(k)}}}\right), \tag{2.5}$$

It follows that the limit, which is characterized by the equality of all the singular values, is reached in at most n iterations [5]. For large matrices, this bound is grossly pessimistic, as shown by a finer analysis of convergence [5] and experiments with simple estimates of the optimum (2.4). It is easy to show from equation (2.5) that the ℓ_2 iteration is monotonic for the associated norm:

$$\sigma_{max}^{(k+1)} \leq \sigma_{max}^{(k)}.$$

The implementation in [3], which uses the approximation

$$\tilde{\gamma}_2^{(k)} = \sqrt[4]{\frac{\| \boldsymbol{X}^{(k)-T} \|_1 \, \| \boldsymbol{X}^{(k)-1} \|_\infty}{\| \boldsymbol{X}^{(k)} \|_1 \, \| \boldsymbol{X}^{(k)} \|_\infty}} \tag{2.6}$$

to $\gamma_2^{(k)}$, delivers an IEEE double-precision solution in about ten or fewer iterations. One can explain this behavior by using a bound in [5] on the distance of the ℓ_2 iterate to the limit for invertible matrices with maximum condition number.

The condition

$$\| \boldsymbol{X}^{(k+1)} - \boldsymbol{X}^{(k)} \|_1 \leq \delta \, \| \boldsymbol{X}^{(k)} \|_1 \tag{2.7}$$

terminates the computation, where $\delta > 0$ is a constant of the order of machine precision. In the following, we refer to this iteration as the "quasi-ℓ_2" iteration.

In practice, little speed of convergence is lost by using an approximate form of the ℓ_2 iteration. What is lost is the important property of monotonicity, which is very useful for iteration control. Unfortunately, a true ℓ_2 implementation of the iteration is not practical at all, and the design of a quasi-ℓ_2 iteration preserving monotonicity seems far from trivial. No such impediment exists for the Frobenius iteration analyzed in the next section.

3 Iteration in the Frobenius norm

In [5], Kenney and Laub report good results with a Frobenius approximation of the ℓ_2 optimum value of the acceleration parameter $\gamma_2^{(k)}$. It is shown in this section that this approximation is actually an optimal choice for the Frobenius norm, and that the corresponding iteration converges as fast as the ℓ_2 iteration.

By squaring equation (2.2),

$$\sigma_j^{(k+1)2} = \frac{1}{4}\left(2 + \gamma^{(k)2}\sigma_j^{(k)2} + \frac{1}{\gamma^{(k)2}\sigma_j^{(k)2}}\right),$$

and summing with respect to j, we get

$$\| \boldsymbol{X}^{(k+1)} \|_F^2 = \frac{1}{4}\left(2n + \gamma^{(k)2} \| \boldsymbol{X}^{(k)} \|_F^2 + \frac{1}{\gamma^{(k)2}} \| \boldsymbol{X}^{(k)-T} \|_F^2\right).$$

Hence, the Frobenius norm of the iterate is minimized by the choice

$$\gamma_F^{(k)} = \sqrt{\frac{\| \boldsymbol{X}^{(k)-T} \|_F}{\| \boldsymbol{X}^{(k)} \|_F}} \tag{3.1}$$

for the acceleration parameter $\gamma^{(k)}$, which is the value used in [5] as an approximation to $\gamma_2^{(k)}$. The associated norm of the iterate is given by

$$\| \boldsymbol{X}^{(k+1)} \|_F^2 = \frac{1}{2}\left(n + \| \boldsymbol{X}^{(k)} \|_F \, \| \boldsymbol{X}^{(k)-T} \|_F\right). \tag{3.2}$$

We shall refer to this iteration as the "Frobenius iteration." Before getting any further in its analysis, we derive the following useful inequalities:

$$\left.\begin{aligned}
&\sigma_i^{(k)} \geq 1 \quad \forall\, i \\
&\sqrt{n}\sigma_{min}^{(k)} \leq \| \boldsymbol{X}^{(k)} \|_F \leq \sqrt{n}\sigma_{max}^{(k)}, \\
&\frac{\sqrt{n}}{\sigma_{max}^{(k)}} \leq \| \boldsymbol{X}^{(k)-T} \|_F \leq \frac{\sqrt{n}}{\sigma_{min}^{(k)}}, \\
&n \leq \| \boldsymbol{X}^{(k)-T} \|_F \, \| \boldsymbol{X}^{(k)} \|_F \leq n\frac{\sigma_{max}^{(k)}}{\sigma_{min}^{(k)}},
\end{aligned}\right\} \quad \forall\, k \geq 1.$$

Using these inequalities and equation (3.2), we establish monotonic convergence:

$$n \leq \| \boldsymbol{X}^{(k+1)} \|_F^2 \leq \frac{1}{2}\left(n + \| \boldsymbol{X}^{(k)} \|_F^2\right) \leq \| \boldsymbol{X}^{(k)} \|_F^2.$$

Numerical experiments comparing the Frobenius and ℓ_2 iterations indicate that the rates of convergence are practically identical. The analysis below, which explains such observations, shows that the two algorithms have essentially the same properties and differ only by details of metrics with little impact on practical computations. For example, the ℓ_2 iteration minimizes the maximum singular value of its iterate, while the Frobenius iteration minimizes the "mean singular value" derived from the associated norm (3.2):

$$\frac{\| \boldsymbol{X}^{(k+1)} \|_F}{\sqrt{n}} = \frac{1}{\sqrt{2}}\left(1 + \frac{\| \boldsymbol{X}^{(k)} \|_F}{\sqrt{(n)}} \frac{\| \boldsymbol{X}^{(k)-T} \|_F}{\sqrt{(n)}}\right)^{1/2}. \tag{3.3}$$

There is also much similarity to be found in the formulas expressing the norms of both types of iterates. In the following, we show that the mean singular value of the Frobenius iterate converges "as fast" as the largest singular value of the ℓ_2 iterate, as defined in the following theorem.

Theorem 3.1 *The mean singular value (3.3) of the Frobenius iterate of a matrix is bounded above by the maximum singular value of the ℓ_2 iterate of the same matrix.*

Proof Let $\boldsymbol{Y} \in \mathcal{R}^{n \times n}$ be a given matrix with singular values $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 1$, to which the Frobenius and ℓ_2 iterations are applied. From equations (3.2) and (2.5), the theorem is expressed by the inequality

$$\frac{1}{n^2}\sum_{i=1}^{n}\frac{1}{\sigma_i^2}\sum_{j=1}^{n}\sigma_j^2 \leq \frac{1}{4}\left(\frac{\sigma_1}{\sigma_n} + \frac{\sigma_n}{\sigma_1}\right)^2. \tag{3.4}$$

We first derive an upper bound for the left-hand side of this inequality. For the reciprocal of the Frobenius acceleration parameter

$$\lambda = \sqrt{\frac{\| \boldsymbol{Y} \|_F}{\| \boldsymbol{Y}^{-T} \|_F}}, \qquad \sigma_n \leq \lambda \leq \sigma_1,$$

we note that the replacement of any singular value smaller than λ with a lower value increases the left-hand side of inequality (3.4), and that the replacement of any singular value greater than λ with a larger value has a similar effect. Based on this property, we obtain our upper bound by substituting σ_n for the m singular values less than λ and σ_1 for the others[1], which yields a sufficient condition for inequality (3.4):

$$\frac{1}{n^2}\left(\frac{m}{\sigma_n^2} + \frac{n-m}{\sigma_1^2}\right)\left[m\sigma_n^2 + (n-m)\sigma_1^2\right] \leq \frac{1}{4}\left(\frac{\sigma_1}{\sigma_n} + \frac{\sigma_n}{\sigma_1}\right)^2 .$$

A little algebra reduces this condition to

$$2(n-2m)^2 \leq (n-2m)^2 \left(\frac{\sigma_1^2}{\sigma_n^2} + \frac{\sigma_n^2}{\sigma_1^2}\right)$$

which is always verified since $\dfrac{\sigma_1^2}{\sigma_n^2} + \dfrac{\sigma_n^2}{\sigma_1^2} \geq 2$. ■

In sum, the Frobenius and ℓ_2 norms converge equally fast under the Frobenius and ℓ_2 iterations, and both iterations are optimum for their own metrics. In practice, the substantial advantage of the Frobenius iteration is found in the low-cost computability of the associated norm, which allows properties such as monotonicity to be used for efficient and precise iteration control.

4 Implementation

Without loss of generality, we assume that $\boldsymbol{A} \in \mathcal{R}^{n \times n}$ is nonsingular—a singularity could be handled as suggested in [4]. Using the same notation as in the previous sections, we describe the implementation of the Frobenius iteration as follows, where ε designates machine precision:

Algorithm 4.1 *Frobenius iteration ($k > 1$)*

Step 1: QR decomposition $\boldsymbol{X}^{(k)} = \boldsymbol{Q}^{(k)}\boldsymbol{R}^{(k)}$

Step 2: if $\| \boldsymbol{R}^{(k)} \|_F \geq \| \boldsymbol{R}^{(k-1)} \|_F$ *or*

[1] These substitutions change λ but not m, which is the variable pertinent to the proof.

if $\| \boldsymbol{R}^{(k)} \|_F \leq (1+\varepsilon)\sqrt{n}$, *end the computation*

otherwise, proceed to Step 3

Step 3: $\gamma_F^{(k)} = \sqrt{\dfrac{\| \boldsymbol{R}^{(k)-T} \|_F}{\| \boldsymbol{R}^{(k)} \|_F}}$,

$$\boldsymbol{X}^{(k+1)} = \frac{1}{2}\left(\gamma_F^{(k)} \boldsymbol{X}^{(k)} + \boldsymbol{Q}^{(k)} \frac{1}{\gamma_F^{(k)}} \boldsymbol{R}^{(k)-T}\right)$$

return to Step 1.

In *Step 2*, the computation is ended when the norm of the iterate no longer decreases, thereby guaranteeing termination independently of machine arithmetic, and precisely when no further improvement should be expected. The iteration is also terminated when the norm of the iterate is sufficiently close to the limit, which usually saves one iteration. These tests are bypassed for the first iteration ($k = 1$) unless the singular values of $\boldsymbol{A}$ are known to be greater than unity.

To reduce computational overhead, all Frobenius norms are evaluated as norms of triangular matrices since

$$\| \boldsymbol{R}^{(k)} \|_F = \| \boldsymbol{X}^{(k)} \|_F, \qquad \| \boldsymbol{R}^{(k)-T} \|_F = \| \boldsymbol{X}^{(k)-T} \|_F.$$

$\boldsymbol{Q}^{(k)}$, which results from Householder triangularization, is not actually computed, but is represented by the $(n-1)$ vectors that define the associated elementary reflectors.

Numerical experiments were performed to compare the Frobenius and quasi-ℓ_2 algorithms for a wide variety of matrices with assigned singular values, including those cited in [3] and [5]. They show that the former is in general more economical by one iteration because the distance between successive iterates in test (2.7) substantially overestimates the distance to the limit. For $\delta = n\varepsilon$, both methods deliver results accurate to about machine precision, as measured by $\| \boldsymbol{I} - \boldsymbol{X}^T \boldsymbol{X} \|$ and $\| \boldsymbol{X}^T \boldsymbol{A} - \boldsymbol{A}^T \boldsymbol{X} \| \, \| \boldsymbol{X}^T \boldsymbol{A} \|^{-1}$.

Finally, the Frobenius iteration was found preferable from a viewpoint of memory management because its convergence test does not require access to two successive matrix iterates.

5 Conclusion

The Frobenius iteration presented here is a much preferable substitute for quasi-ℓ_2 iterations, for reasons both aesthetic and practical: it converges as fast as a true ℓ_2 iteration and enjoys the same properties of monotonicity without the computational complexity. Its use of monotonicity rather than negligibility as a criterion for iteration control is intellectally more pleasing, easier to implement in software applications, and more effective.

Although the implementation recommended in Section 4 does not substantially change the order of the operation count of current practice, it still reduces computational complexity and overhead, simplifies program development and usage, and restricts the number of iterations to the necessary minimum. In addition, it easily lends itself to the type of modification for rank-deficient matrices discussed in [4].

References

[1] J. Borwein and P. Borwein, *π and the AGM—A study in Analytic Number Theory and Computational Complexity*, Wiley, New York, 1987.

[2] G. Golub and C. Van Loan, *Matrix Computations*, The Johns Hopkins University Press, Baltimore MD, 1989.

[3] N. Higham, *Computing the polar decomposition—with applications*, SIAM J. Sci. Stat. Comput., 7(4):1160–1174, 1986.

[4] N. Higham and R. Schreiber, *Fast polar decomposition of an arbitrary matrix*, SIAM J. Sci. Stat. Comput., 11(4):648–655, 1990.

[5] C. Kenney and A. Laub, *On scaling Newton's method for the polar decomposition and the matrix sign function*, SIAM J. Mat. Anal. Appl., 13(3):688–706, 1992.