DOI 10.1007/s10958-017-3460-y
Journal of Mathematical Sciences, Vol. 224, No. 6, August, 2017

LEAST SQUARES METHODS IN KRYLOV SUBSPACES

V. P. Il'in[*]

UDC 519.6

The paper considers iterative algorithms for solving large systems of linear algebraic equations with sparse nonsymmetric matrices based on solving least squares problems in Krylov subspaces and generalizing the alternating Anderson–Jacobi method. The approaches suggested are compared with the classical Krylov methods, represented by the method of semiconjugate residuals. The efficiency of parallel implementation and speedup are estimated and illustrated with numerical results obtained for a series of linear systems resulting from discretization of convection-diffusion boundary-value problems. Bibliography: 12 titles.

1. Introduction

Consider solution of a system of linear algebraic equations (SLAE)

$$Au = \Big\{ \sum_{l' \in \omega_l} a_{l,l'} u_{l'} \Big\} = f, \quad A = \{a_{l,l'}\} \in \mathcal{R}^{N,N}, \quad u = \{u_l\}, \quad f = \{f_l\} \in \mathcal{R}^N, \tag{1}$$

with a large real sparse matrix resulting from grid approximations of multidimensional boundary-value problems by finite element, finite volumes, or other methods. In general, this matrix is unsymmetric and ill-conditioned. In Eq. (1), ω_ℓ denotes the set of indices of nonzero entries in the ℓth row of the matrix A, whose number N_ℓ is assumed to be much smaller than N. The algorithms considered below can easily be extended to the case of complex SLAEs.

The main modern approaches to fast solution of the algebraic equations considered are based on preconditioned iterative methods in Krylov subspaces. Main principles of such methods are presented, for instance, in [2]. In particular, highly efficient computations with scalable parallelism on a multiprocessor computer system (MPS) with distributed and hierarchical shared memory are mostly based on applying domain decomposition methods, see [3] and the references therein.

In the recent paper [1], the authors have suggested special procedures for accelerating the convergence of the Jacobi method as an "efficient alternative" to the classical Krylov methods. In order to solve a linear system, they have used the Anderson extrapolation, which had been originally suggested in [4] for solving systems of nonlinear algebraic equations. A comparative experimental analysis presented in [1] has demonstrated a considerable superiority of the original alternating Anderson–Jacobi (AAJ) method over the popular generalized minimal residual method GMRES as concerns the solution time. The idea of the AAJ method consists of periodically (after a prescribed number of stationary iterations) using an acceleration method based on solving an auxiliary least squares problem not involving orthogonalization of the direction vectors, which is typical for Krylov variational type methods.

The present paper aims at generalizing and studying Anderson's extrapolation algorithm described in [1]. We suggest a nonstationary iterative algorithm and compare it with a typical Krylov approach for solving unsymmetric SLAEs, which is the semiconjugate residual method [5] with periodic restarts. In this context, parallel implementation on multiprocessor computer systems is considered.

[*] Institute of Computational Mathematics and Mathematical Geophysics, SB RAS and Novosibirsk State University, Novosibirsk, Russia, e-mail: ilin@sscc.ru.

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 453, 2016, pp. 131–147. Original article submitted November 11, 2016.

The present paper is organized as follows. In Sec. 2, we present matrix structures for some algorithms of Anderson type, including the naturally generalized variant using Chebyshev acceleration. Section 3 is devoted to analyzing the efficiency of parallel versions of the iterative algorithms considered in comparison with the classical variational method of semicojugate residuals in Krylov subspaces. Section 4 discusses the results of numerical experiments obtained for the algorithms suggested on a series of test SLAEs, resulting from grid approximation of two-dimensional boundary-value problems for the convection-diffusion equation.

2. Extrapolation algorithms in Krylov subspaces

Let the coefficient matrix of system (1) be written in block form as $A = \{A_{k,k'};\ k,k' = 1,\dots,P\}$, where $A = D - C$ and $D = \text{block-diag}\{A_{k,k}\}$ is a nonsingular block diagonal matrix with LU factorization $D = LU$. Then the original system reduces, by left and right preconditionings, to the form

$$\begin{aligned} &L^{-1}(D-C)U^{-1}U\,u = L^{-1}f = \widetilde{A}\widetilde{u} = (I-T)\widetilde{u} = \widetilde{f}, \\ &\widetilde{A} = L^{-1}AU^{-1} = I - T, \quad \widetilde{u} = U\,u, \quad \widetilde{f} = L^{-1}f, \quad T = L^{-1}CU^{-1}. \end{aligned} \tag{2}$$

Here, if the original matrix A is symmetric and positive definite, then both $\widetilde{A}$ and T also are symmetric and positive definite. In this case, by $0 < \lambda_1 < \lambda_N$ we denote the extreme eigenvalues of the preconditioned matrix $\widetilde{A}$. Note that in (2) one can also include one-sided preconditionings, setting $L = D$ or $U = D$.

The linear system (2) can be solved by the following stationary Richardson method (called the weighted Jacobi (WJ) method in [1]):

$$\widetilde{u}^{n+1} = \widetilde{u}^n + \omega(\widetilde{f} - \widetilde{A}\widetilde{u}^n) = \omega(T\widetilde{u}^n + \widetilde{f}) + (1-\omega)\widetilde{u}^n = T_\omega\widetilde{u}^n + \omega\widetilde{f}. \tag{3}$$

If the spectrum $\lambda(T_\omega) = 1 - \omega\lambda(\widetilde{A})$ of the iteration matrix $T_\omega = \omega T + (1-\omega)I$ is real, then its spectral radius $\rho = \max|\lambda(T_\omega)|$ is minimal for $\omega = 2/(\lambda_1 + \lambda_N)$. In the general case, the Richardson iterations converge, provided that $\omega < 2/\|A\|$.

In what follows, for shortness, the symbol "$\sim$" is omitted. Following [1], given an arbitrary initial guess u^0, we consider the following Anderson–Jacobi iterative method, based on supplementing relations (3) with a correction procedure:

$$\begin{aligned} &u^{n+1} = \check{u}^n + \omega\check{r}^n, \quad \check{r}^n = f - A\check{u}^n, \quad \check{u}^n = u^n + W_n c^n, \quad n = 0,1,\dots, \\ &\check{u}^0 = u^0. \end{aligned} \tag{4}$$

Here, $c^n = (c_1^n,\dots,c_{m_n}^n) \in \mathcal{R}^{m_n}$, $m_n \geq 1$, is a certain vector of iterative parameters, and $W_n = (w_1^{(n)},\dots w_{m_n}^{(n)}) \in \mathcal{R}^{N,m_n}$ is the rectangular matrix whose columns are defined as follows:

$$w_s^{(n)} = u^{n-s+1} - u^{n-s}, \quad s = 1,\dots,m_n, \quad 1 \leq m_n \ll N. \tag{5}$$

The corresponding formulas for recomputing the residual vectors is given by

$$\check{r}^n = r^n - R_n c^n, \quad R_n = AW_n \in \mathcal{R}^{N,m_n}. \tag{6}$$

If we want to minimize the residual norm and set

$$c^n = \arg\min_{c_k^n} \|\check{r}^n\|^2, \quad \|\check{r}^n\|^2 = (\check{r}^n, \check{r}^n),$$

then from the associated least squares problem [6] the vector of iterative parameters is found as the solution of the linear equations

$$(R_n^T R_n)c^n = R_n^T r^n. \tag{7}$$

If the matrix R_n is of full rank, then, after substituting the solution $c^n = (R_n^T R_n)^{-1} R_n^T r^n$ into (6) and (4), the iteration formula can be written as

$$u^{n+1} = u^n + B_n^{-1} r^n, \tag{8}$$

where B_n^{-1} formally represents the preconditioning matrix

$$B_n^{-1} = [\omega I + (W_n - \omega R_n)](R_n^T R_n)^{-1} R_n^T. \tag{9}$$

The vector $\check{u}^n$ (4) can be represented as

$$\check{u}^n = u^n + c_1^n(u^n - u^{n-1}) + \cdots + c_{m_n}^n(u^{n-m_n+1} - u^{n-m_n}),$$

and this actually is a procedure for extrapolating the current approximate solution u^n using m_n previous solutions $u^{n-1}, \dots, u^{n-m_n}$, which explains the name of the method. The values $m_n \leq n$ may vary from iteration to iteration.

Observe that in the algorithm considered, the matrices W_n, R_n, and B_n must be recomputed at every iteration. Moreover, the matrix R_n is not necessarily of full rank, in which case the positive semidefinite matrix $Q_n = R_n^T R_n \in \mathcal{R}^{m_n, m_n}$ is ill-conditioned or singular. In the latter case, a generalized inverse matrix should be used. In [1], in particular, the Moore–Penrose inverse is used.

In order to simplify the realization of the above algorithm, the authors of [1] suggest the alternating Anderson–Jacobi (AAJ) method,

$$u^{n+1} = \begin{cases} u^n + \omega r^n, & [a_n]_f \neq 0, \\ u^n + B_n^{-1} r^n, & [a_n]_f = 0, \end{cases} \tag{10}$$

where $[a_n]_f$ denotes the integral part of $a_n = (n+1)/M$ and M is a certain integer. Here, the extrapolation procedure is applied periodically, whereas the remaining steps use the stationary iterative process. In the algorithm (10), the residual vectors satisfy the relations

$$r^{n+1} = \begin{cases} r^n + \omega A r^n, & [a_n]_f \neq 0, \\ r^n + A B_n^{-1} r^n, & [a_n]_f = 0. \end{cases} \tag{11}$$

Obviously, the AAJ method is a dynamically preconditioned iterative process in the “reduced” Krylov subspaces

$$\mathcal{K}_{M,m}(r^0, A) = \text{span}\ (A^{M-m} r^0, \dots, A^M r^0), \tag{12}$$

where it is assumed that $1 \leq m_n = m \leq M$. In the case where $M = m = n$, we have the ordinary “complete” Krylov subspace

$$\mathcal{K}_n(r^0, A) = \text{span}\ (r^0, A r^0, \dots, A^n r^0).$$

From (11) we obtain the relation

$$r^{M+1} = \check{r}^M - A\check{r}^M,$$

where the vector $\check{r}^M$ is defined by (6) for $n = M$. This implies that the optimal coefficient vector $\bar{\alpha}^M$, which is characterized by relation (7), has the following orthogonality property:

$$W_M^T r^{M+1} = 0. \tag{13}$$

The approach considered can be generalized as follows. Consider an iterative process

$$u^{n+1} = u^n + \omega_n (f - A u^n), \tag{14}$$

where ω_n are some iteration parameters, which can be chosen, for instance, using the roots of the Chebyshev polynomials [7]. Let, for a certain $n \geq m > 1$, we have an approximate solution, which can be refined using a linear combination of the previous solutions as in (4):

$$u = u^n + W_n c^n + \delta_n, \quad c^n \in \mathcal{R}^m, \quad W_n \in \mathcal{R}^{N,m}. \tag{15}$$

Here, c^n is an unknown vector; the rectangular matrix W_n is determined in accordance with (4) and (5), and δ_n is the approximation error. Upon substituting (15) into the original linear system, we obtain

$$A(u^n + W_n c^n + \delta_n) = f.$$

If, in this relation, we ignore the error δ_n, then for the coefficient vector we have the equation

$$R_n c^n = r^n, \quad R_n = AW_n \in \mathcal{R}^{N,m}. \tag{16}$$

When (16) is solved, the new approximate solution is computed by the formula

$$u^{n+1} = u^n + W_n c^n \tag{17}$$

rather than by (14).

The system of algebraic equations (16) is overdetermined, and the normal solution of the resulting least squares problem can be found in different ways, see [6]. If to (16) we apply the left Gaussian transformation, i.e., multiply both sides by the transposed matrix R_n^T, then we obtain a compatible system of the form (7), with a symmetric positive-semidefinite or an ill-conditioned symmetric positive-definite matrix $Q_n = R_n^T R_n$. In this case, for the corrected, or extrapolated approximate solution we have, instead of (17), the formula

$$u^{n+1} = u^n + \bar{Q}_n A^T r^n, \quad \bar{Q}_n = W_n (W_n^T A^T A W_n)^{-1} W_n^T \in \mathcal{R}^{N,N}. \tag{18}$$

Note that in the modern terminology, the matrix $\bar{Q}_n$ is a small-rank approximation (see [3] and the references therein) of the inverse matrix $(A^T A)^{-1}$ (or of the generalized inverse of $A^T A$ if the latter matrix is singular).

More numerically stable methods for solving (17) are based on applying the singular value decomposition or QR-factorization directly to (16) because the matrix Q_n obviously has an essentially larger condition number than R_n.

Note that instead of using (14), one can apply the following three-terms algorithm of Chebyshev acceleration:

$$\begin{aligned} u^1 &= u^0 + \tau\, r^n, \quad \tau = 2/(\lambda_1 + \lambda_N), \\ u^{n+1} &= u^n + \tau_n \tau\, r^n + (\tau_n - 1)(u^n - u^{n-1}), \quad \tau_0 = 2, \\ \tau_n &= 4(4 - \tau_{n-1}\gamma)^{-1}, \quad \gamma = (1-c)/(1+c), \quad c = \lambda_1/\lambda_N. \end{aligned} \tag{19}$$

Here, λ_1 and λ_N are the smallest and largest eigenvalues of the symmetric matrix A.

We emphasize that formulas (19) ensure that the current approximate solution is optimal (in the spectral sense) for any value of n. In the case where A is a nonsymmetric matrix, one can apply generalizations of (19), provided that the complex eigenvalues of A lie in an ellipse with known geometric parameters, see [2, 7].

Based on the Chebyshev acceleration method, we consider two iterative periodic processes that consist of M steps performed in accordance with (19) and a subsequent correction of the approximate solution, using one of the following algorithms for solving the least squares problem:

- the first variant, LSM-1, of the least squares method is based on solving the auxiliary linear algebraic system (7), obtained by the preliminary Gaussian transformation;
- the second variant, LSM-2, differs in that the coefficient vector c^n of the correction is computed by solving the overdetermined system (16) itself using the singular value decomposition algorithm SVD (the same algorithm is used for computing the vector c^n in LSM-1).

Upon correction, or extrapolation by formulas of the type (17), the subsequent residual vector is computed from the original equation. In this manner, the cycles consisting of the Chebyshev acceleration and correction are repeated until an approximate solution is computed with a desired accuracy. Observe that in all the above-considered methods, the residual vectors lie in Krylov subspaces of the form (12).

Note that from the theoretic viewpoint, the variants LSM-1 and LSM-2 coincide because, in exact arithmetic, by solving Eqs. (7) and (16) one obtains one and the same vector

$$c^n = (P_n^{(m)})^T S_n^{-1} (P_n^{(N)})^T r^n, \tag{20}$$

where $P_n^{(m)} \in \mathcal{R}^{m,m}$ and $P_n^{(N)} \in \mathcal{R}^{N,N}$ are orthogonal matrices, whereas $S_n^{-1} = \text{diag}\,\{s_k^{-1}; i = 1, \dots, m < N\} \in \mathcal{R}^{N,m}$ is a diagonal matrix whose nonzero entries $s_{i,j}^{-1} = s_i^{-1}\delta_{i,j}$ are the reciprocals of the singular values s_i of the matrix R_n, which follows from the singular value decompositions of the matrices

$$R_n = P_n^{(N)} S_n P_n^{(m)}, \quad Q_n = (P_n^{(m)})^T S_n^T S_n P_n^{(m)}.$$

We conclude this section with the following two remarks. First, construction of extrapolation processes of the form (8), (10) with the matrices B_n actually means polynomial preconditioning, see, e.g., [8]. Second, an approach similar to the one considered above was applied by Montgomery in [9] in solving special systems of linear algebraic equations over a finite field and was referred to as the block Lanczos method.

3. PARALLEL REALIZATIONS OF LEAST SQUARES METHODS AND THE METHOD OF SEMICONJUGATE RESIDUALS IN KRYLOV SUBSPACES

In this section, we consider acceleration in parallel versions of the least squares method and the algorithm of semiconjugate residuals (SCR, [5]) with no preconditioning, which is equivalent, with respect to the convergence rate, to the popular algorithm of generalized minimal residuals (GMRES, [12]) but is described by the following simpler formulas:

$$\begin{aligned} u^{n+1} &= u^n + \alpha_n p^n, \quad \alpha_n = \sigma_n/\rho_n, \quad p^0 = r^0 = f - Au^0, \\ r^{n+1} &= r^n - \alpha_n A p^n, \quad \rho_n = (Ap^n, Ap^n), \quad \sigma_n = (Ar^n, r^n). \end{aligned} \tag{21}$$

This iterative process minimizes the norm $\|r^{n+1}\|$ in the Krylov subspace $\mathcal{K}_n(r_0, A)$, and the resulting residual vectors are semiconjugate, i.e.,

$$(Ar^n, r^k) = \begin{cases} 0, & k < n, \\ \sigma^n, & k = n, \end{cases}$$

provided that the direction vectors p^n are orthogonal in the sense that

$$(Ap^n, Ap^k) = \rho_n \delta_{n,k}, \tag{22}$$

where $\delta_{n,k}$ is the Kronecker symbol. Relations (22) are satisfied if the vectors p^n are computed from the recursive relations

$$p^{n+1} = r^{n+1} - \sum_{k=0}^{n} \beta_{n,k} p^k, \quad \beta_{n,k} = (Ap^k, Ar^{n+1})/\rho_n. \tag{23}$$

Observe that at every iteration of the SCR method only one matrix-vector multiplication is required because from (23) it follows that we also have the relation

$$Ap^{n+1} = Ar^{n+1} - \sum_{k=0}^{n} \beta_{n,k} Ap^k.$$

If the matrix A is symmetric, then the residual vectors become A-orthogonal, and the recursions (23) are short because

$$(Ar^n, r^k) = \sigma_n \delta_{n,k}, \quad \beta_{n,k} = \beta_{n,n}\delta_{n,k}. \tag{24}$$

Thus, we come to the conjugate residual algorithm (CR, see [2]).

In the general case, in computing the vectors u^n and r^n using (20)–(23), it is necessary to store all the vectors $p^n, p^{n-1}, \dots, p^0$. In practice, the SCR methods are realized with periodic restarts every M iterations. This means that the residual vector is computed from the original equation,

$$r^{Ml} = f - Au^{Ml}, \quad l = 0, 1, \dots, \tag{25}$$

rather than using (21), and the subsequent approximations are computed "from the beginning", i.e., for $n > M$ one should change n for $n - Ml$ in the formulas. Here, it is necessary to store only the last $M + 1$ vectors $p^n, p^{n-1}, \dots, p^{n-M}$.

Now we compare parallel realizations of a cycle of M iterations in the methods LSM and SCR. This will suffice for a qualitative comparison of the performances of the algorithms in question because they minimize the same functional in the same Krylov subspace and, consequently, are theoretically equivalent with respect to the convergence rate.

Concerning the methods considered, we assume that they are applied to a block system of linear equations of the form (2), and the block rows $A_k = \{A_{k,l}, l = 1, \dots, P\} \in \mathcal{R}^{N_k,N}$, $N_k \cong N/P$, $N_1 + \cdots + N_p = N$, of the coefficient matrix A are distributed in the memory of the corresponding MPI processes, used for the first level of parallelizing the algorithms, as is done in the domain decomposition methods (where every block row corresponds to a subdomain, see [10]). Note that in fact to different MPI processes different computer processors correspond (though this is not formally necessary).

In the SCR method, the direction vectors $p^n, p^{n-1}, \dots, p^{n-M}$ and also the current vectors u^n and r^n are partitioned into subvectors of lengths N_k, each of which is stored in the corresponding kth MPI process. As the iterations proceed, data exchanges among processes are needed, and their volumes should be minimized. When arithmetic operations are performed in the kth MPI process using a multicore processor, "inner" parallelization (of the second level) can be effected based on multithread computations (here, we omit the details). A similar distributed data structure is formed in the least squares methods, in which case the block partition is used for the vectors $w_s^{(M)}$, $s = 1, \dots, M$. We assume that in all the algorithms the standard double-precision computer arithmetic is used.

For a comparative analysis of the performances of the methods considered, we estimate the time T_P of performing a cycle of M iterations on P MPI processes based on the following simplest model of the computation process:

$$T_P = T_P^a + T_P^c \approx \tau_a V_a + (\tau_0 + \tau_c V_c) N_c. \tag{26}$$

Here, T_P^a and T_P^c are the times for performing arithmetic and communication operations, respectively; τ_a is the average time of a single arithmetic operation, and N_a is the number of such operations (for one processor); N_c is the total number of data transmittings; τ_0 is the delay (tuning) time of a single transaction; τ_c is the average time of transmitting a real number, and V_c is the average volume of one package of data transmitted. Note that in view of the relations $\tau_0 \gg \tau_c \gg \tau_a$, it is natural to attempt to minimize not only the total volume of information to be transmitted but also the number of exchanges. This is important not only from the viewpoint of the time of data transmissions but also in view of high energy costs of communication operations.

Comparing formulas (19) for the Chebyshev acceleration (we will consider their application in LSM-1 only) with relations (21), (23) for the semiconjugate residual method, we arrive at the following conclusions. The method SCR requires, at every iteration, one matrix-vector multiplication and $2(M+1)$ vector operations, as well as computation of $M+2$ inner products of vectors. In the case of a symmetric matrix A, where we come to the CR method, the above-indicated number of vector operations reduces to 4 and 2, respectively. It is of importance that this package of operations, which are carried out at each of the M iterations, can be performed only sequentially.

On the other hand, implementation of every step of Chebyshev acceleration (19) requires one matrix-vector multiplication and five vector operations, but no inner product is computed. However, in implementing LSM-1, it is necessary to compute the M^2 entries of the matrix $Q_n = \{q_{k,\ell} = (Aw_k, Aw_\ell)\}$, but all of them can be computed simultaneously when M iterations are completed. In addition to an M-fold acceleration of the computations, this simultaneously reduces, proportionally, the number of exchanges, which must be performed in computing inner products of vectors distributed over different processors because the partial sums in every CPU necessary for transmission can be assembled in one information buffer.

Thus, without going into technical details, we may claim that the Chebyshev acceleration with correction of approximate solutions by the least squares methods has a considerable advantage over the classical Krylov iterative methods as concerns their parallel implementation on multiprocessor computer systems. Note that the Chebyshev acceleration formulas are not optimal in the case where the coefficient matrix A of (1) has a complex spectrum. But this is of little importance because in fact they are only used in order to construct a basis in the Krylov subspace efficiently.

4. Discussion of numerical results

Consider the Dirichlet problem for the convection-diffusion equation

$$\begin{gathered} -\frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} + p\frac{\partial u}{\partial x} + q\frac{\partial u}{\partial y} = f(x,y), \quad (x,y) \in \Omega, \\ u|_\Gamma = g(x,y), \end{gathered} \tag{27}$$

in a square computational domain $\Omega = (0,1)^2$ with boundary Γ and the convection coefficients p, q, which are assumed to be constant for simplicity.

This boundary-value problem is approximated on a square grid with step size $h = 1/(L+1)$ and the total number of interior nodes $N = L^2$,

$$x_i = ih, \quad y_j = jh, \quad i, j = 0, 1, \ldots, L+1,$$

using the five-point finite-volume monotone approximations of exponential type [11]

$$(Au)_l = a_{l,l}u_l + a_{l,l-1}u_{l-1} + a_{l,l+1}u_{l+1} + a_{l,l-L}u_{l-L} + a_{l,l+L}u_{l+L} = f_l, \tag{28}$$

having the second order of accuracy. Here, ℓ is the "global" number of a grid node in the natural node ordering, $\ell = i + (j-1)L$.

Generally speaking, formulas for the coefficients in Eqs. (28) may be different, and we use the following ones:

$$\begin{gathered} a_{l,l\pm1} = e^{\pm ph/2}/h, \quad a_{l,l\pm L} = e^{\pm qh/2}/h, \\ a_{l,l} = a_{l,l-1} + a_{l,l-L} + a_{l,l+1} + a_{l,l+L}. \end{gathered} \tag{29}$$

Equations (28) are written for the interior nodes of the grid, but for the near-boundary nodes with the subscripts $i = 1, L$ or $j = 1, L$ the values of the solution on the boundary should be substituted into the system of equations and moved to the right-hand side; here, the corresponding coefficients of the left-hand side can be formally set to zero. In our experiments, we have actually solved the normalized equations, which are obtained by the following transformations with the diagonal matrix $D = \text{diag}\,\{a_{\ell,\ell}\}$:

$$\begin{aligned} &D^{-1/2}AD^{-1/2}D^{1/2}u = D^{-1/2}f, \\ &\bar{A}\bar{u} = \bar{f}, \quad \bar{A} = D^{-1/2}AD^{-1/2}, \quad \bar{u} = D^{1/2}u, \quad \bar{f} = D^{-1/2}f. \end{aligned} \tag{30}$$

The numerical experiments have been carried out using the standard double-precision arithmetic for computing the values of the functions $f(x,y) = 0$ and $g(x,y) = 1$ corresponding to the exact solution $u(x,y) = 1$ of the problem (27). Since the convergence rate of iterations depends on the initial error $u - u^0$, its influence has been analyzed by comparing the results for the initial guesses $u^0 = 0$ and $u^0 = P_2(x,y) = x^2 + y^2$. The stopping criterion used has been of the from $(r^n, r^n) \le \varepsilon^2(f,f)$, with $\varepsilon = 10^{-7}$. The computations have been carried out on grids with $N = 7^2$, 15^2, 31^2, 63^2, and 127^2 nodes and for $m = 8$, 16, 32, 64, and 128.

In the tables below, we present the results obtained in solving the problem (27) with the convection coefficients $p = q = 0$ and $p = q = 4$ and for different initial guesses. The algorithms applied differ in the method of forming the auxiliary linear system for finding the coefficient vector of correction c^n (to be exact, the systems obtained in LSM-1 and LSM-2 have been solved using the SVD program from LAPACK, included into the program library MKL Intel [12]).

In Tables 1 and 2, we present the results of numerical experiments for LSM-1 and LSM-2 for the zero convection coefficients and the initial guesses $u^0 = x^2 + y^2$. In every box of the tables, the upper number stands for the total number of iterations, and the lower one stands for the maximal error of the solution computed. The columns with $m = \infty$ correspond to the cases where iterations have been repeated until the stopping criterion was satisfied using the Chebyshev acceleration only, without correction of u^n using a least squares method.

Table 1. Numerical results for LSM-1, $p = q = 0$, $u^0 = P_2(x,y)$.

$N \setminus m$	8	16	32	64	128	∞
7^2	37 $1.8 \cdot 10^{-7}$	27 $4.2 \cdot 10^{-8}$	32 $1.1 \cdot 10^{-15}$	41 $1.2 \cdot 10^{-7}$	41 $1.2 \cdot 10^{-7}$	41 $1.2 \cdot 10^{-7}$
15^2	98 $9.8 \cdot 10^{-7}$	75 $6.8 \cdot 10^{-7}$	56 $1.3 \cdot 10^{-7}$	64 $4.7 \cdot 10^{-10}$	82 $2.0 \cdot 10^{-7}$	82 $2.0 \cdot 10^{-7}$
31^2	313 $3.5 \cdot 10^{-6}$	198 $2.9 \cdot 10^{-6}$	147 $1.7 \cdot 10^{-6}$	112 $3.4 \cdot 10^{-7}$	128 $1.8 \cdot 10^{-8}$	163 $3.0 \cdot 10^{-7}$
63^2	1083 $1.0 \cdot 10^{-5}$	625 $1.0 \cdot 10^{-5}$	389 $8.9 \cdot 10^{-6}$	291 $4.6 \cdot 10^{-6}$	206 $9.6 \cdot 10^{-7}$	327 $3.1 \cdot 10^{-7}$
127^2	3859 $2.9 \cdot 10^{-5}$	2118 $2.8 \cdot 10^{-5}$	1184 $2.8 \cdot 10^{-5}$	746 $2.8 \cdot 10^{-5}$	537 $2.0 \cdot 10^{-5}$	653 $3.5 \cdot 10^{-7}$

Table 2. Numerical results for LSM-2, $p = q = 0$, $u^0 = P_2(x,y)$.

$N \setminus m$	8	16	32	64	128	∞
7^2	37 $1.8 \cdot 10^{-7}$	32 $1.3 \cdot 10^{-7}$	39 $3.3 \cdot 10^{-8}$	41 $1.2 \cdot 10^{-7}$	41 $1.2 \cdot 10^{-7}$	41 $1.3 \cdot 10^{-7}$
15^2	98 $9.8 \cdot 10^{-7}$	75 $6.8 \cdot 10^{-7}$	68 $4.9 \cdot 10^{-7}$	82 $2.0 \cdot 10^{-7}$	82 $2.0 \cdot 10^{-7}$	82 $2.0 \cdot 10^{-7}$
31^2	313 $3.5 \cdot 10^{-6}$	198 $2.9 \cdot 10^{-6}$	146 $1.7 \cdot 10^{-6}$	160 $8.0 \cdot 10^{-8}$	162 $1.3 \cdot 10^{-7}$	163 $3.0 \cdot 10^{-7}$
63^2	1083 $1.0 \cdot 10^{-5}$	624 $1.0 \cdot 10^{-5}$	389 $8.9 \cdot 10^{-6}$	262 $3.0 \cdot 10^{-6}$	298 $7.4 \cdot 10^{-7}$	327 $3.1 \cdot 10^{-7}$
127^2	3859 $2.9 \cdot 10^{-5}$	2117 $2.8 \cdot 10^{-5}$	1183 $2.8 \cdot 10^{-5}$	556 $2.0 \cdot 10^{-5}$	537 $2.0 \cdot 10^{-5}$	653 $3.5 \cdot 10^{-7}$

Tables 3 and 4 present similar results for the convection coefficients $p = q = 4$.
Tables 5 and 6 provide the numerical results for the initial guess $u^0 = 0$.

Table 3. Numerical results for LSM-1, $p = q = 4$, $u^0 = P_2(x, y)$.

$N \setminus m$	8	16	32	64	128	∞
7	30 $1.3 \cdot 10^{-7}$	36 $1.8 \cdot 10^{-9}$	31 $2.5 \cdot 10^{-13}$	32 $8.2 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$
15	73 $9.3 \cdot 10^{-7}$	68 $6.1 \cdot 10^{-7}$	63 $1.1 \cdot 10^{-8}$	64 $1.0 \cdot 10^{-8}$	91 $1.6 \cdot 10^{-7}$	91 $1.6 \cdot 10^{-7}$
31	236 $2.7 \cdot 10^{-6}$	151 $1.5 \cdot 10^{-6}$	125 $1.1 \cdot 10^{-6}$	127 $7.8 \cdot 10^{-9}$	128 $1.4 \cdot 10^{-7}$	184 $2.1 \cdot 10^{-7}$
63	592 $7.9 \cdot 10^{-6}$	472 $8.0 \cdot 10^{-6}$	302 $5.4 \cdot 10^{-6}$	253 $6.7 \cdot 10^{-6}$	247 $2.8 \cdot 10^{-6}$	363 $1.7 \cdot 10^{-7}$
127	2612 $2.3 \cdot 10^{-5}$	1348 $2.3 \cdot 10^{-5}$	900 $1.9 \cdot 10^{-5}$	569 $1.8 \cdot 10^{-5}$	509 $3.7 \cdot 10^{-6}$	719 $1.6 \cdot 10^{-7}$

Table 4. Numerical results for LSM-2, $p = q = 4$, $u^0 = P_2(x, y)$.

$N \setminus m$	8	16	32	64	128	∞
7	36 $1.3 \cdot 10^{-7}$	31 $1.4 \cdot 10^{-7}$	46 $7.5 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$
15	73 $9.3 \cdot 10^{-7}$	68 $6.1 \cdot 10^{-7}$	63 $2.3 \cdot 10^{-7}$	94 $1.0 \cdot 10^{-7}$	91 $1.6 \cdot 10^{-7}$	91 $1.6 \cdot 10^{-7}$
31	236 $2.7 \cdot 10^{-6}$	151 $1.5 \cdot 10^{-6}$	125 $9.7 \cdot 10^{-7}$	163 $7.8 \cdot 10^{-7}$	128 $1.2 \cdot 10^{-13}$	184 $2.1 \cdot 10^{-7}$
63	592 $7.9 \cdot 10^{-6}$	472 $8.0 \cdot 10^{-6}$	302 $5.3 \cdot 10^{-6}$	253 $6.7 \cdot 10^{-6}$	247 $2.8 \cdot 10^{-6}$	363 $1.7 \cdot 10^{-7}$
127	2612 $2.3 \cdot 10^{-5}$	1349 $2.3 \cdot 10^{-5}$	900 $1.9 \cdot 10^{-5}$	569 $1.8 \cdot 10^{-5}$	509 $3.7 \cdot 10^{-6}$	730 $3.0 \cdot 10^{-7}$

Table 5. Numerical results for LSM-1, $p = q = 0$, $u^0 = 0$.

$N \setminus m$	8	16	32	64	128	∞
7^2	22 $2.3 \cdot 10^{-8}$	16 $3.3 \cdot 10^{-16}$	32 $6.6 \cdot 10^{-16}$	41 $1.7 \cdot 10^{-7}$	41 $1.7 \cdot 10^{-7}$	41 $1.7 \cdot 10^{-7}$
15^2	65 $1.2 \cdot 10^{-6}$	59 $3.6 \cdot 10^{-7}$	32 $2.0 \cdot 10^{-9}$	64 $1.3 \cdot 10^{-15}$	83 $2.2 \cdot 10^{-7}$	83 $2.2 \cdot 10^{-7}$
31^2	318 $3.5 \cdot 10^{-6}$	174 $3.4 \cdot 10^{-6}$	109 $1.3 \cdot 10^{-6}$	64 $5.7 \cdot 10^{-10}$	128 $1.3 \cdot 10^{-9}$	167 $3.3 \cdot 10^{-7}$
63^2	1116 $1.0 \cdot 10^{-5}$	635 $1.0 \cdot 10^{-5}$	384 $9.1 \cdot 10^{-6}$	215 $3.0 \cdot 10^{-6}$	128 $1.5 \cdot 10^{-7}$	335 $3.9 \cdot 10^{-7}$
127^2	3987 $2.9 \cdot 10^{-5}$	2190 $2.8 \cdot 10^{-5}$	1217 $2.7 \cdot 10^{-5}$	764 $2.7 \cdot 10^{-5}$	382 $9.9 \cdot 10^{-6}$	670 $3.9 \cdot 10^{-7}$

Based on the numerical results presented, we may draw the following conclusions:

- Application of periodic correction of Chebyshev acceleration using least squares methods provides for a considerable acceleration of the iterative process; moreover, for every grid, or dimension of the linear system, there is an optimal value of the restart parameter m.
- The variants LSM-1 and LSM-2 considered are acceptably stable for different step sizes of the grid and different initial guesses. At some iteration cycles, the matrices R_n do not have full rank, but this has practically no influence on the number of iterations.

Table 6. Numerical results for LSM-2, $p = q = 4$, $u^0 = 0$.

$N \setminus m$	8	16	32	64	128	∞
7^2	37 $5.6 \cdot 10^{-8}$	32 $5.3 \cdot 10^{-8}$	46 $6.4 \cdot 10^{-8}$	45 $7.0 \cdot 10^{-8}$	45 $7.0 \cdot 10^{-8}$	45 $7.1 \cdot 10^{-8}$
15^2	77 $2.5 \cdot 10^{-7}$	76 $2.1 \cdot 10^{-7}$	83 $2.6 \cdot 10^{-7}$	93 $1.6 \cdot 10^{-7}$	91 $1.4 \cdot 10^{-7}$	91 $1.4 \cdot 10^{-7}$
31^2	216 $2.5 \cdot 10^{-6}$	151 $1.6 \cdot 10^{-6}$	153 $1.7 \cdot 10^{-6}$	162 $8.4 \cdot 10^{-7}$	187 $3.0 \cdot 10^{-7}$	184 $2.0 \cdot 10^{-7}$
63^2	507 $7.4 \cdot 10^{-6}$	468 $7.9 \cdot 10^{-6}$	304 $5.2 \cdot 10^{-6}$	253 $2.2 \cdot 10^{-6}$	341 $1.8 \cdot 10^{-6}$	363 $1.6 \cdot 10^{-7}$
127^2	2416 $2.3 \cdot 10^{-5}$	1071 $2.3 \cdot 10^{-5}$	914 $2.3 \cdot 10^{-5}$	569 $1.1 \cdot 10^{-5}$	509 $4.2 \cdot 10^{-6}$	721 $1.7 \cdot 10^{-7}$

- Iterative processes with LSM-1 and LSM-2 provide for an approximately the same convergence rate for the considered symmetric and nonsymmetric linear algebraic systems with various values of the convection coefficients p and q.
- The supplementary numerical experiments, whose results are not presented here for shortness, demonstrate that the Chebyshev acceleration is sufficiently numerically stable with respect to perturbations. For instance, if, in formulas (19), the value of λ_1 is increased several times or even if one sets $\tau_n = 1$, i.e., the stationary Richardson method is used, then the number of iterations somewhat increases but not much.

We conclude the paper with the following remark. Application of least squares methods, considered in [1], should be regarded not as an “efficient alternative” to Krylov iterative methods but as an essential enrichment of the class of algorithms considered, which is especially promising from the standpoint of parallel implementation on multiprocessor computer systems. As to issues concerning the performance and efficiency of these methods, including domain decomposition techniques and various preconditioning methods, further studies are needed.

This work was supported by the Russian Science Foundation (project No. 14-11-00485) and the Russian Foundation for Basic Research (project No. 16-29-15122).

Translated by L. Yu. Kolotilina.

REFERENCES

1. P. P. Pratara, P. Suryanarayana, and J. E. Pask, “Anderson acceleration of the Jacobi iterative method. An efficient alternative to Krylov methods for large, sparse linear systems,” *J. Comput. Phys.*, **306**, 43–54 (2016).
2. Y. Saad. *Iterative Methods for Sparse Linear Systems*, PWS Publ, New York (2002).
3. V. P. Il’in, “Problems of parallel solution of large systems of linear algebraic equations,” *Zap. Nauchn. Semin. POMI*, **439**, 112–127 (2015).
4. D. G. Anderson, “Iterative procedures for nonlinear integral equations,” *J. Assoc. Comput. Mach.*, **12**, 547–560 (1965).
5. V. P. Il’in, “Methods of semiconjugate directions,” *Russ. J. Numer. Anal. Math. Model.*, **23**, No. 4, 369–387 (2008).
6. C. L. Lawson and R. Z Hanson, *Solving Least Squares Problems* [Russian translation], Nauka, Moscow (1986).
7. A. A. Samarskii and E. S. Nikolaev, *Methods for Solving Grid Equations* [in Russian], Nauka, Moscow (1978).

8. D. P. O'Leary, "Yet another polynomial preconditioner for the conjugate gradient algorithm," *Linear Algebra Appl.*, **154**, 377–388 (1991).
9. P. L. Montgomery, "A block Lonczos algorithm for finding dependences over GF(2)," *Advances in Cryptology*, EUROCRYPT 95 (Lect. Notes Comp. Sci., **921**), Springer–Verlag (1995), pp. 106–120.
10. Y. L. Gurieva and V. P. Il'in, "Some parallel methods and technologies of domain decomposition," *Zap. Nauchn. Semin. POMI*, **428**, 89–106 (2014).
11. V. P. Il'in, "On exponential finite volume approximations," *Russ. J. Numer Anal. Math. Model.*, **18**, No. 6, 479–506 (2003).
12. Intel Mathematical Kernel Library from Intel `//http:software.intel.com/en-us/intel-mkl`.