ISSN 1995-4239, Numerical Analysis and Applications, 2025, Vol. 18, No. 2, pp. 148–156. © Pleiades Publishing, Ltd., 2025.
Original Russian Text © The Author(s), 2025, published in Sibirskii Zhurnal Vychislitel'noi Matematiki, 2025, Vol. 28, No. 2, pp. 171–183.

Algebraic-Geometric Multigrid Methods of Domain Decomposition

V. P. Il'in[1,2*]

[1] *Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch, Russian Academy of Sciences, Novosibirsk, Russia*

[2] *Novosibirsk State Technical University, Novosibirsk, Russia*

Received November 29, 2024; in final form, December 6, 2024; accepted January 15, 2025

Abstract—Some iterative processes in Krylov subspaces are considered for solving systems of linear algebraic equations (SLAE) with high-order sparse matrices that arise in grid approximations of multidimensional boundary value problems. The SLAE are preconditioned by a uniform combined method that includes domain decomposition and recursive application of a two-grid algorithm, which are implemented by forming block-tridiagonal algebraic and grid structures inverted by using incomplete factorization and diagonal compensation. Stability and convergence of iterations are studied for some Stieltjes systems. Parallelization and generalization of the methods to wider classes of relevant practical problems are discussed.

DOI: 10.1134/S1995423925020041

Keywords: *preconditioned Krylov methods, multidimensional problems, domain decomposition, multigrid approaches, incomplete factorization, diagonal compensation, parallelization of algorithms.*

Dedicated to the memory of Yuri Alekseyevich Kuznetsov

1. INTRODUCTION

In this paper, some efficient methods are studied for solving high-order SLAEs with sparse matrices that arise in grid approximations of multidimensional boundary value problems. This research area has remained important for many decades, since this branch of mathematical modeling has been most resource-intensive and made a significant contribution to both the efficiency and cost of large-scale computer experiments even on modern supercomputers. Basic computational tools for solving large ill-conditioned algebraic systems (with dimensions $> 10^9$ and condition numbers $\kappa > 10^{13}$) are preconditioned methods in Krylov subspaces. We consider symmetric SLAEs with various variational, projection, and orthogonal properties for which economical algorithms of conjugate directions with short recursions (conjugate gradients, conjugate residuals, and minimal errors) are used.

The convergence rate of the iterations is largely determined by the preconditioned matrix being used. This matrix must be easily invertible and have spectral properties that are sufficiently "close" to those of the matrix of the original SLAE. There exist three main approaches to choosing the matrix: One employs economical matrix decompositions, such as explicit and implicit methods of alternating directions, algorithms for symmetric successive over-relaxation and incomplete factorization with diagonal compensation, or row sum consistency.

The other two approaches are major modern computational methods for preconditioning of algebraic systems: These are domain decomposition methods (DDM) and algebraic multigrid (AMG) approaches. They are apparently the most widely used technological tools for solving SLAEs with sparse high-order matrices resulting from approximations on unstructured grids of multidimensional initial boundary value problems with complex geometry of calculation domains and contrast material properties. In a sense, DDM and AMG are alternative technologies, since the former are the main paralleling tools, and the latter are algorithms with asymptotically optimal order (see monographs and

* E-mail: ilin@sscc.ru

reviews [1–4]). In the present paper, some iterative processes in Krylov subspaces are considered to provide a unified representation of a preconditioned matrix with implementation of DDM and AMG based on a combination of block-tridiagonal recursive algebraic structures and successive use of incomplete factorization with diagonal compensation. These approaches are considered separately in [5, 6]. Note that initially a general idea of joint use of these two approaches was discussed in [7, 8].

The construction of iterative algorithms with a calculation domain divided into a large number of subdomains has two conflicting peculiarities: One is the possibility of parallel solution of the auxiliary SLAEs in the subdomains with an almost linear speedup of calculations. The other is the increase in the number of iterations with increasing number of the subdomains, with conventional use of an additive algorithm of the Schwartz–Jacobi type. This is because at the current iteration each subdomain exchanges information only with its adjacent neighbors.

In recent decades, the technologies of multigrid methods have been based on the principles of SLAE preconditioning. There is a variety of semi-empirical techniques, such as pre- and post-smoothing, reduction and prolongation, V- and W-cycles, coarse grid correction, etc. Such algorithms have their own widely used software implementations, mostly in the form of "black boxes."

The algebraic-geometric multigrid method of domain decomposition developed in the present paper is based on two simple principles: One is the construction of a recursive block-tridiagonal matrix structure by defining some geometric grid objects. For example, in some methods of decomposition (of the top, or zero, level) is divided into four subsets. These are: $\Omega_1^{(0)}$ (macronodes), $\Omega_2^{(0)}$ (macroedges), $\Omega_3^{(0)}$ (macrofaces) and $\Omega_4^{(0)}$ (internal nodes of the subdomains, which are divided into first-level subsets, $\Omega_4^{(0)} = \{\Omega_m^{(1)},\ m = 1, 2, \ldots, M\}$). With an appropriate numbering of the vector components, the matrix of the original SLAE becomes a block-tridiagonal one.

It is important that the right bottom block of this structure is a block-diagonal matrix. An incomplete factorization algorithm (multigrids are most efficient) can be used for approximate inversion of each diagonal block $A_m^{(1)}$, $m = 1, 2, \ldots, M$ [6]. The algorithm is based on the construction of nested grids and introduction of similar four subsets of grid nodes, but only at the "microlevel."

Let a fine grid $\Omega_m^{(1)}$ be constructed in the mth subdomain by halving the steps of a coarser grid $\Omega_m^{(2)}$. In this case, the matrices of the auxiliary SLAEs in the subdomains of $\Omega_m^{(1)}$ can also be brought to block-tridiagonal form and approximately factorized. Each right bottom block of this structure corresponds to an algebraic subsystem on sparse grid $\Omega_m^{(2)}$ in the subdomains. Then the two-grid procedure is implemented recursively: for each grid subset $\Omega_m^{(l)}$, $l = 1, 2, \ldots, L$ (where L is the number of nested grids and l is the level of nesting of each of them) a block-tridiagonal matrix $A_m^{(L)}$ is formed. This matrix is approximately factorized, and the resulting preconditioning matrix has a multi-level structure with nested factorizations. At the bottom L level, an exact LU-decomposition of the matrix $A_m^{(L)}$ is made in each of the M subdomains.

The paper is organized as follows. Section 2 provides some information on preconditioned methods of conjugate directions for solving block-tridiagonal SLAEs. Sections 3 and 4 are devoted to the construction of preconditioners based on domain decomposition and multigrid algorithms. Section 5 presents the results of some preliminary investigations on parallelization of the iterative algorithms on multiprocessor computing systems with distributed and shared memory.

2. ALGORITHMS OF CONJUGATE DIRECTIONS FOR BLOCK-TRIDIAGONAL SLAEs

For simplicity, consider symmetric positive definite (SPD) SLAEs in the following notation:

$$Au = f, \quad A = \{a_{t,s}\} \in \mathcal{R}^{N,N}, \quad u = \{u_t\}, \quad f = \{f_t\} \in \mathcal{R}^N. \tag{1}$$

Transforming, with the help of an SPD preconditioning factorized matrix $A \approx B = L_B U_B$, $U_B = L_B^\top$, the system (1) is brought to a double preconditioned form:

$$\bar{A}\bar{u} = \bar{f}, \quad \bar{A} = L_B^{-1} A U_B^{-1}, \quad \bar{u} = U_B u, \quad \bar{f} = L_B^{-1} f. \tag{2}$$

Let us present the iterative methods of conjugate directions for solving (2) as follows:

$$\bar{r}^0 = \bar{f} - \bar{A}\bar{u}^0, \quad \bar{p}^0 = \bar{r}^0, \quad n = 0, 1, \ldots,$$
$$\bar{u}^{n+1} = \bar{u}^n + \alpha_n^{(\gamma)}\bar{p}^n, \quad \bar{r}^{n+1} = \bar{r}^n - \alpha_n^{(\gamma)}\bar{A}\bar{p}^n, \tag{3}$$

where $\bar{u}^0$ is an arbitrary initial approximation, α_n^γ are iterative parameters, and $\bar{p}^n$ denote directing vectors satisfying the orthogonality conditions $\left(\bar{A}^\gamma\bar{p}^n, \bar{p}^k\right) = (\bar{p}^n, \bar{p}^k)_\gamma = \rho_n^{(\gamma)}\delta_{k,n}$, $\rho_n^{(\gamma)} = (\bar{p}^n, \bar{p}^n)_\gamma$. Here $\delta_{k,n}$ is the Kronecker symbol, and $\gamma = 0, 1, 2$ correspond to the algorithms of minimal errors, conjugate gradients, and conjugate residuals, respectively.

The orthogonality relations minimize functionals $\Phi_\gamma\left(\bar{r}^{n+1}\right) = \left(\bar{r}^{n+1}, \bar{r}^{n+1}\right)_{\gamma-2}$ in Krylov subspaces $\mathcal{K}_{n+1}\left(\bar{r}^0, \bar{A}\right) = \mathrm{Span}\left\{\bar{r}^0, \bar{A}\bar{r}^0, \ldots, \bar{A}^n\bar{r}^0\right\}$, which satisfy the following expressions (for details see [4]):

$$\Phi_\gamma\left(\bar{r}^{n+1}\right) = \Phi_\gamma\left(\bar{r}^0\right) - \sum_{k=0}^{n}\left(\sigma_k^{(\gamma)}\right)^2/\rho_k, \quad \sigma_k^{(\gamma)} = \left(\bar{r}^0, \bar{p}^k\right)_{\gamma-1},$$
$$\alpha_n^{(\gamma)} = \sigma_n^{(\gamma)}/\rho_n^{(\gamma)}, \quad \bar{p}^{n+1} = \bar{r}^{n+1} + \beta_n^{(\gamma)}\bar{p}^n, \quad \beta_n^{(\gamma)} = \sigma_{n+1}^{(\gamma)}/\sigma_n^{(\gamma)}.$$

The algorithms with two-sided preconditioning can be transformed to left or right preconditioning ones. For this it is sufficient to set in (2) $L_B = B, U_B = I$ in the first case and $L_B = I, U_B = B$ in the second case. In particular, an economical method of conjugate gradients with one-sided preconditioning for solving SLAE (1) is described by the formulas

$$r^0 = f - Au^0, \quad p^0 = B^{-1}r^n, \quad n = 0, 1, \ldots,$$
$$u^{n+1} = u^n + \alpha_n p^n, \quad r^{n+1} = r^n - \alpha_n Ap^n,$$
$$p^{n+1} = B^{-1}r^{n+1} + \beta_n p^n, \quad \alpha_n = \sigma_n/\rho_n,$$
$$\beta_n = \sigma_{n+1}/\sigma_n, \quad \sigma_n = (r^n, B^{-1}r^n), \quad \rho_n = (p^n, Ap^n). \tag{4}$$

In the preconditioned methods of conjugate directions, the condition $||r^n|| = (r^n, r^n)^{1/2} \le \varepsilon||f||$ is satisfied at a given $\varepsilon \ll 1$ and the required number of iterations is estimated as follows:

$$n(\varepsilon) \le \sqrt{\kappa}\,[(\log(2\varepsilon^{-1})]/2, \tag{5}$$

where κ is the spectral condition number of the matrix $B^{-1}A$.

To describe the algorithms, let us formally introduce the following family of block-diagonal fourth order matrices:

$$A^{(l)} = \begin{bmatrix} A_{1,1}^{(l)} & A_{1,2}^{(l)} & 0 & 0 \\ A_{2,1}^{(l)} & A_{2,2}^{(l)} & A_{2,3}^{(l)} & 0 \\ 0 & A_{3,2}^{(l)} & A_{3,3}^{(l)} & A_{3,4}^{(l)} \\ 0 & 0 & A_{4,3}^{(l)} & A_{4,4}^{(l)} \end{bmatrix}, \quad l = 0, 1, \ldots, L, \quad A^{(0)} = P_0 A P_0^\top. \tag{6}$$

Here P_0, $P_0^\top$ are some permutation matrices providing the above structure, and l is an index that will be used for a unified representation of DDM and AMG methods in the further discussion. For decomposition we take $l = 0$, and $l = 1, \ldots, L$ are the grid numbers in each of the subdomains.

To approximate matrices of the form (6), we consider the following approximate block-factorized preconditioners:

$$A^{(l)} \approx B^{(l)} = \begin{bmatrix} G_{1,1}^{(l)} & 0 & 0 & 0 \\ A_{2,1}^{(l)} & G_2^{(l)} & 0 & 0 \\ 0 & A_{3,2}^{(l)} & G_3^{(l)} & 0 \\ 0 & 0 & A_{4,3}^{(l)} & G_4^{(l)} \end{bmatrix} \left(G^{(l)}\right)^{-1} \begin{bmatrix} G_1^{(l)} & A_{1,2}^{(l)} & 0 & 0 \\ 0 & G_2^{(l)} & A_{2,3}^{(l)} & 0 \\ 0 & 0 & G_3^{(l)} & A_{3,4}^{(l)} \\ 0 & 0 & 0 & G_4^{(l)} \end{bmatrix},$$

$$G_1^{(l)} = A_{1,1}^{(l)}, \quad G_2^{(l)} = A_{2,2}^{(l)} - \big(A_{2,1}^{(l)}\big(G_1^{(l)}\big)^{-1}A_{1,2}^{(l)}\big)_1 - \theta S_2^{(l)},$$

where the matrices $G_k^{(l)}$ are defined as follows:

$$S_2^{(l)}e_2 = \Big[A_{2,1}^{(l)}\big(G_1^{(l)}\big)^{-1}A_{1,2}^{(l)} - \big(A_{2,1}^{(l)}\big(G_1^{(l)}\big)^{-1}A_{1,2}^{(l)}\big)_1\Big]e_2,$$
$$G_3^{(l)} = A_{3,3}^{(l)} - \big(A_{3,2}^{(l)}\big(G_2^{(l)}\big)^{-1}A_{2,3}^{(l)}\big)_1 - \theta S_3^{(l)},$$
$$S_3^{(l)}e_3 = \Big[A_{3,2}^{(l)}\big(G_2^{(l)}\big)^{-1}A_{2,3}^{(l)} - \big(A_{3,2}^{(l)}\big(G_2^{(l)}\big)^{-1}A_{3,2}^{(l)}\big)_1\Big]e_3,$$
$$G_4^{(l)} = A_{4,4}^{(l)} - \big(A_{4,3}^{(l)}\big(G_3^{(l)}\big)^{-1}A_{3,4}^{(l)}\big)_1 - \theta S_4^{(l)},$$
$$S_4^{(l)}e_3 = \Big[A_{4,3}^{(l)}\Big(\big(G_3^{(l)}\big)^{-1}A_{3,4}^{(l)} - \big(A_{4,3}^{(l)}\big(G_3^{(l)}\big)^{-1}\big)A_{4,3}^{(l)}\Big)_1\Big]e_4. \tag{7}$$

Here $A_{k,k}^{(l)}$, $k = 1,2,3,4$, are square matrices of order $N_k^{(l)}$, and $S_k^{(l)}$ are diagonal matrices. $(C)_1$ denotes the diagonal part of the matrix C, $\theta \in [0,1]$ is a compensation parameter, and e_k are trial vectors of dimension $N_k^{(l)}$ (their components are typically taken equal to one). The use of the preconditioner (7) formally corresponds to an incomplete factorization method (IFM), see [9], with row sum consistency, that is, $B^{(l)}e = A^{(l)}e$ at $\theta = 1$, $e \in \mathcal{R}^N$. When using the factorized preconditioner $B^{(l)}$ in the conjugate gradient method (4) the auxiliary SLAEs of the form $Bq^{(l)} = r^{(l)}$ are solved by the following economical formulas:

$$G_1^{(l)}v_1^{(l)} = r_1^{(l)}, \quad G_2^{(l)}v_2^{(l)} = r_2^{(l)} - A_{2,1}^{(l)}v_1^{(l)}, \quad G_3^{(l)}v_3^{(l)} = r_3^{(l)} - A_{3,2}^{(l)}v_2^{(l)},$$
$$G_4^{(l)}v_4^{(l)} = r_4^{(l)} - A_{4,3}^{(l)}v_3^{(l)}, \quad q_4^{(l)} = v_4^{(l)}, \quad G_3^{(l)}w_3^{(l)} = A_{3,4}^{(l)}q_4^{(l)}, q_3^{(l)} = v_3^{(l)} - w_3^{(l)},$$
$$G_2^{(l)}w_2^{(l)} = A_{2,3}^{(l)}q_3^{(l)}, \quad q_2^{(l)} = v_2^{(l)} - w_2^{(l)}, \quad q_1^{(l)} = v_1^{(l)} - \big(G_1^{(l)}\big)^{(-1)}A_{1,2}^{(l)}q_2^{(l)}. \tag{8}$$

In what follows, when describing different stages of the computational process for DDM and AMG methods, some specific information about the preconditioners $B^{(l)}$ will be used.

As a model SLAE on a cubic grid with step h,

$$\Omega^h : x_i = ih, \quad i = 1,\dots,N_x, \quad y_j = jh, \quad j = 1,\dots,N_y, \quad z_k = kh, \quad k = 1,\dots,N_z,$$

we consider a seven-point system

$$(Au)_{i,j,k} = a_{i,j,k}^{(0)}u_{i,j,k} - a_{i,j,k}^{(1)}u_{i-1,j,k} - a_{i,j,k}^{(2)}u_{i,j-1,k} - a_{i,j,k}^{(3)}u_{i+1,j,k} - a_{i,j,k}^{(4)}u_{i,j+1,k}$$
$$- a_{i,j,k}^{(5)}u_{i,j,k-1} - a_{i,j,k}^{(6)}u_{i,j,k+1} = f_{i,j,k} \tag{9}$$

with a Stieltjes matrix (see [10] for details). It is important that the algebraic systems being considered in the present paper are of the nodal type: each grid node corresponds to one equation and one component of the unknown vector.

3. DOMAIN DECOMPOSITION METHODS ON MACROGRIDS

In classical decomposition algorithms, the solution of a boundary value problem in a "large" calculation domain are reduced to the solution of a set of auxiliary boundary value problems. Boundary conditions of Neumann–Neumann or Dirichlet–Dirichlet type are set at the contact boundaries (see [11–14] and the references therein).

To construct an algebraic domain decomposition method (ADDM) to solve SLAE (9) on grid Ω^h, we define a macrogrid

$$\Omega^h = \Omega^H : x_{i_1},\dots,x_{i_{M_x}};\ y_{j_1},\dots,y_{j_{M_y}};\ z_{k_1}\dots,z_{k_{M_z}},$$
$$1 < i_1 < i_{M_x} < N_x; \quad 1 < j_1 < j_{M_y} < N_y; \quad 1 < k_1 < k_{M_z} < N_z$$
$$\Omega^H = \Omega_V^H \bigcup \Omega_E^H \bigcup \Omega_F^H, \tag{10}$$

that includes sets Ω_V^H, Ω_E^H, Ω_F^H of macronodes, macroedges, and macrofaces, respectively, dividing the other nodes into M_D nonintersecting subdomains: $\Omega_D^h = \cup_m \Omega_m^h$, $m = 1, \ldots, M_D$.

According to the construction of these sets, we order the corresponding grid nodes as follows: first all macronodes, then the macroedge and macroface nodes and, finally, the internal nodes of the subdomains. As a result, the vectors from (1) are divided into sub-vectors $u = \left(u_1^\top, u_2^\top, u_3^\top, u_4^\top\right)^\top$ and $f = \left(f_1^\top, f_2^\top, f_3^\top, f_4^\top\right)^\top$. These, in turn, are divided into sub-vectors corresponding to individual macroedges, macrofaces, and subdomains. With this block structure of the vectors, the initial system (1) may be written as $A^{(0)}u = f$, where in the notation (6) the diagonal blocks $A_{1,1}^{(0)}$, $A_{2,2}^{(0)}$, $A_{3,3}^{(0)}$, $A_{4,4}^{(0)}$ are associated with macronodes, macroedges, macrofaces, and subdomains, respectively.

Let us present in more detail such an algebraic structure for a cubic grid calculation domain where the number of nodes is $N = N_c^3$, $N_c = N_x = N_y = N_z$, the number of macrocoordinates is $M_c = M_x = M_y = M_z$, and all macroedges have $N_e = (N_c - M_c)/(M_c + 1)$ nodes each. In this case, the dimensions of the above sets and diagonal blocks of the matrix $A^{(0)}$ are: $N_1 = M_c^3$, $N_2 = 3N_eM_e$, $N_3 = 3N_e^2M_F$, $N_4 = N_c^3M_D$, where $M_E = 3M_c(M_c - 1)$, $M_F = 3(M_c + 1)^2$, $M_D = (M_c + 1)^3$ are the numbers of macroedges, macrofaces, and subdomains, respectively, which coincide with the orders of the blocks of the matrices $A_{2,2}^{(0)}$, $A_{3,3}^{(0)}$, $A_{4,4}^{(0)}$, which are, in turn, block-diagonal and have three, five, and seven diagonals each. The construction of a preconditioner for this structure and solution of the corresponding SLAE are carried out by formulas (7), (8) at $l = 0$.

Note that the matrices $G_k^{(0)}$ have, in accordance with (7), the same portraits as $A_{k,k}^{(0)}$ for the corresponding values of k. It is natural to calculate their elements once before the start of iterations. The SLAE with the matrix $G_2^{(0)}$ is solved by a double sweep method [12]. It is solved separately for each macroedge and is easily paralleled. The auxiliary five-diagonal subsystems for the macrofaces are also naturally paralleled and can be solved by using an LU decomposition, since their orders are relatively small. The calculation of the elements of $G_3^{(0)}$ (corresponding to the boundary nodes adjacent to the macroedges) requires the calculation of the diagonal elements of inverses of tridiagonal matrices. This can be easily done with double sweep methods.

The solution of the SLAEs with block diagonal matrices $G_4^{(0)}$ for subdomains is the most time-consuming procedure. However, in each of them the seven-point subsystems are independent of each other and can be solved in parallel. Note only that in this case to determine the matrix elements corresponding to the nodes located near the boundaries it is necessary to find the diagonal elements of inverses of five-diagonal matrices.

Remark 1. The error of approximation of incomplete factorization can be reduced if in (2) to determine $G_3^{(0)}$ we use the following tridiagonal banded approximation instead of the diagonal one:

$$G_3^{(0)} = A_{3,3}^{(0)} - \left(A_{3,2}^{(0)}\left(\left(G_2^{(0)}\right)^{-1}A_{2,3}^{(0)}\right)_3 - \theta S_3^{(0)}\right),$$
$$S_3^{(0)}e_3 = \left[A_{3,2}^{(0)}\left(G_3^{(0)}\right)^{-1}A_{2,3}^{(0)} - \left(A_{3,2}^{(0)}\left(G_2^{(0)}\right)^{-1}A_{2,3}^{(0)}\right)_3\right]e_3, \tag{11}$$

which is easily implemented with double sweep methods. The portrait of the matrix $G_3^{(0)}$ (as in (7)) coincides with the portrait of $A_{3,3}^{(0)}$. Similarly, the incomplete factorization for a preconditioner can be corrected in calculating the matrix $G_4^{(0)}$. That is, instead of the diagonal approximation we can use the five-diagonal one:

$$G_4^{(0)} = A_{4,4}^{(0)} - \left(A_{4,3}^{(0)}\left(\left(G_3^{(0)}\right)^{-1}A_{3,4}^{(0)}\right)_5 - \theta S_4^{(0)}\right),$$
$$S_4^{(0)}e_4 = \left[A_{4,3}^{(0)}\left(G_3^{(0)}\right)^{-1}A_{3,4}^{(0)} - \left(A_{4,3}^{(0)}(G_3^{(0)})^{-1}A_{3,4}^{(0)}\right)_5\right]e_4. \tag{12}$$

In this case $\left(G_3^{(0)}\right)^{-1}$ can be approximated by Sparse Approximate Inverse (SPAI) algorithms (see reviews in [16] and [17].

4. MULTIGRID PRECONDITIONED SLAES IN SUBDOMAINS

The SLAE with the matrix $G_4^{(0)}$ can be solved with various direct algorithms. Let $B = B^{(0)}\big(G_4^{(0)}\big)$ be the corresponding preconditioner of the resulting "pure" DDM method, i.e., without the use of AMG. In the present paper, we use an approximation of the matrix $G_4^{(0)}$ (instead of its inversion) with a multigrid preconditioner. The preconditioner is defined by recursively applying the two-grid approach for all subdomains. Since $G_4^{(0)}$ is a block-diagonal matrix with seven-diagonal blocks on the main diagonal, it may be described as

$$G_4^{(0)} = A^{(1)} = \text{block-diag}\big\{A_m^{(1)},\ m = 1, \dots, M_D\big\}, \tag{13}$$

where each of the blocks $A_m^{(1)}$ is in block-tridiagonal form (6). This is done by constructing an appropriate algebraic structure, in accordance with a multigrid incomplete factorization method [6]. In the terminology of [18], we consider a purely algebraic multigrid approach without smoothing and post-smoothing operations, V-cycles and W-cycles, which are typical for the existing technologies described in the extensive literature on this topic (see [18–21] and the references therein).

In each of the subdomains Ω_m^h we construct a sequence of nested grids Ω_m^l, $l = 1, 2, \dots$, such that each finer grid Ω_m^l is obtained from coarser grid $\Omega_m^{(l+1)}$ by halving the spacings of the coarser grid. Let us divide the nodes for a two-grid algorithm into four subsets, defining a multigrid method as a recursive application of the two-grid one:

$$\Omega_m^1 = \Omega_1^{1,m} \bigcup \Omega_2^{1,m} \bigcup \Omega_3^{1,m} \bigcup \Omega_4^{1,m},$$

where $\Omega_4^{1,m} = \Omega_m^2$ is the set of fine grid nodes with spacing $2h$. The elements of the sets $\Omega_k^{1,m}$ belong to various topological primitives: cells, volumes, faces, edges, and nodes of the coarser grid Ω_m^2 for $k = 1, 2, 3, 4$.

Each node of the coarser grid is connected only with nodes from the set $\Omega_3^{1,m}$ (their number does not exceed six) lying on edges of the coarser grid. The set $\Omega_2^{1,m}$ denotes the set of nodes lying at the midpoints of the "coarse" faces, and $\Omega_1^{1,m}$ is the set of centers of "large" cells (each of them is connected with six "face" nodes if the grid is regular). Note also that in this case each "face" node is connected with two "volume" nodes, that is, with the centers of large cells (of the coarser grid), and with four "coarse" nodes, and each "edge" node is connected with four "face" nodes and two "coarse" nodes. Arranging the nodes on Ω_m^1 in accordance with the numbering of the above introduced subsets, for the associated subvectors we obtain, in each of the subdomains, a block-tridiagonal SLAE with a matrix of the form (6) at $l = 1$ (for brevity, the index m is omitted) $A^{(1)}u^{(1)} = f^{(1)}$. Here each of the vectors is represented by four subvectors $u_k^{(1)}$, $f_k^{(1)}$, $k = 1, 2, 3, 4$, and the first thus obtained three diagonal blocks $A_{k,k}^{(1)}$ of the matrix $A^{(1)}$ correspond to "volume," "face," and "edge" subsets, and the block $A_{4,4}^{(1)}$ corresponds to "coarse" nodes.

This matrix is approximated by a preconditioner of the form (7) at $l = 1$, where the formula for $G_4^{(l)}$ is replaced by

$$G_4^{(1)} = A_{4,4}^{(1)} - A_{4,3}^{(1)}\big(G_3^{(1)}\big)^{-1}A_{3,4}^{(1)}. \tag{14}$$

In this case $G_1^{(1)}$, $G_2^{(1)}$, $G_3^{(1)}$ turn out to be diagonal matrices, and $G_4^{(1)}$ is a semidiagonal matrix which corresponds to the system $A^{(2)}u^{(2)} = f^{(2)}$ on coarse grid Ω_m^2. If this SLAE is solved by a direct algorithm, the corresponding two-grid preconditioner is $B = B^{(0)}\big(B^{(1)}\big(G_4^{(1)}\big)\big)$.

Otherwise, a coarser grid, $\Omega^{2,m}$, is constructed in all subdomains. For this grid, a block-tridiagonal matrix $A^{(2)}$ of the form (6) at $l = 2$ is formed. This matrix is approximated by a factorized preconditioner by formulas (7) at $l = 2$, in which the matrix $G_4^{(2)} = A^{(3)}$ is defined similarly to (14). The resulting three-grid approach can be naturally generalized to the case where an arbitrary number, m, of grids is used in each subdomain. The corresponding preconditioner is denoted recursively as

$$B = B^{(0)}\big(B^{(1)} \dots B^{(m-1)}\big(G_4^{(m-1)}\big)\big), \tag{15}$$

where it is assumed that the last matrix, $G_4^{(L)}$, belongs to a SLAE on the coarsest grid and is solved by a direct method.

Remark 2. Strictly speaking, when solving a SLAE with matrix $A^{(2)}$ iteratively, we have a two-level nonlinear computational process. At the upper level, we should use the "more expensive" Flexible Conjugate Gradient (FCG) method [22], which stores all directing vectors $p^0, \ldots, p^n$, instead of the preconditioned conjugate gradient method (5). However, it can, naturally, be assumed that at a sufficiently high accuracy of an iterative solution of a SLAE in the subdomains the perturbation introduced as a result of an early termination of the iteration only weakly affects the convergence of the external iterative process.

The following results may be formulated for the conjugate gradient method with algebraic multigrid domain decomposition (CG-ADD-MG) defined by formulas (4), (7), and (11)–(14):

Theorem 1. *If A is a Stieltjes matrix in SLAE* (1)*, the preconditioning matrix B in* (4) *is also a Stieltjes matrix, and the following inequalities hold for $v \in \mathcal{R}^n$:*

$$\delta(Bv, v) \le (Av, v) \le \triangle(Bv, v), \quad 0 < \delta < \triangle < \infty,$$

where $\delta, \triangle$ are equivalence constants calculated in terms of the elements of the matrix A.

The proof follows directly from estimates of the spectrum of Stieltjes matrices.

Theorem 2. *Under the conditions of Theorem* 1*, the CG-ADD-MG method converges, with a number of required iterations $n(\varepsilon)$ estimated by* (5)*, where $\kappa = \triangle/\delta$ is the spectral condition number of the preconditioned matrix $B^{-1}A$.*

The above follows from the theory of preconditioned methods of conjugate directions.

5. EFFICIENCY AND PERFORMANCE OF PARALLEL MULTIGRID DOMAIN DECOMPOSITION METHODS

In this section, we discuss some necessary conditions for analysis and optimization of the algorithms considered above. The measure of efficiency of an algorithm implies some mathematical qualities that may be called the computational efficiency, which is measured by the number of arithmetic operations, and the amount of memory required for an implementation versus the number of independent variables or the number of degrees of freedom (d.o.f.). Performance is a technological characteristic determined by the execution time of the software code implementing an algorithm on a specific computer configuration. Increasing the efficiency of a method should lead to better performance of the algorithm. However, this is not always the case.

In typical iterative processes, the total volume of calculations depends on the total number of iterations, $n(\varepsilon)$, and the arithmetic complexity of one iteration. The required memory and the number of information exchanges are of great importance. The quality of parallelization of algorithms is estimated by the speedup and efficiency of the processors:

$$S_P = T_1(A)/T_P(A), \quad F_P = S_P/P,$$

where $T_P(A)$ is the execution time of problem A on P processors, and P is their total number. The code execution time T is roughly characterized as follows:

$$T^a = \tau_a N_a n(\varepsilon), \quad T^c = (\tau_0 N_l + \tau_c N_c) n(\varepsilon), \quad T^a + T^c = T.$$

Here $\tau_a \ll \tau_c \ll \tau_0$ are average times of executing an operation, transmitting one number, and waiting (adjusting) a transaction per each iteration. Their number is assumed to be rather large, for example, $n(\varepsilon) \ge 10$. Therefore, the time of one-time operations performed before the beginning of iterations can be neglected.

Let h be a typical spacing of a d-dimensional grid. Then the total number of nodes $N_h = O(h^{-d})$, where $d = 2, 3$ for two- and three-dimensional boundary value problems, respectively, and the number of iterations in a "good" algorithm is $n(\varepsilon) = O(h^{-1/2})$ regardless of dimension d. It is important that in the domain decomposition methods considered in the present paper, in contrast to classical methods,

the amount of data transferred between the subdomains and the macrogrid $N_c = O(h^{d-1})$. The number of subdomains M is, naturally, assumed to be finite and independent of h.

We do not make a deep analysis of the architecture of the multiprocessor computing system (MCS) under study. We only assume that the implementation of the parallel algorithms is performed by using hybrid programming tools: The SLAEs in the subdomains are formed and solved by using the MPI library in a distributed manner on computing nodes connected by buses. Each of these nodes contains several multiprocessor CPUs with shared memory.

The SLAEs in each of the subdomains are solved by using the multigrid incomplete factorization method in accordance with recommendations developed on the basis of some preliminary experimental studies on three or four nested grids ($\Omega_m^{(l)}$, $l = 1, \ldots, L$). In these studies, the calculations are divided into three stages both for the forward and backward factorization sweeps. These can be considered as reduction and prolongation stages. Each of these stages employs only one or two types of nodes (edge, face, or volume ones) and can be naturally parallelized by dividing each grid subdomain into K nonintersecting fragments. If the number of processors, $P = KM$, is relatively small (about several tens) the speedup, S_P, is close to linear.

The set of nodes of a divided macrogrid can be formed as a single subdomain on the central computing node. Since the macrogrid contacts all subdomains, it is reasonable to divide it into several parts to reduce communication load and transmit messages to the subdomains simultaneously.

It remains unclear how to find the optimal number of subdomains. It is obvious that as M increases, the ADDM approach will eventually transform into a multigrid algorithm. However, it does not seem to be the best option. Here the algebraic tool of investigation is rather complicated, and comparative experimental studies have to be performed. The practical results published in [7] and [8] are encouraging. First, the number of iterations in the subdomains depends but slightly on their number, and the macrogrid approach discussed above takes this advantage into account. Second, the multigrid incomplete factorization turns out to be optimal in terms of order. That is, the number of iterations practically does not depend on h, and the total number of calculations is proportional to the number of degrees of freedom, $N = O(h^{-d})$. Note the following property of the algorithm: the minimum number of iterations is in the two-grid version. However, the two-grid version is not optimal in terms of time, since the direct solver on grid $\Omega_m^{(2)}$ is very time consuming.

The results obtained for some special SLAEs of the nodal type offer new prospects for the expansion of the class of problems and the development of new algebraic-geometric approaches. Of great importance are systems of differential equations, high-order approximation schemes with a more complex structure of grid algebraic equations, and technologies of constructing algorithms on unstructured grids typical for practical problems with real data.

FUNDING

This work was supported by the Ministry of Education and Science of the Russian Federation (project no. FSUN-2024-0003).

CONFLICT OF INTEREST

The author of this work declares that he has no conflicts of interest.

REFERENCES

1. Saad, Y., *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM, 2003.
2. Olshanskii, M.A. and Tyrtyshnikov, E.E., *Iterative Methods for Linear Systems, Theory and Applications*, Philadelphia: SIAM, 2014.
3. Laevsky, Yu.M. and Matsokin, A.M., Decomposition Methods for the Solution to Elliptic and Parabolic Boundary Value Problems, *Sib. Zh. Vych. Mat.*, 1999, vol. 2, no. 4, pp. 361–372.
4. Il'in, V.P., Iterative Preconditioned Methods in Krylov Spaces: Trends of the 21st Century, *Zh. Vych. Mat. Mat. Fiz.*, 2021, vol. 61, no. 11, pp. 1786–1813.
5. Gurieva, Ya.L., Il'in, V.P., and Kozlov, D.I., Parallel Domain Decomposition Methods with Graph Preconditioning, *Tr. konf. "Parallel'nye vychislitel'nye tekhnologii (PaVT'2023)"*, Chelyabinsk: Izdatel'skii tsentr YuUrGU, 2023; https://doi.org/10.14529/pct2023

6. Il'in V.P., Multigrid Incomplete Factorization Methods in Krylov Subspaces, *J. Math. Sci.*, 2023, vol. 272, pp. 523–532.
7. Kuznetsov, Yu.A., Algebraic Multigrid Domain Decomposition Methods, Preprint of USSR Acad. Sci., Department of Computational Mathematics, Moscow, 1989, preprint no. 232.
8. Bank, R., Falgout, R., Jones, T., et al., Algebraic Multigrid Domain and Range Decomposition (AMG-DD/AMG-RD), *SIAM J. Sci. Comput.*, 2015, vol. 37, pp. 113–136.
9. Il'in, V.P., *Metody i tekhnologii konechnykh elementov* (Methods and Technologies of Finite Elements), Novosibirsk: Publ. House of ICM and MG SB RAS, 2007.
10. Il'in, V.P., *Matematicheskoe modelirovanie. Ch. 1, Nepreryvnye i diskretnye modeli*, (Mathematical Modeling. Part 1. Continuous and Discrete Models), Novosibirsk: Publ. House of SB RAS, 2017.
11. Agoshkov, V.I. and Lebedev, V.I., Variational Algorithms of the Domain Decomposition Method, *Sov. J. Numer. Anal. Math. Model.*, 1990, vol. 5, no. 1, pp. 27–46.
12. Dolean, V., Jolivet, P., and Nataf, F., *An Introduction to Domain Decomposition Methods: Algorithms, Theory and Parallel Implementation*, Philadelphia: SIAM, 2015.
13. Korneev, V.G., Fast Domain Decomposition Algorithms for Elliptic Problems with Piecewise Variable Orthotropies, in *Advanced Finite Element Methods and Applications*, Apel, T. and Steinbach O. (eds.), Berlin, Heidelberg: Springer, 2013, pp. 57–89 (*Lect. Notes in Applied and Computational Mechanics*, vol. 66).
14. Nepomnyashchikh, S.V., Domain Decomposition and Fictitious Space Methods, Doctoral Dissertation 01.01.07, Novosibirsk, NSU Publishing and Printing Center, 2008.
15. Il'in, V.P. and Kuznetsov, Yu.I., *Trekhdiagonal'nye matritsy i ikh prilozheniya* (Tridiagonal Matrices and Their Applications), Moscow: Nauka, 1985.
16. Kaporin, I.E. and Milyukova, O.Yu., Incomplete Inverse Triangular Factorization in Parallel Algorithms of Preconditioned Conjugate Gradient Methods, Preprint of Keldish Institute of Applied Mathematics, Moscow, 2017, preprint no. 37.
17. Anzt, H., Huckle, T.K., Bräckle, J., and Dongarra, J., Incomplete Sparse Approximate Inverses for Parallel Preconditioning, *Parallel Computing.*, 2018, vol. 71, pp. 1–22.
18. Reusken, A., A Multigrid Method Based on Incomplete Gaussian Elimination, *Numer. Linear. Alg. Appl.*, 1996, vol. 3, pp. 369–390.
19. Bank, R.E., Dupont, T.F., and Yserentant, H., The Hierarchical Basis Multigrid Method, *Numer. Math.*, 1988, vol. 52, pp. 427–458.
20. Hackbusch, W., Khoromskij, B.N., and Kriemann, R., Direct Schur Complement Method by Domain Decomposition Based on h-Matrix Approximation, *Comput. Visual. Sci.*, 2005, vol. 8, no. 3, pp. 179–188; DOI: 10.1007/s00791-005-0008-3
21. Vasilevsky, Yu.V. and Olshansky, M.A., *Kratkii kurs po mnogosetochnym metodam i metodam dekompozitsii oblasti* (A Short Course on Multigrid and Domain Decomposition Methods), Moscow: Moscow State University, 2007.
22. Notay, Y., Flexible Conjugate Gradients, *SIAM J. Sci. Comput.*, 2000, vol. 22, no. 4, pp. 1444–1460.