computing@tanet.edu.te.ua
www.tanet.edu.te.ua/computing

ISSN 1727-6209
International Scientific Journal of Computing

INCOMPLETE CHOLESKY FACTORIZATION IN FIXED MEMORY WITH FLEXIBLE DROP-TOLERANCE STRATEGY

Sergey Saukh

G.Y. Pukhov's Institute of Modelling Problems in Power Engineering, NAS of Ukraine
General Naumov str., 15, 03164 Kiev, Ukraine, e-mail: svetlana@ipme.ua

Abstract: *We propose an incomplete Cholesky factorization for the solution of large positive definite systems of equations and for the solution of large-scale trust region sub-problems. The factorization is based on the two-parameter (m, p) – drop-tolerance strategy for insignificant elements in the incomplete factor matrix. The factorization proposed essentially reduces the negative processes of irregular distribution and accumulation of errors in factor matrix and provides the optimal rate of memory filling with essential nonzero elements. On the contrary to the known p – retain and τ – drop-tolerance strategies, the (m, p) – strategy allows to form the factor matrix in fixed memory.*

Keywords: – *large sparse system, drop-tolerance strategy, preconditioner, conjugate gradients, fixed memory*

1.INTRODUCTION

The main idea of conjugate gradients methods with preconditioner is to form a special $C_Б$ matrix at the preparatory step, which is called A matrix's preconditioner and satisfies the following conditions:

1) $C_Б$ is a good approximation of the matrix A;
2) $C_Б$ is easily obtained from $Б$ and it uses effective memory-save mechanisms;
3) It should be much easier to solve the system of equations $C_Б X = B$ than input system $БЧ = B$.

The preconditioner matrix $C_Б$ is used in order to provide clustering of $Б$'s proper values. Then the convergence of iterative solution of preconditioning equation system $C_Б^{-1} БЧ = C_Б^{-1} B$ is significantly higher than one of the input system $БЧ = B$.

The construction of preconditioner is commonly based on the well-known methods of matrix factorization: *LLT, LDLT* and *LU, LDU* [6 – 7]. These methods traditionally applied to symmetric and asymmetric matrices are implemented in so-called incomplete form, with the addition of any non-zero elements drop-tolerance strategy [1 – 5]. The successfully chosen drop-tolerance strategy allows creating the preconditioner $C_Б$, which satisfies the conditions 1) – 3) in the best way.

Let us consider the problems with large sparse symmetric matrices only. In order to solve them the conjugate gradients iteration methods with preconditioner built on the base of incomplete Cholesky factorization are used.

Initially proposed for positive definite systems, iteration methods with preconditioner on the base of incomplete Cholesky factorization are now being developed to be applicable to indefinite systems [1 – 4]. Indefinite systems may appear in n–measured function minimization problems, where the solution process is reduced to the sequence of solutions of sub problems:

$$\min \left\{ B^T X + \frac{1}{2} X^T A X \ : \| DX \|_2 \le \Delta \right\} \qquad (1)$$

where Δ is the trust region radius, $B \in \mathbb{R}^n$ is the gradient of the function at the current iterate, $A \in \mathbb{R}^{n \times n}$ is an approximation to the Hessian matrix, $D \in \mathbb{R}^{n \times n}$ is a nonsingular scaling matrix [10]. To solve (1) we generally need to solve indefinite system of linear equations $AX + B = 0$.

To find an approximate solution of large-scale problem (1), it is proposed, according to paper [11], a conjugate gradients method with preconditioner, which takes into account the restriction on radius of trust region and the possibility that matrix A is indefinite. As it is indicated in papers [1, 10], if $\| DX_k \|_2 \le \Delta$, then the conjugate gradients method generates sequences $\{X_k\}$ and directions $\{P_k\}$ until one of the next three conditions is satisfied:

$$\| AX_k + B \|_2 \le \sigma \| B \|_2, \; P_k^T A P_k \le 0, \; \| DX_{k+1} \|_2 > \Delta. \quad (2)$$

In all three cases an approximate solution of problem (1) satisfying the three convergence conditions of iteration methods in n-measured trust region is defined. In most problems where Δ is comparatively small, the third condition can be satisfied on the first few iterations. As it is pointed out in papers [1 – 2] for the satisfaction of the first two conditions a higher number of iterations of conjugate gradients method is needed, particularly when A is nearly singular. In this case the second condition can be satisfied only if A is not positively defined. Then P_k is a direction of negative curvature.

Our aim is to reduce the number of iterations needed to satisfy the first two conditions in expression (2), which is comparatively difficult [2].

To solve the problem (1) we, as in paper [1], transform the ellipsoidal trust region in the sphere trust region and obtain the following problem

$$\min \left\{ b^T x + \frac{1}{2} x^T a \, x : \| x \|_2 \le \Delta \right\} \quad (3)$$

where $b = D^{-T} B$, $a = D^{-T} A D^{-1}$. Then the approximate solution x of the problem (3) corresponds to the solution X of the problem (1) with relationship $X = D^{-1} x$.

As the scaled matrix D clusters the proper values of matrix A, the conjugate gradients method provides the solution of problem (3) in a small number of iterations. Matrix D is formed on the base of Cholesky factorization.

2.INCOMPLETE FACTORIZATION

The authors of papers [1 – 3, 5] point out that the factor Cholesky clustering property depends on the choice of the filling pattern S. It is confirmed that matrix L must be a lower-triangle matrix and satisfy the next conditions:

$$A = LL^T + R, \; l_{ij} = 0 \text{ if } (i,j) \notin S \text{ and } r_{ij} = 0 \text{ if } (i,j) \in S. \quad (4)$$

Different strategies of forming filling pattern S are described in papers [2, 5, 12 – 14]. Strategies proposed are based on two following ideas:

- Pattern S is initially fixed;
- Pattern S is formed in the process of forming L.

The first strategy, where S is initially fixed, is rather attractive due to the following reasons:

- Pattern S of matrix L is easily obtained from pattern of matrix A;
- The amount of memory needed for matrix L is predictable;
- There is no need to control drop-tolerance.

There are many variants of pattern S initial fixation. For example, one could define a pattern S so that matrix L has become a band fixed-width matrix. Another way is to get pattern S and pattern matrix A identical. The most promising variant was first used in the paper [14] for $ILU(p)$ factorization of asymmetric matrix A. In this case for parameter p=0 the pattern S is set on the pattern of matrix A. If p>0, additional filling of L and U columns is allowed. Actually, an additional number of non-zero elements placed in L and U during their creation are limited by parameter p.

The disadvantage of initial fixation of pattern S consists in uncontrollability losses of non-zero elements of L and U. These losses in the formation of preconditioners often make it impossible to cluster the proper values of matrix A.

The other idea of S formation strategy in the process of matrix A factorization was first used in paper [13] as a so-called τ – drop-tolerance strategy. In this case on a ν - step formation L and U correspondent elements are considered to be unimportant and they are not stored in memory (as zero elements) if they satisfy the condition $\left| a_{ij}^{(\nu)} \right| \le \tau \sqrt{a_{ii}^{(\nu)} a_{jj}^{(\nu)}}$. The τ parameter defines the limit of losses.

The τ – drop-tolerance strategy has two disadvantages:

- Unpredictability of memory requirements on factor matrices storage, which depends on the value of a chosen parameter τ;
- Elasticity of parameter τ vagueness influence on the ability of factor matrices to cluster the proper values of matrix A.

Different combinations of pattern S formation strategies described earlier are used at present. For example, a modified p – strategy for symmetric matrices is used in papers [2, 12]. The modification is based on forming the filling pattern S independently from the position n_j of non-zero elements in j column of matrix A. Only $n_j + p$ elements of j column of L are stored. The approach presented in paper [12] combines p – and τ – strategies. The idea of this two-parameter strategy is a gradual application of τ – drop-tolerance strategy to the elements of j - column of matrix L formed on j - step. According to the p – strategy the $n_j + p$ elements with the biggest absolute value are stored in memory. There are some other approaches.

There is a considerable disadvantage in traditional strategies for forming pattern S. These strategies generate considerably irregular distribution of errors,

which appears during matrix factorization process in fixed amount of memory. As Cholesky factorization procedure is a gradual procedure and can be performed either by rows or by columns, only errors initially inputted in matrix L in the first few rows or columns correspondingly can be controlled. Once initially set during the first step of factorization, these errors can quickly spread on the rest of matrix L during the next steps of factorization procedure. This process involves not only multiplication of errors but also their accumulation. Besides the assumption that the formed filling pattern S don't satisfy the desirable conditions (4) is a mistake. In general case, the pattern S does not satisfy the last condition, as $r_{ij} \neq 0$ if $(i, j) \in S$.

As an example, let us show this on *jki* version of *LLT* Cholesky factorization implemented in the algorithm 1, so that the elements of lower-triangle part of matrix L are computed at the positions of elements of input matrix A. This algorithm presents no drop-tolerance strategy.

```
for j = 1:n
    a(j,j) = sqrt( a(j,j) )
    for k = 1:(j-1)
        for i = (j+1):n
            a(i,j) = a(i,j) - a(i,k)*a(j,k)
        end
    end
    for i = (j+1):n
        a(i,j) = a(i,j)/a(j,j)
        a(i,i) = a(i,i) - a(i,j)^2
    end
end
```

Algorithm 1 – *jki* version of *LLT* factorization.

Notice, that for symmetric input matrix A the equation is solved gradually by columns respectively to elements l_{ij} of matrix L. During solving procedure of the equation (5) matrix A is being gradually transformed in Cholesky factor L. As a result the matrix $L^{(0)} = LL^T$ can be

$$A = LL^T = L^{(0)} = \begin{bmatrix} l_1 l_{11} & l_2 l_{11} & l_3 l_{11} & l_4 l_{11} & \cdots \\ l_2 l_{11} & l_2 l_{21} + l_{22} l_{22} & l_3 l_{21} + l_{32} l_{22} & l_4 l_{21} + l_{42} l_{22} & \cdots \\ l_3 l_{11} & l_3 l_{21} + l_{32} l_{22} & l_3 l_{31} + l_{32} l_{32} + l_{33} l_{33} & l_4 l_{31} + l_{42} l_{32} + l_{43} l_{33} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (5)$$

presented as following:

$$L^{(1)} = \begin{bmatrix} l_{11} & 0 & 0 & 0 & \cdots \\ l_{21} & l_{22} l_{22} & 0 & 0 & \cdots \\ l_{31} & l_{31} l_{21} + l_{32} l_{22} & l_{32} l_{32} + l_{33} l_{33} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix}, \quad (6)$$

$$L^{(2)} = \begin{bmatrix} l_{11} & 0 & 0 & 0 & \cdots \\ l_{21} & l_{22} & 0 & 0 & \cdots \\ l_{31} & l_{32} & l_{33} l_{33} & 0 & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{bmatrix} \quad (7)$$

and so on till $L^{(n)} = L$.

Assume that in the result of the first step factorization from matrix $L^{(0)}$ we have obtained matrix $L^{(1)}$, with insignificantly small element, for example, l_{21} which should be dropped according to the chosen pattern S formation strategy. Therefore the transformation from $L^{(0)}$ to $L^{(1)}$ introduces a contortion to the diagonal element l_{22} of matrix $L^{(1)}$. During the next step from $L^{(1)}$ to $L^{(2)}$ the small error l_{21} will be multiplied with the lower elements of the first column of matrix $L^{(1)}$ and spread towards the whole second column of matrix $L^{(2)}$ in the form of unconsidered components. During the next steps of factorization the contorted elements of the second column generate deviation of matrix elements at the right lower position from the element l_{21}. Moreover the further operations on the contorted elements result in unpredictable accumulation of errors in elements of final matrix $L^{(n)} = L$. Such errors are non-uniformly spread, so that their lowest level is observed in the first columns of matrix L, and the highest one is in the last columns.

Significant level of accumulated errors is often a reason of the formation of such a matrix L, that is being substituted in the first equation of expression (4), could not guarantee the admissible level of errors. So, the matrix L obtained does not inherit properties of matrix A and so could not be a good preconditioner for conjugate gradients method.

Authors in paper [2] propose an interesting variant of $ILLT(p)$ Cholesky factorization, based on a modified p – drop-tolerance strategy. Implemented here as algorithm 2 this variant of factorization is worth a particular attention, as it partially blocks error distribution in matrix being formed. Algorithm 2 differs from other algorithms as it contains two loops in j-step of Cholesky factorization instead of traditional one.

```
for j = 1:n
    a(j,j) = sqrt( a(j,j) )
    col_len = size( i > j: a(i,j) ≠ 0 )
    for k = 1:(j-1) & a(j,k) ≠ 0
        for i = (j+1):n & a(i,k) ≠ 0
            a(i,j) = a(i,j) - a(i,k)*a(j,k)
        end
    end
    for i = (j+1):n & a(i,j) ≠ 0
        a(i,j) = a(i,j)/a(j,j)
```

```
    end
  Retain the largest col_len + p elements in
a((j+1):n,j).
    for i = (j+1):n & a(i,j) ≠ 0
        a(i,i) = a(i,i) - a(i,j)^2
    end
  end
```

Algorithm 2 – $ILLT(p)$ factorization modified by C. -J. Lin and J. J. Morė.

In the first loop *j* column of forming matrix $L^{(j)}$ are calculated with respect to the elements l_{ij} for $i > j$ and a necessary correction of diagonal elements l_{ii} for $i > j$ is done in the second loop. A drop-tolerance strategy is implemented between two loops so that insignificant elements formed in the first loop do not alter the diagonal elements l_{ii}. Therefore the contortions admissible while dropping unimportant elements do not influence the corresponding diagonal elements and the result errors $R = A - LL^T$ become smaller. In this algorithm an indirect error distribution is possible only through non-diagonal elements. This helps to form better preconditioners for conjugate gradients method.

Algorithm 2 has three essential disadvantages:

- Incomplete usage of limited amount of memory (limit on memory usage is set by *p* parameter) for such a *j* column of matrix $L^{(j)}$, which has a filling less than $n_j + p$, where n_j – is a number of non-zero elements in *j*- column of matrix *A*;
- As parameter *p* can not take negative values, the formed preconditioner *L* can not be placed in less amount of memory than is needed to store matrix *A*;
- Incomplete losses of elements in different columns of matrix *L*, does not permit the exhaustive usage of the memory resources provided for high-quality computation of incomplete Cholesky factorization.

We propose a $ILLT(m, p)$ Cholesky factorization with adaptive τ – drop-tolerance strategy without disadvantages described above. The errors $R = A - LL^T$ are proved to be less than in algorithm 2.

The $ILLT(m, p)$ factorization allows forming matrix *L* within an amount of memory provided without positioning of the initial matrix *A*. The memory size provided to matrix *L* is chosen independently from the memory size occupied by matrix *A* and is set by parameter *m*, which fixes the relation of memory sizes occupied by both matrices. Consider a lower-triangle part of symmetric matrix *A* contains *nnz* non-zero elements. Then the parameter *m* can assume any values so that $m \geq \frac{n}{nnz}$. Obviously in a partial case if $m = \frac{n}{nnz}$, the matrix *L* is diagonal matrix with *n* elements.

The fact that an input matrix *A* and preconditioner *L* are stored in memory simultaneously is not a desirable condition to implement an iterative solution of problem (1) by conjugate gradients method. That is why the refusal to implement factorization in place of positioning of matrix *A* is not only justified, but it also permits even more effective usage of memory resources, provided.

The structural analysis of elements of intermediate matrices $L^{(j)}$ (5) – (7) formed by *LLT* Cholesky factorization confirms the existence of such memory resources.

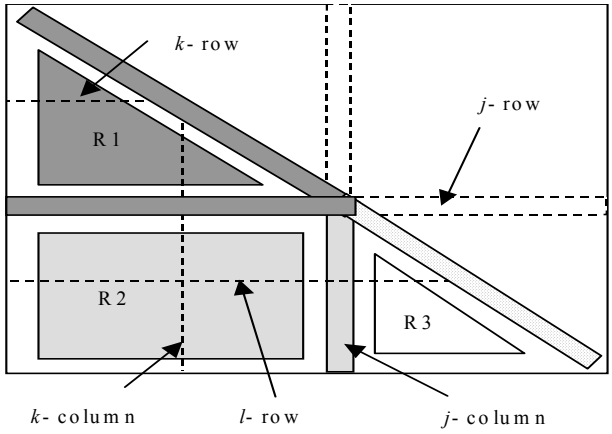

Fig. 1 – Matrix $L^{(j)}$ after *j* steps of factorization.

Without taking into account the positioning of zero elements in matrix $L^{(j)}$, we distinguish three groups of elements R1, R2 and R3, forming the lower-triangle partition of matrix, and lower and upper diagonal elements. Diagonal elements marked in dark-grey, elements of R1, group and elements of row *j*, contain result elements of matrix *L*, which do not change and are not used during the further process of formation of matrices $L^{(j+1)}$, $L^{(j+2)}$, . . . $L^{(n)} = L$. Elements of R2 group marked in light grey and elements of *j* column are also the result elements, which do not change but they are still needed to form the elements of R3 group and lower diagonal elements. Ineffectiveness of memory usage is estimated by measuring the size of R3 group, where corresponding elements of *A* placed initially are stored. It is obvious that the size of R3 group depends on the step *j* of factorization process. During the first steps the size of R3 group is comparable to the size of the whole lower-triangle partition of matrix *A*, but it reduces linearly to zero during the next steps.

As R3 group consists of elements of separately stored matrix *A*, there is no need to store the same elements twice. The released amount of memory can

be used to store a bigger number of elements of matrices $L^{(0)}$, $L^{(1)}$, … $L^{(n)} = L$ and therefore compute an incomplete Cholesky factorization with lower level of losses. We will denote the stored matrices that do not contain R3 group as $\mathbf{L}^{(0)}$, $\mathbf{L}^{(1)}$, … $\mathbf{L}^{(n)} = L$.

Another special feature of $ILLT(m, p)$ Cholesky factorization proposed is the usage of an adaptive $\tau-$drop-tolerance strategy. This drop-tolerance strategy allows coordinating memory requirements for matrices $\mathbf{L}^{(0)}$, $\mathbf{L}^{(1)}$, … $\mathbf{L}^{(n)} = L$ formed gradually with the possibility to place the elements of these matrices in memory actually provided. Normally, during the first steps of factorization memory requirements for formed $\mathbf{L}^{(0)}$, $\mathbf{L}^{(1)}$, … matrices storage do not exceed a provided memory limit. Consequently, at the beginning any element losses may be admissible and we may actually start with *LLT* Cholesky factorization. Factorization with $\tau = 0$ continues until the deficit of memory for distributed of following $\mathbf{L}^{(j)}$ matrix is arisen. Then if we increase τ and set $l_{ik} = 0$ for the such elements l_{ik} that

$$\left\{ \left| l_{ik} \right| \le \tau \cdot l_{kk};\ i \in (k+1, n), k \in (1, j) \right\} \qquad (8)$$

we can decrease $\mathbf{L}^{(j)}$ matrix's filling to an appropriate level and provide an opportunity to place the rest of non-zero elements in provided memory. It is obvious that τ should be increased gradually and filling of matrix $\mathbf{L}^{(j)}$ with non-zero elements should be strictly observed. A new value of τ parameter should be selected so that we can find an admissible filling of matrix $\mathbf{L}^{(j)}$ as soon as possible and, from the other side, do not permit redundant losses and unjustified big errors in matrices $\mathbf{L}^{(j)}$, $\mathbf{L}^{(j+1)}$, …, $\mathbf{L}^{(n)} = L$. All these conflicting requirements satisfy the following correspondence for τ

$$\Delta\tau = \max\left(\tau_s,\ \tau \cdot \frac{n-j}{n} \right). \qquad (9)$$

Here the initial value of incremental growth $\Delta\tau = \tau_s \neq 0$ is set by expert way. Usually it is sufficiently small and does not lead to any significant losses. Further, the relative growth rate of τ parameter is set to be equal to $\frac{\Delta\tau}{\tau} = \frac{n-j}{n}$ for during the first steps of factorization for $j << n$ and the low value of τ closed to τ_s we can set admissible level of losses quickly but not accurately. During the last steps of factorization if $j \to n$ and the value of τ is high one must not allow unjustified losses.

Two blocking strategies, current and retrospective, resist against further distribution and accumulation of errors occurring as a result of using the adaptive $\tau-$losses strategy. We use the blocking strategies to decrease negative influence of dropped non-zero elements of R2 group, as they directly influence the elements of the lower diagonal group and indirectly influence the elements of adjacent columns from the left of R2 group.

The current blocking strategy is needed on the current j step of factorization with regard to the elements of j column and drop-tolerance criteria $\left\{ \left| l_{ij} \right| \le \frac{\tau}{2} \cdot l_{jj};\ i \in (j+1, n) \right\}$ can be used, which differs from criteria (8), as τ is twice less. The correction of diagonal elements l_{ii} on the loss value l_{ij} is not allowed according to the algorithm 2 and the explanations in paper [2]. Note that the drop-tolerance strategy with regard to elements of *j* row is constant.

The retrospective blocking strategy is used after the next incrementation of τ parameter. This strategy supposed a partial recovering value of the elements of lower-diagonal group $\{ l_{kk};\ k \in (j+1, n) \}$. The square values of l_{ki} elements, which now satisfy the condition $\{ l_{ki} \le \tau;\ k \in (j+1, n);\ i \in (1, j) \}$, were subtracted from $\{ l_{kk};\ k \in (j+1, n) \}$ earlier. The recovering of corresponding values of l_{kk} involves the addition of values l_{ki}^2.

The details of implementation of these blocking strategies are shown in algorithm 3 presented below. Here $\tau-$ and $p-$ are drop-tolerance strategies used concurrently. The $p-$strategy is additional and used in special cases to influence the amount of calculations for $ILLT(m, p)$ factorization. If p parameter is small, then $p-$ drop-tolerance strategy predominates and algorithm 3 becomes close to algorithm 2 in respect of its possibilities. If parameter p satisfies condition $p + n_j = n - j$, $p-$ drop-tolerance strategy is inoperative. Such an unusual application of $p-$strategy could be ineffective in combination with equation- and variable- sorting algorithms, which are not considered.

Select $\tau_s > 0$
Set mt = m*nnz
Set $\tau = \tau_s$
Copy the diagonal elements from *A* to *L*
for j = 1:n

```
L(j,j) = sqrt( L(j,j) )
Copy the j-column from A to L
col_len = size( i > j: L(i,j) ≠ 0 )
for k = 1:(j-1) & L(j,k) ≠ 0
      for i = (j+1):n & L(i,k) ≠ 0
            L(i,j) = L(i,j) - L(i,k)* L(j,k)
      end
end
for i = (j+1):n & L(i,j) ≠ 0
      L(i,j) = L(i,j)/L(j,j)
end
Retain the largest col_len + p elements in
L((j+1):n,j).
Erase the smallest q nonzero elements on
conditions:
      abs( L((j+1):n,j) ) ≤ τ *L(j,j)/2;
      abs( L(j,1:j-1) ) ≤ τ *L(j,j).
mt = mt – ( col_len + p ) + q
while( mt < 0 )
   τ = τ + τ *(n-j)/n
   for k = 1:(j-1)
         Erase the smallest q nonzero elements
         on conditions:
            abs( L(k,1:k-1) ) ≤ τ *L(k,k),
            abs( L(J=(k+1):n,k) ) ≤ τ *L(k,k)
         and for J > j restore diagonal
elements:
            L(J,J) = L(J,J) + L(J,k)^2
   end
   mt = mt + q
end
for i = (j+1):n & L(i,j) ≠ 0
      L(i,i) = L(i,i) - L(i,j)^2
end
end
```

Algorithm 3 – $ILLT(m,p)$ **factorization with adaptive** $\tau-$ **drop-tolerance strategy**

3. SCALING AND SHIFTING

The accumulation of errors often results in losses of positive definiteness of formed matrices $\mathbf{L}^{(j)}$ on the step j of factorization. In this case, at least one negative element appears among the elements of the lower diagonal group. There are a few methods to preserve a positive definiteness property of matrices $\mathbf{L}^{(j)}$ formed [1 – 2, 5, 10]. We use only one method, based on scaling and shifting procedure.

Choose $\alpha_s > 0$ and $p \geq 0$.

Compute $A = N^{-1/2} A N^{-1/2}$ where $N = diag\ (\|Ae_i\|_2)$.

Set $\alpha_0 = 0$ if $\min(a_{ii}) > 0$ else $\alpha_0 = -\min(a_{ii}) + \alpha_s$.

For $k = 0, 1, \ldots,$

Use algorithm $ILLT(m,p)$ on

$A_k = A + \alpha_k I$

if successful set $\alpha_F = \alpha_k$ and exit.

Set $\alpha_{k+1} = \max(2\alpha_k, \alpha_s)$.

Algorithm 4 – $ILLT(m,p)$ **factorization with scaling and shifting procedure.**

In this algorithm scaling is based on l^2-norm of columns for the input symmetric matrix A normalization, and shifting by the value $\alpha_F \geq 0$ of proper values of the normalized matrix A. Shifting is used to compute the $ILLT(m,p)$ factorization of newly formed positive definite matrix $A_F = A + \alpha_F I$.

4.CONJUGATE GRADIENTS METHOD

As a result of scaling and shifting the matrices and vectors corresponding the problem (1) can be rewritten as following:

$$\min\left\{ B^T X + \frac{1}{2} X^T A X : \left\| DX \right\|_2 \leq \Delta \right\}, \quad (10)$$

where $B = N^{-1/2} B$, $X = N^{1/2} X$, $D = N^{-1/2} D$, $A = N^{-1/2} A N^{-1/2}$, $N = diag(\|Ae_i\|_2)$. Then, according to Steihaug assumption [11], correspondence for the trust region in problem (10) is:

$$\left\| DX \right\|_2 = \left(X^T A_F X \right)^{1/2} = \|x\|_2 \leq \Delta \quad (11)$$

Here $A_F = LL^T$; $x = L^T X$; matrices L and L^T are defined by algorithms 3 – 4.

With respect to (11) the problem (10) is

$$\min\left\{ \left(L^{-1}B\right)^T x + \frac{1}{2} x^T \left(L^{-1} A L^{-T}\right) x : \| x \|_2 \leq \Delta \right\}, \quad (12)$$

where preconditioner L provides proper values of A matrix clustering. Algorithm 5 reflects the main features of the computation for the solution of problem (12), with a preconditioner L [1, 5, 6, 11].

Choose initial approximation $x_0 \in \mathbb{R}^n$.

Set $R_0 = -(L^{-1}B + L^{-1} A L^{-T} x_0)$ and $P_0 = R_0$.

For $k = 0, 1, \ldots n,$

Compute $\eta_k = P_k^T L^{-1} A L^{-T} P_k$;

In case $\eta_k \leq 0$ exit loop;

Compute $\mu_k = \|R_k\|^2 / \eta_k$;

Define more precisely $x_{k+1} = x_k + \mu_k P_k$;

Define more precisely

$R_{k+1} = R_k - \mu_k L^{-1} A L^{-T} P_k$;

In case $\left\| N^{1/2} \cdot R_{k+1} \right\|^2 \le \|\sigma \cdot B\|^2$ exit loop;

Compute $\nu_k = \|R_{k+1}\|^2 / \|R_k\|^2$;

Define more precisely $P_{k+1} = R_{k+1} + \nu_k P_k$.

Algorithm 5 – Conjugate gradients method with preconditioner *L*

Here the exit loop conditions correspond to (2). Using the correspondence $X = N^{-1/2} L^{-T} x$ the final solution x of the problem (12) can be easily transformed to the solution X of the problem (1).

5.CONCLUSION

The $ILLT(m,p)$ factorization algorithm was used for solving of the system of form $AX+B=0$ with matrices A taken from "Harwell-Boeing" collection and with the unity vector B.

The choice for comparing of the factorization $ILLT(p)$ algorithm was made due to its superficiality over the algorithms implemented in the code *ma31* from the Harwell program library (Release 10) and in the routine *cholinc* from the *Matlab* package (Version 5) [2].

The experiments showed the unpredictability of the memory consumed in fact by the $ILLT(p)$ algorithm at different values of p. On the contrary, the memory demand of the $ILLT(m,p)$ algorithm correlated well with the memory actually consumed, moreover the memory needed was substantially less than the memory occupied by matrices A. The convergence of the iterative procedures in the conjugate gradients method with the preconditioner L formed by $ILLT(m,p)$ algorithm was substantially higher than those preconditioner L formed by the $ILLT(p)$ algorithm. Such a difference was especially noticeable for badly conditioned matrices.

6.REFERENCE

[1] *A. Bouaricha, J. Morė, Z. Wu. Preconditioning Newton's Method, Rice University, Center for Research on Parallel Computation, Houston, TX, May 1998, 21 p.*

[2] *C. Lin, J. Morė. Incomplete Cholesky factorizations with limited memory, SIAM Journal on Sci. Comput., No.1 (1999). pp. 24–45.*

[3] *E. Chow, Y. Saad. Experimental study of ILU Preconditioners for indefinite matrices, Department of Computer Science and Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, June 1997. p. 32.*

[4] *N. Li, Y. Saad, E. Chow. Crout versions of ILU for general sparse matrices. Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN; Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA, April 2002. p. 17.*

[5] *Y. Saad. Iterative Methods for Sparse Linear Systems. University of Minnesota, Department of Computer Science and Engineering, Minneapolis, MN, 2000, 447 p.*

[6] *A. George, J. Liu. Computer Solution of Large Sparse Positive Definite Systems, Moscow: Mir, 1984. p. 333.*

[7] *S. Pissanetzky. Sparse Matrix Technology. Moscow: Mir, 1988. p. 410.*

[8] *B. Averick, J. Morė. Evaluation of large-scale optimization problems on vector and parallel architectures, SIAM Journal on Optimization, No.4, 1994, pp.708-721.*

[9] *C. Schelthoff, A. Basermann, Polynomial Preconditioning for the Conjugate Gradient Method on Massively Parallel Systems, Informatik-Bericht, No.1, 1995, pp. 150-167.*

[10] *J. Dennis, R. Schnabel. Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Moscow: Mir, 1988. p. 440.*

[11] *T. Stehiaug, The conjugate gradient method and trust regions in large-scale optimization, SIAM Journal on Numerical Analyses, No.20, 1983. pp.626–637.*

[12] *M. Jones, P. Plassmann. An improved incomplete Cholesky factorizations. ACM Trans. Math. Software, No.21, 1995, pp.5-17.*

[13] *N. Munksgaard. Solving sparse symmetric sets of linear equations by preconditional conjugate gradients, ACM, Trans. Math. Software, No.6, 1980, pp.206–219.*

[14] *I. Gustafsson, A class of first order factorization methods. BIT. No.18, 1978. pp.142–156.*

Sergey Saukh *was graduated from Civil Aviation University, Kiev, Ukraine in 1978 as M.S. in Radio Engineering.*

1981 - Ph.D. in Electrical Engineering, Power Engineering University,Moscow, Russia

Specialization: operational numerical methods in analysis of digital filtration processes.

1991- Dr.S. in Numerical Mathematics, Ukrainian Academy of Sciences,Kiev, Ukraine

Specialization: block operational numerical methods in the theory of mathematical modeling and simulation.

1997- International Monetary Fund (IMF) Certificate on Macroeconomic Analysis and Budget Forecastin.

Areas of interests:

- *theory of similarity of processes with different physical nature;*
- *numerical operational methods for solving ordinary and partial differential equations;*
- *technology for sparse matrix and optimization methods;*
- *spectral theory, especially time-frequency signal analysis and digital signal processing;*
- *macroeconomic dynamic theory and methods of econometric modeling, in financial engineering and derivative analytics*