ISSN 0965-5425, Computational Mathematics and Mathematical Physics, 2025, Vol. 65, No. 6, pp. 1220–1231. © Pleiades Publishing, Ltd., 2025.

GENERAL NUMERICAL METHODS

Multigrid Methods of Macrogrid Domain Decomposition

V. P. Il'in*

Institute of Computational Mathematics and Mathematical Geophysics, Siberian Branch, Russian Academy of Sciences, Novosibirsk, 630090 Russia

**e-mail: ilin@sscc.ru*

Received February 17, 2025; revised March 18, 2025; accepted March 27, 2025

Abstract—Integrated domain decomposition multigrid methods (DDM-MG) for solving large systems of linear algebraic equations (SLAEs) with sparse symmetric or asymmetric matrices are considered. Such systems are obtained as a result of grid approximations of multidimensional boundary value problems. The proposed algorithms are based on the construction of single-layer or two-layer macrogrids and a special ordering of nodes according to their belonging to various topological primitives of the macrogrid: macronodes, macroedges, macrofaces, and subdomains. With a consistent numbering of vector components, the SLAE matrix in the three-dimensional case takes a block-tridiagonal form of the fourth order. To solve it, an iterative preconditioned method in Krylov subspaces is used. In this case, the solution of auxiliary systems in subdomains is carried out by multigrid methods of block incomplete factorization based on a similar topologically oriented ordering of nodes but at the microlevel rather than at the macrolevel, as a result of which a single preconditioner of a recursively nested type is formed. The justification of the proposed methods is carried out for matrices of the Stieltjes type.

Keywords: large sparse SLAEs, multigrid methods, domain decomposition, macrogrids, preconditioned algorithms, Krylov subspaces

DOI: 10.1134/S0965542525700460

1. INTRODUCTION

Domain decomposition methods (DDM) and multigrid approaches (MG) are among the main computational technologies for solving large systems of linear algebraic equations (SLAEs) with sparse matrices arising in grid approximations of multidimensional boundary value problems. Historically, they evolved as alternative directions: DDM were oriented mainly toward efficient parallelization of iterative algorithms on multiprocessor computing systems (MCS), and MG solvers turned out to be asymptotically optimal in terms of order; in these methods, for a characteristic grid step $h \to 0$, the amount of required arithmetic operations is directly proportional to the SLAE dimension, while the acceleration of parallel computations is not high. The theoretical analysis of these computational tools is completely different, and their software implementations are available in many libraries (see an extensive bibliography in [1–18] and the references therein).

In [19], an algebraic-geometric decomposition method was proposed for a parallelepipedal grid computational domain based on constructing a separating macrogrid and dividing the nodes into four nonintersecting subsets according to their belonging to different topological primitives: macronodes, macroedges, macrofaces, and subdomains. With the appropriate ordering of the nodes and vector components, the SLAE matrix is represented in a block-tridiagonal form, and preconditioned iterative algorithms in Krylov subspaces using approximate matrix triangular decompositions are used to solve it.

Note that in this approach the first stage of the algorithm is associated with the analysis of the geometric properties of grid objects, and the second stage is only associated with transformations of the constructed block matrices, and there are no concepts of solutions to auxiliary boundary value problems in subdomains, which is the basis of the classical DDMs. In [20, 21], a similar approach was proposed for the multigrid method, but at the microlevel rather than at the macrolevel. For the case of two three-dimensional nested grids consisting, e.g., of parallelepipeds or tetrahedra, the nodes of the dense grid are divided into four subsets according to their belonging to different topological primitives of the sparse grid: nodes, edges, faces and cells. With the appropriate numbering of the vector components, the algebraic system is also reduced to a block-tridiagonal form of the fourth order, to the solution of which the algo-

rithm of approximate factorization in Krylov subspaces can be effectively applied. Multigrid methods are defined as a recursive application of two-grid methods with the matrix structures taking a nested block-tridiagonal form.

A natural step in developing these two approaches is to combine DDM and MG into a single-level algebraic-topological method with the construction of a Krylov-type iterative process with a single preconditioning matrix constructed by approximating the general recursive block-tridiagonal structure for the original SLAE. In this case, the nodes of the macrogrid implementing the decomposition of the domain are numbered first, and then the internal nodes in the subdomains are numbered using for each of them the ordering of the algebraic-topological type described above for MG.

In contrast to the single-layer separating macrogrid considered in [19], we also assume that the grid computational domain is decomposed using a two-layer macrogrid. In this case, the formal division into subdomains is carried out by breaking the edge connections between individual neighboring nodes. The set of near-boundary nodes thus obtained is excluded from the subdomains and combined into a macrogrid consisting, for the case of a cubic grid, of two-layer macrofaces, "thick" macroedges (four nodes in a perpendicular section) and "fat" macronodes (eight nodes in each). The block structure of the SLAE obtained in this algebraic-topological version of DDM retains its general form, but the configuration of the blocks themselves becomes more complex.

Macrogrid subdomain separators ensure high speed of DDM iterations since the solution of the corresponding "carcass" SLAEs ensures the propagation of the disturbance throughout the computational domain at each step.

The implementation of the MG stage of the combined approach does not change, neither does the principle of the preconditioned iterative method change: approximation of the block-tridiagonal matrix using approximate factorization plus construction of any of the Krylov-type algorithms. Note that the first approaches to the joint use of DDM and MG were considered in [22, 23].

This paper is organized as follows. In Section 2, we give a brief description of iterative preconditioned methods in Krylov subspaces for symmetric and asymmetric SLAEs. Section 3 presents an algebraic-topological version of DDM with two-layer macrogrids, and Section 4 presents a similar description of the multigrid method for subdomains. Section 5 presents the general structure of the proposed DDM–MG iterative process and some of its properties. In the last section, possible generalizations of the described approaches and directions for further research are discussed.

2. PRECONDITIONED ITERATIVE METHODS IN KRYLOV SUBSPACES

Consider a real nonsingular SLAE

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

which is assumed to be obtained as a result of approximation of a boundary value problem on a grid Ω^h. We assume that system (1) is of the node type, i.e., each grid node is associated with one component of the unknown vector u. Below, to illustrate the proposed algorithms, we use a seven-point or seven-diagonal SLAE that is an approximation of a diffusion-convection equation for the cubic grid

$$\Omega^h : x_i = ih, \quad i = 1,\ldots,N_x, \quad y_j = jh, \quad j = 1,\ldots,N_y, \quad z_k = kh, \quad k = 1,\ldots,N_z, \tag{2}$$

which in multi-index form is written as

$$\begin{aligned}(Au)_{i,j,k} &= a^{(0)}_{i,j,k}u_{i,j,k} - a^{(1)}_{i,j,k}u_{i-1,j,k} - a^{(2)}_{i,j,k}u_{i,j-1,k} - a^{(3)}_{i,j,k}u_{i+1,j,k} \\ &\quad - a^{(4)}_{i,j,k}u_{i,j+1,k} - a^{(5)}_{i,j,k}u_{i,j,k-1} - a^{(6)}_{i,j,k}u_{i,j,k+1} = f_{i,j,k};\end{aligned} \tag{3}$$

for methodological purposes, we will use symmetric positive definite (s.p.d.) matrices, including the class of Stieltjes matrices (indecomposable that have the properties of diagonal dominance, positive diagonal and non-positive off-diagonal elements, see [24]).

If B is an easily invertible (preconditioning) s.p.d. matrix, then with its help preconditioned iterative processes of the Krylov type are formulated in the general form:

$$\begin{gathered}r^0 = f - Au^0, \quad p^0 = B^{-1}r^n, \quad n = 0,1,\ldots, \\ u^{n+1} = u^n + \alpha_n p^n, \quad r^{n+1} = r^n - \alpha_n Ap^n,\end{gathered} \tag{4}$$

here p^n are the direction vectors, r^0 is the initial residual, and α_n, β_n are the iterative parameters to be determined. If the direction vectors are A^γ-orthogonal

$$(A^\gamma p^n, p^k) = \rho_k^{(\gamma)}\delta_{n,k}, \quad \rho_k^{(\gamma)} = (A^\gamma p^k, p^k) = \left\|p^k\right\|_\gamma^2, \tag{5}$$

where $\gamma = 0,1,2$, and $\delta_{n,k}$ is the Kronecker symbol, then relations (4) define the residual functional

$$\Phi_\gamma(r^n) = (A^{\gamma-2}r^n, r^n) = (r^0, r^0)_{\gamma-2} - \sum_{k=0}^{n-1}[2\alpha_k(r^0, A^{\gamma-1}p^k) - \alpha_k^2\rho_k], \tag{6}$$

which under the conditions

$$p^{n+1} = B^{-1}r^{n+1} + \beta_n p^n, \quad \alpha_n = \sigma_n/\rho_n, \quad \beta_n = \sigma_{n+1}/\sigma_n, \quad \sigma_n = (r^n, B^{-1}r^n), \quad \rho_n = (p^n, A^\gamma) \tag{7}$$

attains its minimum

$$\Phi_\gamma(r^n) = \left\|r^0\right\|_{\gamma-2} - \sum_{k=0}^{n-1}(r^0, p^k)_{\gamma-1}^2/\rho_k. \tag{8}$$

In this case, to satisfy the condition $\left\|r^n\right\| = (r^n, r^n)^{1/2} \le \varepsilon\|f\|$ at the given $\varepsilon \ll 1$, the number of required iterations satisfies the bound

$$n(\varepsilon) \le \sqrt{\kappa}[(\log(2\varepsilon^{-1})]/2, \tag{9}$$

where κ is the spectral condition number of the matrix $B^{-1}A$. In this family of preconditioned conjugate direction methods (CD) the value $\gamma = 0$ corresponds to the minimum error algorithm, and for $\gamma = 1$ and $\gamma = 2$ we have the conjugate direction and conjugate residual methods. More precisely, the two-term formulas (7) define the Hestenes–Stiefel orthogonalization for the vectors p^n. An alternative are the three-term Lanczos relations, which are more sensitive to rounding errors.

If the original SLAE is not symmetric, then instead of CD, semi-conjugate direction (SCD) methods with long vector recursions are defined, which are more resource-intensive. We will consider these algorithms in a generalized form with multi-preconditioning, when a new iterative approximation is calculated using several direction vectors rather than a single direction vector p^n; these direction vectors form the matrix P_n:

$$r^0 = f - Au^0, \quad n = 0,\ldots: \quad u^{n+1} = u^n + P_n\overline{\alpha}_n,$$
$$r^{n+1} = r^n - AP_n\overline{\alpha}_n = r^q - AP_q\overline{\alpha}_q - \ldots - AP_n\overline{\alpha}_n, \quad 0 \le q \le n, \tag{10}$$
$$P_n = (p_1^n,\ldots,p_{M_n}^n) \in \mathcal{R}^{N,M_n}, \quad \overline{\alpha}_n = (\alpha_{n,1},\ldots,\alpha_{n,M_n})^\top \in \mathcal{R}^{M_n}.$$

Here, $\overline{\alpha}_n \in \mathcal{R}^{M_m}$ are the iterative parameter vectors and p_k^n are the direction vectors satisfying the semi-conjugation condition (it is essential that following relations $n' \le n$):

$$(Ap_k^n, A^\gamma p_{k'}^{n'}) = \rho_{n,k}^{(\gamma)}\delta_{n,n'}^{k,k'}, \quad \rho_{n,k}^{(\gamma)} = (Ap_k^n, A^\gamma p_k^n), \quad \gamma = 0,1, \quad n' = 0,1,\ldots,n-1, \quad k,k' = 1,2,\ldots,M_n. \tag{11}$$

If we set the coefficients $\overline{\alpha}_n = \{\alpha_{n,l}\}$ in (10) by the formulas

$$\alpha_{n,l} = \sigma_{n,l}/\rho_{n,n}^{(\gamma)}, \quad \sigma_{n,l} = (r^0, A^\gamma\overline{p}_l^n), \tag{12}$$

then the residual functional

$$\Phi_n^{(\gamma)}(r^{n+1}) \equiv (r^{n+1}, A^{\gamma-1}r^{n+1}) = ((r^q, A^{\gamma-1}r^q) - \sum_{k=q}^{n}\sum_{l=1}^{M_n}(r^q, A^\gamma p_l^k)^2/\rho_{k,l}^{(\gamma)}, \quad q = 0,1,\ldots,n, \tag{13}$$

attains its minimum in the block Krylov spaces:

$$\mathcal{K}_M = \mathrm{Span}\{p_1^0,\ldots,p_{M_0}^0, Ap_1^1,\ldots,Ap_{M_1}^1,\ldots,Ap_1^n,\ldots,Ap_{M_n}^n\}, \quad M = M_0 + M_1 + \ldots + M_n. \tag{14}$$

The orthogonal properties of the direction vectors p_j^n are generally determined using various preconditioning matrices $B_{n,l}$; as a result, we obtain

$$p_l^0 = B_{0,l}^{-1} r^0, \quad p_l^{n+1} = B_{n+1,l}^{-1} r^{n+1} - \sum_{k=0}^{n} \sum_{l=1}^{M_k} \beta_{n,k,l}^{(\gamma)} p_l^k, \quad n = 0,1,\ldots;$$

$$B_{n,l} \in \mathcal{R}^{N,N}, \quad i = 1,\ldots,M_n; \quad \gamma = 0,1,2, \quad \overline{\beta}_{n,k}^{(\gamma)} = \{\beta_{n,k,l}^{\gamma}\} = \left(\beta_{n,k,1}^{(\gamma)} \ldots \beta_{n,k,M_n}^{(\gamma)}\right)^{\top} \in \mathcal{R}^{M_n}, \tag{15}$$

$$\beta_{n,k,l}^{(\gamma)} = -\left(A^{\gamma} p_l^k, AB_{n+1,l}^{-1} r^{n+1}\right) / \rho_{n,l}^{\gamma}, \quad n = 0,1,\ldots; \quad k = 0,\ldots,n; \quad l = 1,\ldots,M_n.$$

Note that particular versions of the considered methods for asymmetric SLAEs are equivalent in convergence rate to CMRES-type algorithms based on Arnoldi orthogonalization, including those with dynamic or flexible preconditioning (FGMRES, see [25] for more details).

Note also that asymmetric SLAEs can be solved by Krylov's methods with short recursions if we use biconjugate direction algorithms or symmetrization of the original system with the preliminary Gaussian transformation (see [26, 27] and the references therein).

3. MACROGRID DOMAIN DECOMPOSITION METHODS

The construction of the proposed macrogrid DDMs consists of two stages. At the first stage, a block three-diagonal representation of the original SLAE based on the topology oriented ordering of the nodes of the grid computational domain is formed, and the second stage consists of constructing an efficient preconditioning matrix.

3.1. Topology Oriented Domain Decompositions

In [14], the nonoverlapping subdomains were decomposed using the separating macrogrid Ω^H embedded into the original grid Ω^h:

$$\Omega^H : x_{i_1} < i_1 < i_{M_x} < N_x; \quad 1 < j_1 < j_{M_y} < N_y; \quad 1 < k_1 < k_{M_z} < N_z, \quad \Omega^H = \Omega_V^H \cup \Omega_E^H \cup \Omega_F^H. \tag{16}$$

In this case, the dimensions of each macronode, macroedge, and macroface are zero, one, and two, respectively; i.e., they are a simple node, grid segment, and a fragment of the grid plane. The corresponding geometric (or topological) grid objects are shown by the symbols •, ×, ∘ in Fig. 1 depicting a fragment of a cubic grid.

By numbering all the macronodes, then the macroedge nodes, then macroface nodes, and finally the internal nodes of the subdomains, we obtain four subsets of the nodes of the original grid and four sub-

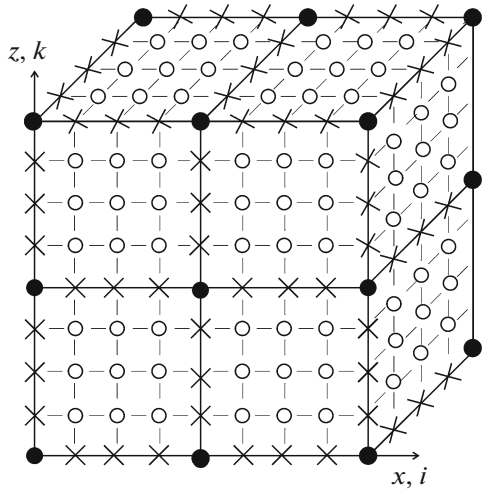

Fig. 1. A fragment of a cubic grid with macrogrid objects.

vectors of the original grid solution to SLAE (1), which then takes a block tridiagonal form of the fourth order:

$$Au = \begin{bmatrix} A_{1,1} & A_{1,2} & 0 & 0 \\ A_{2,1} & A_{2,2} & A_{2,3} & 0 \\ 0 & A_{2,3} & A_{3,3} & A_{3,4} \\ 0 & 0 & A_{4.3} & A_{4,4} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}. \tag{17}$$

Here, the diagonal blocks $A_{l,l}$ and the subvectors u_l, f_l correspond to the subsets Ω_V^h, Ω_E^h, Ω_f^h, and Ω_D^h for $l = 1,2,3,4$. The matrix $A_{1,1}$ in (17) is a diagonal one, its size equals the number of macronodes M_V, and $A_{2,2}$, $A_{3,3}$, and $A_{4,4}$ are block diagonal matrices with the block sizes M_E, M_F, and M_D equal to the number of macroedges, macrofaces, and subdomains, respectively. If we assume for simplicity that all macroedges have N_E nodes each, then the sizes of the resulting grid subdomains and the corresponding subvectors are

$$N_1 = M_V, \quad N_2 = M_E N_E, \quad N_3 = M_F N_E^2, \quad N_4 = M_D N_E^3, \quad N = \sum_{l=1}^{4} N_l. \tag{18}$$

The set of separating nodes described above is called a simple (or single-layer) macrogrid. In this case, each diagonal block of the matrix $A_{2,2}$ is a tridiagonal matrix, and the diagonal blocks of $A_{3,3}$ and $A_{4,4}$ are five-diagonal and seven-diagonal matrices, respectively.

Now consider the decomposition of the grid domain Ω^h using "virtual" coordinate planes that do not include grid nodes. In this case, to each grid line we assign the macrogrid pairs of adjacent nodes separated the plane:

$$\Omega^H : x_{i_1}, x_{i_1+1}, \ldots, x_{M_x}, x_{M_{x+1}}; \quad y_{j_1}, y_{j_1+1}, \ldots, y_{M_y}, x_{M_x+1}; \quad z_{ki_1}, z_{k_1+1}, \ldots, z_{M_z}, z_{M_{z+1}}. \tag{19}$$

Then the macroface is a fragment of a "double" grid plane, and a three-dimensional macroedge is a sequentially located "quadruples" of nodes of the original grid. The corresponding macronodes, which are intersections of a triple of macroedges, consist of eight nodes. The set of nodes Ω^H obtained in this way will be called, in contrast to (10), a double or two-layer macrogrid. For clarity, Fig. 2 shows a fragment of the obtained structure in the two-dimensional case, where the symbols •, ×, ∘ denote macronodes, macroedges, and internal nodes of subdomains, respectively.

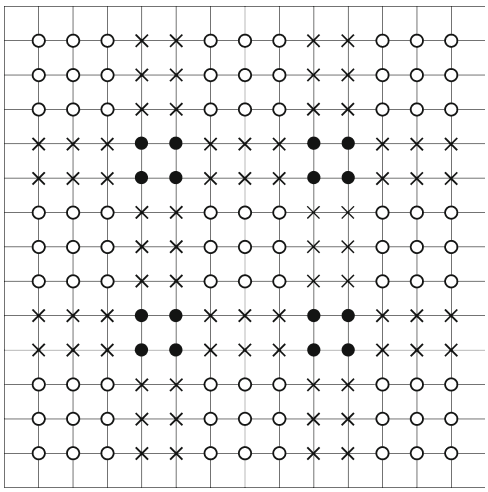

Fig. 2. A fragment of a double macrogrid for the two-dimensional case, $M_x = M_y = 2$.

In this case, the SLAE block structure (16) has the third, rather than fourth, order. It is easy to verify that, for the model example with identical macroedge lengths N_E, the sizes of the diagonal blocks and subvectors are

$$N_1 = 4M_V, \quad N_2 = 2M_E N_E, \quad N_3 = M_D N_E^2, \quad N = N_1 + N_2 + N_3. \tag{20}$$

For a three-dimensional double separation macrogrid, the corresponding quantities are defined as follows:

$$N_1 = 8M_V, \quad N_2 = 4M_E N_E, \quad N_3 = 2M_E N_E^2, \quad N_4 = M_D N_E^3, \quad N = \sum_{l=1}^{4} N_l. \tag{21}$$

In the last case, each diagonal block of the matrix $A_{1,1}$ is a four-diagonal matrix, and each diagonal block of $A_{2,2}$ and $A_{3,3}$ is a six-diagonal matrix. The structure of the matrix $A_{4,4}$ is the same as that in the case of a simple macrogrid.

3.2. Construction of "Macrogrid" Preconditioners

In Section 2, we presented formulas for preconditioned methods of conjugate or semi-conjugate directions for iterative solution of symmetric or asymmetric SLAEs, respectively. In this section, we focus only on the issues of constructing and implementing preconditioners that generalize the decomposition of domains using single-layer and double-layer macrogrids that have some analogy, at least terminological, with the potentials of simple and double layers in the theory of integral equations. To approximate the matrix of the original SLAE (17), two methods of its approximate block-triangular decomposition can be used. The first is the method of incomplete factorization with diagonal compensation:

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

where the matrices G_k are given by the formulas

$$\begin{gathered} G_1 = A_{1,1}, \quad G_2 = A_{2,2} - (A_{2,1}G_1^{-1}A_{1,2})_1 - \theta S_2, \quad S_2 e_2 = \left[A_{2,1}G_1^{-1}A_{1,2} - (A_{2,1}G_1^{-1}A_{1,2})_1\right] e_2, \\ G_3 = A_{3,3} - (A_{3,2}G_2^{-1}A_{2,3})_1 - \theta S_3, \quad S_3 e_3 = \left[A_{3,2}G_2^{-1}A_{2,3} - (A_{3,2}G_2^{-1}A_{3,2})_1\right] e_3, \\ G_4 = A_{4,4} - (A_{4,3}G_3^{-1}A_{3,4})_1 - \theta S_4, \quad S_4 e_3 = \left[A_{4,3}G_3^{-1}A_{3,4} - (A_{4,3}G_3^{-1}A_{4,3})_1\right] e_4. \end{gathered} \tag{22}$$

Here, S_k are diagonal matrices, $(C)_1$ denotes the diagonal part of the matrix C, $\theta \in [0,1]$ is the compensation parameter, and the trial vectors (typically, their components are equal to unity) e_k have dimension N_k. These formulas correspond to the condition of full compensation of the row sums $Be = Ae$ for $\theta = 1$, $e \in \mathcal{R}^N$, and they come from exact factorization of $(A = B)$, for which $G_k = A_{k,k} - A_{k,k-1}C_{k+1}^{-1}A_{k+1,k}$.

The second method for constructing a factorized preconditioner is the symmetric or asymmetric (depending on the properties of the original SLAE) method of successive upper relaxation (SSOR or USSOR), when the matrices G_k in (22) are defined simply as

$$G_k = \omega^{-1}A_{k,k}, \quad \omega \in (0,2), \tag{23}$$

where ω is the relaxation parameter.

Note that for the implementation of preconditioned iterative methods, an auxiliary SLAE $Bq = r$ must be solved at each step n, which is done using the following efficient formulas:

$$\begin{gathered} G_1 v_1 = r_1, \quad G_2 v_2 = r_2 - A_{2,1}v_1, \quad G_3 v_3 = r_3 - A_{3,2}v_2, \quad G_4 v_4 = r_4 - A_{4,3}v_3, \quad q_4 = v_4, \\ G_3 w_3 = A_{3,4}q_4, \quad q_3 = v_3 - w, \quad G_2 w_2 = A_{2,3}q_3, \quad q_2 = v_2 - w_2, \quad q_1 = v_1 - (G_1)^{(-1)}A_{1,2}q_2. \end{gathered} \tag{24}$$

To calculate the matrices G_k in (22), the simple approximation $C \approx (C)_1$ is used. More accurate versions will be discussed later.

In these relations, the components of the vector v_4 correspond to a subset of nodes of M_D in subdomains in each of which an algebraic subsystem must be solved, for which purpose we will use multigrid methods described in the following section. Note that the matrix G_4 in (22) has the same seven-point portrait as $A_{4,4}$.

Iterative algorithmic capabilities in the macrogrid DDM arise if a special block tridiagonal structure of the matrix block $A_{2,2}$ is used instead of (22) for constructing the preconditioned matrix B. In this case, from the first two block rows of SLAE (17) it is possible to form the subsystem

$$\overline{A}\overline{u}\begin{bmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{bmatrix}\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \end{bmatrix} = f_2 - A_{3,3}u_3, \tag{25}$$

which can be solved (or its solution can be expressed in terms of u_3) using the block sweeping method. Let us consider, for illustration, the system of equations for a double macrogrid in the two-dimensional case shown in Fig. 2:

$$\begin{gathered} -a_t'v_{t-1} + b_t'v_t - c_t'v_{t+1} + d_t'w_t = f_t', \quad -a_t''w_{t-1} + b_t''w_t - c_t''w_{t+1} + d_t''v_t = f_t'', \\ t = 1,\ldots,N_e, \quad v_0 = v^b, \quad w_0 = w^b, \quad v_{N_e} = v^e, \quad w_{N_e} = w^e. \end{gathered} \tag{26}$$

t is the local index of macroedge section located between two macronodes the values v_b, $w - B$, V_l, W_l of the desired solution at which can be considered as boundary conditions for the grid problem

$$\begin{gathered} -A_iu_{i-1} + B_iu_i - C_iu_{i+1} = f_i, \quad i = 1,\ldots,N_e, \quad u_i = \begin{pmatrix} v_i \\ w_i \end{pmatrix}, \quad f_i = \begin{pmatrix} f_i' \\ f_i'' \end{pmatrix}, \quad u_0 = \begin{pmatrix} v^b \\ w^b \end{pmatrix}, \quad u_{N_e+1} = \begin{pmatrix} v^e \\ w^e \end{pmatrix}, \\ B_1u_1 - C_1u_2 = f_1, \quad -A_{N_e}u_{N_e-1} + B_{N_e}u_{N_e} = f_{N_e}, \quad A_i = \begin{bmatrix} a_i' & 0 \\ 0 & a_i'' \end{bmatrix}, \quad B_i = \begin{bmatrix} b_i' & d_i' \\ d_i'' & b_i'' \end{bmatrix}, \quad C_i = \begin{bmatrix} c_i' & 0 \\ 0 & c_i'' \end{bmatrix}. \end{gathered} \tag{27}$$

Its solution can be found using forward or back version of the sweep algorithm

$$\hat{u}_{N_e} = \hat{z}_{N_e}, \quad \hat{u}_t = \hat{B}_t\hat{u}_{t+1} + \hat{z}_t, \quad t = N_e - 1,\ldots,1, \quad \breve{u}_1 = \breve{z}_1, \quad \breve{u}_t = \breve{B}_t\breve{u}_{t-1} + \breve{z}_t, \quad t = 2,\ldots,N_e, \tag{28}$$

where the vectors $\hat{z}_t$, $\breve{z}_t$ and the matrices $\hat{B}_t$, $\breve{B}$ are found from the equations

$$\begin{gathered} \hat{\beta}_1 = C_1\hat{D}_1, \quad \hat{D}_1 = B_1^{-1}, \quad \hat{\beta}_t = C_t\hat{D}_t, \hat{D}_t = \left(B_t - A_t\hat{\beta}_{t-1}\right), \quad t = 2,\ldots,N_e, \\ \breve{\beta}_{N_e} = A_{N_e}\breve{D}_{N_e}, \quad \breve{D}_{N_e} = B_{N-e}^{-1}, \quad \breve{\beta}_t = A_t\breve{D}_t, \quad \breve{D}_t = \left(B_t - C_t\breve{\beta}_{t+1}\right)^{-1}, \quad t = N_e - 1,\ldots,1, \\ \hat{z}_1 = (f_1 + A_1u_0)\hat{D}_1^{-1}, \quad \hat{z}_t = (f_t + A_t\hat{z}_{t-1})\hat{D}_t^{-1}, \quad t = 2,\ldots,N_e, \\ \breve{z}_{N_e} = (f_{N_e} + C_{N_e}u_{N_e+1})\breve{D}_{N_e}^{-1}, \quad \breve{z}_t = (f_t + C_t\breve{z}_{t+1})\breve{D}_t^{-1}, \quad t = N_e - 1,\ldots,1. \end{gathered} \tag{29}$$

Using these formulas, the components of the subvector u_t can be explicitly expressed in terms of the macronode values $\hat{z}_t$, $\breve{z}_t$ and $\hat{B}_t$, $\breve{B}_t$ after the substitution into the corresponding macronode equations and eliminated from the system. The resulting macronode low-order subsystem can be solved by a direct method after which the values at macroedge nodes are calculated using (28).

The matrix relations (27), (28) remain valid for the three-dimensional case; only the matrices and vectors become not of the second but of the fourth order, and vice versa; and when using a simple macrogrid, the sweep formulas become scalar for both the two-dimensional and three-dimensional cases. Note without going into details that this approach allows one to relatively easily calculate the main diagonal of the inverse matrix A^{-1} from (25), which provides additional opportunities for improving the preconditioned matrix.

Let us consider another direction of development of this class of subdomain decompositions, which we will illustrate on the simplest two-dimensional case with a single-layer macrogrid shown in Fig. 3. In this case, the block matrices A and B have order three, and the fourth block rows and columns are absent. The new algorithm is constructed by replacing the diagonal approximation of the matrix $A_{3,2}G_2^{-1}A_{2,3}$ in (22) with a more accurate approximation, which has a four-point portrait $(A_{3,2}G_2^{-1}A_{3,3})_4$. The changes concern

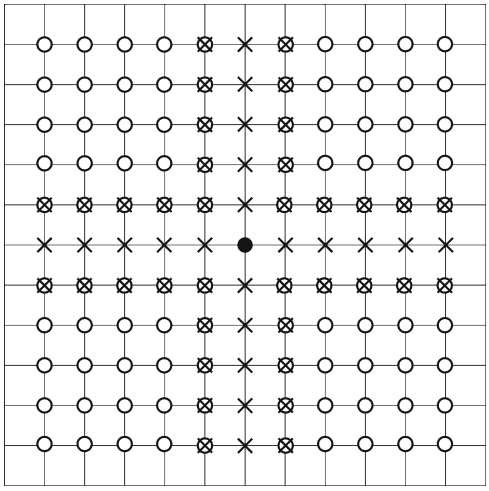

Fig. 3. A fragment of a two-dimensional grid with a secondary macrogrid and a simple primary macrogrid.

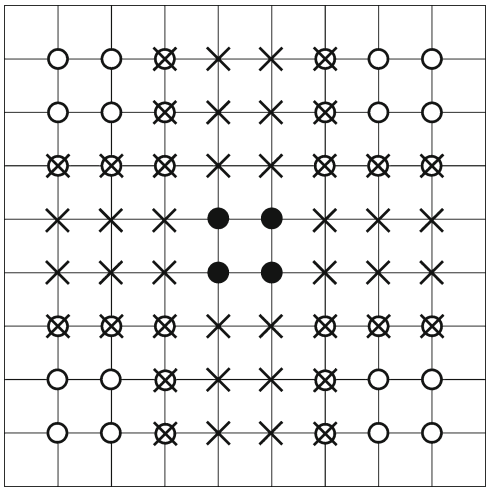

Fig. 4. A fragment of a two-dimensional grid with a secondary macrogrid and a two-layer primary macrogrid.

only the near-boundary nodes of the subdomains designated in Fig. 3 by the symbol $\otimes$, each node has connections with two nearest nodes from its subdomain and with one or two from the neighboring ones (some rows of the matrix have not four, but five nonzero elements).

It is clear that the secondary macrogrid is structurally two-layered and its image will be topologically equivalent to the graph in Fig. 2 if the nodes of the type $\times$, $\bullet$ are removed from Fig. 3 (the dimensions of the new subdomains are reduced by one layer). In this case, the matrix G_3 (which we will redesignate as $A^{(2)} = G_3$) acquires a block form similar to the original matrix $A = A^{(1)}$, and to solve the corresponding SLAE (approximate calculation of v_3 in (24)) we can recursively apply the incomplete factorization procedure (22).

Let us now make some remarks on the construction of preconditioners for macrogrid decompositions of subdomains.

Remark 1. The computational scheme for constructing a secondary macrogrid has the same block-matrix form if the original (primary) macrogrid is not simple but has two-layers (see Fig. 4).

Remark 2. The described scheme for constructing the secondary macrogrid can be recursively continued to form m-ary (tertiary, quaternary, etc.) macrogrids, while for any m the macrogrids have no more than two-layers and with each such step the subregions are reduced by one grid layer.

Remark 3. Recursive procedures for constructing macrogrid DDMs have the same block-matrix structure in the three-dimensional case as in the two-dimensional case both for the original simple and double macrogrids.

4. MULTIGRID PRECONDITIONING IN SUBDOMAINS

The purpose of this section is to specify methods for solving SLAE for calculating the subvector v_4 in formulas (24). More strictly speaking, we describe an approximate representation for the matrix G_4, which completes the construction of a single-level combined DDM-MG method in Krylov subspaces and provides a justification for the proposed approach.

Since G_4 is a block-diagonal matrix of a large order M_D, the problem can be formulated as a parallel solution of systems of the form

$$A_m^{(l)} v_m^{(l)} = r_m^{(l)}, \quad m = 1,\ldots,M_D, \quad l = 1. \tag{30}$$

Here, we assume that $v_4 = \{v_m^{(1)}\}$ and that in all M_D subdomains the original grids $\Omega_m^h = \Omega_m^{(1)}$ have approximately the same size (for the purpose of efficient parallelization), and in each of them a sequence of nested grids $\Omega_m^{(1)} \supset \cdots \supset \Omega_m^{(l)} \supset \cdots \supset \Omega_m^{(L)}$ is constructed.

Below, we will construct identical algorithms for subdomains; for this reason, we will omit the index m in the following formulas. We will consider in more detail the proposed methods for the case of two grids $\Omega^{(1)} \supset \Omega^{(2)}$ (dense and sparse), and we will define multigrid algorithms as a recursive application of two-grid ones.

We divide the set of nodes of the original (dense) grid $\Omega^{(1)}$ in the subdomain into four subsets according to their belonging to different topological primitives of the sparse grid $\Omega^{(2)}$—nodes, edges, faces and cells. In Fig. 5, the nodes of the corresponding subsets are designated by $\bullet$, $\times$, $\circ$.

Note that these four subsets

$$\Omega_1^{(1)} \cup \Omega_2^{(1)} \cup \Omega_3^{(1)} \cup \Omega_4^{(1)} = \Omega^{(1)}$$

have an analogy with the partition in the case of macrogrid decomposition but at the microlevel rather than at the macrolevel.

By further sequentially ordering the nodes from these subsets, as well as the corresponding components of the vector $v_4 = \{v_m^{(1)}\}$ in (30) (first we renumber all the nodes in the first subset, then in the second, etc.), we obtain a matrix $A^{(1)}$ for the initial grid that has a block-tridiagonal form with a block of fourth order

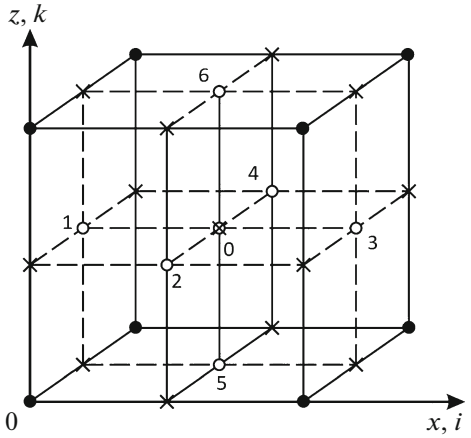

Fig. 5. Illustration of node classification for two external grids.

like SLAE (17). For convenience, we below will formally write the resulting algebraic system with a superscript $A^{(l)}$, keeping in mind that in our case $l = 1$:

$$A^{(l)}u^{(l)} = \begin{bmatrix} A^{(l)}_{1,1} & A^{(l)}_{1,2} & 0 & 0 \\ A^{(l)}_{2,1} & A^{(l)}_{2,2} & A^{(l)}_{2,3} & 0 \\ 0 & A^{(l)}_{3,2} & A^{(l)}_{3,3} & A^{(l)}_{3,4} \\ 0 & 0 & A^{(l)}_{4,3} & A^{(l)}_{4,4} \end{bmatrix} \begin{bmatrix} u^{(l)}_1 \\ u^{(l)}_2 \\ u^{(l)}_3 \\ u^{(l)}_4 \end{bmatrix} = \begin{bmatrix} f^{(l)}_1 \\ f_2 \\ f^{(l)}_3 \\ f^{(l)}_4 \end{bmatrix}. \tag{31}$$

Note that for the chosen "topological" ordering for the SLAE (3), the diagonal blocks $A^{(l)}_{k,k}$ are diagonal matrices, the submatrix $A^{(l)}_{1,2}$ is a six-diagonal one, $A^{(l)}_{2,4}$ and $A^{(l)}_{3,4}$ are a four-diagonal matrix. It is clear that the matrix of the algebraic system can be approximately factorized ($A^{(l)} \approx B^{(l)}$), and for the diagonal blocks $G^{(l)}_4$ it holds that

$$G^{(l)}_4 = A^{(l)}_{4,4} - A^{(l)}_{4,3}(G^{(l)}_3)^{-1}A^{(l)}_{3,4}.$$

In this case, this matrix has a seven-point portrait of the same type as that of $A^{(l)}$ but only for a coarser grid, due to which we can introduce the notation $G^{(l)}_4 = A^{(l+1)}$. In turn, if $l + 1 < L$, then the matrix $A^{(l+1)}$ can be reduced to a block-tridiagonal form (17) and approximately factorized; next the recursive process can be continued. Recall that when implementing formulas for incomplete factorization with diagonal compensation, an auxiliary SLAE $Bq = r$ must be solved at each iteration of the preconditioned conjugate gradient method; this is easily accomplished using formulas (24).

Based on the outlined principles of constructing algebraic-geometric multigrid methods, the following three variants of combined DDM-MG algorithms can be proposed.

The first option is to use nested incomplete factorizations by analogy with the work [19]. We define the matrix block G_4 in the DDM decomposition (22) as the SLAE matrix $G_4 = A^{(1)}$ on the original (dense) grid in the subdomain and perform a similar approximate triangular decomposition $A^{(1)} \approx B^{(1)} = L^{(1)}_B U^{(1)}_B$ for it. Then we continue this recursive process in each domain, as a result of which we obtain a sequence of matrices

$$A^{(1)} \approx B^{(1)} = L^{(1)}_B U^{(1)}_B \to G^{(1)}_4 = \cdots = A^{(l)} \approx B^{(l)} = L^{(l)}_B U^{(l)}_B \to G^{(l)}_4 = \cdots = G^{(L)}_4 \to A^{(L)} = L^{(L)}_A U^{(L)}_A.$$

Thus, the matrix A acquires a factorized combined preconditioner, and we obtain a single-level iterative process in Krylov subspaces, which we denote by DDM-MG.

This process satisfies the following theorem.

Theorem. *Let the matrix A in SLAE* (1) *be a Stieltjes matrix. Then the preconditioner B in the DDM-MGa method is an s.p.d. matrix, and the number of iterations* $n(\varepsilon)$ *of the CG method satisfies bound* (9), *where* κ *is the condition number of the preconditioned matrix* $B^{-1}A$.

The second option is based on a two-level iterative process, which includes solving the auxiliary system $G_4v_4 = r_4 - A_{4,3}v_3$ from (14) with the required accuracy using the multigrid method described above similar to [20]. The preconditioner implemented in this case turns out to be dynamic, and the use of the CG algorithm as an external solver for DDM is, strictly speaking, unjustified. A natural way here, is, e.g., to use the "flexible" conjugate gradient method FCG [28].

The third option is a development of the second one and includes a complement of the algebraic-geometric method with iterative smoothing operations used in the classical MG algorithms. Here, *V*-, *W*-, and *K*-cycles can be used which are described in extensive literature, and we will not dwell on these issues here.

5. CONCLUSIONS: TECHNOLOGICAL ISSUES AND PROSPECTS OF THE DDM-MG

Based on publications [19–21] and preliminary experimental studies, it can be concluded that the algebraic-geometric macrogrid domain decomposition methods and multigrid approaches developed separately but based on common structural principles, when combined into a unified algorithm, promise to provide a synergetic effect with a significant increase in the efficiency and performance of software. In

particular, these issues are relevant for solving practical problems with real data that have large condition numbers (up to 10^{14}) and high orders (10^{10}–10^{12} and greater). Note that often, when repeatedly solving SLAEs in interdisciplinary problems with nonlinear, nonstationary and singular effects, the calculation times on modern computers can reach tens of hours.

The developed computational methods and technologies are naturally applied to multiprocessor computing systems (MCS) with distributed and hierarchical shared memory, and they have a high level of parallelization and efficiency of processor usage

$$S_p = T_1(\mathcal{A})/T_p(\mathcal{A}), \quad F_p(\mathcal{A}) = S_p(\mathcal{A})/p,$$

where $T_p(\mathcal{A})$ is the computation time for problem $\mathcal{A}$ on p processors.

A feature of data structures and algorithms in the methodologies under examination is the logical complexity of operations with algebraic-topological objects, especially for problems with non-trivial configurations of the computational domain and unstructured grids. The implementation of DDM-MG algorithms is, in principle, possible on matrices represented in sparse-compressed forms such as GSR, but the computational complexity of such implementations will be much higher than when geometric grid information is used. In fact, this means the need to develop new-generation tools for the stages of grid generation and the construction of discrete models for solving problems. When constructing flexible and expandable algebraic data structures, using artificial intelligence tools [29], both the efficiency of developing software solvers and the performance of the resulting code can be significantly increased.

The next important step is the expansion of the class of problems to be solved and the development of the DDM-MG algorithms themselves. This includes, e.g., the solution of new types of SLAEs obtained from the discretization of systems of differential equations and from the application of higher-order approximate schemes. The solution of such problems puts on the agenda the formation of a new paradigm of applied research, some issues of which are discussed in [30].

FUNDING

This work was supported by the Russian Foundation for Basic Research, project no. 24-21-00402.

CONFLICT OF INTEREST

The author declares that he has no conflicts of interest.

REFERENCES

1. V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory and Parallel Implementation* (SIAM, Philadelphia, 2015).
2. Y. M. Laevsky and A. M. Matsokin, "Decomposition methods for solving elliptic and parabolic boundary value problems," Sib. J. Comput. Math. **2**, 361–372 (1999).
3. Y. Vassilevski and M. A. Olshanskii, *Short Course on Multi-Grid and Domain Decomposition Methods* (MAKS, Moscow, 2007).
4. Y. Notay, "Algebraic multigrid for Stokes equations," SIAM J. Sci. Comput. **39**, 88–111 (2017).
5. A. Brandt, "Algebraic multigrid theory: The symmetric case," J. Appl. Math. Comput. **19**, 23–56 (1986).
6. M. A. Olshanskii, *Lecture Notes on Multigrid Methods* (Inst. Vychisl. Mat., Ross. Akad. Nauk, Moscow, 2012).
7. Y. Saad, *Iterative Methods for Sparse Linear Systems,* 2nd ed. (SIAM, 2003).
8. M. A. Olshanskii and E. E. Tyrtyshnikov, *Iterative Methods for Linear Systems Theory and Applications* (SIAM, Philadelphia, 2014).
9. S. V. Nepomnyaschikh, "Decomposition and fictitious domains methods for elliptic boundary value problems," *Proc. of the Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Norfolk, VA, 1991 (SIAM, 1992), pp. 62–72.
10. V. G. Korneev, "Fast domain decomposition algorithms for elliptic problems with piecewise variable orthotropies," in T. Apel and O. Steinbach O. (eds), *Advanced Finite Element Methods and Applications. Lecture Notes in Applied and Computational Mechanics*, Ed. by T. Apel and O. Steinbach (Springer, Berlin, 2013). Vol. 66. DOI.org/10.1007/978-3-642-30316-6_3.
11. V. P. Il'in, "Iterative Preconditioned methods in Krylov spaces: trends of the 21st century," Comput. Math. Math. Phys. **61** (11), 1750–1775 (2021).
12. A. Reusken, "A multigrid method based on incomplete Gaussian elimination," Numer. Linear Alg. Appl. **3**, 369–390 (1996).

13. W. Hackbusch, B. N. Khoromskij, and R. Kriemann. "Direct Schur complement method by domain decomposition based on H-matrix approximation," Comput. Visual. Sci. **8** (3), 179–188 (2005). https://doi.org/10.1007/s00791-005-0008-3
14. F. A. Bornemann and P. Deuflhard, "The cascadic multigrid method for elliptic problems," Numer. Math. **75** (2), 135–152 (1996). https://doi.org/10.1007/s002110050234
15. V. V. Shaidurov, "Some estimates of the rate of convergence for the cascadic conjugate-gradient method," J. Comput. Math. Appl. **31** (4–5), 161–171 (1996). https://doi.org/10.1016/0898-1221(95)00228-6
16. Y. Notay and A. Napov, "A massively parallel solver for discrete Poisson-like problems," J. Comput. Phys. **281**, 237–250 (2015). https://doi.org/10.1016/j.jcp.2014.10.043
17. J. W. Ruge and K. Stüben, "Algebraic multigrid," in *Multigrid Methods* (SIAM, Philadelphia, 1987), pp. 73–130. https://doi.org/10.1137/1.9781611971057.ch4
18. J. Xu and L. Zikatanov, "Algebraic multigrid methods," Acta Numerica **26**, 591–721 (2017). https://doi.org/10.1017/S0962492917000083
19. V. P. Il'in, Y. I. Gurieva, and D. I. Kozlov, "Parallel domain decomposition methods with graph preconditioning," in *Parallel Computational Technologies* (South Ural State Univ., Chelyabinsk, 2023), pp. 215–228.
20. V. P. Il'in, "Multigrid incomplete factorization methods in Krylov subspaces," J. Math. Sci. **272**, 523–532 (2023).
21. M. Batalov and V. Il'in, "Multigrid incomplete factorization methods in Krylov subspaces on unstructured grids" in *Parallel Computational Technologies,* Ed. by L. Sokolinsky (Springer Nature, Cham, 2024), pp. 163–176.
22. Yu. A. Kuznetsov, "Algebraic multigrid domain decomposition methods," Preprint No. 232, OVM Akad. Nauk SSSR (Department of Computational Methods. SSSR Academy of Sciences, 1989). https://doi.org/10.1007/978-3-030-36592-9_41
23. R. Bank, R. Falgout, T. Jones, T. Manteuffel, S. McCormick, and J. Ruge, "Algebraic multigrid domain and range decomposition," SIAM J. Sci. Comput. **37**, 113–136 (2015).
24. V. P. Il'in, *Finite Element Methods and Technologies* (Inst. of Computational Mathematics and Mathematical Geophysics, Russian Academy of Sciences, 2007) [in Russian].
25. L. A. Knizhnerman, "Error bounds for the Arnoldi method: A set of extreme eigenpairs," J. Numer. Linear Algebra Appl. **296**, 191–211 (1999).
26. V. P. Il'in, "Biconjugate direction methods in Krylov subspaces," J. Appl. Indust. Math. **4** (1), 65–78 (2010).
27. V. P. Il'in, D. I. Kozlov, and A. V. Petukhov, "On the minimal residual methods for solving diffusion-convection SLAEs," J. Phys.: Conf. Ser. **2099**, 012005 (2021).
28. Y. Notay, "Flexible conjugate gradients," SIAM J. Sci. Comput. **22**, 1444–1460 (2000). https://doi.org/10. 1137/S1064827599362314
29. V. P. Il'in, "Parallel intelligent computing in algebraic problems," in *Parallel Computational Technologies,* Ed. by L. Sokolinsky and M. Zymbler, 2021, pp. 108–117.
30. V. P. Il'in, "Programming for the near future: Concepts and pragmatic considerations," Herald Russ. Acad. Sci. **93** (2), 92–102 (2023). https://doi.org/10.31857/S086958732302007X

Translated by A. Klimontovich

Publisher's Note. Pleiades Publishing remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.
AI tools may have been used in the translation or editing of this article.