DOI 10.1007/s10958-018-3907-9
Journal of Mathematical Sciences, Vol. 232, No. 6, August, 2018

ON COARSE GRID CORRECTION METHODS IN KRYLOV SUBSPACES

Y. L. Gurieva* **and V. P. Il'in*** UDC 519.6

Two approaches using coarse grid correction in the course of a certain Krylov iterative process are presented. The aim of the correction is to accelerate the iterations. These approaches are based on an approximation of the function sought for by simple basis functions having finite supports. Additional acceleration can be achieved if the iterative process is restarted and the approximate solution is refined. In this case, the resulting process turns out to be a two-level preconditioned method. The influence of different parameters of the iterative process on its convergence is demonstrated by numerical results. Bibliography: 6 *titles.*

1. Introduction

Algorithms of coarse-grid correction have originally been proposed as a method for accelerating iterative domain decomposition methods (DDMs), which, in their turn, provide the main approach to scalable parallelization in solving large systems of linear algebraic equations (SLAEs) with sparse coefficient matrices arising from finite volume or finite element approximations of multidimensional boundary-value problems on unstructured grids, see [1] and the references therein. Parallel implementation of DDMs is traditionally carried out using two-level Krylov processes, the outer one realizing block iterations over subdomains, and the inner one effecting the synchronous solution of algebraic subsystems in subdomains. From the standpoint of the overall complexity of the algorithms under consideration, it is urgent to have a fast method for solving relatively small SLAEs in subdomains, which must be solved repeatedly, i.e., at each outer iteration. It is quite natural to solve each of the subsystems on its "own" multi-core processor using multithreaded computations on a shared memory. Since, in this case, application of algorithms of incomplete triangular factorization for preconditioning the inner Krylov iterations implies difficulties in parallelization [2], the present paper considers other approaches that are based on coarse-grid correction and least squares methods (LSMs) [3, 4].

Coarse grid correction is accomplished using finite basis functions of the first, second, and third orders on a coarse grid, connected with a set of subdomains in the DDM concept, and it can be interpreted, if desired, as a variant of the multigrid approach. The coefficients of a linear combination of the basis functions are computed from the condition of residual minimization for the initial SLAE in the resulting preconditioned Krylov iterations. Specifically, we use the conjugate gradient algorithm (even for unsymmetric SLAEs) with restarts and a periodic optimization of the solution approximations by the LSM, which actually generates a preconditioner as a low-rank approximation of the original matrix. The approach applied can be regarded as deflation or augmentation in the sense of extending the basis for the initial iteration process. In order to accelerate the convergence of the algorithm, we also use an "outer" level of acceleration by the LSM, which is based on combining residuals at "restarts."

The paper is organized as follows. In Sec. 2, the algorithms proposed and their basic properties are described. In Sec. 3, we discuss the construction of the basis functions. In Sec. 4, the efficiency of the approaches suggested is investigated numerically on a series of grid SLAEs for a two-dimensional convection-diffusion Dirichlet boundary-value problem. In

*Institute of Computational Mathematics and Mathematical Geophysics SO RAS, Novosibirsk, Russia; Novosibirsk State University, Novosibirsk, Russia, e-mail: yana@lapasrv.sscc.ru, ilin@sscc.ru.

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 463, 2017, pp. 44–57. Original article submitted November 1, 2017.

the Conclusion, the prospects of using the iterative processes constructed in solving practical problems are analyzed.

2. Conjugate gradient methods with coarse-grid correction of various orders

The aim of this paper is to construct and study numerically fast and parallelizable iterative algorithms for solving large SLAEs of the form

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

with sparse ill-conditioned coefficient matrices, including unsymmetric ones, which arise from finite volume or finite element approximations of multidimensional boundary-value problems on unstructured grids. More specifically, we consider real and not "superlarge" algebraic systems that fit into the shared memory of a single multi-core processor, so that the software implementation of the algorithms is performed with multithreading technologies without energy consuming and relatively slow data exchanges. It is assumed, in particular, that the approaches proposed can be used in two-level domain decomposition methods with scalable parallelism in solving subsystems in grid subdomains.

On the one hand, we will consider an algebraic method of coarse-grid correction as a preconditioning method for Krylov iterations. On the other hand, the method construction is based on approximation principles. In abstract form, this approach is as follows. Given a system of basis functions, or vectors $\phi_1, \ldots, \phi_m$, we will refine an approximate solution u^n of the linear system (1) using the representation

$$\widetilde{u}^n = u^n + c_1\phi_1 + \cdots + c_m\phi_m = u^n + \Phi c, \quad m \ll N, \tag{2}$$

where $c = (c_1, \ldots, c_m)^T$, the subscript T means transposition, and the columns of the rectangular matrix $\Phi = (\phi_1 \ldots \phi_m) \in \mathcal{R}^{N,m}$ are the vectors ϕ_k, $k = 1, \ldots, m$. The corresponding residual vector is written in the form

$$\widetilde{r}^n = f - A\widetilde{u}^n = r^n - \Psi c, \quad \Psi = A\Phi. \tag{3}$$

Then the vector c and the corresponding corrected approximate solution $\widetilde{u}^n$ can be computed from the condition of minimization of the residual norm $\|\widetilde{r}^n\|_2 = (\widetilde{r}^n, \widetilde{r}^n)^{1/2}$. By using the LSM [4], we obtain a SLAE

$$\widehat{B}\widehat{c} = \Psi^T\Psi\widehat{c} = \Psi^T r^n \equiv \widehat{g}^n, \quad \widehat{B} \in \mathcal{R}^{m,m}, \tag{4}$$

which can be regarded as a consequence of the orthogonality relation $\Psi^T\widetilde{r}^n = 0$, and $\widehat{c}$ is a solution of system (4) in a generalized sense. The matrix $\widehat{B}$ is nonsingular whenever Ψ has full rank m. In the general case, even if $\widehat{B}$ is singular, system (4) is consistent, and the vector $\widehat{c}$ sought for can formally be defined in terms of the generalized inverse matrix B^+ by the formula $\widehat{c} = \widehat{B}^+\Psi^T r^n$. In practice, SLAE (4) can always be solved, for example, using the singular value decomposition method [4]. Further, relations (2) and (3) are readily brought to the form

$$\begin{aligned} &\widehat{u}^n = u^n + \Phi\widehat{B}^+\Psi^T r^n, \quad \widehat{B} = \Psi^T\Psi = \Phi^T A^T A\Phi, \\ &\widehat{r}^n = r^n - \Psi(\Psi^T\Psi)^+\Psi^T r^n = \widehat{H}r^n, \quad \widehat{H} = I - A\Phi(\Phi^T A^T A\Phi)^+\Phi^T A^T. \end{aligned} \tag{5}$$

Note that instead of using the LSM (4), (5), one can also use the correction obtained from the condition of orthogonality of the residual $\widetilde{r}^n$ to the vectors $\phi_1, \ldots, \phi_m$. In matrix form, this is written as $\Phi^T\widetilde{r}^n = 0$. Thus, as a result of (3), we obtain

$$\breve{B}\breve{c} = \Phi^T A\Phi\breve{c} = \Phi^T r^n \equiv \breve{g}^n. \tag{6}$$

In this case, instead of relations (5) we have

$$\breve{u}^n = u^n + \Phi\breve{B}^+\Phi^T r^n, \quad \breve{B} = \Phi^T A\Phi, \quad \breve{r}^n = \breve{H}r^n, \quad \breve{H} = I - A\Phi(\Phi^T A\Phi)^+\Phi^T. \tag{7}$$

Recall that the two approaches considered, namely, (4), (5) and (6), (7) have different orthogonality properties: for the first approach, $\Phi^T A^T \widetilde{r}^n = 0$, whereas for the second one, $\Phi^T \widetilde{r}^n = 0$. It is also essential that the approaches have different optimization characteristics: relations (4), (5) provide for minimization of the functional $((r^n - A\Phi\widetilde{c}), (r^n - A\Phi\widetilde{c}))$, whereas conditions (6), (7) ensure minimization of the functional $(A^{-1}(r^n - A\Phi\widetilde{c}), (r^n - A\Phi\widetilde{c}))$ but only in the case where A is a symmetric positive definite (s.p.d.) matrix. Note that in the cases considered, the matrices $\widehat{B}$, $\breve{B} \in \mathcal{R}^{N,N}$ are actually low-rank approximations of the matrices $A^T A$ and A, respectively.

So far, we have not specified by which method the vectors u^n and r^n, to which the above-described coarse-grid correction applies, are obtained. In general, one can choose any of the Krylov type processes as the initial iterative algorithm. We consider the deflated conjugate gradient method (DCG) proposed in [5] for solving SLAEs with s.p.d. coefficient matrices, and we will apply it to unsymmetric systems as well. This algorithm is based on choosing a rectangular deflation matrix $W = (w_1 \dots w_k) \in \mathcal{R}^{N,k}$ with linearly independent columns, which form a basis of the corresponding deflation subspace, and initial vectors $u^0, r^0 = f - Au^0$, and p^0 satisfying the orthogonality relations

$$W^T r^0 = 0, \quad W^T A p^0 = 0. \tag{8}$$

Given an arbitrary vector u^{-1}, conditions (8) are fulfilled if we set $r^{-1} = f - Au^{-1}$ and

$$u^0 = u^{-1} + WB^{-1}W^T r^{-1}, \quad p^0 = r^0 - WB^{-1}W^T A r^0, \quad B = W^T A W. \tag{9}$$

Then, for $n = 0, 1, \dots$, the approximate solutions are computed by the formulas

$$\begin{aligned} u^{n+1} &= u^n + \alpha_n p^n, \quad \alpha_n = \rho_n/(p^n, Ap^n), \\ r^{n+1} &= r^n - \alpha_n A p_n, \quad \rho_n = (r^n, r^n), \\ p^{n+1} &= r^{n+1} + \beta_n p^n - WB^{-1}WAr^{n+1}, \quad \beta_n = \rho_{n+1}/\rho_n. \end{aligned} \tag{10}$$

In this case, for all n, the relations

$$W^T r^{n+1} = 0, \quad W^T A p^{n+1} = 0, \tag{11}$$

analogous to (8), hold. In order to select the deflation vectors w_k, we will use the approximation principle of coarse-grid correction from [3], see Sec. 3 below.

Now we describe the second, outer, level of acceleration of an iterative process with restarts. Let $n_0 = 0$ and let n_k, $k = 1, \dots, M$, be the numbers of the "restart" iterations, at which the approximate solutions and the corresponding residual vectors,

$$u^{n_0}, u^{n_1}, \dots, u^{n_m}, \qquad r^{n_0}, r^{n_1}, \dots, r^{n_m}, \tag{12}$$

are computed. In particular, if the length of the restart period $m = n_k - n_{k-1}$ is constant, then $n_k = km$. From these vector sequences we form the rectangular matrices

$$\begin{aligned} V_k &= (v_1 = u^{n_1} - u^{n_0} \dots v_k = u^{n_k} - u^{n_{k-1}}) \in \mathcal{R}^{N,k}, \\ W_k &= (w_1 = Av_1 \dots w_k = Av_k) \in \mathcal{R}^{N,k}. \end{aligned} \tag{13}$$

We will refine every restart approximation by using a linear combination of the previous similar iterations in the following way:

$$\widehat{u}^{n_k} = u^{n_k} + c_1 v_1 + \dots + c_k v_k = u^{n_k} + V_k \bar{c}_k. \tag{14}$$

Here, $\bar{c}_k = (c_1, \dots, c_k)^T$ is the vector of unknown coefficients. The corresponding modified residual vectors can be written in the form

$$\widehat{r}_{n_k} = f - A\widehat{u}_{n_k} = r^{n_k} - W_k \bar{c}_k. \tag{15}$$

In order to obtain the coefficients c_k by the LSM, we solve the auxiliary SLAE

$$B_k\bar{c}_k \equiv W^T A W\bar{c}_k = W^T r^{n_k} \equiv g_k, \quad B_k \in \mathcal{R}^{k,k}, \tag{16}$$

and the corrected solution is computed by formula (14).

3. Boundary-value problem and basis functions

For a convection-diffusion equation in the square domain $\Omega = [0,1] \times [0,1]$ with boundary Γ, we consider the two-dimensional boundary-value problem with the Dirichlet boundary conditions

$$-\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y} = f(x,y), \quad (x,y) \in \Omega, \quad u|_\Gamma = 1, \quad \Gamma = \overline{\Omega} \setminus \Omega. \tag{17}$$

Discretization is carried out by the finite volume method of the second order (for details, see [4]) on the square grid with $L+1$ steps along the x-axis and $M+1$ steps along the y-axis. The number of unknowns in system (1), i.e., the number of computation nodes is LM if the boundary conditions are taken into account and the boundary nodes are eliminated.

In the domain Ω, a macro-grid is constructed. Its macro-elements (subdomains) cover all the computation nodes. The macro-grid also is a square one and is specified by the numbers of macro-steps, P_x and P_y, in each of the directions. The number of subdomains in Ω is $P = P_x P_y$. The macro-grid is constructed in such a way that its grid lines interlace the grid lines of the original grid, whence every node belongs to exactly one subdomain. Denote the grid macro-coordinates in the x- and y-directions by $X_0, \ldots, X_{P_x}$ and $Y_0, \ldots, Y_{P_y}$, respectively. Every subdomain is defined by four values of the macro-coordinates, or macro-vertices, and has its own number (which corresponds to the column number of the deflation matrix W). In this case, the number of columns in the matrix W is equal to the number of basis functions defined on the macro-grid, and the number of rows in this matrix equals the number of computation nodes.

In applying coarse-grid correction, we will use basis functions of the zero, first, and second orders. Dwell on describing the functions of the three types.

The zero-order basis functions, which form the first type, are the so-called "shelves" in the subdomains. They are defined as follows. Let the coordinates (x_i, y_j) of the nth node be known (we assume that all the nodes are ordered) and let X_k, X_{k+1}, $Y_l, Y_{l,+1}$ be the macro-coordinates defining a certain subdomain with number p. Then for the matrix entry we have $W(q,p) = 1$, provided that

$$X_k < x_i < X_{k+1}, Y_l < y_j < Y_{l+1}, \tag{18}$$

where $q = q(i,j)$ is the number of the matrix row and, simultaneously, that of the corresponding node. In the matrix column that corresponds to the subdomain with number p, the number of unit entries equals that of the nodes lying inside this subdomain, and the remaining entries are zero. In each of the subdomains, only one basis function is nonvanishing, whence every row of W contains a single nonzero entry.

The second type of functions consists of basis functions of the first order, the so-called "caps". These functions are functions with finite support bilinear in subdomains. Each of them is a product of a linear function in x and a linear function in y, and it is defined on its support, which is a doubled macro-edge $[X_{k-1}, X_{k+1}]$ and $[Y_{l-1}, Y_{l+1}]$, respectively. This implies that a basis function is computed in one of the four possible ways, depending on the

macro-cell to which the node belongs:

$$\begin{aligned}
&\text{if } X_{k-1} < x_i < X_k, Y_{l-1} < y_j < Y_l, \text{ then } W(q,p) = \big(\frac{x_i - X_{k-1}}{X_k - X_{k-1}}\big)\big(\frac{y_j - Y_{l-1}}{Y_l - Y_{l-1}}\big);\\
&\text{if } X_{k-1} < x_i < X_k, Y_l < y_j < Y_{l+1}, \text{ then } W(q,p) = \big(\frac{x_i - X_{k-1}}{X_k - X_{k-1}}\big)\big(1 - \frac{y_j - Y_l}{Y_{l+1} - Y_l}\big);\\
&\text{if } X_k < x_i < X_{k+1}, Y_{l-1} < y_j < Y_l, \text{ then } W(q,p) = \big(1 - \frac{x_i - X_k}{X_{k+1} - X_k}\big)\big(\frac{y_j - Y_{l-1}}{Y_l - Y_{l-1}}\big);\\
&\text{if } X_k < x_i < X_{k+1}, Y_l < y_j < Y_{l+1}, \text{ then } W(q,p) = \big(1 - \frac{x_i - X_k}{X_{k+1} - X_k}\big)\big(1 - \frac{y_j - Y_l}{Y_{l+1} - Y_l}\big).
\end{aligned} \tag{19}$$

Each of the functions of a piecewise linear basis can be associated with one of the macro-grid nodes, so that the total number of columns in the corresponding matrix W is equal to $(P_x + 1)(P_y + 1)$. In a separate subdomain, the desired solution is approximated by a linear combination of four basis functions, whence every row of the corresponding deflation matrix W has four nonzero entries.

The third type of functions consists of the second-order basis functions, which are tensor products of one-dimensional B-splines. These functions also have finite supports, namely, the triple macro-edges $[X_{k-2}, X_{k+1}]$ and $[Y_{l-2}, Y_{l+1}]$ in each of the directions. The spline (a function of one variable) is "sewn" from three quadratic functions, computed in their own way on each single macro-edge that is a part of the support. Denote these partial functions by $b1, b2, b3$. Then, formally, one can write

$$\begin{aligned}
b1(x_0, x_1, x_2, x) &= \frac{(x - x_0)}{(x_2 - x_0)}\frac{(x - x_0)}{(x_1 - x_0)},\\
b2(x_0, x_1, x_2, x_3, x) &= \frac{(x - x_0)}{(x_2 - x_0)}\frac{(x_2 - x)}{(x_2 - x_1)} + \frac{(x_3 - x)}{(x_3 - x_1)}\frac{(x - x_1)}{(x_2 - x_1)},\\
b3(x_1, x_2, x_3, x) &= \frac{(x_3 - x)}{(x_3 - x_1)}\frac{(x_3 - x)}{(x_3 - x_2)},
\end{aligned} \tag{20}$$

where x_0, x_1, x_2, x_3 are given coordinates of the macro-nodes, and x is the coordinate of the computed node. Since we consider the direct product of the supports of one-dimensional splines, the node belongs to one of the nine macro-cells that form the support of the resulting spline. Therefore, the value of the basis spline function is computed in one of nine ways, depending on a specific macro-cell to which the grid node with coordinates (x_i, y_j) belongs:

$$\begin{aligned}
&\text{if } X_{k-2} < x_i < X_{k-1}, Y_l < y_j < Y_{l+1},\\
&\quad \text{then } W(q,p) = b1(X_{k-2}, X_{k-1}, X_k, x_i)b3(Y_{l-1}, Y_l, Y_{l+1}, y_j),\\
&\text{if } X_{k-1} < x_i < X_k, Y_l < y_j < Y_{l+1},\\
&\quad \text{then } W(q,p) = b2(X_{k-2}, X_{k-1}, X_k, X_{k+1}, x_i)b3(Y_{l-1}, Y_l, Y_{l+1}, y_j),\\
&\text{if } X_k < x_i < X_{k+1}, Y_l < y_j < Y_{l+1},\\
&\quad \text{then } W(q,p) = b3(X_{k-1}, X_k, X_{k+1}, x_i)b3(Y_{l-1}, Y_l, Y_{l+1}, y_j),\\
&\text{if } X_{k-2} < x_i < X_{k-1}, Y_{l-1} < y_j < Y_l,\\
&\quad \text{then } W(q,p) = b1(X_{k-2}, X_{k-1}, X_k, x_i)b2(Y_{l-2}, Y_{l-1}, Y_l, Y_{l+1}, y_j),\\
&\text{if } X_{k-1} < x_i < X_k, Y_{l-1} < y_j < Y_l,\\
&\quad \text{then } W(q,p) = b2(X_{k-2}, X_{k-1}, X_k, X_{k+1}, x_i)b2(Y_{l-2}, Y_{l-1}, Y_l, Y_{l+1}, y_j),\\
&\text{if } X_k < x_i < X_{k+1}, Y_{l-1} < y_j < Y_l,\\
&\quad \text{then } W(q,p) = b3(X_{k-1}, X_k, X_{k+1}, x_i)b2(Y_{l-2}, Y_{l-1}, Y_l, Y_{l+1}, y_j),
\end{aligned}$$

$$
\begin{aligned}
&\text{if } X_{k-2} < x_i < X_{k-1}, Y_{l-2} < y_j < Y_{l-1},\\
&\quad \text{then } W(q,p) = b1(X_{k-2}, X_{k-1}, X_k, x_i) b1(Y_{l-2}, Y_{l-1}, Y_l, y_j),\\
&\text{if } X_{k-1} < x_i < X_k, Y_{l-2} < y_j < Y_{l-1},\\
&\quad \text{then } W(q,p) = b2(X_{k-2}, X_{k-1}, X_k, X_{k+1}, x_i) b1(Y_{l-2}, Y_{l-1}, Y_l, y_j),\\
&\text{if } X_k < x_i < X_{k+1}, Y_{l-2} < y_j < Y_{l-1},\\
&\quad \text{then } W(q,p) = b3(X_{k-1}, X_k, X_{k+1}, x_i) b1(Y_{l-2}, Y_{l-1}, Y_l, y_j). \qquad (21)
\end{aligned}
$$

4. Numerical experiments

Below, we present results of numerical experiments on solving the boundary-value problem (17) for various numbers of nodes, values of the convective coefficients, numbers of subdomains, and parameters of restarts. The experiments were aimed at studying the rate of convergence of iterations for relatively small two-dimensional grid boundary-value problems. For this reason, they were conducted in a sequential mode. The exact solution of the system was $u = 1$. As the base iterative method for solving SLAE (1), the deflated conjugate gradient method was chosen, the stopping criterion being $\|r^n\|_2 \le 10^{-7}\|f\|_2$. Upon the termination of iterations, the error of the approximate solution $\delta = \|1 - u^n\|_\infty$ was computed. In all the computations, the initial guess was $u^0 = x^2 + y^2$. In Tables 1–5 below, the piecewise constant basis functions were used.

In every cell of the following Table 1, the number of iterations and the maximal error norm are indicated. The columns correspond to the restart values $m = 8, 16, 32, 64$, and the rows correspond to the total number of nodes $N = 16^2, 32^2, 64^2, 128^2$. In Table 1, as well as in the subsequent two Tables 2 and 3, the results correspond to application of the DCG algorithm using formulas (10) and the one-level least-squares method applied for refining the restart approximations in accordance with (7).

Table 1. Numerical results for the number of subdomains $P = 2 \times 2$ and the convective coefficients $p = q = 0$.

$N \setminus m$	8	16	32	64
16^2	55 $4.2 \cdot 10^{-7}$	48 $1.4 \cdot 10{-}7$	39 $1.3 \cdot 10^{-7}$	36 $7.1 \cdot 10^{-8}$
32^2	157 $1.1 \cdot 10^{-6}$	101 $5.9 \cdot 10^{-7}$	86 $5.2 \cdot 10^{-7}$	73 $2.2 \cdot 10^{-7}$
64^2	517 $2.7 \cdot 10^{-6}$	282 $2.0 \cdot 10^{-6}$	189 $8.8 \cdot 10^{-7}$	161 $9.2 \cdot 10^{-7}$
128^2	1798 $7.7 \cdot 10^{-6}$	965 $5.4 \cdot 10^{-6}$	514 $4.7 \cdot 10^{-6}$	344 $2.9 \cdot 10^{-6}$

As is seen from the data presented, the resulting error is acceptable and matches the stopping criterion used. This shows a satisfactory stability of the algorithms investigated. This conclusion is also confirmed by other numerical experiments.

Table 2 provides the results for the same data as in Table 1, except for the number of subdomains, which is equal to $P = 4 \times 4$, 8×8, 16×16. These three numbers correspond to the three numbers of iterations indicated (from left to right) in every cell of the table.

In the next Table 3, the results are provided for the same data as in Table 2, except for the values of the convective coefficients from the problem (17), which are nonzero, namely,

Table 2. The results of experiments for the number of subdomains $P = 4 \times 4$, 8×8, 16×16 and convective coefficients $p = q = 0$.

$N \setminus m$	8	16	32	64
16^2	27 14 1	26 14 1	26 14 1	26 14 1
32^2	60 28 14	52 26 14	50 26 14	48 26 14
64^2	172 61 27	105 50 26	94 49 26	92 48 26
128^2	563 171 60	306 102 49	193 92 49	175 91 47

$p = q = 4$. Note that in this case, the iterative process sometimes diverges for large periods of restarts (in Table 3, this is indicated by the symbol "∞"). Recall that for unsymmetric SLAEs, the "pure" DCG method does not necessarily converge, and the LSM has a stabilizing effect for relatively small m only. The difference in the number of iterations for zero and nonzero values of p and q proves to be insignificant and oscillates. Observe that if the number of subdomains is equal to that of nodes, then the process converges in one iteration because, in this case, the matrix W is square, and the method turns out to be a direct one.

Table 3. The results of experiments for the number of subdomains $P = 4 \times 4$, 8×8, 16×16 and convective coefficients $p = q = 4$.

$N \setminus m$	8	16	32	64
16	43 25 1	71 39 1	167 98 1	517 198 1
32	69 51 25	94 115 49	198 516 99	651 ∞ 259
64	150 74 51	132 119 131	188 455 840	398 ∞ ∞
128	440 158 74	289 145 125	249 238 716	356 3288 ∞

Tables 4 and 5 present data similar to those provided in Tables 2 and 3, with the difference that the two-level LSM is applied, i.e., the restart approximations are additionally corrected in accordance with formulas (13)–(16).

Table 4. The results of experiments for the number of subdomains $P = 4 \times 4$, 8×8, 16×16, convective coefficients $p = q = 0$, and the two-level LSM acceleration.

$N \setminus m$	8	16	32	64
16^2	27 14 1	26 14 1	26 14 1	26 14 1
32^2	53 27 14	49 26 14	50 26 14	48 26 14
64^2	98 55 26	97 49 26	94 49 26	92 48 26
128^2	185 98 55	177 98 48	181 92 49	177 91 47

As one can see, using the LSM for the second level, one can significantly reduce the number of iterations and ensure convergence in all the cases. The results presented also show that as the number of subdomains increases, the number of iterations decreases almost linearly. The influence of the length m of the restart period on the rate of convergence is not crucial, and it manifests itself in different ways: for symmetric SLAEs, the number of iterations generally decreases, as m grows, whereas for unsymmetric systems it increases.

Tables 6 and 7 provide similar data on the efficiency of applying piecewise-linear basis functions computed by formulas (19).

Table 5. The results for the two-level LSM acceleration for $p = q = 4$.

$N \setminus m$	8	16	32	64
16^2	38 23 1	67 37 1	135 97 1	324 195 1
32^2	71 45 22	89 85 37	169 164 97	394 450 194
64^2	139 73 45	140 109 95	178 212 163	342 393 392
128^2	281 415 72	273 152 111	275 238 208	346 463 400

Table 6. The results of computations using the piecewise-linear basis (19) and the two-level LSM acceleration for $p = q = 0$.

$N \setminus m$	8	16	32	64
16^2	24 12 1	22 12 1	21 12 1	21 12 1
32^2	45 26 12	45 27 11	42 25 11	41 25 11
64^2	81 49 27	81 48 27	83 50 26	78 47 26
128^2	153 90 49	145 93 49	150 92 51	152 92 49

Table 7. The results of computations using the piecewise-linear basis (19) and the two-level LSM acceleration for $p = q = 4$.

$N \setminus m$	8	16	32	64
16^2	27 14 1	36 17 1	54 20 1	84 20 1
32^2	47 27 12	51 30 12	72 40 12	104 68 12
64^2	88 50 27	87 52 27	97 60 32	142 81 32
128^2	160 96 51	159 96 51	166 96 52	181 113 65

Comparing the results presented in Tables 6 and 7 with those in Tables 4 and 5, we see that as the order of the basis functions increases by one, the number of iterations reduces by about 10% in the case of symmetric SLAEs and almost twice in the case of unsymmetric systems, i.e., in presence of convection.

5. Conclusion

The approximation principles proposed for constructing coarse-grid correction with two-level application of the least-squares method appear to be a promising approach to accelerating the convergence of iterative processes in Krylov subspaces, which additionally provides for the possibility of using multi-preconditioning algorithms. It is necessary to make a comment concerning parallelization of the algorithms suggested, which is related to fast and efficient computation of the matrices $\widehat{B}$ and $\breve{B}$. If we have a system of equations of order $N \approx 10^6$ or higher, the number of subdomains is relatively small ($P \approx i \cdot 10$, $i = 1, 2, \dots$), and the number of restarts is $m \approx i \cdot 10$, $i = 1, 2, \dots$, then, in order to compute the entries of the matrix B, one must multiply "long" matrices W by the matrix A repeatedly. This process can be significantly accelerated by performing this multiplication in parallel, e.g., by using efficient functions from the MKL INTEL library. The same concerns the computation of the entries of the matrix W via repeated computations of inner products of vectors, which can be done concurrently on various devices of a multiprocessor computer system.

This work was supported by the Russian Science Foundation (project No. 14-11-00485p) and the RFBR (grant No. 16-29-1522/17 ofi-m).

Translated by Y. L. Gurieva.

REFERENCES

1. Y. L. Gurieva, V. P. Il'in, and D. V. Perevozkin, "Algebro-geometric and information structures of domain decomposition methods," *Vychisl. Met. Program.*, **17**, 132–146 (2016).
2. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS Publ., New York (1996)
3. Y. L. Gurieva and V. P. Il'in, "On technologies of acceleration of parallel domain decomposition methods," *Vychisl. Met. Program.*, **16**, 146–154 (2015).
4. V. P. Il'in, "Least squares methods in Krylov subspaces," *Zap. Nauchn. Semin. POMI*, **453**, 131–147 (2016).
5. Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h, "A deflated version of the conjugate gradient algorithm," *SIAM J. Sci. Comput.*, **21**, No. 5, 1909–1926 (2000).
6. V. P. Il'in, *Methods and Technologies of Finite Element Methods* [in Russian], IVMMG Publ., Novosibirsk (2007).