Verification of preordering numerical methods for reduction of fill-ins number in LU decomposition

Jagoda Plata
Numerical Methods Laboratory, "Politehnica" University of Bucharest,
Splaiul Independentei 313, Bucharest, ROMANIA
plata@lmn.pub.ro

Michał Dobrzyński
Numerical Methods Laboratory, "Politehnica" University of Bucharest,
Splaiul Independentei 313, Bucharest, ROMANIA
d_michal@lmn.pub.ro

Abstract

Decomposition of large matrices appearing in process of modeling EM effects in electrical circuits causes high load of memory, extended computing time and large number of numerical faults. Those three main problems can be reduced, especially in sparse matrices, improving number of zeros appearance in process of matrix factorization. Choosing a proper preordering algorithm to decompose matrix can be a basic method to reach satisfying effects.
Our problem was to decompose an unsymmetrical, sparse and complex matrix, with as few fill-ins as possible, with good accuracy, that could be a base for further calculations. We decided to implement our own code for Markowitz preordering algorithm in every LU decomposition iteration.
The preoredering impact was illustrated with several examples of matrix A. Markowitz algorithm compared to UMFPACK and MATLAB build-in method with COLAMD preordering gave us a minor number m of fill-ins in matrices L and U. Our measure of error in LU decomposition was the number n of elements with values above numerical zero in matrix $\delta = |LU - PAQ|$ and its maximum value $max(\delta)$. Number n proved an efficient use of our algorithm and the $max(\delta)$ indicated a good accuracy for tested algorithm.
Unfortunately the cost of computation for Markowitz algorithm method was very high. This may be partly caused by no optimal code implementation. However, precision of results and fewer number of fill-ins in matrices L and U in Markowitz algorithm suggest, that in order to receive satisfying results it is essential to choose a proper preordering method, adapted to a unique matrix type.
Improvement of computation time can be reached by code optimization and changing a programming platform from MATLAB to C++, and will be one of the next steps in our further research.

References

[1] L.N. Trefethen, D. Bau III, *Numerical linear algebra.* SIAM, Philadelphia, 1997.