Project: Computing the Polar Decomposition

Nicholas J. Higham*

October 26, 2022

The aim of this project is to do some basic numerical linear algebra, programming in MATLAB, and LaTeX. You should prepare a LaTeX document summarizing all this work and including the MATLAB M-file listings.

Sources of Information

For the polar decomposition and its computation: Higham [2], [3], Higham and Papadimitriou [4]. For the SVD, norms, etc.: Golub and Van Loan [1], Stewart [7] or Watkins [8].

Tasks

Throughout, $A \in \mathbb{C}^{n\times n}$.

1. Familiarize yourself with the polar decomposition $A = UH$ and its properties (from the above references).

2. Prove that the singular values of A are the eigenvalues of H.

3. Prove that A is normal ($A^*A = AA^*$) iff U and H commute.

4. Verify the formula
$$U = \frac{2}{\pi} A \int_0^\infty (t^2 I + A^*A)^{-1} dt$$
for full rank A by using the singular value decomposition (SVD) of A to diagonalize the formula.

5. Derive Newton's method for computing U by considering the equations $(X + E)^*(X + E) = I$, where E is a "small perturbation". (Newton's method is $X_{k+1} = (X_k + X_k^{-*})/2$, $X_0 = A$.)

6. Prove that Newton's method converges, and at a quadratic rate, by using the SVD of A.

*School of Mathematics, University of Manchester, Manchester, M13 9PL, UK (nick.higham@manchester.ac.uk).

7. (More difficult). Use the SVD to analyze the convergence of the Newton–Schulz iteration for computing U:

$$X_{k+1} = \tfrac{1}{2}X_k(3I - X_k^* X_k), \qquad X_0 = A.$$

Note that this iteration uses only matrix multiplication.

8. Evaluate the operation count for one step of Newton's method and one step of the Newton–Schulz iteration (taking account of symmetry). Ignoring operation counts, how much faster does matrix multiplication have to be than matrix inversion for Newton–Schulz to be faster than Newton (assuming both take the same number of iterations)?

9. Write a MATLAB M-file `poldec` that computes the polar decomposition of a nonsingular $A \in \mathbb{C}^{n\times n}$ as follows: it starts with the Newton iteration and switches to the Newton–Schulz iteration when the latter iteration "is certain" to converge (the motivation being that matrix multiplication is very fast on high-performance computers). You will have to choose a suitable convergence test.

   The first lines of the M-file should be exactly as follows:

```
function [U, H, its] = poldec(A)
%POLDEC    Polar decomposition.
%   [U, H, ITS] = poldec(A) computes the polar decomposition A = U*H
%   of the square, nonsingular matrix A.  ITS is the number of
%   iterations for convergence.
```

   On each iteration print out $\|X_k - X_{k-1}\|_\infty / \|X_k\|_\infty$ and $\|I - X_k^* X_K\|_\infty$.

   Test your M-file on these matrices, and comment on the results:

   (a) `randn(n)` for various n to get the routine working.

   (b) `eye(8)`

   (c) `hilb(6)`

   (d) `magic(6)`

   (e) `hadamard(8)`

10. Write another routine that computes the square root of a symmetric positive definite matrix by doing a Cholesky decomposition and calling `poldec`.

11. Read [5] and [6] to get up to date with some recent developments.

Format

- When preparing the LaTeX document please follow the "Some specific suggestions" under the LaTeX heading at https://nla-group.org/need-to-know/, and in particular use the LaTeX template there.

- Keep all the files in a Git repository and commit to it after each major change.

- Create a GitHub account and push the repository there (you can do this once everything is finished if you want).

- Hand in by giving me the link to your GitHub repository.

References

[1] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. Fourth edition, Johns Hopkins University Press, Baltimore, MD, USA, 2013. xxi+756 pp. ISBN 978-1-4214-0794-4.

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

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

[4] Nicholas J. Higham and Pythagoras Papadimitriou. A parallel algorithm for computing the polar decomposition. *Parallel Comput.*, 20(8):1161–1173, 1994.

[5] Yuji Nakatsukasa and Nicholas J. Higham. Backward stability of iterations for computing the polar decomposition. *SIAM J. Matrix Anal. Appl.*, 33(2):460–479, 2012.

[6] Yuji Nakatsukasa and Nicholas J. Higham. Stable and efficient spectral divide and conquer algorithms for the symmetric eigenvalue decomposition and the SVD. *SIAM J. Sci. Comput.*, 35(3):A1325–A1349, 2013.

[7] G. W. Stewart. *Introduction to Matrix Computations*. Academic Press, New York, 1973. xiii+441 pp. ISBN 0-12-670350-7.

[8] David S. Watkins. *Fundamentals of Matrix Computations*. Third edition, Wiley, New York, USA, 2010. xvi+644 pp. ISBN 978-0-470-52833-4.