DOI 10.1007/s10958-019-04395-7
Journal of Mathematical Sciences, Vol. 240, No. 6, August, 2019

PROJECTION METHODS IN KRYLOV SUBSPACES

V. P. Il'in* UDC 519.6

The paper considers preconditioned iterative methods in Krylov subspaces for solving large systems of linear algebraic equations with sparse coefficient matrices arising in solving multidimensional boundary-value problems by finite volume or finite element methods of different orders on unstructured grids. Block versions of the weighted Cimmino methods, based on various orthogonal and/or variational approaches and realizing preconditioning functions for two-level multi-preconditioned semi-conjugate residual algorithms with periodic restarts, are proposed. At the inner iterations between restarts, additional acceleration is achieved by applying deflation methods, providing low-rank approximations of the original matrix and playing the part of an additional preconditioner. At the outer level of the Krylov process, in order to compensate the convergence deceleration caused by restricting the number of the orthogonalized direction vectors, restarted approximations are corrected by using the least squares method. Scalable parallelization of the methods considered, based on domain decomposition, where the commonly used block Jacobi–Schwarz iterative processes is replaced by the block Cimmino–Schwarz algorithm, is discussed. Hybrid programming technologies for implementing different stages of the computational process on heterogeneous multi-processor systems with distributed and hierarchical shared memory are described. Bibliography: 20 *titles.*

1. Introduction

In 1937, a Polish mathematician C. Kaczmarz presented in [1] an iterative method for solving systems of linear algebraic equations (SLAEs), having a remarkable generality and an elegant geometric interpretation. Every equation of the system

$$(Au)_i \equiv \sum_{j=1}^{N} a_{i,j}u_j = f_i, \quad i = 1, 2, \ldots, N, \tag{1}$$

can be associated with a hyperplane in the N-dimensional space $\mathcal{R}^N$. Given an arbitrary initial guess u^0, which is the radius vector of a point in $\mathcal{R}^N$, we determine the orthogonal projection $u_1^1 = P_1(u^0)$ onto the first hyperplane, where P_1 is the corresponding projection operator. Next we project the point obtained onto the second hyperplane and find $u_2^1 = P_2(u_1^1)$. The process is continued until the values

$$u_i^1 = P_i(u_{i-1}^1), \quad i = 2, \ldots, N, \tag{2}$$

are computed. The vector $u^1 = (u_1^1, \ldots, u_N^1)^T$ obtained in this way is called the first iterative approximation of the Kachmarz algorithm, which can be written in the form

$$\begin{aligned} &u_i^{n+1} = P_1(u^n), \quad u_i^{n+1} = P_i(u_{i-1}^n), \quad i = 2, \ldots, N, \\ &u^{n+1} = (u_1^{n+1}, u_2^{n+1}, \ldots, u_N^{n+1}), \quad n = 0, 1, \ldots. \end{aligned} \tag{3}$$

If SLAE (1) has a unique solution, then all the related hyperplanes intersect at a point u, to which the sequence of the points u_i^{n+1} obviously converges.

In 1938, an Italian mathematician G. Cimmino [2] suggested an iterative algorithm, which is also based on orthogonal projections onto hyperplanes, but projection operations are performed simultaneously rather than successively. In other terms, the point u^0 is projected onto all the

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

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 472, 2018, pp. 103–119. Original article submitted November 7, 2018.

 1072-3374/19/2406-0772

hyperplanes, and a new approximation is determined by averaging all the values $u_i^1 = P_i(u^0)$. As a result, Cimmino's method can be described as follows:

$$u^{n+1} = \frac{1}{N}\sum_{i=1}^{N} P_i(u^n), \quad i = 0, 1, \ldots. \tag{4}$$

For many years, the projection methods of Kaczmarz and Cimmino have remained almost unnoticed in the literature on numerical methods, especially, in monographs and textbooks, except for a brief presentation in [3]. Note also that the paper [1] was published in English for the second time in 1993. In 1985, V. Hackbusch [4] demonstrated that the Kaczmarz algorithm coincided with the point Gauss–Seidel iterative method (the method of successive displacements) applied to the equations obtained from system (1) by the left Gaussian transformation, i.e., by multiplying (1) on the left by the transpose A^T of the coefficient matrix,

$$A^T A u = A^T f. \tag{5}$$

This implies that the rate of convergence of the Kaczmarz iterative method is even slower than that of the "slow" Seidel algorithm because the condition number of the coefficient matrix in (5) is the squared condition number of the coefficient matrix A of the original system (1).

The Kaczmasz and Gauss–Seidel algorithms belong to the class of iterative methods of successive displacements [3], whereas the Cimmino algorithm can be associated with the point Jacobi method (the method of simultaneous displacements), in which all the components of a new approximation u^{n+1} explicitly depend on the components of the current approximation u^n only. On the one hand, methods of simultaneous displacements are, in general, slower than the corresponding methods of successive displacements. On the other hand, methods of the first group possess the advantage of allowing for a natural parallelization when implemented on multiprocessor computer systems.

As to the Kaczmarz method, several directions for its development were suggested in [5], namely, the introduction of a relaxation parameter ω, which is analogous to the passage from the Seidel method to the Successive Over Relaxation (SOR) method, the passage to the alternating variant with the symmetrizing backward iterations – a counterpart of the Symmetric Successive Over Relaxation (SSOR) method, and also a variational approach to accelerating the resulting iterative method in Krylov subspaces. Obviously, the latter approach is also applicable to the Cimmino method.

It should also be mentioned that the Kaczmarz and Cimmino methods can be generalized to the block case, where the matrix A is partitioned into m block rows,

$$A_q \in R^{N_q, N}, \quad q = 1, \ldots, m, \quad N_1 + \cdots + N_m = N,$$

and the numbers of matrix rows in different blocks can be different. For simplicity, in what follows we assume that all the block sizes are equal, i.e., $N_q = M = N/m$. Naturally, in order to write block variants of the methods (2)–(4), it is sufficient to change the indices i and N for q and M, respectively.

The idea of constructing orthogonal or skew projections underlies the construction of many methods for solving systems of linear algebraic equations with symmetric or unsymmetric matrices. To such methods one can refer various iterative processes in Krylov subspaces, see [6, 7], which are frequently connected with variational properties of functionals, as well as algorithms of deflation, aggregation, coarse-grid correction, and some others, which are actually based on low-rank approximations of matrices, whose remarkable and promising properties are presented in [8, 9].

Note that algorithms of Kaczmarz and Cimmino types, possessing some specific projective features, are sometimes referred to as row projection and row action methods and also as Projected Aggregation Methods (RAM), see the surveys [10, 11].

The aim of the present paper is to construct and investigate weighted Cimmino algorithms for solving systems of linear algebraic equations with symmetric and unsymmetric matrices, which are based on residual minimization, see Sec 2. In Sec. 3, the approaches suggested, regarded as preconditioned iterative algorithms in Krylov subspaces, are developed. Section 4 is devoted to methods for accelerating the iterative algorithms constructed, and in the Conclusion we discuss the results obtained and prospects of further investigations along the directions under consideration.

For simplicity, below we only consider SLAEs with real square coefficient matrices, although the methods suggested can be extended to the case of complex rectangular matrices.

2. Weighted Cimmino algorithms

Represent the original SLAE (1) in the following block form:

$$A_q u = f_q, \quad q = 1, \ldots, m, \quad f_q \in \mathcal{R}^{N_q}, \quad u \in \mathcal{R}^N; \tag{6}$$

here, we assume that $m \ll N$, and f_q are the subvectors of the right-hand-side vector $f = (f_1^T, \ldots, f_m^T)^T$. Consider the vectors $\widehat{u}_q \in \mathcal{R}^N$ as solutions of m underdetermined and uncoupled SLAEs

$$A_q \widehat{u}_q = f_q, \quad q = 1, \ldots, m. \tag{7}$$

The vectors $\widehat{u}_q$ are sought for in the form

$$\widehat{u}_q^0 = u^0 + A_q \check{u}_q^0, \quad \check{u}_q^0 \in \mathcal{R}^{N_q}, \quad q = 1, \ldots, m, \tag{8}$$

where u^0 is an arbitrary vector. By substituting representation (8) into Eqs. (7) (which actually is the right Gaussian transformation), we obtain the following symmetric positive semidefinite system for computing the "short" vectors $\check{u}_q^0$:

$$B_q \check{u}_q^0 = r_q^0, \quad B_q = A_q A_q^T \in \mathcal{R}^{N_q}, \quad r_q^0 = f_q - A_q u^0. \tag{9}$$

The solution of (9) in terms of the generalized inverse B_q^+ is written as

$$\check{u}_q^0 = B_q^+ r_q^0, \quad q = 1, \ldots, m. \tag{10}$$

Note that in the case where the rectangular matrix A_q is of full rank, the matrix B_q^+ is nonsingular and $B_q^+ = B_q^{-1}$. Otherwise the generalized inverse can be computed, for instance, applying Greville's formula [12]. Using (8) and (10), we write the "long" vectors $\widehat{u}_q$ as

$$\widehat{u}_q^0 = u^0 + A_q^T B_q^+ r_q^0 = u^0 + A_q^T (A_q A_q^T)^+ r_q^0. \tag{11}$$

Observe that the qth block component of the residual corresponding to the vector $\widehat{u}_q^0$ from (11) is zero because

$$\widehat{r}_q^0 = (f - A u_q^0 - A u_q^0)_q = r_q^0 - A_q A_q^T (A_q A_q^T)^+ r_q^0 = 0.$$

Now we construct an iterative process for solving SLAE (6), using an arbitrary vector $u^0 \in \mathcal{R}^N$ as the initial guess. Find the first iterative approximation u^1 in the form

$$\begin{aligned} &u^1 = u^0 + c_1^0 \widehat{u}_1^0 + \cdots + c_m^0 \widehat{u}_m^0 = u^0 + V_0 c^0, \\ &c^0 = (c_1^0, \ldots, c_m^0)^T \in \mathcal{R}^m, \quad V_0 = (\widehat{u}_1^0, \ldots, \widehat{u}_m^0) \in \mathcal{R}^{N,m}. \end{aligned} \tag{12}$$

The unknown coefficients c_q^0 in (12) will be determined from the condition of minimization of the residual r^1 of the vector u^1, i.e.,

$$r^1 = f - A u^1 = r^0 - W_0 c^0, \quad W_0 = A V_0 \in \mathcal{R}^{N,m}.$$

By formally setting $r^1 = 0$ in (13), we obtain the overdetermined system

$$W_0 c^0 = r^0 \tag{13}$$

for computing the vector c^0. The generalized solution of (13) can be computed using, for instance, the Singular Value Decomposition (SVD) or the QR algorithm [7]. We will look for the normal generalized solution of SLAE (13), which has the smallest norm and minimizes the norm of the residual r^1. To this end, we apply the left Gaussian transformation,

$$W_0^T W_0 c^0 = W^T r^0, \quad c^0 = (W_0^T W_0)^+ W_0^T r^0. \tag{14}$$

As a result of the first iteration in accordance with (14), we find the vectors

$$\begin{aligned} &u^1 = u^0 + V_0 (W_0^T W_0)^+ W_0^T r^0, \\ &r^1 = T_0 r^0, \quad T_0 = I - P_0, \quad P_0 = W_0 (W_0^T W_0)^+ W_0^T. \end{aligned} \tag{15}$$

Note that P_0 and T_0 are orthogonal projection operators because, as is readily verified, they satisfy the relations

$$P_0 = P_0^T = P_0^2, \quad T_0 = T_0^T = T_0^2.$$

Thus, formulas (15) determine the iterative process

$$\begin{aligned} &r^0 = f - Au^0, \quad r^{n+1} = T_n r^n, \quad T_n = I - P_n, \\ &P_n = W_n (W_n W_n)^+ W_n^T, \quad u^{n+1} = u^n + V_n (W_n^T W_n)^+ W_n^T r^n, \end{aligned} \tag{16}$$

which possesses the following orthogonality properties:

$$W_n r^{n+1} = 0, \quad W_n = AV_n, \quad V_n = (\widehat{u}_1^n, \ldots, \widehat{u}_m^n)^T, \quad n = 0, \ldots; \tag{17}$$

here, $\widehat{u}_q^n \in \mathcal{R}^N$, whereas the matrices of orthogonal projection P_n and T_n are defined by formulas similar to (11) and (15), respectively, which result from changing the superscript "0" for "n". The iteration formula from (16) can be written as follows:

$$u^{n+1} = u^n - B_n^{(1)} A (f - Au^n), \quad B_n^{(1)} = V_n (V_n^T A^T A V_n)^+ V_n^T. \tag{18}$$

From (18) we see that this is an algorithm with dynamically varying preconditioning matrices $B_n^{(1)}$applied to the equation obtained from the original one via the left Gaussian transformation. Obviously, it follows that the condition number of the coefficient matrix grows quadratically. In order to avoid this undesirable effect, one can consider another algorithm for computing the generalized solution of the overdetermined system (1) for the coefficient vector c^0. To this end, we multiply both sides of (1) by the matrix V_0^T (rather than by W_0^T) and obtain the equation

$$V_0^T A V_0 c^0 = V_0^T r^0. \tag{19}$$

Therefore, the solution is given by

$$c^0 = (V_0^T A V_0)^+ V_0^T r^0, \tag{20}$$

whereas instead of (15) we have the relations

$$u^1 = u^0 + B_0^{(2)} r^0, \quad B_0^{(2)} = V_0 (V_0^T A V_0)^+ V_0^T. \tag{21}$$

Here, $B_0^{(2)}$ is a preconditioner, different from $B_0^{(1)}$, which actually is a low-rank approximation to the inverse (or generalized inverse) of A. It is symmetric whenever the matrix of the original SLAE is symmetric. If the rectangular matrix $V_0 \in \mathcal{R}^{N,m}$ has rank m, then $B_0^{(2)}$ is singular or nonsingular simultaneously with A.

Notice that the matrices $B_n^{(1)}$ from (18) are low-rank approximations (in the corresponding bases V_n) of the matrix $(A^TA)^+$. Thus, in computing the vector c^0 in accordance with (21) instead of (16), we arrive at the following iterations:

$$\begin{aligned} &r^0 = f - Au^0, \quad r^{n+1} = T_n^{(2)} r^n, \quad T_n^{(2)} = I - P_n^{(2)}, \quad P_n^{(2)} = AB_n^{(2)}, \\ &u^{n+1} = u^n + B_n^{(2)} r^n, \quad B_n^{(2)} = V_n (V_n^T A V_n)^+ V_n^T. \end{aligned} \tag{22}$$

Here, as is nondifficult to realize, the matrices $P_n^{(2)}$ and $T_n^{(2)}$ are projectors, as well as P_n and T_n from (16); however, they are symmetric only if the original matrix A possesses this property. In this case, the residual vectors also possess the orthogonality property (17).

3. Cimmino type methods in Krylov subspaces

Based on the projection algorithms (16) or (22), one can construct various multi-preconditioned iterative methods in Krylov subspaces, possessing certain variational properties. We will present them in a unified framework with some preconditioners B_n, which can take the form of $B_n^{(1)}$, or $B_n^{(2)}$, or some other in every specific case. For generality, we assume that the matrices A and $B_{n,l}$ are unsymmetric. The formulas of the dynamic Multi-Preconditioned Semi-Conjugate Direction (MPSCD) methods, see [13], which are equivalent, in the rate of convergence, to the Flexible Generalized Minimal Residual (FGMRES) method [6], are as follows:

$$\begin{aligned} &r^0 = f - Au^0, \qquad n = 0, 1, \cdots : \quad u^{n+1} = u^n + P_n \bar{\alpha}_n, \\ &r^{n+1} = r^n - AP_n\bar{\alpha}_n = r^q - AP_q\bar{\alpha}_q - \cdots - AP_n\bar{\alpha}_n, \qquad 0 \le q \le n, \\ &P_n = (p_1^n \dots p_{M_n}^n) \in \mathcal{R}^{N,M_n}, \qquad \bar{\alpha}_n = (\alpha_{n,1} \dots \alpha_{n,M_n})^T \in \mathcal{R}^{M_n}, \\ &(Ap_k^n, A^\gamma p_{k'}^{n'}) = \rho_{n,k}^{(\gamma)} \delta_{n,n'}^{k,k'}, \qquad \rho_{n,k}^{(\gamma)} = (Ap_k^n, A^\gamma p_k^n), \\ &\gamma = 0, 1; \quad n' = 0, 1, \dots, n-1; \quad k, k' = 1, 2, \dots, M_n. \end{aligned} \tag{23}$$

Here, P_n are the direction matrices, having M_n columns (direction vectors), whose number can vary from iteration to iteration. The vectors $\bar{\alpha}_n \in \mathcal{R}^{Mn}$ are generalizations of the corresponding scalar coefficients for the preconditioned semi-conjugate direction methods (for $M_n = 1$ and all n), whereas the superscript takes the values 0 and 1 for semi-conjugate gradient and semi-conjugate residual methods, respectively. In the general case, the multi-preconditioned direction vectors are determined (in view of the orthogonality conditions from (23)) from the "long" recurrence relation by the formulas

$$\begin{aligned} &p_l^0 = B_{0,l}^{-1} r^0, \quad p_l^{n+1} = B_{n+1,l}^{-1} r^{n+1} - \sum_{k=0}^{n} \sum_{l=1}^{M_k} \beta_{n,k,l}^{(\gamma)} p_l^k, \quad n = 0, 1, \dots; \\ &B_{n,l} \in \mathcal{R}^{N,N}, \quad l = 1, \dots, M_n; \quad \gamma = 0, 1, \\ &\bar{\beta}_{n,k}^{(\gamma)} = \{\beta_{n,k,l}^{(\gamma)}\} = (\beta_{n,k,1}^{(\gamma)} \cdots \quad \beta_{n,k,M_n}^{(\gamma)})^T \in \mathcal{R}^{M_n}. \end{aligned} \tag{24}$$

In this case, for the coefficients of the recurrence relations (24) and for the functionals of the residual we have the following formulas: for $q = 0, \dots, n$,

$$\begin{aligned} &\beta_{n,k,l}^{(\gamma)} = -(A^\gamma p_l^k, AB_{n+1,l}^{-1} r^{n+1}) / \rho_{n,l}^{(\gamma)}, \quad n = 0, 1, \dots; \quad k = 0, \dots, n; \quad l = 1, \dots, M_n. \\ &\Phi_n^{(\gamma)}(r^{n+1}) \equiv (r^{n+1}, r^{n+1}) = (r^q, r^q) - \sum_{k=q}^{n} \sum_{l=1}^{M_n} (r^q, A^\gamma p_l^k)^2 / \rho_{k,l}^{(\gamma)}. \end{aligned} \tag{25}$$

Note that if $\gamma = 1$ or $A = A^T$, then formulas (23)–(24) ensure minimization of the functional $\Phi_n^\gamma(r^n)$ in the following Krylov subspace of dimension $M = M_0 + \cdots + M_n$, see [13]:

$$\mathcal{K}_M(r^0, A) = \mathrm{Span}\Big\{B_{0,1}^{-1}r^0, \ldots, B_{0,M_0}^{-1}r^0, \ldots, AB_{1,1}^{-1}r^0, \ldots, AB_{1,M_1}^{-1}r^0, \ldots, AB_{n,1}^{-1}r^0, \ldots, AB_{n,M_n}^{-1}r^0\Big\}. \quad (26)$$

For a symmetric matrix A, (24) yields a short recurrence relation for the direction vectors ($\beta_{n,k,l}^{(\gamma)}=0$ for $k < n$), whereas for MPSCD we obtain multi-preconditioned conjugate gradient or conjugate residual methods (for $\gamma = 0$ and $\gamma = 1$, respectively).

If the case where $A \neq A^T$, it is seemingly most reasonable to apply the Multi-Preconditioned Semi-Conjugate Residual (MPSCR) method. In solving ill-conditioned unsymmetric SLAEs, which requires a large number of iterations, it is necessary to shorthen long recurrences, which are very expensive (mainly because of an increase of the storage needed). This is done either by introducing the so-called restart procedure or by limiting the number of direction vectors (or matrices, in the case of multi-preconditioning) to be orthogonalized, or by using both approaches simultaneously. In all the cases, the rate of convergence slows down, sometimes quite considerably, which is an inevitable price to pay for reducing storage requirements.

In order to overcome this difficulty, we consider application of the Least Squares Method (LSM, [14]), confining ourselves to using restarts in their "pure" form. For simplicity, we assume that the restarts are repeated every m iterations. This means that at every iteration step with number

$$n_t = mt, \quad t = 0, 1, \ldots,$$

the residual vector is computed not from the recurrence relations (13) but from the original equation, i.e.,

$$r^{n_t} = f - Au^{n_t}, \quad (27)$$

and then the recurrence relations are used in the standard way. More exactly, it is convenient to write such an iterative process using two superscripts, which correspond to the numbers of successive approximations, namely,

$$u^{n_t} = u^{t,0}, \quad u^n = u^{t,k}, \quad k = n - n_t, \qquad n \in [n_t, n_{t+1}].$$

Here, the interrelation between neighboring restart approximations can formally be described in terms of suitable transition operators or, in matrix-vector from, as

$$u^{n_{t+1}} = B_t u^{n_t} + g_t, \quad (28)$$

where B_t and g_t can be written as closed-form expressions, depending on a specific iterative method applied.

4. Methods of inner and outer acceleration

Let the restart approximations u^{n_0}, $u^{n_1}, \ldots, u^{n_t}$, $n_0 = 0$, be already known. In order to correct the current approximation, consider a linear combination of the form

$$\begin{aligned} \widehat{u}^{n_t} &= u^{n_t} + b_1 v_1 + \cdots + b_t v_t = u^{n_t} + V_t\bar{b}, \quad \bar{b} = (b_1, \ldots, b_t)^T, \\ V_t &= \{v_k = u^{n_k} - u^{n_{k-1}};\ k = 1, \ldots, t\} \in \mathcal{R}^{N,t}, \end{aligned} \quad (29)$$

and find the coefficients b_n from the generalized solution of the overdetermined linear algebraic system obtained by premultiplying Eqs. (29) by the original matrix A, i.e.,

$$W_t\bar{b} = r^{n_t} = f - Au^{n_t}, \quad W_t = AV_t. \quad (30)$$

Here, as in formulas (14)–(20) for the weighted Cimmino methods, one can apply a number of methods for solving SLAE (30), for instance, use the singular value or QR decomposition

of the matrix W_t, or compute the generalized inverse W_t^+ by Greville's formulas, or determine the generalized normal solution by the least squares method, or reduce the problem to the less ill-conditioned system

$$V_t^T A V_t \bar{b} = V_t^T r^{n_t} \tag{31}$$

with a square coefficient matrix. In all the cases, upon computing the vector $\vec{b}$ and correcting the current approximation in accordance with formula (29), the next restart begins with the computation of the residual

$$r^{n_t+1} = f - A u^{n_t+1}, \quad u^{n_t+1} = \widehat{u}^{n_t}. \tag{32}$$

As is nondifficult to see, since the vectors u^{n_t} in (28) are computed successively, instead of the above approach, requiring that all the restart approximations be stored, one can use another algorithm, which is, in a sense, a generalization of the conjugate residual method [17]. We consider it here as a particular case of formulas (23)–(24) for $\gamma = 1$ and $M_n = 1$, i.e., without multi-preconditioning. If the vectors u^{n_t} and r^{n_t} are known, then the restart begins with the formulas

$$\begin{aligned} u^{n_{t+1}} &= u^{n_t} + \alpha_t p^t = u^0 + \alpha_0 p^0 + \cdots + \alpha_t p^t, \\ r^{n_{t+1}} &= r^{n_t} + \alpha_t A p^t = r^0 - \alpha_0 A p^0 - \cdots - \alpha_t A p^t, \end{aligned} \tag{33}$$

conventional for Krylov processes. In (33), α_t and p^t are some coefficients and direction vectors (as a rule, $p^0 = r^0$). Relations (33) possess the following remarkable property: If the vectors p^t satisfy the orthogonality conditions

$$(Ap^k, Ap^t) = \rho_t \delta_{k,t}, \quad \rho_t = (Ap^t, Ap^t), \tag{34}$$

where $\delta_{k,t}$ is the Kronecker symbol, then, for any value of t, formulas (33) ensure minimization of the residual r^{n_t+1}, provided that the coefficients α_t are determined by the relations

$$\alpha_t = \sigma_t / \rho_t, \quad \sigma_t = (Ar^{n_t}, r^{n_t}). \tag{35}$$

The restart direction vectors, in their turn, possess the properties (34) whenever they are computed from the long recurrence relation

$$p^{t+1} = r^{n_t+1} - \sum_{k=0}^{t} \beta_{t,k} p^k, \quad \beta_{t,k} = (Ap^k, Ar^{n_t+1})/\rho_t, \tag{36}$$

which actually realizes the Gram–Schmidt orthogonalization process. Note that in order to ensure numerical stability in solving systems with ill-conditioned matrices, the formulas for $\beta_{t,k}$ in (36) should be changed for the formulas of the Modified Gram–Schmidt (MGS) method, see [17] and the references therein.

Since the restart approximations u^{n_t} are connected via relation (28), which realizes m inter-restart iterations, formulas (33)–(36) can be regarded as an outer iteration process of a general two-level method in the Krylov subspace with a pseudopolynomial preconditioner because the operator B_t involves, in the general case, the preconditioners $B_{n,l}$ from (24).

Now consider the possibility of accelerating the above-described block iterative methods of Krylov type based on the deflation approach suggested by Nicolaides [16] in 1987 and developed since then by many authors, see the survey [17]. In this case, in addition to traditional variational and/or orthogonal properties of successive numerical approximations, they also satisfy conditions of orthogonality to a specially selected fixed m-dimensional deflation subspace associated with a rectangular matrix

$$V = (v_1 \dots v_m) \in \mathcal{R}^{N,m}. \tag{37}$$

We start by preliminarily discussing the application of deflation to the conjugate residual algorithm for solving a SLAE with a symmetric coefficient matrix A. In this case, first, there

is an approach to optimizing (in a sense) the initial guess u^0. Let an arbitrary vector u^{-1} be given. Determine the vectors

$$u^0 = u^{-1} + Vc, \quad r^0 = r^{-1} - AVc. \tag{38}$$

The vector $c = (c_1, \ldots, c_m)^T$ of unknown coefficients occurring in (38) will be determined by solving the overdetermined system

$$Wc = AVc = r^{-1}, \tag{39}$$

which is obtained by formally setting $r^0 = 0$ in (38). Applying the least squares method to (39), we find the normal solution

$$c = (W^T W)^+ W^T r^{-1}, \tag{40}$$

which ensures the smallest norm of the residual vector:

$$\begin{aligned} r^0 &= T_0 r^{-1}, \quad T_0 = I - W(W^T W)^{-1} W^T, \\ (r^0,\ r^0) &= (r^{-1},\ r^{-1}) - (W(W^T W)^{-1} W^T r^{-1},\ r^{-1}) \\ &= (W^T W z,\ z) - ((W^T W)^{-1} z,\ z), \quad z = W^T r^{-1}. \end{aligned} \tag{41}$$

In (41), the matrix T_0 is a symmetric orthogonal projector with the properties

$$T_0 = T_0^T = T_0^2, \quad W^T T_0 = T_0 W = 0,$$

i.e., the space $\mathrm{Span}(W)$ belongs to the kernel $\mathcal{N}(T_0)$. Then we define the initial direction vector by the relations

$$p^0 = r^0 - V(W^T W)^{-1} W^T A r^0 = B r^0, \quad B = I - V(W^T W)^{-1} W^T A, \tag{42}$$

and for the vectors r^0 and p^0 we obtain the deflation orthogonality conditions

$$W^T r^0 = 0, \quad W^T A p^0 = 0. \tag{43}$$

The matrix B introduced in (42) satisfies the following readily verifiable orthogonality conditions:

$$W^T AB = 0, \quad BV = 0.$$

Subsequent iterations of the corresponding Deflated Conjugate Residual (DCR) algorithm are performed in accordance with the standard formulas of the preconditioned conjugate residual method

$$\begin{aligned} u^{n+1} &= u^n + \alpha_n p^n, \quad \alpha_n = \sigma_n/\rho_n,\ \rho_n = (Ap^n, Ap^n), \\ r^{n+1} &= r^n - \alpha_n A p^n, \quad \sigma_n = (ABr^n,\ r^n), \\ p^{n+1} &= B r^{n+1} + \beta_n p^n, \quad \beta_n = \sigma_{n+1}/\sigma_n, \end{aligned} \tag{44}$$

where the preconditioner B is defined in (42). This algorithm, at every iteration step, minimizes the norm $||r^n||$ of the current residual in the preconditioned Krylov subspace

$$\mathcal{K}_n(A,\ r^0, B) = \mathrm{Span}(r^0,\ ABr^0,\ \ldots,\ (AB)^{n-1} r^0), \tag{45}$$

and the vectors computed satisfy the orthogonality conditions

$$\begin{aligned} (ABr^k,\ r^n) &= \sigma_n \delta_{k,n}, \\ (Ap^k, Ap^n) &= \rho_n \delta_{k,n}, \quad k = 0, 1, \ldots, n-1, \\ W^T r^n &= 0, \quad W^T A p^n = 0, \quad n = 0, 1, \ldots. \end{aligned} \tag{46}$$

In this case, the following relations hold:

$$\begin{aligned}(r^{n+1},\ r^{n+1}) &\le (r^n,\ r^n) - \frac{(ABr^n,\ r^n)^2}{(Ap^n,\ Ap^n)} \\ &\le (r^k, r^k) - \frac{(ABr^k,\ r^k)^2}{(Ap^k,\ Ap^k)} - \cdots - \frac{(ABr^n,\ r^n)^2}{(Ap^n,\ Ap^n)}, \quad k = 0,\ 1\ ,..,\ n.\end{aligned} \tag{47}$$

It should be mentioned that the preconditioning matrix B introduced above is singular because $BW = 0$. However, relations (43) and (46) imply that all the residual vectors are orthogonal to the kernel of the matrix $\bar{A} = AB$, $\mathcal{N}(\bar{A}) = \mathrm{Span}(W)$, which ensures that the iterations (41)–(44) converge.

The above formulas of the DCR method can be modified and simultaneously simplified if one takes into account the specific properties of the preconditioner B, namely,

$$B^2 = B, \quad AB = B^T A = B^T AB, \tag{48}$$

as is done in the DCG algorithm. More exactly, the residual vector determined in the preconditioned Krylov subspaces (45) by formulas (44) is represented as

$$r^{n+1} = P_n(AB)r^n = P_n(B^T AB), \tag{49}$$

where P_n is a matrix polynomial of degree n. It follows that the iterations considered can be written in the following form:

$$\begin{aligned}&p^0 = Br^0, \quad \alpha_n = \sigma_n/\rho_n, \quad q^0 = p^0, \quad s^0 = Ap^0, \\ &u^{n+1} = u^n + \alpha_n p^n, \quad \rho_n = (s^n,\ s^n), \\ &r^{n+1} = r^n - \alpha_n Ap^n, \quad \sigma_n = (Aq^n,\ r^n), \quad q^n = Br^n, \\ &p^{n+1} = q^{n+1} + \beta_n p^n, \quad \beta_n = \sigma_{n+1}/\sigma_n, \\ &s^{n+1} \equiv Ap^{n+1} = Aq^{n+1} + \beta_n s^n,\end{aligned} \tag{50}$$

where the initial residual vector r^0 is obtained from (42). Every iteration performed in accordance with (50) involves one multiplication of each of the vectors q^n and r^n by the matrices A and B, respectively.

Now consider a formal modification of the above algorithm, which will be referred to as the Deflated Conjugate Direction (DCD) algorithm. Let two distinct families of vectors v_k and w_k, which are the columns of some matrices

$$V = (v_1 \dots v_{m_v}) \in \mathcal{R}^{N,m_v}, \quad W = (w_1 \dots w_{m_w}) \in \mathcal{R}^{N,m_w}, \tag{51}$$

be given. In (51), the matrix W is a generalization of the above-considered matrix $W = AV$, and $m_w = m_v$. As above, the initial guess is chosen in accordance with the formulas

$$u^0 = u^{-1} + Vc, \quad r^0 = r^{-1} - AVc, \quad c = (c_1,\ \dots,\ c_{m_v})^T \in \mathcal{R}^{m_v}.$$

However, the vector c is determined as the generalized normal solution of the SLAE

$$Mc \equiv W^T AVc = W^T r^{-1}, \quad c = (W^T AV)^+ W^T r^{-1}, \quad M \in \mathcal{R}^{m_w,m_v}, \quad M^+ \in \mathcal{R}^{m_v,m_w}, \tag{52}$$

which is compatible but has a rectangular coefficient matrix. In this case, the initial residual is determined by

$$r^0 = \widehat{B}r^{-1}, \quad \widehat{B} = I - AV(W^T AV)^+ W^T, \tag{53}$$

and the following orthogonality conditions are fulfilled:

$$W^T r^0 = 0, \quad W^T \widehat{B} = 0, \quad \widehat{B}AV = 0. \tag{54}$$

The initial direction vector is determined, using a new preconditioning matrix, by the formula

$$p^0 = \check{B}r^0, \quad \check{B} = I - V(W^T AV)^+ W^T A, \tag{55}$$

which ensures that

$$\check{V} = 0, \quad W^T A\check{B} = 0, \quad W^T A p^0 = 0. \tag{56}$$

The subsequent iterations in the DCD algorithm are performed in accordance with the formulas

$$\begin{aligned} &u^{n+1} = u^n = \alpha_n p^n, \quad \alpha_n = \sigma_n/\rho_n, \quad \rho_n = (Ap^n, Ap^n), \\ &r^{n+1} = r^n - \alpha_n A p^n, \quad \sigma_n = (A\check{B}r^n, r^n), \\ &p^{n+1} = \check{B}r^{n+1} + \beta_n p^n, \quad \beta_n = \sigma_{n+1}/\sigma_n. \end{aligned} \tag{57}$$

Then, under conditions (52) and (53), for all n we have the following orthogonality properties:

$$\begin{aligned} &W^T r^{n+1} = 0, \quad W^T A p^{n+1} = 0, \\ &(A\check{B}r^k r^n) = \sigma_n \delta_{k,n}, \quad (Ap^k, Ap^n) = \rho_n \delta_{k,n}, \quad k = 0, 1, \ \ldots, \ n-1. \end{aligned} \tag{58}$$

For $m_v, m_v < N$ and for V and W introduced in (51), the matrices V^T and W can be interpreted as a restriction and a prolongation operators but relative to spaces of different dimensions. This allows one to consider numerous formal generalizations of the deflated algorithms presented.

5. Conclusion

The weighted block versions of the Cimmino iterative method naturally develop the conventional representations of projection algorithms of the class under consideration, which can efficiently be applied in solving a wide range of SLAEs with symmetric, as well as (and especially) unsymmetric matrices. The presented approaches to constructing multi-preconditioned iterations in Krylov subspaces and also to their two-level acceleration by deflation and least-squares methods look promising for solving ill-conditioned linear algebraic systems with large sparse coefficient matrices. Here, as a theoretical foundation one can use the results on analyzing properties of projection and deflation methods presented in [18–20] and the references therein.

An important feature of the iterative methods considered in this paper is their natural parallelism. A high scalable efficiency here can be achieved based on using domain decomposition that realizes block variants of Cimmino–Schwartz methods in Krylov subspaces, when applying the above-described approaches to accelerating inner and outer iterations. The two-level nature of the resulting computational process provides for an efficient application of technologies of hybrid programming on heterogeneous supercomputers with distributed and hierarchical shared memory, including MPI processes, multithreaded computations, and vectorization of operations. The main part here is played by intensive numerical testing, which is an object of further investigations along these directions.

This work was supported by the Russian Foundation for Fundamental Research (projects 16-29-15122 ofi-m and 18-01-00295).

Translated by L. Yu. Kolotilina.

REFERENCES

1. S. Kaczmarz, "Angenaherte Auflosung von Systemen lineare Gleichungen," *Bull. Int. Acad. Polon. Sci. Lett.*, **35**, 355–357 (1937).
2. G. Cimmino, "Calcolo approssimati de soluzioni dei sistemi di equazioni linear," *Ric. Sci.*, **11.9**, 326–333 (1938).
3. V. P. Il'in, *Methods and Technologies of Finite Elements* [in Russian], IVMIMG SO RAN, Novosibirsk (2007).

4. W. Hackbush, *Multigrid Methods and Applicatios*, Springer-Verlag, Berlin (1985).
5. V. P. Il'in, On the Kaczmarz iterative method and its generalizations," *Sib. Zh. Industr. Mat.*, **9**, No. 3, 39–40 (2006).
6. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS (1996).
7. G. H. Golub and Y. C. F. Loan, *Matrix Computations*, John's Hopkins Univ. Press (1996).
8. M. T. Chu, R. E. Funderlic, and G. H. Golub, "A rank-one reduction formula and its application to matrix factorization," *SIAM Rev.*, **37**, 512–530 (1999).
9. L. Hubert, J. Meulman, and W. Heise, "Two purposes for matrix factorization: A historical appraisal," *SIAM Rev.*, **42**, No. 1, 68–82 (2000).
10. G. Appleby and D. S. Smolarski, "A linear acceleration row action method for projecting onto subspaces," *Eectr. Trans. Numer. Anal. Kent State Univ.*, **20**, 253–275 (2005).
11. H. Scolnik, N. Echebest, M. T. Guardarrdarucci, and M. C. Vacachino, "A class of optimized row projection methods for solving large nonsymmetric linear systems," *Appl. Numer. Math.*, **41**, 499–513 (2002).
12. F. P. Gantmakher, *The Theory of Matrices* [in Russian], Nauka, Moscow (1968).
13. V. P. Il'in, "Problems of parallel solution of large systems of linear algebraic equations," *Zap. Nauchn. Semin. POMI*, **439**, 112–127 (2015).
14. V. P. Il'in, "Two-level least-squares methods in Krylov subspaces," *Zap. Nauchn. Semin. POMI*, **463**, 224–239 (2017).
15. V. P. Il'in, "Methods of semiconjugate directions," *Russ. J. Numer. Anal. Math. Model.*, **23**, No. 4, 369–387 (2008).
16. R. A. Nicolaydes, "Deflation of conjugate gradients with application to boundary-value problems," *SIAM J. Numer. Anal.*, **24**, 335–365 (1987).
17. Y. Gurieva and V. P. Il'in, "On parallel computational technologies of augmented domain decomposition methods," in: *Parallel Computing Technologies,* 13*th International Conference PaCT*, Petrozavodsk, Russia (2015), pp. 33–46.
18. J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, "Convergence estimates for product iterative methods with applications to domain decomposition," *Math. Comp.*, **57**, No. 195, 1–21 (1991).
19. L. Yu. Kolotilina, "Two-fold deflation preconditioning of linear algebraic systems. I. Theory," *Zap. Nauchn. Semin. POMI*, **229**, 95–152 (1995).
20. J. Liesen, M. Rozložnik, and Z. Strakoš, "Least squares resudials and minimal residual methods," *SIAM J. Sci. Comput.*, **23**, No. 5, 1503–1525 (2002).