DOI 10.1007/s10958-020-04926-7
Journal of Mathematical Sciences, Vol. 249, No. 2, August, 2020

ON MULTIGRID METHODS FOR SOLVING TWO-DIMENSIONAL BOUNDARY-VALUE PROBLEMS

Y. L. Gurieva,* V. P. Il'in,† and A. V. Petukhov* UDC 519.6

Various methods for constructing algebraic multigrid type methods for solving multidimensional boundary-value problems are considered. Two-level iterative algorithms in Krylov subspaces based on approximating the Schur complement obtained by eliminating the edge nodes of the coarse grid are described on the example of two-dimensional rectangular grids. Some aspects of extending the methods proposed to the multilevel case, to nested triangular grids, and also to three-dimensional grids are discussed. A comparison with the classical multigrid methods based on using smoothing, restriction (aggregation), coarse-grid correction, and prolongation is provided. The efficiency of the algorithms suggested is demonstrated by numerical results for some model problems.

1. Introduction

Solution of systems of linear algebraic equations (SLAEs) of high order (up to 10^{10} and more) with sparse ill-conditioned coefficient matrices arising from approximation of multidimensional boundary-value problems by finite difference, finite volume, finite element, and discontinuous Galerkin methods of various orders of accuracy on unstructured grids is an important problem of computational algebra because it is a bottle-neck of mathematical modeling and requires huge computer resources (the number of arithmetic operations and the memory required), increasing nonlinearly as the system dimension grows. Multi-scale and multi-phase problems with contrasting material properties, for which condition numbers can exceed 10^{13} (which is the maximally admissible value when using the standard double-precision arithmetic), are especially hard.

In addition to solving stationary boundary-value problems, systems of linear algebraic equations must also be solved in modeling dynamic processes using stable implicit schemes.

Modern approaches to solving SLAEs are based on preconditioned iterative methods in Krylov subspaces, see [1–3] and the references therein. The urgent problem of scaled parallelization of algorithms is mainly solved by applying additive domain decomposition methods, whose description can be found in [3–5]. The best theoretical estimates of the amount of computations are obtained for the multigrid algorithms. They appeared in the pioneering works by R. P. Fedorenko and N. S. Bakhvalov and since then have been intensively studied by many authors. An in-depth analysis of multigrid methods can be found in the monographs [6–7] and many other publications, among which we mention [8–14]. These methods can be subdivided into geometric (GMG) and algebraic (AMG) multigrid methods. The geometric methods can effectively use interpolation approaches based on geometric characteristics of grids. On the other hand, the algebraic methods are, in a sense, more universal because they only exploit algebraic properties of SLAEs. In general, both approaches are interpreted as construction of a multigrid preconditioner for an iterative Krylov type process. In this case, every step, i.e., application of a two-grid algorithm includes smoothing, restriction, prolongation, and coarse-grid correction operations, which are iteratively implemented in different versions of V-cycles or W-cycles. The transition to a multigrid version consists in applying the two-grid method recursively.

*Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia, e-mail: yana@lapasrv.sscc.ru, petukhov@lapasrv.sscc.ru.

†Novosibirsk State University, Novosibirsk, Russia, e-mail: ilin@sscc.ru.

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 482, 2019, pp. 13–27. Original article submitted October 17, 2019.

In this paper, a new version of the two-grid algebraic method is proposed, which can readily be extended to the multigrid case. The main idea is to construct the restriction and prolongation operators that approximate, in a sense, a Schur complement and are similar to the Uzawa algorithm. The main idea is exposed on the example of two-dimensional rectangular grids, with account for the specific information structure for the objects under consideration. Also we discuss the possibilities of extending the approach in question to nested triangular grids and to different types of three-dimensional grids.

The paper is organized as follows. In Sec. 2, we describe the general scheme of the Uzawa type multigrid methods. Section 3 contains a comparison of the algorithm implementation for the two-dimensional boundary-value problem with the "classical" multigrid approaches, which are based on the operations of restriction, smoothing, coarse-grid correction, and prolongation. The last section presents and discusses numerical results for some model problems.

2. The general scheme of the Uzawa type multigrid methods

Consider a symmetric system of linear five-point grid equations

$$(Au)_{i,j} \equiv -a_{i,j}u_{i-1,j} - b_{i,j}u_{i,j-1} - a_{i+1,j}u_{i+1,j} - b_{i,j+1}u_{i,j+1} + e_{i,j}u_{i,j} = f_{i,j},$$
$$i = 1, \dots L_1, \quad j = 1, \dots M_1, \quad L_1 M_1 = N_1, \tag{1}$$

approximating, on a rectangular grid Ω^h with N_1 nodes, the boundary-value problem for an elliptic equation in a rectangular computational domain by a finite difference, a finite volume, a finite element, or a discontinuous Galerkin method, see [15]. Assume that a sequence of m nested grids $\widehat{\Omega}_1 = \Omega^h, \widehat{\Omega}_2 = \Omega^{2h}, \dots, \widehat{\Omega}_m = \Omega^{2^{m-1}h}$ is given and let the number of nodes of the lth grid be equal to $N_l = L_l M_l$, where $L_l = (L_{l-1} - 1)/2 + 1 = (L_1 - 1)/2^{l-1} + 1$, $l = 1, \dots, m-1$, and all the numbers are integers (e.g., $L_1 = M_1 = 2^m + 1$).

We assume that the matrix A is positive semidefinite, all the coefficients occurring in Eqs. (1) are nonnegative, and A is (nonstrictly) diagonally dominant, i.e.,

$$|e_{i,j}| \geq |a_{i,j}| + |b_{i,j}| + |a_{i+1,j}| + |b_{i,j+1}|.$$

Note that for the near-boundary nodes, some coefficients are "zeroed" in order to take into account the boundary conditions, i.e., $a_{1,j} = b_{i,1} = a_{L+1,j} = b_{i,M+1} = 0$.

The system of equations (1) is symmetrically scaled in such a way that all the diagonal matrix entries are equal to unity. As a result, we obtain a symmetric SLAE of the form

$$\bar{u}_{i,j} - \bar{a}_{i,j}\bar{u}_{i-1,j} - \bar{b}_{i,j}\bar{u}_{i,j-1} - \bar{a}_{i+1,j}\bar{u}_{i+1,j} - \bar{b}_{i,j+1}\bar{u}_{i,j+1} = \bar{f}_{i,j},$$
$$\bar{a}_{i,j} = a_{i,j}\left(\frac{e_{i,j}}{e_{i-1,j}}\right)^{-1/2}, \quad \bar{b}_{i,j} = b_{i,j}\left(\frac{e_{i,j}}{e_{i,j-1}}\right)^{-1/2}, \quad \bar{u}_{i,j} = u_{i,j}e_{i,j}^{1/2}, \quad \bar{f}_{i,j} = f_{i,j}e_{i,j}^{-1/2}. \tag{2}$$

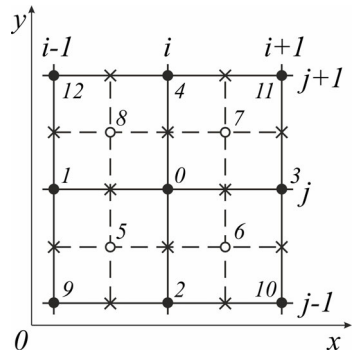

Fig. 1. The local node numbering for the multigrid method.

The multigrid methods result from recursive application of the two-grid algorithm, which is described in accordance with the notation presented in Fig. 1. In Fig. 1, the black dots denote the nodes of the coarse grid Ω^{2h}, whereas the circles and crosses are the nodes of the fine grid Ω^h, which are located, respectively, in the middles of the cells and at the centers of the edges of the coarse grid Ω^{2h}. The corresponding subsets of nodes are denoted by Ω_1, Ω_2, and Ω_3, and the resulting geometric types of points with respect to Ω^{2h} will be referred to as nodal, edge, and facial ones.

Assume that $\bar{u}_{i,j}$ are the values of the solution of SLAE (1) on the fine grid. Upon eliminating the edge unknowns from SLAE (2) using relations of the form

$$\begin{aligned}\bar{u}_{i-1,j} &= \bar{f}_{i-1,j} + \bar{a}_{i-1,j}\bar{u}_{i-2,j} + \bar{b}_{i-1,j}\bar{u}_{i-1,j-1} + \bar{a}_{i,j}\bar{u}_{i,j} + \bar{b}_{i-1,j+1}\bar{u}_{i-1,j+1},\\ \bar{u}_{i,j-1} &= \bar{f}_{i,j-1} + \bar{a}_{i,j-1}\bar{u}_{i-1,j-1} + \bar{b}_{i,j-1}\bar{u}_{i,j-2} + \bar{a}_{i+1,j-1}\bar{u}_{i+1,j-1} + \bar{b}_{i,j}\bar{u}_{i,j}\end{aligned} \tag{3}$$

(similar expressions are also used for $\bar{u}_{i+1,j}$ and $\bar{u}_{i,j+1}$, where $(i \pm 1, j), (i, j \pm 1) \in \Omega_3$), we obtain the following symmetric system of nine-point equations at the "circle" type nodes:

$$\begin{aligned}&p^0_{i,j}u_{i,j} - p^1_{i,j}u_{i-2,j} - p^2_{i,j}u_{i,j-2} - p^3_{i,j}u_{i+2,j} - p^4_{i,j}u_{i,j+2} - p^5_{i,j}u_{i-1,j-1} - p^6_{i,j}u_{i+1,j-1}\\ &- p^7_{i,j}u_{i+1,j+1} - p^8_{i,j}u_{i-1,j+1} = \tilde{f}_{i,j}, \quad (i,j) \in \Omega_v.\end{aligned}$$

Here, the coefficients introduced are determined by the following formulas:

$$\begin{aligned}&p^0_{i,j} = 1 - \bar{a}^2_{i,j} - \bar{b}^2_{i,j} - \bar{a}^2_{i+1,j} - \bar{b}^2_{i,j+1},\\ &p^1_{i,j} = \bar{a}_{i,j}\bar{a}_{i-1,j}, \quad p^2_{i,j} = \bar{b}_{i,j}\bar{b}_{i,j-1}, \quad p^3_{i,j} = \bar{a}_{i+1,j}\bar{a}_{i+2,j} = p^1_{i+2,j},\\ &p^4_{i,j} = \bar{b}_{i,j+1}\bar{b}_{i,j+2} = p^2_{i,j+2}, \quad \bar{p}^5_{i,j} = \bar{a}_{i,j}\bar{b}_{i-1,j} + \bar{a}_{i,j-1}\bar{b}_{i,j}, \quad p^6_{i,j} = \bar{a}_{i+1,j-1}\bar{b}_{i,j} + \bar{a}_{i+1,j}\bar{b}_{i+1,j},\\ &p^7_{i,j} = \bar{a}_{i+1,j}\bar{b}_{i+1,j+1} + \bar{a}_{i+1,j+1}\bar{b}_{i,j+1}, \quad p^8_{i,j} = \bar{a}_{i,j}\bar{b}_{i-1,j+1} + \bar{a}_{i,j+1}\bar{b}_{i,j+1}\\ &\tilde{f}_{i,j} = \bar{f}_{i,j} + \bar{a}_{i,j}\bar{f}_{i-1,j} + \bar{b}_{i,j}\bar{f}_{i,j-1} + \bar{a}_{i+1,j}f_{i+1,j} + \bar{b}_{i,j+1}f_{i,j+1}.\end{aligned} \tag{4}$$

Similar nine-point equations are also obtained for the nodes of the coarse grid, i.e., for $(i,j) \in \Omega_1$. Now represent the above transformations in matrix-vector form. By u_1, u_2, u_3 we denote the subvectors whose components correspond to the nodes of the black dot, circle, and cross types, i.e., we set

$$u_l = \{u_{i,j} : (i,j) \in \Omega_l, \quad l = 1,2,3\}.$$

Then the original SLAE takes the form

$$\begin{aligned}&A_{1,1}u_1 + A_{1,3}u_3 = f_1,\\ &A_{2,2}u_2 + A_{2,3}u_3 = f_2,\\ &A_{3,1}u_1 + A_{3,2}u_2 + A_{3,3}u_3 = f_3.\end{aligned} \tag{5}$$

We will assume that this system is obtained after performing the scaling operations (2). In this case, the diagonal blocks are identity matrices ($A_{1,1} = A_{2,2} = A_{3,3} = I$); the off-diagonal matrix entries and the components of the subvectors u_1, u_2, u_3 are the values $\bar{a}_{i,j}$, $\bar{b}_{i,j}$, and $\bar{u}_{i,j}$ defined in (2).

Upon eliminating the subvector u_3 from (5) using the relation

$$u_3 = A^{-1}_{33}(f_3 - A_{3,1}u_1 - A_{3,2}u_2),$$

which follows from the last equation in (5), we arrive at the following system for the subvectors u_1 and u_2:

$$\bar{A}u \equiv \begin{cases}\bar{A}_{1,1}u_1 + \bar{A}_{1,2}u_2 = \bar{f}_1,\\ \bar{A}_{2,1}u_1 + \bar{A}_{2,2}u_2 = \bar{f}_2.\end{cases} \tag{6}$$

Here, the new matrices and vectors are defined by the relations

$$\bar{A}_{1,1} = I - A_{1,3}A_{3,1}, \quad \bar{A}_{1,2} = -A_{1,3}A_{3,2}, \quad \bar{f}_1 = f_1 - A_{1,3}A_{3,3}^{-1}f_3,$$
$$\bar{A}_{2,1} = -A_{2,3}A_{3,1}, \quad \bar{A}_{2,2} = I - A_{2,3}A_{3,2}, \quad \bar{f}_2 = f_2 - A_{2,3}A_{3,3}^{-1}f_3. \tag{7}$$

Note that the entries of the matrices $\bar{A}_{1,1}$ and $\bar{A}_{1,2}$ involve the coefficients $p_{i,j}^0$, $p_{i,j}^1, \ldots, p_{i,j}^4$ and $p_{i,j}^5, \ldots, p_{i,j}^8$, respectively, for $(i,j) \in \Omega_1$, see (4). If $(i,j) \in \Omega_2$, then these two groups of coefficients correspond to the matrices $\bar{A}_{2,2}$ and $\bar{A}_{2,1}$, respectively. The matrices $\bar{A}_{1,1}$ and $\bar{A}_{2,2}$, occurring in these relations, are five-diagonal, and they will be referred to as the Schur complements of the first level.

Note that the diagonal blocks $\bar{A}_{l,l}$ in (7) are five-diagonal matrices, and their condition numbers are independent of the grid step size h. For example, for the model problem corresponding to the approximation of the Laplace operator on a square grid, in which case all the nonzero off-diagonal entries of the matrix A equal $1/4$, the eigenvalues λ of the matrices $\bar{A}_{l,l}$ belong to the interval $[\lambda_{\min} = 1/2,\ \lambda_{\max} = 1]$.

From the algebraic system (6) we again eliminate some unknowns. Using the second equation, we express the subvector u_2 as

$$u_2 = \bar{A}_{2,2}^{-1}(\bar{f}_2 - \bar{A}_{2,1}u_1)$$

and substitute this expression into the first equation. In this way, we obtain the following SLAE for the nodes of the coarse grid:

$$\widehat{A}u_1 \equiv (\bar{A}_{1,1} - \bar{A}_{1,2}\bar{A}_{2,2}^{-1}\bar{A}_{2,1})u_1 = \widehat{f} \equiv \bar{f}_1 - \bar{A}_{1,2}\bar{A}_{2,2}^{-1}\bar{f}_2. \tag{8}$$

Observe that the matrix $\widehat{A}$, which is the second-level Schur complement, is dense. However, multiplication of vectors by this matrix can be implemented in an efficient way. Indeed, the most time-consuming operation here is solution of a SLAE with the well-conditioned coefficient matrix $\bar{A}_{2,2}$, which can be done using the Chebyshev acceleration.

In order to solve SLAE (8), consider the following modified stationary Jacobi type block method:

$$B(u^{n+1} - u^n) = \bar{f} - \bar{A}u^n \equiv r^n,$$
$$B = \text{block} - \text{diag}\{\widehat{A}_{l,l} :\ l = 1,2\}, \tag{9}$$
$$\widehat{A}_{l,l} = \bar{A}_{l,l} - \theta D_l, \quad \theta \in [0,1].$$

Here, D is the diagonal matrix determined by the condition

$$Be = \widehat{A}e \quad \text{for} \quad \theta = 1, \quad \text{i.e.,} \quad De = \bar{A}_{1,2}\bar{A}_{2,2}^{-1}\bar{A}_{2,1}e,$$

where e is the vector with unit components. The parameter θ should be selected in such a way that the preconditioner B is optimized, i.e., the condition number of the matrix $B^{-1}\widehat{A}$ and the number of iterations in the algorithm (9) are minimized. (The problem of how to choose θ in practice still needs a special investigation.) Here and below, the stopping criterion for terminating the iterative process is as follows:

$$||r^n||_2^2 \equiv (r^n, r^n) \ \le \varepsilon^2 (f,f), \quad \varepsilon \ll 1, \tag{10}$$

where r^n is the residual vector.

The iterative process (9) can be accelerated by applying a preconditioned conjugate direction method in Krylov subspaces or a Chebyshev type spectral algorithm, which can be written in

the following form, see [16] (in the sequel, for convenience, the vector f and the matrix A are written without the symbol " $\wedge$ "):

$$\begin{aligned} & r^0 = f - Au^0, \quad p_0 = B^{-1}r^0, \\ n = 1, 2, \cdots : & \\ & u^n = u^{n-1} + \alpha_{n-1}p^{n-1}, \\ & r^n = r^{n-1} - \alpha_{n-1}Ap^{n-1}, \\ & p^n = B^{-1}r^n + \beta_n p^{n-1}. \end{aligned} \tag{11}$$

Here, the coefficients α_n and β_n in the Chebyshev algorithm are computed by the formulas

$$\begin{aligned} & \alpha_0 = \tau, \quad \alpha_n = \gamma_n\tau, \quad \beta_n = (\gamma_n - 1)\alpha_{n-1}/\alpha_n, \quad \tau = 2/(\lambda_1 + \lambda_N), \\ & \gamma_n = 4/(4 - \gamma_{n-1}\gamma^2), \quad \gamma_0 = 2, \quad \gamma = (C-1)/(1+C), \quad C = \lambda_N/\lambda_1, \end{aligned} \tag{12}$$

where λ_1 and λ_N are the smallest and largest eigenvalues of the matrix $B^{-1}A$. In the Krylov methods, the formulas

$$\alpha_n = \sigma_n/\rho_n, \quad \sigma_n = (B^{-1}r^n, A^\delta r^n), \quad \rho_n = (Ap^n, A^\delta p^n), \quad \beta_n = \sigma_{n+1}/\sigma_n \tag{13}$$

are used. Here, $\delta = 1$ corresponds to the conjugate residual algorithm, whereas $\delta = 0$ corresponds to the conjugate gradient algorithm.

In both cases, the number of iterations required to satisfy the stopping criterion (10) is bounded from above as follows:

$$n(\varepsilon) \le 1 + |\ln(\varepsilon/2)|\sqrt{c}/2, \quad c = \lambda_N/\lambda_1. \tag{14}$$

In the case of a singular matrix A, for the conjugate direction method, the constant c in (14) is the effective condition number, and λ_1 is the smallest nonzero eigenvalue, see [17]. However, for the method (11)–(12) to converge and the bound (14) to be valid, the linear algebraic system to be solved must be compatible, i.e., the equality $Au = f$ must hold for one vector u at least.

Note that at every iteration of the form (9), algebraic systems with the matrices B and $\bar{A}_{2,2}$ must be solved, each of which is of order about four times smaller than that of the original SLAE (1). The extremal eigenvalues $\lambda_{\min}$, $\lambda_{\max}$ of these matrices, which are well-conditioned for $\theta < 1$, can be estimated using, e.g., the Gerschgorin circle theorem [16]. For this reason, here it is reasonable to apply the efficient Chebyshev acceleration method, which involves no inner products of vectors and is well parallelizable. The resulting two-level iterative process can be optimized by selecting a suitable value of the parameter θ and an appropriate degree of the Chebyshev polynomial. These two parameters determine the number of inner iterations, which, in general, can vary from iteration to iteration.

Note that for the outer and inner iterative processes for solving SLAEs with the coefficient matrices $\widehat{A}_{1,1}$ and $\bar{A}_{2,2}$ one can use distinct values ε_1 and ε_2 in the stopping criterion (10). Obviously, the subvectors u_2 and u_3 can be computed only once upon terminating iterations for the SLAE (8). It should also be mentioned that if at both levels of the iterative process under consideration the conjugate direction method is applied, then, strictly speaking, even for symmetric SLAEs it is necessary to use "flexible" preconditioning [18] with long recursions, which makes the algorithm significantly more expensive.

The method in question is, in a sense, similar to the Uzawa algorithm because it is based on preliminary elimination of subvectors. It can also be interpreted as a non-standard two-grid approach, which requires solution of the subsystems for the vectors u_1 and u_2, corresponding to the two coarse grids Ω_1 and Ω_2 with step size $2h$. It is important to emphasize that if the sparse system (8) is solved sufficiently accurately, then the subvectors u_1, u_2, and u_3

themselves provide a good approximate solution of the SLAE (1), which is in contrast with the conventional multigrid method.

Based on the above-described two-grid method, multigrid iterative algorithms can readily be constructed in a recursive manner. Indeed, in solving Eqs. (8), a coarser grid with step size $4h$ can be used for the subvector u_1. In this case, each of the subsystems with the coefficient matrices B and $\bar{A}_{2,2}$ reduces to two subsystems of halved order, and then the process of grid coarsening can be continued, the number of subsystems being doubled. Thus, if an algorithm with m levels is used, then, on the coarsest grid, it is necessary to solve 2^m independent algebraic subsystems, which can be performed concurrently on different processors. In this way, the degree of parallelism of computations can be increased.

A similar hierarchical approach can be transferred to nested triangular grids, including the case of finite element approximations with higher-order Lagrangian basis functions. In this case, the block structure of the original matrix A for the two-grid version is still of the form (5), and the subvectors u_1, u_2, and u_3 still correspond to the node, edge, and face types of unknowns, respectively, which are denoted by the symbols $\bullet$, $\circ$, and $\times$. Therefore, the matrix-vector form (8) of the iterative algorithm remains the same. When passing to three-dimensional nested polygonal grids of different types (with "parallelepiped," "tetrahedron," "prism," etc. finite elements), the principles of classifying the unknowns and matrix partitioning remain the same: in the course of hierarchical refinement of a coarse grid, the volume type nodes appear, whence the block order of the original matrix increases by one.

3. Comparison with the classical multigrid approaches

In the up-to-date interpretation, the general two-grid AMG method can be represented as a sequence of the following stages, see [8–10]:

1. On a fine grid Ω_1, given an initial guess u_1^0, the residual vector is computed,

$$r_1^0 = f - A_1 u_1^0, \quad A_1 = A.$$

2. For the vector r_1^0, preprocessing (preliminary smoothing) is performed, as a rule, by carrying out a few iterations of a simple algorithm,

$$r_1^1 = S_1 r_1^0, \tag{15}$$

where S_1 is an operator (or a matrix) of this stage (pre-smoothing). More specifically, this stage is implemented in two steps. The first one computes the direction vector

$$\tilde{A}_1 p_1^0 = r_1^0, \tag{16}$$

where $\tilde{A}_1$ is a certain approximation of the matrix A; at the second step, the corresponding residual is computed,

$$r_1^1 = f - A p_1^0. \tag{17}$$

3. From the vector r_1^1, corresponding to the fine grid Ω_1, the residual vector r_2^1 for the coarse grid Ω_2 is computed,

$$r_2^1 = R r_1^1, \quad R \in \mathcal{R}^{N_1,N_2}, \quad r_2^1 \in \mathcal{R}^{N_2}, \tag{18}$$

where R is a certain restriction operator (restriction stage).

4. On the coarse grid, the direction vector p_2^1 is computed from the solution of the SLAE

$$A_2 p_2^1 = r_2^1, \quad A_2 \in \mathcal{R}^{N_2,N_2}, \quad p_2^1,\ r_2^1 \in \mathcal{R}^{N_2}, \tag{19}$$

where A_2 is the coefficient matrix of the SLAE for the grid Ω_2.

5. The vector p_2^1 found from the solution of system (19) is prolongated from the coarse grid Ω_2 to the fine grid Ω_1 (prolongation stage),

$$p_1^1 = Pp_2^1, \quad P \in \mathcal{R}^{N_1,N_2}, \quad p_1^1 \in \mathcal{R}^{N_1}. \tag{20}$$

6. For the vector p_1^1, the corresponding residual vector on the fine grid is computed (residual update),

$$r_1^2 = r_1^1 - Ap_1^1, \quad r_1^1 \in \mathcal{R}^{N_1}. \tag{21}$$

7. "Post-processing" is carried out for the newly obtained residual vector on the fine grid Ω_1; simultaneously, the new direction vector p_1^2 is computed from the solution of the auxiliary SLAE with the matrix $\tilde{A}_1$ (the post-smoothing stage),

$$\tilde{A}_1 p_1^2 = r_1^2. \tag{22}$$

8. The resulting direction vector is obtained as the sum

$$p_1 = p_1^0 + p_1^1 + p_1^2 = Br_1^0,$$

where B is the preconditioning matrix of the two-grid method under consideration.

Specific variants of multigrid approaches, which have already become classical, differ in the ways of choosing the matrix operators that determine the successive stages of the above computational scheme. In general, the preconditioning matrix for a two-grid method for solving SLAE (1) can be represented in the form

$$B = S_2 P A_c^{-1} R S_1 \in \mathcal{R}^{N_1,N_1}, \tag{23}$$

where S_1 and S_2 are the pre-smoothing and post-smoothing operators defined in Ω^h; $R \in \mathcal{R}^{N_1,N_2}$ and $P \in \mathcal{R}^{N_2,N_1}$ are the restriction and prolongation matrices; $A_c \in \mathcal{R}^{n_2,N_2}$ is the matrix that determines the coarse grid correction; N_1 and N_2 are the dimensions of vectors defined on the fine and coarse grids Ω^h and Ω^{2h}, respectively. If the matrix A of the original SLAE is symmetric, then it is natural to use a symmetric preconditioning matrix B. In this case, it is reasonable to set $P = R^T$, $A_c = A_c^T$, and $S_1 = S_2^T = S = S^T$. In the special case of the so-called Galerkin approximation, one also sets $A_c = P^T AP$. As a result, the preconditioning matrix has the form

$$B = SP(P^T AP)^{-1} P^T S. \tag{24}$$

Below, as an example, we consider the following version, based on the approach proposed in [12].

Algorithm 1. In solving a nine-point symmetric SLAE approximating a two-dimensional boundary-value problem for an elliptic equation on a rectangular grid, as a smoother one uses the Iteration Line LU (ILLU) iterative method, which is nothing else than the implicit incomplete factorization algorithm proposed in [19, 20]. The operation of prolongation from a coarse grid to a fine one is based on bilinear interpolation, applied to the neighboring grid nodes, and the restriction operator is defined by the rule $R = P^T$. The matrix for the coarse grid is defined by $A_c = P^T AP$. As a result, the preconditioning matrix is of the form (24). The multigrid method is obtained by recursively applying the two-grid algorithm $\bar{m}$ times, where $\bar{m}$ is a given number. The outer iterative process is the Jacobi method (without Krylov acceleration). On the coarsest grid, the SLAE with the coefficient matrix A_c is solved by a direct or an iterative algorithm, which is of no practical importance for large values of m.

4. Numerical results

In this section, we consider the results of preliminary experimental studies of the algebraic two-grid and multigrid approaches on simple model examples, including implementation of the described iterative algorithms for the finite difference solution of the Poisson equation in a square computational domain on a uniform rectangular grid. The computations were carried out for relatively small numbers of nodes equal to $N = 64^2, 128^2, 256^2, 512^2, 1024^2$. For this reason, we aimed at evaluating the mathematical efficiency of the methods in dependence of the values of the problem and algorithms parameters rather than at achieving the best performance. All computations were carried out in the standard double-precision arithmetic on the Intel(R) Core(TM) i7-770HQ CPU @ 2.80 GHz 2.80 GHz computer.

In Table 1, we present the results of numerical solution of the Neumann model problem for the Poisson equation in the square computational domain $\Omega = [0,1] \times [0,1]$ approximated by the standard five-point scheme [16] on square grids with $N = 64^2$, 128^2, 256^2, 512^2, 1024^2 cells. We used the one-dimensional solution $u = \sin \pi x$ as the test one and $u = 0$ as the initial guess. In this case, in accordance with [12], Algorithm 1 with $m = 2$, 3, 4, 5, 6 nested grids was used. In every case, the SLAE on the coarsest grid was solved by the iterative conjugate residual method with $\varepsilon = 10^{-8}$ in the stopping criterion (10) of the iterations (11). For the outer iterative (Jacobi) method, the same stopping criterion was used. Every cell of Table 1 contains two numbers: the total computation time (including the preparatory operations) and the number of outer iterations.

Table 1. Numerical results for Algorithm 1.

$m \setminus N$	64^2	128^2	256^2	512^2	1024^2
2	0.028 6	0.152 5	0.968 5	7.41 5	62.2 5
3	0.010 6	0.052 5	0.341 5	2.45 5	20.4 5
4	0.008 6	0.042 6	0.256 5	1.78 5	15.0 5
5	0.007 6	0.04 6	0.246 5	1.68 5	14.3 5
6	0.007 6	0.04 6	0.249 6	1.82 5	14.3 5

As is seen from the data presented, the number of outer iterations is practically independent of the problem dimension and of the number of nested grids, which shows the importance of choosing a high-quality iterative process. The total computation time decreases as the number of grids grows, but only up to a certain limit (in the above computations, it is not advisable to use more than five grids).

Note that the computation times in Table 1 are rather large because the contribution of preparatory operations is essential. The overall efficiency of the algorithm in question would increase significantly if we were solving a series of SLAEs with one and the same coefficient matrix and many right-hand sides.

Table 2 presents similar numerical results obtained by applying the two-grid version of the Uzawa type algorithm (6)–(13) from Sec. 2. In this case, the outer iterative process also was carried out by the conjugate residual method, and, at every iteration, the auxiliary SLAEs with the matrices B and $\bar{A}_{2,2}$ were solved using the Chebyshev acceleration (with fixed degrees

of polynomials and the bounds for the matrix eigenvalues determined numerically from the Gerschgorin circles).

Every cell of Table 2 also contains the computation time in seconds and the number of outer iterations. The same grid sizes are considered, but the values of m, in this case, are the orders of the polynomials used in the Chebyshev acceleration. The value of the damping iterative parameter θ was chosen experimentally and, in the above computations, was equal to $\theta = 0.975$ almost everywhere (strictly speaking, its optimal value tends to unity as N grows).

Table 2. Numerical results for the Uzawa type algorithm.

$m \setminus N$	64^2	128^2	256^2	512^2	1024^2
	0.021	0.122	0.576	4.9	50.3
4	23	31	57	106	198
	0.020	0.112	0.454	3.37	35.4
8	17	22	37	64	118
	0.022	0.078	0.472	2.84	30.5
12	15	21	34	48	89

The above-presented preliminary results of experimental studies demonstrate the potential of applying the Uzawa type iterative algorithm. However, the important issue of optimizing the choice of the iterative parameters needs a special study.

This work was supported by the Russian Foundation for Basic Research (project No. 18-01-00295).

Translated by the authors.

REFERENCES

1. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed., SIAM (2003).
2. V. P. Il'in, *Finite Element Methods and Technologies* [in Russian], ICM & MG SBRAS Publ., Novosibirsk (2007).
3. M. A. Olshanskii and E. E. Tyrtyshnikov, *Iterative Methods for Linear Systems. Theory and Applications*, SIAM (2014).
4. V. Dolean, P. Jolivet, and F. Nataf, *An Introduction to Domain Decomposition Methods: Algorithms, Theory and Parallel Implementation*, SIAM (2015).
5. V. P. Il'in, "Problems of parallel solution of large systems of linear algebraic equations," *Zap. Nauchn. Semin. POMI*, **439**, 112–127 (2015).
6. W. Hackbusch, *Multi-Grid Methods and Applications* (Springer Ser. Comput. Math., **4**), Springer, Berlin (1985).
7. V. V. Shaidurov, "Some estimates of the rate of convergence for the cascadic conjugate gradient method," *Comput. Math. Appl.*, **31**, No. 4-5, 161–171 (1996).
8. G. Lu, X. Jiao, and N. Missirlis, "A hybrid geometrical algebraic multigrid method with semi-iterative solution," *Numer. Linear Algebra Appl.*, **21**, No. 2, 212–258 (2014).
9. Y. Notay and A. Napov, "Further comparison of additive and multiplicative coarse grid correction," *Appl. Numer. Math.*, **65**, 53–62 (2013).
10. A. Napov and Y. Notay, "An efficient multigrid method for graph Laplacian systems. II: Robust aggregation," *SIAM J. Sci. Comput.*, **39**, No. 5, 379–403 (2017).
11. Y. Notay, "Analysis of two-grid methods. The nonnormal case," Universite Libre de Bruxelles, Report GANMN 18-01 (2018).

12. P. M. de Zeeuw, "Matrix-dependent prolongations and restrictions in a blackbox multigrid solver," *J. Comput. Appl. Math.*, **33**, 1–27 (1990).
13. Y. Shapira, M. Israeli, and A. Sidi, "Towards automatic multigrid algorithms for spd nonsymmetric and indefinite problems," *SIAM J. Sci. Comput.*, **17**, No. 2, 439–453 (1996).
14. Y. Notay, "Algebraic two-level convergence theory for singular systems," *SIAM J. Matrix Anal. Appl.*, **37**, 1419–1439 (2016).
15. V. P. Il'in, *Mathematical Modeling*. Part 1. *Continous and Discrete Models* [in Russian], SD RAS Publ., Novosibirsk (2017).
16. V. P. Il'in, *Finite Difference and Finite Volume Methods for Elliptic Equations* [in Russian], Inst. of Math. Publ., Novosibirsk (2001).
17. R. Nabben and C. Vuik, "A comparison of deflation and coarse grid correction applied to a porous media flows," *SIAM J. Numer. Anal.*, **42**, 1631–1647 (2004).
18. Y. Notay, "Flexible conjugate gradients," *SIAM J. Sci. Comput.*, **22**, No. 4, 1444-1460 (2000).
19. O. Axelsson, S. Brinkemper, and V. P. Il'in, "On some versions of incomplete block-matrix factorization iterative method," Catholic Univ. of Nijmegen, Preprint 8392 (1983).
20. P. Concus, G. Golub, and G. Meurant, "Block preconditioning for the conjugate gradient method," *SIAM J. Sci. Statist. Comput.*, **6**, No. 1, 220–252 (1985).