# Nested Alternating - Triangular Incomplete Factorization Methods

# Nested Alternating - Triangular Incomplete Factorization Methods

**S V Gololobov[1,2], V P Il'in[1,2], A M Krylov[1] and A V Petukhov[1]**

[1] Institute of Computational Mathematics and Mathematical Geophysics SB RAS, pr. Akademika Lavrentjeva, 6, Novosibirsk, 630090, Russia
[2] Novosibirsk State University, Pirogova st., 1, Novosibirsk, Russian Federation, 630090

E-mail: ilin@sscc.ru

**Abstract.** We consider several versions of incomplete nested factorization methods for solving the large systems of linear algebraic equations (SLAEs) with sparse matrices which arise in grid approximations of the multi-dimensional boundary value problems. Our approach is based on the two-level iterative process in the Krylov subspaces in 3D case. Corresponding hierarchical incomplete factorization is applied to the block tridiagonal matrix structure. At the upper level, the diagonal blocks correspond to 2D grid subproblems which are factorized in the line-by-line framework. Instead of the low and upper triangular matrix factors, the alternating triangular matrices are used, which allows to apply the parallel counter sweeping approaches. The improvement of preconditioners is made by means of generalized compensation principles. To solve SLAE iterative conjugate direction methods in Krylov subspaces are applied. The efficiency of the proposed methods are demonstrated on the set of representative test problems.

## 1. Introduction

The incomplete factorization methods both explicit and implicit constitute a classical approach for construction of effective iterative preconditioned methods in Krylov subspaces for solving large systems of linear algebraic equations (SLAEs) with sparse ill-conditioned matrices that arise in grid approximations of multi-dimensional boundary value problems (BVP) using finite difference, finite volumes, finite elements algorithms or discontinuous Galerkin methods, see [1]. The existing wide variety of relevant constructive ideas can be found in books [2]-[7] as well as in reviews [8]-[10].

In this paper, we consider the combined use of three different algorithmic techniques for constructing economical, easily implementable, and parallelizable iterative processes. The first one is the nested factorization method proposed in 1981 by J. Appleyard and I. Cheshire in [11], and then successfully applied and developed by various authors, see [12]-[16]. In particular, such algorithms are actively used in [17] and other software packages for modeling oil production processes. The second trick is to reorder the unknowns in tridiagonal (scalar or block) SLAEs in such a way as to switch from ordinary recursive runs to counter sweeping [18]. Such approach is called twisted decomposition in [16] and is considered in [19]. In particular, such approach provides the opportunity to obtain high performance based on multi-thread technologies. The third method used to improve the preconditioner is reconciliation of matrix row sums, or compensation principle, which is known from [2], [4], [20], and actively used, in particular,

in the variant of matching column matrix sums and called the low-frequency filtering method [19].

The aim of our research is a comparative experimental analysis of the effectiveness of the constructed algorithms using model examples for solving 2D and 3D grid boundary value problems. In Section 2, we describe the iterative methods under consideration. Section 3 is devoted to the analysis of their computational resource consumption and scalability of parallelization, and presents the results of numerical experiments. In conclusion, an outlook for the methods studied and plans for future research are discussed.

## 2. Description of two-level methods of incomplete factorization

We consider the problems of constructing and studying fast converging and easily parallelizable iterative methods in Krylov subspaces for solving SLAEs

$$Au = f, \quad A \in \mathcal{R}^{N,N}, \quad u, f \in \mathcal{R}^N, \tag{1}$$

with high-order real sparse matrices ($N \approx 10^{10}$ and higher), having large condition numbers ($10^{13}$ and above), the implementation of which on modern multiprocessor computing systems (MCS) is a lasting problem. In particular, when solving direct and inverse interdisciplinary problems of mathematical modeling with real data including non-linear and non-stationary, this computational stage can take about 80 % of the time of a machine experiment since here the volumes of spent computer resources grow non-linearly with an increase in the degrees of freedom.

We are mainly interested in SLAEs that arise from approximation of multidimensional initial-boundary value problems with variable coefficients and contrasting material properties using the methods of finite differences, finite volumes, finite elements, or discontinuous Galerkin algorithms [1]. It is assumed that in such cases special techniques such as fast Fourier transform are not directly applicable. The main approaches are based on preconditioned iterative algorithms in Krylov subspaces. A typical form of an easily invertible preconditioned matrix is approximate factorization of the form

$$B = (G + L)G^{-1}(G + U), \tag{2}$$

where $L$ and $U$ are the lower and upper triangular parts of the original matrix $A = D + L + U$, and $D$ is some block-diagonal or diagonal matrix.

Various methods of symmetric successive over relaxation (SSOR) as well as explicit and implicit incomplete factorization are constructed based on formulas (2), see the review in [3]. One of the general approaches to accelerating iterations is the compensation principle, or the match of row sums which consists in selecting the matrix $G$ in such a way to satisfy the conditions

$$By^{(l)} = Ay^{(l)}, \quad l = 1, \ldots, m, \tag{3}$$

on a given set of $m$ test vectors, see [1, 3] and the literature cited there. In some works, for example, in [14] - [16], this technique is also called filtering.

To satisfy the conditions $B \approx A$ and (3), the matrix $G$ is sought in the form

$$G = D - \overline{LG^{-1}U} - \theta S, \tag{4}$$

where $\theta \in [0, 1]$ is some compensation parameter, the bar over the matrix means some band approximation, and $S$ is the block-diagonal matrix formed by the following conditions (the implementation for $m = 2$ is described in [20]):

$$Sy^{(l)} = (LG^{-1}U - \overline{LG^{-1}U})y^{(l)}, l = 1, \ldots, \bar{m}. \tag{5}$$

Let us note that if $y^{(l)} = e$ ($e$ is the vector with unit elements on all positions) the condition (2) is called row sum criteria. In some papers the column sum criteria $eB = eA$ is used for the non-symmetric matrices $B, A$ instead of this one.

Note that in [16] two-thread block version of the alternating triangular factorization of the original matrix was proposed when left and right factors in [2] are not lower or upper triangular matrices, but they consist of block rows of different placement: some are lower triangular and the remaining ones are upper triangular (such decomposition is called "twisted decomposition" by the authors, and in Russian papers this approach is traditionally called as an algorithm of counter sweeping, see the reviews in [4, 18]).

We give a description of alternating triangular factorization methods using five-point approximations of two-dimensional boundary value problems for diffusion-type equations on a rectangular grid with the number of nodes $N = N_x N_y$, see [1] - [4].

With the natural ordering of the grid nodes (for definiteness, we choose their global numbering by the index $k = j + (i-1)N_y$, $i = 1, \ldots, N_x$, $j = 1, \ldots, N_y$), the matrix of the system takes a block-tridiagonal form

$$A = \begin{bmatrix} D_1 & U_1 & & & 0 \\ L_2 & D_2 & U_2 & & \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ & & & & U_{N_x-1} \\ 0 & & & L_{N_x} & D_{N_x} \end{bmatrix}. \tag{6}$$

Here $D_i$, $L_i = U_{i-1}^T$ are the tridiagonal and diagonal matrices.

In formulas (2), (4), the matrices $L$ and $U$ are defined as alternating triangular which for $N_x = 7$ have the following form:

$$L = \begin{bmatrix} 0 & 0 & & & & & \\ L_2 & 0 & 0 & & & 0 & \\ & L_3 & 0 & 0 & & & \\ & & L_4 & 0 & U_4 & & \\ & & & 0 & 0 & U_5 & \\ & 0 & & & 0 & 0 & U_6 \\ & & & & & 0 & 0 \end{bmatrix}, \; U = \begin{bmatrix} 0 & U_1 & & & & & \\ 0 & 0 & U_2 & & & 0 & \\ & 0 & 0 & U_3 & & & \\ & & 0 & 0 & 0 & & \\ & & & L_5 & 0 & 0 & \\ & 0 & & & L_6 & 0 & 0 \\ & & & & & L_7 & 0 \end{bmatrix}. \tag{7}$$

In this case, the matrix $G$ is defined as block-diagonal, and its blocks $G_i$ are defined as tridiagonal calculated from counter recursions which we write for an arbitrary odd $N_x = 2m+1$

$$\begin{gathered} G_1 = D_1, \; G_i = D_i - L_i \bar{G}_{i-1}^{-1} U_{i-1} - \theta S_i, \\ S_i y^{(l)} = L_i (G_{i-1}^{-1} - \bar{G}_{i-1}^{-1}) U_{i-1} y^{(l)}, \; i = 2, \ldots, m, \\ G_{N_x} = D_{N_x}, \; G_i = D_i - U_i \bar{G}_{i+1}^{-1} L_{i+1} - \theta S_i, \\ S_i y^{(l)} = U_i (G_{i+1}^{-1} - \bar{G}_{i+1}^{-1}) L_{i+1} y^{(l)}, \; i = N_x - 1, \ldots, m+2, \\ G_{m+1} = D_{m+1} - L_{m+1} \bar{G}_m^{-1} U_m - U_{m+1} \bar{G}_{m+2}^{-1} L_{m+2} - \theta S_{m+1}, \\ S_{m+1} y^{(l)} = [L_{m+1} (G_m^{-1} - \bar{G}_m^{-1}) U_m + U_{m+1} (G_{m+2}^{-1} - \bar{G}_{m+2}^{-1}) L_{m+2}] y^{(l)}. \end{gathered} \tag{8}$$

Here $\bar{G}_{i-1}^{-1}$ means the tridiagonal part of the matrix $G_{i-1}^{-1}$, and in the simplest case with one compensating or filtering vector we have $l = 1$ (usually $y^{(1)} = e$) and two test vectors $l = 1, 2$, see [20].

When using matrix B as a preconditioner for any iterative process, it is required at each step to solve an auxiliary system of the form

$$Bp = (G + L)G^{-1}(G + U)p = r. \tag{9}$$

Its solution can be found from equations

$$(G + L)v = r,\ (I + G^{-1}U)p = v, \tag{10}$$

which are implemented using counter matrix sweeps according to the following formulas:

$$\begin{aligned}
&v_1 = G_1^{-1}r_1,\ v_i = G_i^{-1}(r_i - L_i v_{i-1}),\ i = 2, \dots, m,\\
&v_{N_x} = G_{N_x}^{-1},\ v_i = G_i^{-1}(r_i - U_i v_{i+1}),\ i = N_x - 1, \dots, m + 2,\\
&v_{m+1} = G_{m+1}^{-1}(r_{m+1} - L_{m+1}v_m - U_{m+1}v_{m+2}) = p_{m+1},\\
&p_i = v_i - G_i^{-1}U_i v_{i+1},\ i = m, \dots, 1,\\
&p_i = v_i - G_i^{-1}L_i v_{i-1},\ i = m + 2, \dots, N_x.
\end{aligned} \tag{11}$$

In recursions (8) and (11), the stages of the calculations with an increasing and decreasing index $i$ are called direct and inverse sweeps, respectively. Obviously, their implementations can be performed in parallel on two threads.

To solve SLAEs with symmetric positive definite matrices A and B, we use left or right preconditioned conjugate direction methods

$$\begin{array}{ll}
p_0 = r_0 = B^{-1}(f - Au_0), & p_0 = r_0 = f - AB^{-1}u_0,\\
\alpha_i = \frac{(r_i,(B^{-1}A)^\gamma r_i)_B}{(B^{-1}Ap_i,(B^{-1}A)^\gamma p_i)_B}, & \alpha_i = \frac{(r_i,(AB^{-1})^\gamma r_i)_{B^{-1}}}{(AB^{-1}p_i,(AB^{-1})^\gamma p_i)_{B^{-1}}},\\
u_i = u_{i-1} + \alpha_{i-1}p_{i-1}, & u_i = u_{i-1} + \alpha_{i-1}B^{-1}p_{i-1},\\
r_i = r_{i-1} - \alpha_{i-1}B^{-1}Ap_{i-1}, & r_i = r_{i-1} - \alpha_{i-1}AB^{-1}p_{i-1},\\
\beta_i = \frac{(r_i,(B^{-1}A)^\gamma r_i)_B}{(r_{i-1},(B^{-1}A)^\gamma r_{i-1})_B}, & \beta_i = \frac{(r_i,(AB^{-1})^\gamma r_i)_{B^{-1}}}{(r_{i-1},(AB^{-1})^\gamma r_{i-1})_{B^{-1}}},\\
p_i = r_i + \beta_i p_{i-1}, & p_i = r_i + \beta_i p_{i-1},
\end{array} \tag{12}$$

where $\gamma = 0$ corresponds to the conjugate gradient method, and $\gamma = 1$ corresponds to conjugate residual, see [2], [3].

Now we consider nested factorization method for solving seven-diagonal algebraic equations approximating a three-dimensional Dirichlet boundary value problem of diffusion type in a parallelepiped computational domain on a uniform grid in Cartesian coordinates ($x = x_i$, $i = 0, 1, \dots, N_x + 1$; $y = y_j$, $j = 0, 1, \dots, N_y + 1$; $z = z_k$, $k = 0, 1, \dots, N_z + 1$), see [1].

We represent the matrix of the dimension $N = N_x N_y N_z$ in the form

$$A = D + L_1 + U_1 + L_2 + U_2 + L_3 + U_3, \tag{13}$$

where $D$ is diagonal, and $L_l$ and $U_l$, $l = 1, 2, 3$, are the lower and upper triangular matrices. Here, the indexes $l = 1, 2, 3$ correspond to the axes $x, y, z$, respectively. We define a preconditioning matrix $B$ by nested factorization of the form (9):

$$\begin{aligned}
B &= (P + L_3)P^{-1}(P + U_3) = P + L_3 + U_3 + L_3P^{-1}U_3,\\
P &= (T + L_2)T^{-1}(T + U_2) = T + L_2 + U_2 + L_2T^{-1}U_2,\\
T &= (M + L_1)M^{-1}(M + U_1) = M + L_1 + U_1 + L_1M^{-1}U_1,
\end{aligned} \tag{14}$$

resulting in

$$B = M + A - D + L_1 M^{-1} U_1 + L_2 T^{-1} U_2 + L_3 P^{-1} U_3. \tag{15}$$

With the natural ordering of the grid nodes, the matrices $M$, $T$, and $P$ are diagonal, tridiagonal, and pentadiagonal, respectively, and the preconditioner can be defined by the formulas

$$\begin{aligned} M &= D - L_1 M^{-1} U_1 - \theta_1 S_1 - \theta_2 S_2, \\ B &= A + L_2 T^{-1} U_2 + L_3 P^{-1} U_3 - \theta_1 S_1 - \theta_2 S_2. \end{aligned} \tag{16}$$

Here $\theta_1$ and $\theta_2$ are iterative (relaxing) parameters, and $S_1$ and $S_2$ are diagonal matrices determined from the conditions for matching the row sums of matrices $A$ and $B$:

$$S_1 e = L_2 T^{-1} U_2 e, \quad S_2 e = L_3 P^{-1} U_3 e. \tag{17}$$

The proposed three-level method of nested factorization can be structurally simplified by reducing it to a two-level one. To do this, we rewrite the original matrix A from (13) in the form

$$A = D_3 + L_3 + U_3, \quad D_3 = D_2 + L_2 + U_2, \quad D_2 = D + L_1 + U_1. \tag{18}$$

In this case, $D_3$ is a block-diagonal matrix with five-diagonal blocks $D_{3,i}$ of dimension $N_y N_z$, each of which corresponds to a plane problem in the cross section $x = const$ and has the structure similar to that of matrix A in (6). Then the matrix P in (14) is defined by the formula

$$P = \{G_i = D_{3,i} - \theta S_i\}, \quad S_i e = L_{3,i} G_{i-1}^{-1} U_{3,i} e, \tag{19}$$

which corresponds to the definition of $\overline{L_3 P^{-1} U_3} = 0$ in (14). Note that $L_3$ and $U_3$ can be defined as alternating triangular, and then the implementation of the algorithm can be parallelized using counter block sweeping.

If we put $\theta = 0$ in (19), then we arrive at the block symmetric successive over relaxation method (BSSOR, [3], [4]), which is an alternative to the compensation, or filtration, principle considered above. In this case, it is necessary in (14) to replace $P$ with $\omega^{-1} P$, where the relaxation parameter $\omega$ has an optimal value on the interval [1,2].

In fact, in this case, in (11), the inversion of the matrices $G_i$ is made iteratively by formulas similar to (12), which generates a two-level process. Note that each auxiliary two-dimensional SLAE has a strict diagonal dominance and a finite condition number, and the boundaries of its spectrum can be estimated using Gershgorin circles.

## 3. Numerical experiments

For a comparative analysis of the effectiveness of the iterative processes under consideration for various values of the computational parameters, we present the results of numerical experiments on a series of 2D and 3D BVPs for the Poisson equation with Dirichlet boundary conditions approximated by common schemes of the second order of accuracy on square or cubic grids.

All demonstrated calculations were performed with double precision. The criterion for stopping the iterations is based on the residual vector

$$(r^n, r^n) \le \varepsilon^2 (f, f), \quad \varepsilon \ll 1. \tag{20}$$

In the first Table, we present the number of iterations for the 3D problem in cube $\omega = [0,1]^3$ using the basic nested factorization method with the preconditioner (13) - (17), $\theta_1 = \theta_2 = \theta$, and conjugate gradient solver. Exact solution is $u(x,y,z) = \sin \pi x \sin \pi y \sin \pi z$, stopping criteria $\epsilon = 10^{-6}$ in (20), and initial guess is $u^0 = 0$. In the Table 2, we present similar results for the BSSOR method with the preconditioner

$$B = (G + L_3)G^{-1}(G + U_3), G = \frac{1}{\omega}D_3, \tag{21}$$

where $D_3$ is defined in (18) and $\omega \in [1, 2)$ is the relaxation parameter. Here, the 5-diagonal matrix $G$ is inverted approximately using iterative method IMIF531, see below.

In all cases, the number of internal iterations for $\epsilon_{int} = 10^{-8}$ were $n = 4, 5$ because of strong diagonal dominance of the matrix $D_3$. The minimal number of external iterations for the different grids are the following: for $N = 32, n = 17$ for $\omega_0 \in [1.56, 1.62]$; for $N = 64, n = 22$ for $\omega_0 \in [1.69, 1.80]$, and for $N = 128, n = 30$ for $\omega_0 \in [1.88, 1.91]$. If the relaxation parameter goes above $\omega_0$, then the number of iterations $n$ increases fast. In the Table 2, we denote such value as $n > 40$.

**Table 1.** The number of iterations for 3D problem. The classic method of nested factorization with conjugate gradients, $\varepsilon = 10^{-6}$.

| $N\backslash\theta$ | 0 | 0.95 | 0.975 | 0.9875 | 0.99375 | 1.0 |
|---|---|---|---|---|---|---|
| 32 | 17 | 11 | 11 | 11 | 12 | 14 |
| 64 | 29 | 16 | 15 | 15 | 15 | 20 |
| 128 | 54 | 26 | 25 | 22 | 21 | 30 |

**Table 2.** The results for $3D$ problem, SSOR two-level method, with IMIF531 internal iteration.

| $N\backslash\omega$ | 1 | 1.4 | 1.6 | 1.8 | 1.9 | 1.95 |
|---|---|---|---|---|---|---|
| 32 | 24 | 20 | 17 | >40 | >40 | >40 |
| 64 | 39 | 33 | 25 | 22 | >40 | >40 |
| 128 | 76 | 69 | 42 | 33 | 30 | >40 |

In the remaining Tables, the results of numerical experiments for 2D problem are demonstrated on square $[0, 1]^2$ with exact solution $u = \sin \pi x$ on uniform grids with $N^2$ nodes, $N = 128, 256, 512, 1024$. The Table 3 presents the number of iterations for the method IMIF531 (Implicit Incomplete Factorization, 5-diagonal matrix A, 3-diagonal matrix G, one filtration vector $e$ with unit entries in compensation principle (3)) with stopping criteria $\epsilon = 10^{-8}$, and standard triangular matrices $L, U$ in (2), for varying values of parameter $\theta$.

These results demonstrate that for any grid with the mesh size $h$, there is an optimal value of compensation parameter which can be estimated as $\theta_0 = 1 - O(h)$.

In the Table 4, we demonstrate the results for counter sweeping method with alternating triangular matrices $L, U$, see formulas (6)-(11). As we can see, the Counter Block Sweeping (CBS) method demonstrates an advantage versus conventional one, in the sense potential parallelizating for selected examples.

In the research conducted, we compared the number of iterations for the simple problems only. To speedup parallel computations, we suggest to apply the counter sweeping in three directions following the axes $x, y, z$ in 3D computational domain. The corresponding parallel implementation can be implemented based on the multi-threaded technologies using modern multi-core CPU together with the vectorization.

**Table 3.** The number of iterations for 2D problem. IMIF531 method with conjugate residuals, $\varepsilon = 10^{-8}$.

| $N\backslash\theta$ | 0 | 0.8 | 0.9 | 0.95 | 0.975 | 0.99 | 0.995 | 1.0 |
|---|---|---|---|---|---|---|---|---|
| 128 | 41 | 27 | 26 | 27 | 29 | 34 | 37 | 46 |
| 256 | 73 | 47 | 42 | 37 | 36 | 38 | 41 | 65 |
| 512 | 121 | 82 | 73 | 65 | 57 | 50 | 50 | 95 |
| 1024 | 208 | 137 | 121 | 108 | 98 | 84 | 75 | 132 |

**Table 4.** The number of iterations for a two-dimensional model problem. IMIF531_CBS method with conjugate residuals, $\varepsilon = 10^{-8}$.

| $N\backslash\theta$ | 0 | 0.8 | 0.9 | 0.95 | 0.975 | 0.99 | 0.995 | 1.0 |
|---|---|---|---|---|---|---|---|---|
| 128 | 36 | 34 | 34 | 39 | 47 | 59 | 70 | 94 |
| 256 | 65 | 57 | 53 | 52 | 57 | 73 | 89 | 193 |
| 512 | 104 | 101 | 97 | 92 | 89 | 93 | 112 | 416 |
| 1024 | 194 | 164 | 157 | 161 | 160 | 160 | 163 | 908 |

## 4. Conclusion

We present the results of numerical experimental analysis of several versions of nested factorization methods for solving 3D and 2D BVPs approximated on the structured grids. The classic nested factorization algorithm is combined with compensation principal, or generalized row sum criteria, which provide the band approximation of the matrix which is inverse to the original band matrix. This approach is responsible for the improvement of the condition number of the preconditioned SLAEs to be solved. The second approach consists in the reordering of the unknowns to obtain alternating triangular matrices which provide parallel three level implementation of the algorithm based on multi-threaded technologies and vectorization of the operations.

## Acknowledgments

The work is supported by RFBR grant $N$18-01-00295 in theoretical part an RSF grant $N$19-11-00048 in computational experiments part.

## References

[1] Il'in V P 2017 Mathematical Modeling. Part I. Continuous and Discrete Models (Novosibirsk: SBRAS Publ.) [in Russian]
[2] Axelsson O 1994 Iterative Solution Methods (Cambridge: Cambridge Univ. Press)
[3] Saad Y 2003 Iterative Methods for Sparse Linear Systems 2nd edn. SIAM
[4] Il'in V P 2007 Finite Element Methods and Technologies (Novosibirsk: ICM&MG SBRAS Publ.) [in Russian]
[5] Vassilevski P S 2008 Multi-Level Block Factorization Preconditioners (New York: Springer)
[6] Elman H C, Silvester D J and Wathen A J 2014 Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics Numerical Mathematics and Scientific Computation second edition , Oxford University Press (Oxford University Press)
[7] Olshanskii M A and Tyrtyshnikov E E 2014 Iterative Methods for Linear Systems Theory and Applications (Philadelphia: SIAM)
[8] Wathen A 2015 Preconditioning Acta Numer. **24** 329–376
[9] Hiptmair R 2006 Operator Preconditioning *J. Comput. Math. Appl.* **52 (5)** 699–706

[10] Benzi M 2002 Preconditioning Techniques for Large Linear Systems: a Survey *J. Comput. Phys.* **82 (2)** 418–477
[11] Appleyard J R, Cheshire I M and Pollard R K 1981 Special Techniques for Fully-Implicit Simulators *Proc. European Symposium on Enhanced Oil Recovery Bournemouth* pp 395-408.
[12] Appleyard J R and Cheshire I M 1983 Nested factorization *Proc. Seventh SPE Symposium on Reservoir Simulation* pp 315-324
[13] Kuznetsova N N, Diyankov O V, Kotegov S V, Krasnogorov I V, Pravilnikov V Y and Maliassov S Y 2007 The Family of Nested Factorizations *Russian Journal of Numer. Anal. Math. Mod.* **22** 393–412
[14] Kumar P, Grigori L, Nataf F and Nui Q 2016 On Relaxed Nested Factorization and Combination Preconditioning *International J. Comput. Math.* **93 (1)** 178–199
[15] Kumar P, Grigori L, Niu Q and Nataf F 2010 Fourier Analysis of Modified Nested Factorization Preconditioner for Three-Dimensional Isotropic Problems. [Research Report] INRIA - 00448291
[16] Wang R, Nu Q and Lu L 2009 A Twisted Block Tangential Filtering Decomposition Preconditioner *Math. Prob. In Engineering* 282307
[17] Using ECLIPSE Parallel. Parallel computing, 2003.
`www.sis.slb.com/me-dia/about/whitepaper_parallelcomputing.pdf`
[18] Il'in V P and Kuznetsov Yu I 1985 Three-diagonal matrices and their applications (M.: Nauka) [In Russian]
[19] Achdou Y and Nataf F 2007 Low frequency tangential filtering decomposition *Num. Lin. Alg. with Appl.* **14** 129–147
[20] ll'in V P and Laevsky K Yu 1997 Generalized Compensation Principle in Incomplete Factorization Methods *Russian J. Numer. Anal. and Mathem. Model.* **12 (5)** 399–412