How and How not to Compute the Moore-Penrose Inverse?

Alicja Smoktunowicz and Iwona Wróbel

Warsaw University of Technology,
Faculty of Mathematics and Information Science,
Pl. Politechniki 1, 00-661 Warsaw, Poland,
e-mails: smok@mini.pw.edu.pl, wrubelki@wp.pl

Abstract

In the last years a number of fast algorithms for computing the Moore-Penrose inverse of structured and block matrices have been designed. There is a variety of new papers dealing with numerical algorithms, whose authors neglect the issue of numerical stability of their algorithms and focus only on complexity (number of arithmetic operations). However, very often the proposed algorithms are not accurate up to the limitations of data and conditioning of the problem.

We present a comparison of certain algorithms for computing the Moore-Penrose inverse $A^\dagger$, of $A \in \mathbb{R}^{m\times n}$, where A has full column rank, i.e. $rank(A) = n \leq m$. In this case $A^\dagger$ is the left inverse of A, i.e. $A^\dagger A = I_n$, where I_n denotes the $n \times n$ identity matrix.

We study the numerical stability and algebraic complexity of considered algorithms. Numerical experiments in MATLAB are given to compare the performance of some methods for computing the Moore-Penrose inverse.

References

[1] A. Ben-Israel and T.N.E. Greville, *Generalized Inverses: Theory and Applications, 2nd edn.*, Springer, New York, 2003.

[2] Å. Björck, *Numerical Methods for Least Squares Problems*, SIAM, Philadelphia, PA, USA, 1996.

[3] N.J. Higham, *Accuracy and Stability of Numerical Algorithms*, SIAM, Philadelphia, 1996.

[4] A. Smoktunowicz and I. Wróbel, *Numerical aspects of computing the Moore-Penrose inverse of full column rank matrices*, BIT Numerical Mathematics. Published online: 14 October 2011, doi: 10.1007/s10543-011-0362-0.