CR- factorization method for sparse matrices

Sergiy Ye. Saukh
National Academy of Sciences of Ukraine
saukh@svitonline.com

Abstract

Sergiy Ye. Saukh. CR- factorization method for sparse matrices. *The column-row (CR-) factorization method is offered. The CR-factorization method differs from the LU- factorization method by property of adaptive to the placement of pivoting entries. Suggested method allows to give up implementation of actions by transposition of columns and rows in the process of matrix factorization and therefore to accelerate the solving of the large-scale systems of linear algebraic equalizations.*

Introduction

Triangular decomposition (or $LU-$factorization) method is one of the most frequently used variant of Gaussian elimination method. In this case matrix $\boldsymbol{A}$ is presented as the product of lower triangular matrix $\boldsymbol{L}$ and upper triangular matrix $\boldsymbol{U}$. The necessary conditions to realize $LU-$factorization are regularity of $\boldsymbol{A}$ matrix and nonzero entry u_{kk} of matrix $\boldsymbol{U}$ on the $k-$th step of factorization. The triangular decomposition of regular matrix $\boldsymbol{A}$ can always be achieved by permuting of rows and columns so that conditions $u_{kk} \neq 0$ are held true. For that purpose permutation matrices $\boldsymbol{P}$ and $\boldsymbol{Q}$ are used such that $\boldsymbol{P} \cdot \boldsymbol{A} \cdot \boldsymbol{Q} = \boldsymbol{L} \cdot \boldsymbol{U}$. In the conditions of calculations with finite number of digits the Gaussian elimination method realization features require such selection of pivoting entries a_{ij} that the growth of absolute values of entries of $\boldsymbol{L}$ and $\boldsymbol{U}$ matrices is staying limited during the factorization process [3-8]. In the case of the sparse matrix $\boldsymbol{A}$ the permutations influence on the level of filling by nonzero entries of $\boldsymbol{L}$ and $\boldsymbol{U}$ matrices are also taken into account. This level may raise significantly in comparison with initial value of the level of filling by nonzero entries of matrix $\boldsymbol{A}$. So the selection of the pivoting entries is based on the combination of demands to support of numerical stability of the Gaussian elimination and to minimize the filling levels of the $\boldsymbol{L}$ and $\boldsymbol{U}$ matrices. In accordance with these demands the different strategies of pivoting entries selection are realized in modern software [3-6].

For storage nonzero entries of sparse matrices are used special formats of data so called row-wise or linked list representation formats. Using these formats significantly complicate procedures of the permutation of rows and columns [3-6]. Although the permutation operations don't need really displacements of the values of entries in the computer memory, but they require searching and changing index values of the rows and columns. In general the sub arrays of permuting column and row indexes don't coincide by lengths and it complicates the realization of index displacements. Our researches show the commensurability of the volumes of operations which are necessary for execution of the permutations and the LU – factorization. We suggest the new method of column-row factorization or CR – factorization briefly, which is distinguished on principle from the method of LU – factorization by the adaptation to the selection of pivoting entries. It permits to refuse from the row and column permutations in the process of matrix factorization and therefore to accelerate the process of calculation of matrix factors.

CR–factorization method

The basic relation of the method is the expression:

$$\boldsymbol{A} = \boldsymbol{C}_j \boldsymbol{R}_i + \boldsymbol{A}_1\langle\langle i, j\rangle\rangle \tag{1}$$

in which the initial $n \times n$ matrix $\boldsymbol{A}$ is presented by the sum of $n \times n$ matrix $\boldsymbol{A}_1\langle\langle i, j\rangle\rangle$ and the product of $n \times 1$ column-vector $\boldsymbol{C}_j$ and $1 \times n$ row-vector $\boldsymbol{R}_i$.

If the entry a_{ij} of matrix $\boldsymbol{A}$ doesn't equal to zero then from the relation (1) can determine all the entries of the column $\boldsymbol{C}_j$ and row $\boldsymbol{R}_i$ accurate to a certain factor, under the assumption made that the entries of row i and column j of matrix $\boldsymbol{A}_1\langle\langle i, j\rangle\rangle$ equal to zero. Indeterminacy "accurate to a certain factor" in determination of entry column $\boldsymbol{C}_j$ and row $\boldsymbol{R}_i$ is conditioned by impossibility to find two required values c_{ij} and r_{ij} from one equation $c_{ij} \cdot r_{ij} = a_{ij}$.

For the elimination of indeterminacy in c_{ij} and r_{ij} values determination it ought to accept one of the eventual assumptions, for example,

$$c_{ij} = 1, \tag{2}$$

$$r_{ij} = 1, \tag{3}$$

$$|c_{ij}| = |r_{ij}|. \tag{4}$$

Notice that the assumptions (2) and (3) are analogous to the assumptions which are accepted in point of diagonal entries of one of the $\boldsymbol{L}$ and $\boldsymbol{U}$ factor matrices in LU – factorization method. The assumption (4) also takes place in the special variant of LU – factorization method for the symmetrical positive determinate matrices and are known as the Cholesky elimination method (LL^T – factorization) or the square root method [7, 8].

The relation (1) determines the partial one-step column-row factorization of $\boldsymbol{A}$ matrix relative to the a_{ij} pivoting entry. Here the column $\boldsymbol{C}_j$ and row $\boldsymbol{R}_i$ are multipliers or factors, and $\boldsymbol{A}_1\langle\langle i, j\rangle\rangle$ matrix is a residual matrix. Its entries, except those which are in $i-$th row and $j-$th column, form an active $(n-1)\times(n-1)$ submatrix.

Analyzing the $\boldsymbol{A}_1\langle(i,j)\rangle$ matrix structure in the relation (1), we notice that the obtained entries of column $\boldsymbol{C}_j$ and row $\boldsymbol{R}_i$ naturally disposes in the matrix $\boldsymbol{A}_1\langle(i,j)\rangle$ on the place of zero entries its $i-$th row and $j-$th column. This superposition of the entries of column $\boldsymbol{C}_j$ and row $\boldsymbol{R}_i$ with the entries of matrix $\boldsymbol{A}_1\langle(i,j)\rangle$ leads to the superposed matrix forming as

$$\boldsymbol{A}_1^{CR}\langle(i,j)\rangle = \boldsymbol{A}_1\langle(i,j)\rangle \oplus (\boldsymbol{C}_j, \boldsymbol{R}_i) \qquad (5)$$

where the symbol $\oplus$ is the superposition operation.

Thus the relations (1) – (5) determine the first step of a row-column factorization of matrix $\boldsymbol{A}$ relatively a_{ij} pivoting entry without the rows and columns permutation. The following analogous step of factorization is realized relative to the pivoting entry, which is chosen in the active part of matrix $\boldsymbol{A}_1\langle(i,j)\rangle$. It coincides with the analogous part of matrix $\boldsymbol{A}_1^{CR}\langle(i,j)\rangle$.

If the pivoting entry is $a_{pq} - c_{pi}r_{jq} \neq 0$ which is situated on the intersection of $p-$row and $q-$column of matrix $\boldsymbol{A}_1\langle(i,j)\rangle$ then taking into account the identity (1), we can write

$$\boldsymbol{A}_1\langle(i,j)\rangle = \boldsymbol{C}_q\boldsymbol{R}_p + \boldsymbol{A}_2\langle(i,j),(p,q)\rangle \qquad (6)$$

where $n\times 1$ vector-column $\boldsymbol{C}_q$ and $1\times n$ vector-row $\boldsymbol{R}_p$ contain the zero entries $c_{iq}=0$ and $r_{pj}=0$ and $n\times n$ matrix $\boldsymbol{A}_2\langle(i,j),(p,q)\rangle$ has two zero rows and two zero columns with indexes i, p and j, q correspondingly.

We receive after the substitution (6) to (1):

$$\boldsymbol{A} = \boldsymbol{C}_j\boldsymbol{R}_i + \boldsymbol{C}_q\boldsymbol{R}_p + \boldsymbol{A}_2\langle(i,j),(p,q)\rangle \qquad (7)$$

or in the combined form

$$\boldsymbol{A}_2^{CR}\langle(i,j),(p,q)\rangle = \boldsymbol{A}_2\langle(i,j),(p,q)\rangle \oplus (\boldsymbol{C}_j,\boldsymbol{R}_i,\boldsymbol{C}_q,\boldsymbol{R}_p). \qquad (8)$$

Using the expressions (1) and (6), which determine partial one- or two-steps column-row factorization, we receive the full formula of $CR-$factorization of regular $n\times n$ matrix $\boldsymbol{A}$ for the case when the pivoting entry co-ordinates (i,j) are chosen dynamically in the process of factorization and form certain ordered set $\boldsymbol{P}$:

$$\boldsymbol{A} = \sum_{(i,j)\in P} \boldsymbol{C}_j \cdot \boldsymbol{R}_i\ . \qquad (9)$$

Here the summing is realized in the entry sequence in set $\boldsymbol{P}$. Notice that in contrast to (1) and (7) the right part of this expression don't contain the summand type matrix $\boldsymbol{A}_n\langle\boldsymbol{P}\rangle$ of entries those n rows and n columns in the intersection of which the pivoting entries are chosen. These entries are replaced by zero in the process of forming $\boldsymbol{A}_n\langle\boldsymbol{P}\rangle$ for n steps. Just these places vector-columns $\boldsymbol{C}_j$ and vector-rows $\boldsymbol{R}_i$ are situated at the forming of the coincidence matrix $\boldsymbol{A}^{CR}\langle\boldsymbol{P}\rangle$. So $\boldsymbol{A}_n\langle\boldsymbol{P}\rangle \equiv 0$.

Obviously the expression (9) can be presented in the matrix form:

$$\boldsymbol{A} = \boldsymbol{C}\langle\boldsymbol{P}\rangle \cdot \boldsymbol{R}\langle\boldsymbol{P}\rangle, \tag{10}$$

where the matrices $\boldsymbol{C}\langle\boldsymbol{P}\rangle$ and $\boldsymbol{R}\langle\boldsymbol{P}\rangle$ are aggregated from the vector-columns $\boldsymbol{C}_j$ and vector-rows $\boldsymbol{R}_i$ accordingly. The situation order of these vectors corresponds to the order of their indexes in $\boldsymbol{P}$ set. Therefore the situation orders of vector-columns $\boldsymbol{C}_j$ and vector-rows $\boldsymbol{R}_i$ in the coincidence matrix

$$\boldsymbol{A}^{CR}\langle\boldsymbol{P}\rangle = \boldsymbol{C}^{CR} \oplus \boldsymbol{R}^{CR} \tag{11}$$

correspond to their j and i indexes so in fact in the matrix $\boldsymbol{A}^{CR}\langle\boldsymbol{P}\rangle$ are superposed two matrices $\boldsymbol{C}^{CR}\langle\boldsymbol{P}\rangle$ and $\boldsymbol{R}^{CR}\langle\boldsymbol{P}\rangle$ instead of the matrices $\boldsymbol{C}\langle\boldsymbol{P}\rangle$ and $\boldsymbol{R}\langle\boldsymbol{P}\rangle$.

In general case the matrices $\boldsymbol{C}\langle\boldsymbol{P}\rangle$ and $\boldsymbol{R}\langle\boldsymbol{P}\rangle$ don't take triangular form. Only when the pivoting entries are chosen sequentially in the co-ordinate $\boldsymbol{P}\langle(1,1),(2,2),\ldots,(n,n)\rangle$ in the process $CR-$factorization, then forming vector-columns $\boldsymbol{C}_j$ and vector-rows $\boldsymbol{R}_i$ form the lower triangular and upper triangular matrices $\boldsymbol{C}\langle\boldsymbol{P}\rangle = \boldsymbol{L}$ and $\boldsymbol{R}\langle\boldsymbol{P}\rangle = \boldsymbol{U}$, which are situated in the coincidence matrix $\boldsymbol{A}^{CR}\langle\boldsymbol{P}\rangle$ so that $\boldsymbol{C}^{CR}\langle\boldsymbol{P}\rangle = \boldsymbol{C}\langle\boldsymbol{P}\rangle = \boldsymbol{L}$ and $\boldsymbol{R}^{CR}\langle\boldsymbol{P}\rangle = \boldsymbol{R}\langle\boldsymbol{P}\rangle = \boldsymbol{U}$. The basic structural peculiarity of general matrices $\boldsymbol{C}\langle\boldsymbol{P}\rangle$ and $\boldsymbol{R}\langle\boldsymbol{P}\rangle$ is such as it in principle can be converted to the triangular form by the way of rows and columns permutations.

The results of the experimental researches

It is evident that the sample-comparison operations over the indexes into rows and columns lists of nonzero matrix entries, which are executed in the process of rows and columns permutation, cannot be directly correlated with the arithmetical operations executed under the nonzero entries in the factorization process. So the number of sample-comparison operations should be estimated experimentally by the time expenses for the fulfillment of those operations and to determine their specific weight in the general time expenses on the matrices factorization. The description of the experiments conditions and its results are represented below.

For calculations the computer Intel P4 (Chipset Intel 865 PE, FSB 800 MHz, CPU 3.0GHz with HT, Dual Channel Memory 1GB: 2 x 512MB, DDR400) running under operational system Microsoft Windows XP are used. The program code was written by the author on the C++ language in the Microsoft Visual Studio.Net and was taken as a base for the experimental researches. Sequentially three algorithms are realized for solution of linear algebraic equation $\boldsymbol{A} \cdot \boldsymbol{X} = \boldsymbol{B}$ with a vector of a right part $\boldsymbol{B} = \boldsymbol{A} \cdot \boldsymbol{1}$, where $\boldsymbol{1}$ is unit vector. For $CR-$factorization of matrix $\boldsymbol{A}$ the first algorithm realizes the method with the improved generalized Markovits's strategy of the pivoting entries choice [4]. The second algorithm realizes the method of $CR-$factorization of the same matrix $\boldsymbol{A}$, using the set $\boldsymbol{P}$ of pivoting entries co-ordinates on their choice order, which was formed by the first algorithm. The third algorithm differs from the second one only the $LU-$factorization method realization. Notice that in the both realization of $CR-$factorization method for matrix $\boldsymbol{A}$ the assumption (3) is used.

The implementation of a written program code allows to estimate the summary time expenses for the search of $\boldsymbol{X}$ vector-solution by the given vector $\boldsymbol{B}$ and to differentiate the time expenses on $CR-$ and $LU-$factorization.

In all our experiments the equality fulfillment $nnz\left(\boldsymbol{C}^{CR} \oplus \boldsymbol{R}^{CR}\right) = nnz(\boldsymbol{L} \oplus \boldsymbol{U})$ is supported, this is the coincidence of the numbers of nonzero entries in the $\boldsymbol{C}$ and $\boldsymbol{R}$ factor matrices and in the $\boldsymbol{L}$ and $\boldsymbol{U}$ factor matrices which are received by the methods of $CR-$ and $LU-$factorization.

For the supporting of correct experimental researches the influence of control procedures of nonzero matrix entries located in working arrays was slacked maximally. First of all the sizes of arrays of nonzero entries and their rows and columns indexes were chosen equal $45 \cdot 10^6$. This is so large that not to need to initiate the special procedure "cleaning the refuse" in them [4, 5]. In addition the improving extended Markovits's strategy is used in all tests, where the search of pivoting entries is limited by the set S_p of p rows of active submatrix with the minimal numbers of nonzero entries. The pivoting is chosen the entry a_{ij} with the least Markovits's value but that the absolute value of which not more than in τ times less than maximal entry on absolute value $a_{max} \in S_p$.

Table 1. The test matrices, their dimensions n, the numbers of nonzero entries in initial matrices $nnz(\boldsymbol{A})$ and in the coincidence factor matrices $nnz(\boldsymbol{C} \oplus \boldsymbol{R})$, mistakes ε and the test systems solution time T_{CR}.

name	n	$nnz(\boldsymbol{A})$	$nnz(\boldsymbol{C} \oplus \boldsymbol{R})$	ε	T_{CR}, *c*
ASIC_680k	682862	3871773	8220972	4.57E-08	1024.531
ASIC_680ks	682712	2329176	6170719	9.51E-09	47.953
ASIC_320ks	321671	1827807	5082048	2.89E-12	54.875
scircuit	170998	958936	2408645	8.39E-11	2.328
circuit_4	80209	307604	707803	1.92E-09	1.359
bayer01	57735	277774	7283567	4.33E-08	40.453
mark3jac060	27449	170695	28775871	3.42E-11	2345.125
ex35	19716	228208	2474829	6.82E-08	7.360
poisson3Da	13514	352762	10697886	1.10E-13	162.281
circuit_3	12127	48137	76708	2.07E-12	0.141
cry10000	10000	49699	501518	4.04E-06	0.828
gemat11	4929	33185	77616	2.21E-13	0.047
lns_3937	3937	25407	568884	1.60E-12	1.719

psmigr_1	3140	543162	6534812	9.40E-13	122.531
orani678	2529	90158	336551	3.07E-15	0.469

Table 1. Continue

name	n	$nnz(\boldsymbol{A})$	$nnz(\boldsymbol{C} \oplus \boldsymbol{R})$	ε	T_{CR}, c
adder_trans_01	1814	14579	20541	2.11E-16	0.016
orsirr_1	1030	6858	57892	1.42E-13	0.063
orsirr_2	886	5970	45146	1.37E-13	0.031

The calculation experiments are fulfilled with the matrices which was taken from the Web sites [9, 10]. The general characteristics of test matrices and the results of experimentation with them are shown in the tables 1-2, where

$$\varepsilon = \sqrt{\frac{\sum_{z=1}^{n}(x_z - 1)^2}{n}} .$$

The parameters p and τ are formed equal depending on the tested matrix: in the test ASIC_680k – $p = 100$ and $\tau = 0.0005$, in the test mark3jac060 – $p = n$ and $\tau = 0.001$. The parameters values were $p = 1$ and $\tau = 1$ in the rest all tests.

Described conditions of experiments realization permitted to avoid the initialization of procedures “cleaning the refuse” in the blocks. However, in the algorithm which realizes the method of LU – factorization we applied one of the well-known procedure of rows and columns permutation minimizing the volume of sample-comparison operations which are needed for that [11]. This procedure leads not only to change the values indexes of the permuted rows an columns in the arrays of rows and columns indexes but also is accompanied the permutation of the separate entries in the arrays and also in the array of nonzero entries. In the result of entries permutation in the array their placement on the present step of the LU – factorization method realization differs from the placement which can observe in the case of realization of CR – factorization method. The necessity of placement the new appearing nonzero entries reinforces these differences in the placement of entries. It generates the different dynamic processes of replacement the entries in rows and columns. Thus we failed to achieve a full equality of realization experiments conditions with the CR – and LU – factorization algorithms. However we minimized difference of conditions as far as possible.

Table 2. The computation time expenses of factorization (T_{CR-F}, T_{LU-F}) and solution (T_{CR-S}, T_{LU-S}) of the test systems of equations by the methods of CR – and LU – factorization; relative decelerations of the LU – factorization processes $\delta T_F = \left(\frac{T_{LU-F}}{T_{CR-F}} - 1\right) \cdot 100\%$.

name	T_{CR-F}, c	T_{LU-F}, c	δT_F, %	T_{CR-S}, c	T_{LU-S}, c
ASIC_680k	1004.328	1539.047	53.2	0.172	0.156
ASIC_680ks	46.656	49.204	5.5	0.188	0.140
ASIC_320ks	54.875	59.921	9.2	0.171	0.141

scircuit	2.000	3.907	95.4	0.125	0.078
circuit_4	1.172	1.562	33.3	0.016	0.032
bayer01	39.062	51.563	32.0	0.094	0.094

Table 2. Continue

name	T_{CR-F}, *c*	T_{LU-F}, *c*	δT_F, %	T_{CR-S}, *c*	T_{LU-S}, *c*
mark3jac060	903.953	1106.750	22.4	0.297	0.281
ex35	7.016	10.578	50.8	0.047	0.031
poisson3Da	159.109	204.328	28.4	0.110	0.110
circuit_3	0.093	0.109	17.2	0.000	0.016
cry10000	0.781	1.171	49.9	0.000	0.016
gemat11	0.031	0.047	51.6	0.000	0.000
lns_3937	1.625	2.313	42.3	0.015	0.000
psmigr_1	120.437	147.562	22.5	0.063	0.063
orani678	0.437	0.656	50.1	0.016	0.000
adder_trans_01	0.015	0.016	6.7	0.000	0.000
orsirr_1	0.047	0.078	66.0	0.000	0.000
orsirr_2	0.047	0.063	34.0	0.000	0.000

The conclusions follow from these results:

1. The time expenses of execution of the rows and columns permutation in the active submatrices are found to compare with the time expenses of execution of the matrix factorizations. The realization of permutations in the $LU-$factorization method leads to the essential deceleration of the calculating process on the average value $\delta T_F = 37.3\%$ for the given tests set. It follows from the table 2 where the values $\delta T_F = \left(\frac{T_{LU-F}}{T_{CR-F}} - 1 \right) \cdot 100\%$ are presented.
2. The calculating time expenses of the solution of linear equation systems $\boldsymbol{C}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{R}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{X} = \boldsymbol{B}$ received by the $CR-$factorization method are greater than the calculating time expenses of the solution of linear equation systems $\boldsymbol{L} \cdot \boldsymbol{U} \cdot \boldsymbol{X} = \boldsymbol{B}$ formed by the $LU-$factorization method. The increase of the calculating time expenses are stipulated by the algorithms peculiarities of linear equation systems $\boldsymbol{C}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{V} = \boldsymbol{B}$ and $\boldsymbol{R}\langle \boldsymbol{P} \rangle \cdot \boldsymbol{X} = \boldsymbol{V}$ with the non-triangular matrices $\boldsymbol{C}\langle \boldsymbol{P} \rangle$ and $\boldsymbol{R}\langle \boldsymbol{P} \rangle$. However, as a whole the summarized calculating time $T_F + T_S$ expenses of the solution of initial equation systems is essential greater in the case of use the $LU-$factorization method.

Conclusions

As opposed to the LU-factorization method with choosing pivot entries, where, in general case, rows and columns reordering are executed obligatorily, CR-factorization method doesn't demand such operations and therefore executes the factorization more quickly. The computation acceleration is significant in the case of difficult access to the matrix entries allocated in arrays according to special formats.

The CR-factorization method for sparse matrices can be used not only as the direct method of quick solving large-scale system of linear equations but also as the method of quick construction of the preconditioning matrices for the iterative solving of the systems [12-14]. Obviously the preconditioning matrices can be gotten by using incomplete CR-factorization algorithms. Such algorithms can be executed quickly without any displacements of elements in the arrays. Thus the suggested CR-factorization method becomes the basis of new algorithms of the effective solving of the large-scale linear algebraic systems of equations.

References

1. www.cise.ufl.edu/research/sparse/codes/
2. www.srcc.msu.su/num_anal/lib_na/cat/cat552.htm
3. Pissanetzky S. Sparse Matrix Technology. – Moscow: Mir, – 1988. – 410 p.
4. Esterbyu O., Zlatev Z. Direct Methods for Sparse Matrices. – Moscow: Mir, – 1987. – 120 p.
5. Ikramov H.D. Computational Methods of Linear Algebra. Moscow : Znaniye, 1989. – 48 p.
6. Davis T.A. Direct Methods for Sparse Linear Systems. – Philadelphia.: SIAM, – 2006. – 226 p.
7. Samarsky A.A., Gulin A.V. Numerical Methods. – Moscow: Nauka, – 1989. – 432 p.
8. George A., Lui J. W-H. Computer Solution of Large Sparse Positive Definite Systems. – Moscow: Mir, – 1984. – 334 p.
9. www.cise.ufl.edu/research/sparse/matrices/index.html
10. http://math.nist.gov/MatrixMarket/matrices.html
11. www.netlib.org/y12m/index.html
12. Saad Y. Iterative Methods for Sparse Linear Systems, University of Minnesota, Department of Computer Science and Engineering, Minneapolis, MN, 2000. – 448 p.
13. Ortega J.M. An Introduction to Parallel and Vector Methods for Solution of Linear Systems. – Moscow: Mir, – 1991. – 386 p.
14. Saukh S.Ye. Incomplete Cholesky factorization in fixed memory // Parallel Processing and Applied Mathematics: 5th International Conference, PPAM 2003, Czestochowa, Poland, September 7-10, 2003. Revised Papers. – Springer-Verlag Heidelberg: Lecture Notes in Computer Science. –Volume 3019 / 2004, pp. 1042 - 1051.\

Дата надходження до редакції 25.10.2007 р.