Preordering and symbolic factorization for reduction of fill-ins in LU decomposition of large sparse matrices

Jagoda Plata and Michal Dobrzynski
Numerical Methods Laboratory
Politehnica University of Bucharest
Bucharest, Romania
plata@lmn.pub.ro, d_michal@lmn.pub.ro

***Abstract*—Linear systems arise in large-scale scientific and engineering calculations. In many cases, coefficient matrices tend to be very large and sparse. Given a system of linear equations, direct solutions can be obtained using Gaussian elimination. The paper describes the application of the Gaussian elimination method in conjunction with reordering algorithms. Our main goal is to present an overview of issues arising in the symbolic factorization phase.**

I. Introduction

Factorization of large matrices appearing in process of modeling electromagnetic effects in electrical circuits is the most resources consuming stage of numerical simulation. It causes high load of memory, extended computing time and large number of numerical instabilities or faults. Those three main problems can be diminished, especially in sparse matrices, controlling the fill-in in the process of matrix factorization. A proper preordering algorithm is a basic method to reach desired effects.

Our goal is to improve the solution procedure of the EM numerical simulations in the frequency domain of radio frequency devices modeled by Finite Integrals Technique – FIT. This approach generates very sparse matrices (with maxim five nonzero entries per line, sparsity approximately 0.07%), but unsymmetrical, unstructured and complex [1].

II. The Computational Methodology

A. Gaussian Elimination

Gaussian elimination algorithm proceeded on A in finite number of steps of decomposition results in LU factorization:

$$A = L \cdot U \qquad ( )$$

where U is upper-triangular and L is lower-triangular matrix.

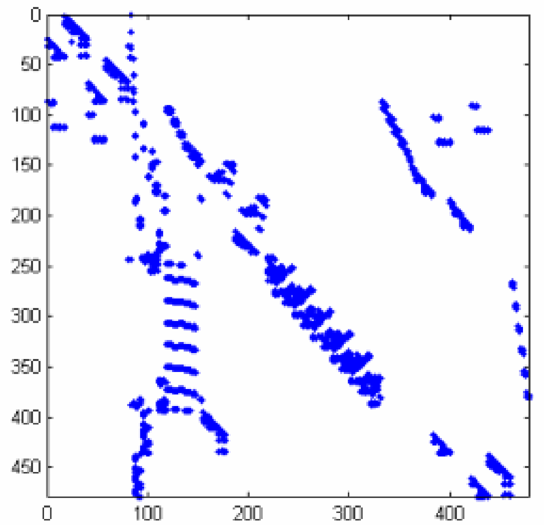

Figure 1. Example of unsymmetric, real matrix A_1, size: 479x479, Westerberg, chemical process simulation problem.

The significant impact for the algorithm of elimination has the structure of matrix A. For sparse and not reordered matrices, a process of Gaussian elimination can destroy the zero entries in A, i.e. fill-in will occur in L and U. Specifically, after one step of Gaussian elimination, the L and U parts of LU factorization can become dense; this results in disastrous usage of time and memory in further computations. For this reason, it is important to find permutations of the matrix that will have the effect of reducing fill-ins during the Gaussian elimination process.

B. Ordering

As the sparsity structure of the LU factors depends on the sparsity structure of A, controlling the number of zero entries that turn into nonzero can be performed with ordering

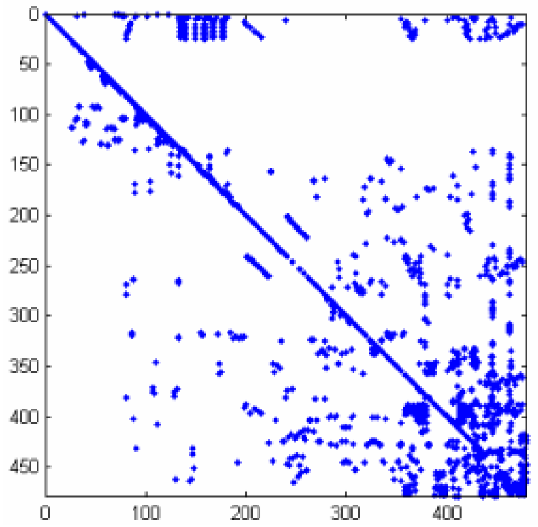

Figure 2. Matrix A_1, preordered with MARKMOD algorithm.

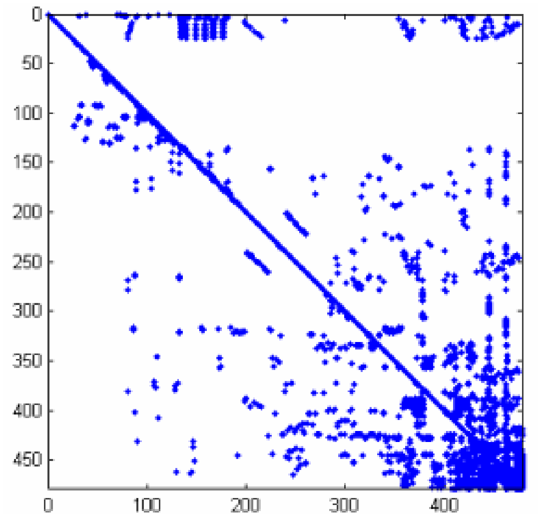

Figure 3. Matrices L and U as a result of decomposition of preordered matrix A_1. It can be observed, that its pattern is similar to PA_1Q. Number of fill-ins has been reduced. The process of Gaussian elimination results stable.

techniques, i.e. permuting rows with row permutation matrix P and columns with the column permutation matrix Q:

$$P \cdot A \cdot Q = L \cdot U \qquad (\ )$$

However, designing an ordering algorithm it is wise to predict which zero entries will turn into nonzero in the next step of elimination, which means that a symbolic factorization should be performed (factorization processed only symbolically, i.e. without numerical values).

C. Pivoting

The main problem of sparse Gaussian elimination is choosing row and column permutation matrices P and Q to preserve not only the sparsity of L and U, but also to maintain numerical stability of the system. For general matrices, the LU factorization is neither stable nor backward stable. The instability can be controlled by pivoting, however, in a code where numerical pivoting is necessary, the symbolic phase cannot be separated from the numerical factorization.

III. RESULTS

Our main problem was to develop a preordering algorithm combining the phase of pivoting and symbolic factorization and giving satisfying fill-in quantitative results. As numerical factorization is the most time-consuming, whereas symbolic factorization takes very little time and ordering can be relatively inexpensive too, the perfect solution would be finding such preordering, processed without numerical values, that stability of elimination would be maintained without the need of applying pivoting.

For nonsymmetric ordering, due to previous research results [4], Markowitz scheme has been investigated and applied to all nonzero entries in matrix. The developed preordering algorithm MARKMOD performs at each stage of decomposition Markowitz measures, predicting the entries filled-in by the elimination process. From the multiple choices of elements with minimum Markowitz cost, after the phase of breaking the ties, a row and column to be permuted are selected taking into consideration the location of previously inserted diagonal entries and their corresponding columns and rows.

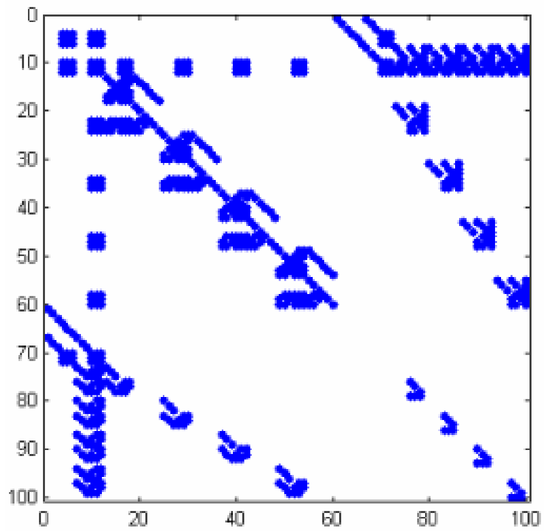

Figure 4. Example of unsymmetric, real matrix A_2, size: 100x100, Marco Morandini, small helicopter rotor model.

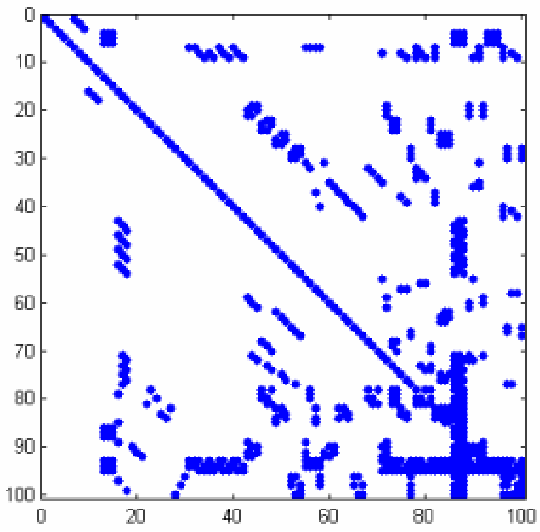

Figure 5. Matrix A_2, preordered with MARKMOD algorithm.

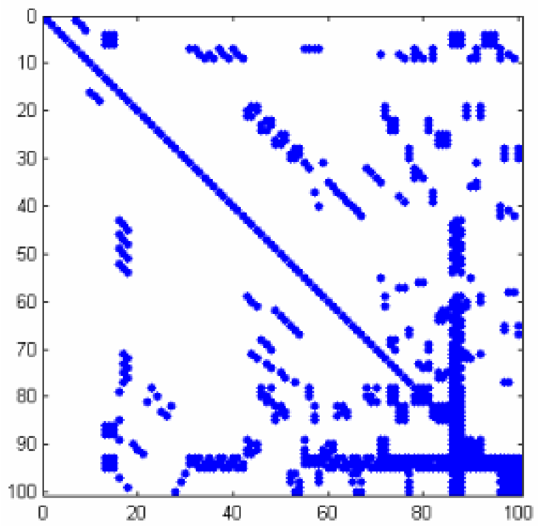

Figure 6. Matrices L and U as a result of decomposition of preordered matrix A_2. Even though, that the number of fill-ins has been profitably reduced, the Gaussian elimination results unstable.

IV. Conclusions

The researches prove very good quantitative and stability results comparing with other preordering methods (Table I). However, still, in some cases, despite an effective reduction of fill-ins, the problem of instability arises (Fig. 4, 5, 6). The basic reason for this behaviour is the difficulty of predicting pivots in the phase of symbolical factorization. Nevertheless the carried out researches are an excellent base for future studies of heuristic approach.

As matrices generated in FIT modeling have a defined and predictive pattern, it is necessary in further researches to adapt the algorithm to the specific matrix. This, and studies of other methods for improvement of stability will be the next stage of our future research work.

TABLE I. Algorithms' Performances With Some Unsymmetric Matrices

Matrix title size / nnz(A)	**Preordering algorithm and corresponding nnz[e](L)+nnz(U)**				
	COLAMD[a]	***COLPERM***[b]	***SYMRCM***[c]	***UMFPACK***[d]	***MARKMOD***
Chemical process, Bogle, 1982 59x59 / 271	534	542	668	433	402
Chemical process, Grund, 1994 87x87 / 230	480	507	631	442	394
Fluid dynamics, Meerbergen, 1994 100x100 / 396	786	782	784	498	496
Pajek networks, Garfield, 2002 396x396 / 994	1 576	1 390	2 109	1 389	1 383
Chemical process, Westerberg, 1983 479x479 / 1.887	6 422	7 083	14 579	4 246	3 245
Web connectivity, Moler, 2002 500x500 / 2.636	14 767	3 143	3 752	3 137	3 138
Fluid dynamics, Godet-Thobie, 1991 1090x1090 / 3.546	4 890	4 636	6 592	4 636	4 636
Circiut simulation, Bomhof, 2000 2624x2624 / 35.823	~1M	47 304	~6M	44 879	43 021

a. Column Approximate Minimum Degree Permutation

b. Column-count Permutation

c. Sparse Reverse Cuthill-McKee Ordering

d. Unsymmetric Multifunctional Sparse LU Factorization Package

e. Number of nonzero elements

References

[1] Gabriela Ciuprina, Daniel Ioan, Dragos Niculae, Jorge Fernandez Villena, L. Miguel Silveira, "Parametric models based on sensitivity analysis for passive components", ISEF 2007 - XIII International Symposium on Electromagnetic Fields in Mechatronics, Electrical and Electronic Engineering, Prague, Czech Republic, 2007.

[2] Lloyd N. Trefethen, David Bau III, Numerical linear algebra, SIAM, Philadelphia, USA, 1997.

[3] Yousef Saad, Iterative methods for sparse linear systems, PWS Publishing Company, Boston, USA, 1996.

[4] Jagoda Plata, Michal Dobrzynski, "Solving methods for sparse matrices in modeling of EM effects in nano IC", IP:M&S 2008 - IV International Conference on Inverse Problems: Modeling & Simulation, Fethiye, Turkey, 2008, in press.