DOI 10.1007/s10958-016-2945-4
Journal of Mathematical Sciences, Vol. 216, No. 6, August, 2016

PROBLEMS OF PARALLEL SOLUTION OF LARGE SYSTEMS OF LINEAR ALGEBRAIC EQUATIONS

V. P. Il'in* UDC 519.6

The paper considers some modern problems arising in developing parallel algorithms for solving large systems of linear algebraic equations with sparse matrices occurring in mathematical modeling of real-life processes and phenomena on a multiprocessor computer system (MCS). Two main requirements to methods and technologies under consideration are fast convergence of iterations and scalable parallelism, which are intrinsically contradictory and need a special investigation. The paper analyzes main trends is developing preconditioned iterative methods in Krylov's subspaces based on algebraic domain decomposition and principles of their program implementation on a heterogeneous MCS with hierarchical memory structure. Bibliography: 24 *titles.*

1. Introduction

As is well known, solution of large Systems of Linear Algebraic Equations (SLAEs) is a bottleneck in problems of mathematical modeling because the costs of solving a SLAE grow nonlinearly as the number of degrees of freedom increases. Large and extremely large SLAEs are systems of the largest size that can be solved for the time being. Today, extremely large systems have orders 10^{10}–10^{12} and can be solved on a Multiprocessor Computer System (MCS) with about 10^5–10^7 processing units. It is clear that in practice such SLAEs can only be solved by iterative methods because direct methods require too much computer resources, especially, computer memory.

The main approaches to solving the problems under consideration are based on using preconditioned iterations in Krylov subspaces and algebraic domain decomposition into subdomains with intersections parametrized using their width and various interface conditions on the interior boundaries of adjacent subdomains. The related auxiliary problems in the subdomains are solved concurrently on the corresponding computer units by either direct or iterative methods.

The basic requirements to the algorithms considered are intrinsically contradictory. On the one hand, in order to increase the acceleration factor, one must have a large number of subdomains, which allows one to exploit a large number of processing units concurrently. However, if, in this case, the simplest "standard" approaches are used, then the number of outer iterations over the subdomains grows quite considerably. This circumstance gives rise to a number of methods for accelerating such an iterative process, which, as a rule, result in making the related algorithms considerably more complicated from the logical standpoint.

In the last decades, the iterative solution methods have mainly been developed in three directions. The first one is related to generalizing the concept of Krylov's subspaces, in which approximate solutions are constructed based on variational, orthogonal, and/or projection principles. In this connection, one can mention semi-conjugute direction methods [1–3] for solving nonsymmetric SLAEs, rational (in contrast to the classical polynomial) Krylov subspace methods [4, 5], methods of Induced Dimension Reduction (IDR(s)) [6, 7], and also deflated and augmented Krylov subspace techniques [8, 9], in which the Krylov basis is supplemented with some additional vectors possessing special properties.

The second direction is related to decreasing the condition numbers of the SLAEs to be solved by using new types of preconditioners and also by applying nested dissection methods

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

Translated from *Zapiski Nauchnykh Seminarov POMI*, Vol. 439, 2015, pp. 112–127. Original article submitted October 23, 2015.

and multifrontal algorithms of matrix decomposition associated with special orderings of unknowns. Here, promising approaches are based on the tensor apparatus [10] and representation of matrices in the Hierarchically Semi-Separable (HSS) form [11], which yield efficient approximate and/or exact algorithms of matrix factorization. In addition, relatively recently it has been suggested to accelerate Krylov iterations with the use of multipreconditioning (see [12–14]), which involves several preconditioners at every iteration. This technique provides an extension of the known dynamic (or flexible) preconditioning [15], where at every iteration an only one preconditioner is used, but it changes from iteration to iteration.

The third direction is related to the first two but is specially oriented to scalable parallelism, with a mapping of algorithms onto a MCS with a specific architecture. In this case, the main tool is the method of decomposition of grid domains in its geometric or algebraic interpretation. It is intuitively clear that a multidimensional grid domain should be decomposed into subdomains of the same dimension in a balanced way. The ultimate algorithm performance is determined, to a significant extent, by computing and information technologies and also by the quality of program implementation. Thus, the efficiency of parallelization must be characterized not only by theoretic estimates but also by results of numerical experiments.

The present paper is organized as follows. In Sec. 2, block semi-conjugate direction methods with multipreconditioning are described in an abstract vector-matrix form. In the next Sec. 3, these methods are specialized for parallel domain decomposition algorithms, and a comparative analysis of different structural approaches (like FETI and BNN, see [16, 17]) and variants free of separating subdomains suggested in this paper is presented. Also some principles of acceleration of iterative methods in Krylov's subspaces based on aggregation, deflation, and coarse-grid correction, which are different methods of low-rank approximation of matrices described in [18–21], are considered. The last section is devoted to a brief presentation of the concept of the program library KRYLOV [22] as an open integrated toolkit environment oriented at users interested in practical application of fast parallel algebraic solvers. At the same time, this library provides means for automatically constructing algorithms in developing and experimentally studying new methods. In the Conclusion, some ideas concerning realization of the science-technology chain from generation of an idea and theoretical investigation of the related methods to trial versions and experimental approbation, and to highly efficient program implementation of algorithms and their practical promotion are presented.

2. Block semi-conjugate direction methods

The semi-conjugate gradient and semi-conjugate direction (SCG and SCR, see [1, 2]) methods are developed for solving systems of real linear algebraic equations with unsymmetric positive-definite coefficient matrices,

$$Au = f, \quad A = \{a_{i,j}\} \in \mathcal{R}^{N,N}, \quad u = \{u_i\}, \quad f = \{f_i\} \in \mathcal{R}^N, \tag{1}$$

satisfying the conditions

$$(Av, v) \geq \delta ||v||^2, \quad \delta > 0, \quad (v, w) = \sum_{i=1}^{N} v_i w_i, \quad ||v||^2 = (v, v) \tag{2}$$

for all real vectors v.

In contrast to the generalized minimum residual methods [3], the family of semi-conjugate direction algorithms does not require that a Hessenberg matrix be explicitly stored, which considerably simplifies the numerical algorithm and its program implementation. As is known, the above-presented inequalities ensure that the symmetric part of the matrix $A = \{a_{i,j}\}$ is

positive definite, i.e.,

$$(A^s u, u) \geq \delta(u, u), \quad A^s = (A + A^T)/2, \quad A^T = \{a_{j,i}\},$$

and also that the real parts of the eigenvalues of A are positive (the superscript T means matrix transposition).

In order to solve Eq. (1), we apply the following iterative method of the generalized (or block) Krylov type:

$$\begin{gathered} r^0 = f - Au^0, \quad u^{n+1} = u^n + P_n \bar{\alpha}_n, \quad n = 0, 1, \ldots, \\ r^{n+1} = r^n - AP_n\bar{\alpha}_n = r^q - AP_q\bar{\alpha}_q - \cdots - AP_n\bar{\alpha}_n, \quad 0 \leq q \leq n. \end{gathered} \tag{3}$$

Here, $P_n = (p_1^n \ldots p_{M_n}^n) \in \mathcal{R}^{N,M_n}$ is the matrix composed of the direction vectors p_k^n, and $\bar{\alpha}_n = (\alpha_{n,1} \ldots \alpha_{n,M_n})^T$ is the vector of iteration parameters, which will be determined from the following orthogonality relations:

$$\begin{gathered} (Ap_k^n, A^\gamma p_{k'}^{n'}) = \rho_{n,k}^{(\gamma)} \delta_{n,n'}^{k,k'}, \quad \rho_{n,k}^{(\gamma)} = (Ap_k^n, A^\gamma p_k^n), \\ \gamma = 0, 1; \quad n' = 0, 1, \ldots, n-1; \quad k, k' = 1, 2, \ldots, M_n. \end{gathered} \tag{4}$$

In (4), $\delta_{n,n'}^{k,k'}$ is the Kronecker symbol, which is equal to unity if $n = n'$ and $k = k'$, and to zero otherwise; the values $\gamma = 0, 1$ will specify, in the sequel, either the semi-conjugate gradient ($\gamma = 0$) or semi-conjugate residual method, respectively. Observe that in contrast to the conventional semi-conjugate direction methods, in relations (3) at every nth iteration M_n direction vectors, rather than one, are involved, and, in general, their number can vary from iteration to iteration.

If, in (3), we introduce the integrated direction vectors p^n via

$$\alpha_n p^n = P_n \bar{\alpha}_n, \quad n = 0, 1, \ldots, \tag{5}$$

where the scalar coefficients α_n have not yet been determined, then, instead of (3), we obtain the classical representation of the residual vectors

$$r^{n+1} = r^n - \alpha_n A p^n = r^q - \alpha_q A p^q - \cdots - \alpha_n A p^n, \quad 0 \leq q \leq n, \tag{6}$$

in a certain Krylov subspace

$$\widetilde{K}(r^0, A) = \operatorname{span}\{r^0, Ap^0, \ldots, A^n p^0\}, \tag{7}$$

which will be specified after determining the vectors $p^0, \ldots, p^n$.

Obviously, we can alternatively write (3) without using the matrices P_n and vectors $\bar{\alpha}_n$ as follows:

$$r^{n+1} = r^n - A\sum_{l=1}^{M_n} \alpha_{n,l} p_l^n = r^q - A\Big(\sum_{l=1}^{M_q} \alpha_{q,l} p_l^q - \cdots - \sum_{l=1}^{M_n} \alpha_{n,l} p_l^n\Big). \tag{8}$$

This implies that under conditions (4), we can define the functional

$$\begin{aligned} \Phi_n^{(\gamma)}(r^{n+1}) &\equiv (r^{n+1}, A^{\gamma-1} r^{n+1}) \\ &= (r^q, A^{\gamma-1} r^q) - \sum_{l=1}^{M_n} \sum_{k=q}^{n} \alpha_{k,l} [2(r^q, A^\gamma p_q^k) - \alpha_{k,l} (Ap_l^k, A^\gamma p_l^k)]. \end{aligned} \tag{9}$$

Thus, from the extremum condition

$$\partial \Phi_n^{(\gamma)} / \partial \alpha_{k,l} = 0,$$

for $q = 0, 1, \ldots, n$ we obtain

$$\alpha_{k,l}^{(\gamma)} = (r^q, A^\gamma p_l^k)/\rho_{k,l}^{(\gamma)}, \quad l = 1, \ldots, M_n; \quad k = q, q+1, \ldots, n. \tag{10}$$

Relations (3), (4), and (10) imply that the vectors r^{n+1} and $p_k^{n'}$ satisfy the conditions of A^γ-semi-conjugacy in the following sense:

$$(r^{n+1}, A^\gamma p_k^{n'}) = 0, \quad n' = 0, 1, \ldots, n; \quad k = 1, \ldots, M_n; \quad \gamma = 0, 1. \tag{11}$$

Remark 1. As is nondifficult to see, the corresponding value $\Phi_n^{(\gamma)}(r^{n+1})$ of the functional is of the form

$$\Phi_n^{(\gamma)}(r^{n+1}) = (r^q, r^q) - \sum_{k=q}^{n} \sum_{l=1}^{M_n} (r^q, A^\gamma p_l^k)^2 / \rho_{k,l}^{(\gamma)} \tag{12}$$

and attains its minimum, provided that $\gamma = 1$ or the matrix A is symmetric. We emphasize, in particular, that in the semi-conjugate gradient method, the functional $\Phi_n^{(0)}(r^{n+1}) = (A^{-1}r^{n+1}, r^{n+1})$ does not, in general, attains its minimum if A is unsymmetric.

The direction vectors p_l^n will be determined from the orthogonality conditions (5) for $\gamma = 0, 1$ in the following form:

$$\begin{aligned} &p_l^0 = B_{0,l}^{-1} r^0, \quad p_l^{n+1} = B_{n+1,l}^{-1} r^{n+1} + \sum_{k=o}^{n} \sum_{l=1}^{M_k} \beta_{n,k,l}^{(\gamma)} p_l^k, \quad n = 0, 1, \ldots; \\ &B_{n,l} \in \mathcal{R}^{N,N}, \quad l = 1, \ldots, M_n; \quad \gamma = 0, 1. \end{aligned} \tag{13}$$

Here, $\bar{\beta}_{n,k}^{(\gamma)} = \{\beta_{n,k,l}^{(\gamma)}\} = (\beta_{n,k,1}^{(\gamma)} \ldots \beta_{n,k,M_n}^{(\gamma)})^T \in \mathcal{R}^{M_n}$ are the coefficient vectors, and $B_{n,l} \in \mathcal{R}^{N,N}$ are preconditioners, which are chosen with account for the requirements of nonsingularity, cheap invertibility, and efficient acceleration of the iterative methods under construction. Note that the preconditioners $B_{n,l}$ are dynamic, or flexible (as in the FGMRES methods [3]) because, for a given l, they depend on the iteration number n.

Remark 2. In formulas (12), it is assumed that the initial approximation u^0 and the corresponding residual vector are the same for all values of l. Obviously it is nondifficult to construct block versions of the multiply preconditioned methods suggested by prescribing M_0 distinct initial vectors $U^0 = (u_1^0 \ldots u_{M_0}^0)^T$. Moreover, the approach considered can readily be extended to the case of solving several systems with the same coefficient matrix A and different right-hand sides $\bar{f}_m = (f_{1,m} \ldots f_{N,m})^T$, $m \le M_0$. In the latter case, one can define the rectangular matrices of right-hand sides and residuals,

$$F = (\bar{f}_1 \ldots \bar{f}_{M_0}), \quad R^0 = F - AU^0 \in \mathcal{R}^{N,M_0}.$$

In the case where $m < M_0$, some $M_0 - m$ columns in the matrix F will be identical. However, in what follows, such extensions will not be dealt with.

On substituting expressions (13) into the orthogonality conditions (4), we obtain the following formulas for the coefficients:

$$\begin{aligned} &\beta_{n,k,l}^{(\gamma)} = -(A^\gamma p_l^k, AB_{n+1,l}^{-1} r^{n+1}) / \rho_{n,l}^{(\gamma)}, \\ &\quad n = 0, 1, \ldots; \quad k = 0, \ldots, n; \quad l = 1, \ldots, M_n. \end{aligned} \tag{14}$$

Now we are ready to state the result below.

Theorem 1. *Under assumptions* (2)*, the iterative methods* (3), (11), (13), (14) *with* $\gamma = 0, 1$ *and nonsingular preconditioners* $B_{n,l}$, $n = 0, 1, \ldots$, $l = 1, \ldots, M_n$, *ensure that the extremum conditions* (10) *are fulfilled. Conditions* (10) *correspond for* $\gamma = 1$ *to minimization of the*

residual norm $||r^{n+1}||$ *in the multipreconditioned Krylov subspace*

$$\begin{aligned}&\mathcal{K}_{\sum\limits_{n+1}}(r^0, A)\\ &=\operatorname{span}\Big\{B_{0,1}^{-1}r^0, \dots, B_{0,M_0}^{-1}, AB_{1,1}^{-1}r^1, \dots, AB_{1,M_1}^{-1}r^1, \dots, A^nB_{n,1}^{-1}r^n, \dots, A^nB_{n,M_n}^{-1}r^n\Big\}.\end{aligned} \tag{15}$$

In the methods in question, the residual vectors are semi-conjugate for $\gamma = 0, 1$ *in the following generalized sense:*

$$(A^\gamma B_{k,l}^{-1}r^n, r^k) = \begin{cases}0, & k < n,\\ \sigma_n^{(\gamma)} = (A^\gamma B_{n,l}^{-1}r^n, r^n), & k = n,\end{cases} \qquad l = 1, \dots M_k, \tag{16}$$

whereas the coefficients $\alpha_{n,l}^{(\gamma)}$ *in* (3) and (11) *are given by the formula*

$$\alpha_{n,l}^{(\gamma)} = (A^\gamma B_{n,l}^{-1}r^n, r^n)/\rho_{n,l}. \tag{17}$$

Proof. For $k < n$, (16) follows from the relations

$$\begin{aligned}(A^\gamma B_{k,l}^{-1}r^k, r^n) &= ((A^\gamma p_l^k - \sum_{i=0}^{k-1}\beta_{k,i,l}A^\gamma p_l^i), (r^0 - \sum_{i=0}^{n-1}\alpha_{i,l}^{(\gamma)}Ap_l^i))\\ &= (A^\gamma p_l^k, r^0) - \alpha_{k,l}\rho_{k,l}^{(\gamma)} - \sum_{i=0}^{k-1}\beta_{k,i,l}[(A^\gamma p_l^i, r^0) - \alpha_{i,l}^{(\gamma)}\rho_{i,l}^{(\gamma)}] = 0.\end{aligned}$$

For $k = n$, from similar relations we have

$$(A^\gamma B_{n,l}^{-1}r^n, r^n) = (A^\gamma p_l^n, r^0),$$

which, in view of (11), yields (16). □

Note that in the particular case where $\gamma = 1$ and $B_{n,l} = I$ (the identity matrix), the residual vectors with properties (16) are said to be right A-semi-conjugate. In [1], the corresponding method is called the Generalized Conjugate Residual (GCR) method.

From the algebraic viewpoint, for $\gamma = 1$, formulas (13), (14) for $n = 0, 1, \dots$ correspond to transformation of the linearly independent vectors $B_{0,l}^{-1}r^0, \dots, B_{n,l}^{-1}r^0$ into the vectors $p_l^0, \dots, p_l^n$, which are A^TA-orthogonal, by using the Gram–Schmidt orthogonalization process. As is known (see [3] and the references therein), for large n, this orthogonalization process can prove to be numerically unstable with respect to round-off errors. For this reason, it is recommended to apply the modified Gram–Schmidt orthogonalization method. In the case considered, the latter method is realized by the following algorithm.

We suggest to compute $\beta_{n,k,l}^{(\gamma)}$ by the formula

$$\beta_{n,k,l}^{(\gamma)} = -(A^\gamma p_l^k, Ap_l^{n,k})/\rho_{n,l}^{(\gamma)} \tag{18}$$

(rather than (14)), in which the vectors $p_l^{n,k}$ are determined from the relations

$$\begin{aligned}p_l^{n,k} = p_l^{n,k-1} + \beta_{n,k-1,l}^{(\gamma)}p_l^{k-1} &= B_{n+1,l}^{-1}r^{n+1} + \sum_{i=0}^{k-1}\beta_{n,i,l}^{(\gamma)}p_l^n,\\ l = 1, \dots, M_n, \quad k = 0, 1, \dots, n+1;&\end{aligned} \tag{19}$$

$$p_l^{n,0} = B_{n+1,l}^{-1}r^{n+1}, \quad p_l^{n,n+1} = p_l^{n+1}.$$

Here, if $\gamma = 1$, then the choice of the coefficients $\beta_{n,i,l}^{(\gamma)}$ in (19) ensure that the conditions

$$\partial ||p_l^{n,k}|| / \partial \beta_{n,i} = 0, \quad i = 0, 1, \ldots, k-1,$$

of minimizing the norm of the vector $p_l^{n,k}$ are fulfilled. These conditions imply a formula coinciding with (14) up to notation. As is readily seen, by virtue of the orthogonality relations

$$(Ap_l^{n,k}, A^\gamma p_l^i) = 0, \quad i = 0, \ldots, k-1,$$

which hold for all l, the values of the coefficients $\beta_{n,k,l}^{(\gamma)}$ computed (in the exact arithmetic) by formulas (14) and (18) coincide because

$$(A^\gamma p_l^k, AB_{n+1,l}^{-1} r^{n+1}) = (A^\gamma p_l^k, Ap_l^{n,k}).$$

Observe that in the case $\gamma = 1$ (the semi-conjugate residual method), computations in accordance with formulas (19) require no additional matrix-vector operations, which follow from the recurrent relations

$$Ap_l^{n,k} = Ap_l^{n,k-1} + \beta_{n,k-1,l}^{(\gamma)} Ap_l^{k-1},$$

valid for all l.

Since the semi-conjugate direction methods for solving linear systems with nonsymmetric matrices are based on using long-term recurrences of the form (19), for large n, they require too much RAM similarly to the GMRES method.

In order to alleviate this limitation, we will use two commonly accepted approaches, one of which is related to restarting the algorithm in a given number of iterations m_{res}, whereas the other one is based on a limited orthogonalization, in the course of which the direction vectors from a prescribed number of last iterations m_{lim} are preserved, see [1–3]. Both of these forced approaches result in an unavoidable reduction of the convergence rate of the iterations. But the situation can be somewhat relaxed by choosing appropriate preconditioners, for which the number of iterations turns out to be moderate.

3. Domain decomposition algorithms with coarse grid correction in Krylov's subspaces

In this section, the multipreconditioned methods of semi-conjugate directions, considered above in a rather abstract form, are specified in application to constructing naturally parallel algebraic domain decomposition methods based on two types of preconditionings. The first of them is connected with the additive Schwarz method, or the block Jacobi method, in which inversion of each of the blocks means solution of an auxiliary system of linear algebraic equations in the corresponding subdomain. In order to accelerate the convergence of iterations for the subdomains, another type of preconditioning is used. The latter consists in a coarse-grid correction of the approximate solutions. Specific variants of such a correction are know as aggregation and deflation, whereas the general principle of constructing them consists in small-rank approximation of the inverse matrices. In this case, multipreconditioning in Krylov's subspaces is aimed at studying the most efficient combinations of two approaches, specifically, the additive Schwarz method and aggregation.

Initially, the domain decomposition methods are described on the continuous level. A computational domain Ω, in which a certain boundary value problem for a differential equation is solved, is decomposed into P subdomains Ω_s, in each of which the corresponding subproblem is stated. However, we will only consider the discretized problem, i.e., the problem in terms of the computational grid domain having the total number of nodes N and composed of some nonoverlapping computational subdomains, $\Omega^h = \bigcup_{s=1}^{P} \Omega_s^h$, the number of grid nodes in each

of the subdomains being equal to N_s, $N_1 + N_2 + \cdots + N_P = N$. Below, for shortness, the superscript h will be omitted.

It should be emphasized that we consider decomposition into grid subdomains without interface nodes belonging to two or more adjacent subdomains. Such a decomposition of the computational grid domain into disjoint subdomains (we do not dwell on decomposition methods and criteria of their quality, which is a separate important question) is only the first step toward constructing grid subdomains with parametrized intersections described below.

For a grid subdomain Ω_s, by $\Gamma_s = \Gamma_s^0$ we denote its boundary, i.e., the collection of nodes that are exterior with respect to Ω_s but have at least one neighboring node in Ω_s ($\bar{\Omega}_s^0 = \Omega_s \cup \Gamma_s^0$ is the closure of the original grid subdomain Ω_s). In what follows, Γ_s^1 denotes the first extended boundary, or the first exterior front $\bar{\Omega}_s$, i.e., the collection of nodes not belonging to $\bar{\Omega}_s$ but having at least one neighboring node in $\bar{\Omega}_s^0$ ($\bar{\Omega}_s^1$ is the first extension of $\bar{\Omega}_s^0$). The subsequent stages of extension of a grid subdomain are defined similarly, and the number Δ of such stages will be referred to as the parameter of the extended subdomain $\bar{\Omega}_s = \bar{\Omega}_s^\Delta = \Omega_s^\Delta \bigcup \Gamma_s^\Delta$, where the nodes from Γ_s^Δ do not belong to Ω_s^Δ; the number of nodes in $\bar{\Omega}_s$ is denoted by $\bar{N}_s$.

Below, for simplicity, we assume that there is an isomorphism between the grid and algebraic problem statements in the sense that to every node i there correspond an equation and the component u_i of the unknown solution vector u. The subvectors u_s and $\bar{u}_s$ of dimensions N_s and $\bar{N}_s$ consist of the components belonging to Ω_s and $\bar{\Omega}_s$, respectively.

In order to construct an iterative domain decomposition method in Krylov's subspaces, we define two types of preconditionings. The first of them can be characterized as the Restricted Additive Schwarz (RAS) method and is described as follows:

$$B_{RAS}^{-1} = R\widehat{A}^{-1}W, \;\; \widehat{A} = W^T A W = \text{ block-diag } \{\bar{A}_{s,s} \in \mathcal{R}^{\bar{N}_s,\bar{N}_s}\}. \tag{20}$$

Here, $W = [w_1 \dots w_P] \in \mathcal{R}^{N,P}$ is a rectangular matrix in which every column w_s has unit components at the nodes from $\bar{\Omega}_s$ and zero components elsewhere; the matrix $R \in \mathcal{R}^{N,N}$ consists of block rows $R_s \in \mathcal{R}^{N_s,N}$ each of which represents the restriction operator from Ω onto Ω_s, i.e., $R_s u = u_s$. Note that even in the case where the original coefficient matrix is symmetric, the preconditioner B_{RAS} defined in (20) is, in general, nonsymmetric. Inversion of the blocks $A_{s,s}$ of the matrix $\widehat{A}$ de facto reduces to solution of independent subsystems in the corresponding extended subdomains, which is the foundation of parallelizing the additive Schwarz, or the block Jacobi method.

The second preconditioner establishes "distant" connections among the subdomains and is defined as follows:

$$B_c^{-1} = \Phi \check{A}^{-1} \Phi^T, \quad \check{A} = \Phi^T A \Phi \in \mathcal{R}^{N_c,N_c}. \tag{21}$$

Here, $N_c \ll N$ is the dimension (the number of nodes) of a certain coarse grid; $\Phi = [\varphi_1 \dots \varphi_{N_c}] \in \mathcal{R}^{N,N_c}$ is a rectangular matrix the sth column of which, $s = 1, \dots, N_c$, is composed of the values of a certain basis function φ_s at the nodes of the original grid Ω. In what follows, for simplicity, we assume that $N_c = P$ and that the ith component of the column vector φ_s equals unity if the respective node belongs to the non-extended domain Ω_s and zero otherwise. From (21) it is seen that multiplication by the preconditioning matrix B_c^{-1}, to which the method of coarse-grid correction actually reduces, essentially consists of solving an auxiliary linear system with an "aggregated" matrix $\check{A}$ of small order, which establishes, at every iteration, interconnections among the subdomains. Obviously, if Φ is a matrix of full rank, which is assumed, and A is nonsingular, then $\check{A}$ also is nonsingular. The matrix $\check{A}$ (more exactly, the matrix $\Phi^T \check{A} \Phi \in \mathcal{R}^{N_c,N_c}$) is called the small-rank approximation of the original matrix A.

Introduce the notation

$$B_{RAS}^{-1} = B_{n,1}^{-1}, \quad B_c^{-1} = B_{n,2}^{-1}.$$

Then the preconditioners (20) and (21) can be used in the semi-conjugate direction methods (13)–(19). If the auxiliary linear systems with the matrices $\widehat{A}$ and $\check{A}$ are solved by direct methods, then the resulting preconditioners prove to be independent of the iteration number n. In the case where the auxiliary systems are solved by iterative methods (inner iterations), we arrive at dynamic, or flexible preconditioning.

Domain decomposition methods provide a foundation for various computational technologies of parallelizing algorithms. These issues are considered in a huge number of publications, see [23], and we do not dwell on them here.

4. On the concept of the KRYLOV library of algorithms

Theoretical estimation of the convergence rate of iterative methods for solving systems of linear algebraic equations and also a comparative analysis of the efficiencies of different algorithms, as well as elaboration of recommendations concerning the choice of the best algorithm for solving a specific linear system or a class of problems, are quite difficult problems. This is only more true for evaluation of real-life performance of program implementations of methods on heterogeneous multiprocessor computer systems with complicated hierarchies of the shared and distributed memories. Therefore, the development of new numerical methods is absolutely impossible without their experimental study, which requires that massive systematic computations be performed on a representative series of test and real-life problems. This labor and time consuming problem includes the development of specialized tools for automatically testing and verifying program implementations, including creation of collections of specific examples. It should be mentioned that in computational algebra, investigations along this direction have already been conducted for many years, and they have resulted in different collections of test matrices, e.g., MatrixMarket, Florida, and Boeing, which are widely used and available from the e-net.

Also it should be said that up to now, a lot of software related to solution of linear algebraic problems has been accumulated in the world, including, in particular, software for solving sparse linear algebraic systems, which exists either in the form of specialized program libraries or is included in applied systems, both commercial and open source. As examples, one can mention MATLAB, NETLIB, PETSc, Hypre, Trilinos, MKL, SparseKit, and so on. This list can be considerably extended. It is important to note that in the field in question, the main standards for matrix data structures have already been formed, and sufficiently representative collections of efficiently implemented matrix-vector operations can be found in the widely-used packages BLAS and SparseBlas.

In view of what has been said above, we will consider a concept of creating an integrated program environment including computational toolkits not only for efficient solution of algebraic problems on modern multiprocessor computer systems but also for prompt development of new algorithms. Some aspects of such an environment (the KRYLOV library) are presented in [22].

This superproject is motivated by the following circumstances. First, there is a very wide class of urgent problems of linear algebra, which is regularly replenished. On the one hand, this class can be characterized by matrix properties (matrices can be real and complex, Hermitian and non-Hermitian, symmetric and nonsymmetric, positive definite and indefinite and can possess a number of more specific properties). On the other hand, various linear algebraic systems result from old and new methods for solving differential and/or integral equations (Maxwell, Lamé, D'Arcy, Navier–Stokes, etc.), and their structural and spectral properties should be taken into consideration.

Another motivation is provided by a permanent and intensive development of new methods of computational linear algebra, for which efficient technologies of experimental investigations

is a separate fundamental problem (similarly, no progress in theoretical physics is possible without experimental physics). Finally, the third motivation is as follows. Applied software should be adapted to explosive evolution of supercomputer architectures and platforms. A characteristic requirement to such projects is the absence of program restrictions on the number of degrees of freedom in a problem to be solved and on the number of computer threads and/or cores used.

With account for what has been said above, we suggested a concept of the KRYLOV library as an integrated toolkit environment for solving a large class of problems of numerical linear algebra open for its development by coordinated efforts of different groups of scientists and allowing for various user interfaces and modes of use.

The functional content of the library is supposed to include the possibility of performing an automated, balanced geometric or algebraic problem decomposition, of using different types of iterative solvers in Krylov's subspaces with preconditionings of various kinds and different stopping criteria, and also programs for generating matrices from specific classes of problems.

Program implementation is based on unified data formats, with an option of their convertation and with forming MPI processes and multithread computations on heterogeneous clusters with distributed and shared memory of the CPU or GPGPU. Inner and outer interfaces are constructed based on the CCA (Common Component Architecture, [24]) technologies, which support multilingualism and cross-platform development and also the possibility of using exterior program products. We do not go into the details of this large project because it is a topic for a separate investigation.

5. Conclusion

This paper presents a brief overview of modern approaches to the urgent problem of solving large systems of linear algebraic equations and also some results obtained by the author. We have aimed at demonstrating the variety of existing algorithms, which are intensively developed both theoretically and from the implementation standpoint. It should be pointed out that different aspects of the directions considered are intimately associated parts of numerical algebra. As a whole, they form the science-technology chain from generation of an idea and theoretical study of methods to trial versions and experimental approbation, and to highly efficient program implementation of algorithms and their practical promotion.

This work was supported by the Russian Science Foundation (project No. 14-11-00485) and Russian Foundation for Basic Research (project No. 14-07-00128).

Translated by L. Yu. Kolotilina.

REFERENCES

1. S. C. Eisenstat, H. C. Elman, and M. H. Schultz, "Variational iterative methods for nonsymmetric systems of linear equations," *SIAM J. Numer. Anal.*, **20**, No. 3, 345–357 (1983).
2. J. Y. Yuan, G. H. Golub, R. J. Plemmons, and W. A. Cecilio, "Semi-conjugate direction methods for real positive definite systems," *BIT*, **44**, No. 1, 189–207 (2004).
3. V. P. Il'in and E. A. Itzkovich, "Semi-conjugate direction methods with dynamic preconditioning," *Sib. Zh. Industr. Mat.*, **10**, No. 4, 41–54 (2007).
4. A. Ruhe, "Rational Krylov sequence methods for eigenvalue computation," *Linear Algebra Appl.*, **58**, 391–405 (1984).

5. V. Druskin, L. Knizhnerman, and V. Simoncini, "Analysis of the rational Krylov subspace and ADI methods for solving the Lyapunov equation," *SIAM J. Numer. Anal.*, **49**, No. 5, 1875–1898 (2011).
6. M. van Gijzen and P. Sonneveld, "Algorithm 913: An elegant IDR(s) variant that efficiently exploits biorthogonality properties," *ACM Trans. Math. Software*, **38**, No. 1 (2011), Article 5.
7. M. B. van Gijzen, G. L. G. Sleijpen, and J.-P. M. Zemke, "Flexible and multi-shift induced dimension reduction algorithms for solving large sparse linear systems," *Numer. Linear Algebra Appl.*, **22**, 1–25 (2015).
8. A. Chapman and Y. Saad, "Deflated and augmented Krylov subspace techniques," *Numer. Linear Algebra Appl.*, **4**, No. 1, 43–66 (1997).
9. Y. L. Gurieva and V. P. Il'in, "Some parallel methods and technologies of domain decomposition," *Zap. Nauchn. Semin. POMI*, **428**, 89–106 (2014).
10. I. V. Oseledets and E. E. Tyrtyshnikov, "Breaking the curse of dimensionality. Or how to use SVD in many dimensions," *SIAM J. Sci. Comput.*, **31**, Iss. 5, 3744–3759 (2009).
11. W. Hackbusch, *Hierarchische Matrizen: Algorithmen and Analysis*, Springer (2009).
12. R. Bridson and C. Greif, "A multipreconditioned conjugate gradient algorithm," *SIAM J. Matrix Anal. Appl.*, **27**, No. 4, 1056–1068 (2006).
13. C. Greif, T. Rees, and D. B. Szyld, *MPGMRES: a generalized minimum residual method with multiple preconditioners.* Tech. Rep. 11-12-23, Department of Mathematics, Temple University (2011).
14. C. Greif, T. Rees, and D. B. Szyld, "Additive Schwarz with variable weights," *Lect. Notes Comput. Sci. Eng.*, **98**, 779–787 (2014).
15. Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Society for Industrial and Applied Mathematics (2003).
16. V. Dolean, P. Jolivet, and F. Nataf, "An introduction to domain decomposition methods: algorithms, theory and parallel implementation," `https://hal.archives-ouvertes.fr/cel-01100932.v4`
17. A. Toselli and O. Widlund, *Domain Decomposition Methods – Algorithms and Theory* (Springer Ser. Comput. Math., **34**), Springer (2005).
18. 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 (1995).
19. L. Yu. Kolotilina, "Eigenvalue bounds and inequalities using vector aggregation of matrices," *Linear Algebra Appl.*, **271**, 139–167 (1998).
20. O. Dubois, M. J. Gander, A. St.-Cyr, S. Loisel, and D. Szyld, "The optimized Schwarz method with a coarse grid correction," *SIAM J. Sci. Comp.*, **34**, No. 1, 421–458 (2012).
21. Y. Efendiev, J. Galvis, R. Lazarov, and J. Willems, "Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms," *Esaim Math. Model. Numer. Anal.*, **46**, No. 5, 1175-1199 (2012).
22. D. S. Butyugin, Y. L. Gurieva, V. P. Il'in, D. V. Perevozkin, A. V. Petukhov, and I. N. Skopin, "Functionality and technologies of algebraic solvers in the KRYLOV library," *Vestn. YuUrGU*, **2**, No. 3, 92–105 (2013).
23. `URL: http://www.ddm.org`
24. `CCA: The Common Component Architecture Forum. www.cca-forum.org/.`