On the parallel least square approaches in the Krylov subspaces

V.P.Il'in

[1)] *Institute of Computational Mathematics and Mathematical Geophysics RAS, 6, pr. Lavrentieva, 630090, Novosibirsk, Russia,*

[2)] *Novosibirsk State University, e-mail:ilin@sscc.ru*

Abstract. We consider different parallel versions of the least squares methods in the Krylov subspaces which are based on computing various basis vectors. These algorithms are used for solving very large real, non-symmetric, in gerenal, sparse systems of linear algebraic equations (SLAEs) which arise in grid approximations of multi-dimensional boundary value problems. In particular, the Chebyshev acceleration approach, steepest descent and minimal residual, conjugate gradient and conjugate residual are applied as preliminary iterative processes. The resulting minimization of residuals is provided by the block, or implicit, orthogonalization procedures. The properties of the Krylov approaches proposed are analysed in the "pure form", i.e. without preconditioning. The main criteria of parallelezation are estimated. The convergence rate and stability of the algorithms are demonstated on the results of numerical experiments for the model SLAEs which present the exponential fitting approximation of diffusion-convection equations on the meshes with various steps and with different coefficients.

Keywords: Large sparse systems of linear algebraic equations, non-symmetric matrices, block implicit least squares methods, Krylov subspaces, parallel technologies, numerical experiments.

1 Introduction

The mathematical modeling in real extremal interdisciplinary problems includes the solution of the multi-dimensional direct and inverse tasks, linear and nonlinear, stationary and non-stationary, which are approximated by various order numerical schemes on the non-structured grids in the complicated computational domains. In any case, at a low level of these procedures, the multi-fold solution to the systems of linear algebraic equations (SLAEs) is required. The practical high resolution demands very large degrees of freedom (dof). So, the solution of the corresponding ill-conditioned SLAEs is the bottle-neck of the general numerical process, because necessary computational resources grow nonlinearly at this stage if the dimension of the system increases (for example, 10^{10} and higher).

In this case the road map to provide a high performance consists in parallel implementation of modern multi-preconditioned iterative processes in the Krylov subspaces based on the domain decomposition methods (DDM) (see [1], [2] and the references therein). The main achivements are based on the combination of efficient mathematical discoveries and scalable parallel technologies on the multi-processor systems (MPS) with distributed and hierarchical shared memory.

This paper deals with just one particular side of the general problem. Namely, we consider the possibility of parallel "implicit" construction of the iterative methods in the Krylov subspace "in the pure form", i.e. without preconditioning, which is supposed to be a separate problem.

Let us consider the solution of the SLAE

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

with a large real sparse matrix resulting from grid approximations of multi-dimensional boundary value problems by finite element, finite volume, or other methods. In general, this matrix is non-symmetric and ill-conditioned. In equation (1), ω_ℓ denotes a set of indices of nonzero entries in the ℓ-th 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.

In [3], the authors have offered 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 acceleration, which had been originally proposed in [4] for solving systems of nonlinear algebraic equations, A comparative experimental analysis presented in [3] 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 in periodical (after a prescribed number of stationary iterations) use of an acceleration method based on solving an auxilary least squares problem not involving successive orthogonalization of the direction vectors, which is typical of the Krylov variational type methods.

The present paper aims at generalization and experimental study of the similar approaches. We apply several non-stationary iterative algorithms as a preliminary tool for constructing some basis vectors in the Krylov subspaces and further minimization of the residual vector norm by means of the least squares method. In this context, parallel implementation of the approaches proposed is considered.

This paper is organized as follows. In Section 2, we present the idea of implicit, or block, least squares method in the Krylov subspaces which uses a preliminary consruction of the basis vectors. Section 3 is devoted to analyzing the efficiency of parallel versions of the iterative algorithms considered in comparison with the classical variational method of semi-conjugate residuals in the Krylov subspaces. Section 4 discusses the results of numerical experiments obtained for the algorithms offered on a series of the test SLAEs, resulting from the grid approximation of two-dimensional boundary value problems for the convection-diffusion equation. In conclusion, we observe the efficiency of the algorithms presented and discuss some plans for future studies.

2 Versions of the least squares methods in the Krylov subspaces

The wide class of iterative processes for solving SLAE (1) can be written in the form

$$\begin{aligned} &u^{n+1} = u^n + \alpha_n p^n = u^0 + \alpha_0 p^0 + ... + \alpha_n p^n, \\ &r^{n+1} = r^n - \alpha_n A p^n = r^0 + \alpha_0 A p^0 + ... + \alpha_n A p^n. \end{aligned} \tag{2}$$

Here u^0 and $r^0 = f - Au^0$ are the initial guess and the corresponding residual vector, and p^n, α^n are some direction vectors (usualy $p^0 = r^0$) and the iterative parameters which are defined from the additional conditions in the different approaches.

If A is a symmetric positive definite (spd) matrix, then the following conjugate direction (CD) methods [1], [5]:

$$p^{n+1} = r^{n+1} + \beta_n^{(s)} p^n,$$
$$\alpha_n^{(s)} = \frac{(A^s r^n, r^n)}{(Ap^n, A^s p^n)}, \;\; \beta_n^{(s)} = \frac{(A^s r^{n+1}, r^{n+1})}{(A^s r^n, A^s r^n)}, \tag{3}$$

for $s = 0, 1$ present the classical conjugate gradient (CG) and conjugate residual (CR) algorithms, respectively, which minimize the functionals $\Phi_n^{(s)}(r^0) = (A^{-s} r^{n+1}, r^{n+1})$ in the Krylov subspaces

$$\mathcal{K}_n(r^0, A) = \text{span } (r^0, Ar^0, ..., A^n r^0). \tag{4}$$

The residual and direction vectors in these approaches for all k, n satisfy the orthogonal properties

$$(A^s r^k, r^n) = (A^s r^n, r^n)\delta_{k,n}, \;\; (A^s p^k, Ap^n) = (A^s p^n, Ap^n)\delta_{k,n} \tag{5}$$

where $\delta_{k,n}$ is the Kronecker symbol.

However, if A is a non-symmetric matrix, then these methods have no such variational and orthogonal properties. In such cases, the global minimization of the functionals $\Phi_n^{(s)}$ is provided by the general minimized residual type (GMRES) approaches or by the equivalent, in some sense, semi-conjugate direction(SCD) methods [6]

$$p^{n+1} = r^{n+1} - \sum_{k=0}^{n} \beta_{n,k}^{(s)} p^k, \;\; \beta_{n,k}^{(s)} = (Ap^k, A^s r^{n+1})/(Ap^n, A^s p^n). \tag{6}$$

Let us remark that the formulas (6) realize the orthogonal properties (5) by Gram–Schmidt procedure. It fact, this procedure should be changed by more stable modified Gram–Schmidt (MGS) orthogonalization [7]. If $\alpha_n^{(s)}$ are defined by (3) then for $s = 0, 1$ from (6) we provide the extremum conditions

$$\partial \Phi_n^{(s)} / \partial \alpha_n = 0, \;\; \Phi_n^{(s)} = (r^{n+1}, A^{s-1} r^{n+1}), \tag{7}$$

and for $s = 1$ the functional $\Phi_n^{(s)}$ has the minimum in the Krylov subspace (4).

In this case the resulting residual vectors are not conjugate, but semi-conjugate only, i.e.,

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

and for $s = 0, 1$ we have a semi-conjugate gradient and a semi-conjugate residual (SCG and SCR) methods, respectively.

Let us remark, that for spd - matrix A, the CD methods (both CG and CR), as well as SCD approaches (SCG and SCR) have the same theoretical number of iterations, see [1], [5]:

$$n(\varepsilon) \approx 0.5 |\ell n(\varepsilon/2)| (cond A)^{-1/2},$$

where $cond A$ is the condition number of A and $\Phi_n^{(s)} \le \varepsilon^2 \Phi_{n-1}^{(s)}$, $0 < \varepsilon \ll 1$. But if A is non-symmetric, the same estimate is valid for SCD but not for CD methods.

In the general case, to compute the vectors u^n and r^n using (2)–(6), it is necessary to store all the vectors $p^n, p^{n-1}, ..., p^0$ and $Ap^n, Ap^{n-1}, ..., Ap^0$. In practice, these methods are realized with periodic restarts every m iteration. This means that the residual vector is computed from the original equation

$$r^{ml} = f - Au^{ml}, \;\; \ell = 0, 1, ..., \tag{8}$$

rather than using (2), 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 $p^n, p^{n-1}, ..., p^{n-m}$, and $Ap^n, Ap^{n-1}, ..., Ap^{n-m}$. The restarted versions of SCD methods, similar to restarded GMRES, have lower convergence rate, but this is the cost for the memory saved.

The most expensive stage of the SCD methods consists in successive computations of the direction vectors p^{n+1} by means of long recursions (6). In accord with the Anderson acceleration approach, we can simplify (6) and use in the sum the last direction vector p^n only (but save the vectors $p^n, ..., p^{n-m}$ and $Ap^n, ..., Ap^{n-m}$). In these cases, the minimization of the residual norm $||r^{n+1}||_2 = (r^{n+1}, r^{n+1})^{1/2}$ in the Krylov subspace

$$K_{n,m}(r^n, A) = \text{ span } (r^n, Ar^n, ..., A^m r^n) \tag{9}$$

can be provided by the following least squares method:

$$\begin{array}{l} r^{n+m} = r^n - W_{n,m}\bar{\gamma}_{n,m} \approx 0, \;\; W_{n,m} = (w_n w_{n+1} ... w_{n+m}) \in \mathcal{R}^{N,m+1}, \\ w_{n+k} = A^k p^n, \;\; \bar{\gamma}_{n,m} = (\gamma_n, \gamma_{n+1}, ..., \gamma_{n+m})^T \in \mathcal{R}^{m+1}. \end{array} \tag{10}$$

The coefficient vector $\bar{\gamma}_{n,m}$ can be computed from the over-determined SLAE

$$W_{n,m}\bar{\gamma}_{n,m} = r^n, \tag{11}$$

which can be solved, for example, by means of the singular value decomposition (SVD) or an other approach (see [7]). In particular, the left-hand Gauss transformation procedure

$$B_{n,m}\bar{\gamma}_{n,m} = g_{n,m}, \;\; B_{n,m} = W_{n,m}^T W_{n,m} \in \mathcal{R}^{m+1,m+1}, \;\; g_{n,m} = W_{n,m} r^n \in \mathcal{R}^{m+1} \tag{12}$$

can be here efficiently applied.

In fact, the computing vectors p^k, Ap^k in such algorithms can be realized by formulas (2), (3), and we call them CD-LSM-ℓ (CG-LSM-ℓ and CR-LSM-ℓ for $s = 0, 1$, respectively) where the integer $\ell = 1, 2$ corresponds to application of formulas (11) or (12).

If the coefficent vector $\bar{\gamma}_{n,m}$ is known, the improved numerical solution can be computed by the formulae

$$u^{n+m} = u^n + \gamma_n p^n + ... + \gamma_{n+m} p^{n+m}. \tag{13}$$

The considered algorithms can be simplified even to a greater extent if we use instead CG or CR method, the two-terms formulas of the steapest descent (SD) or the minimal residual (MR) method, which can be formaly described (for $s = 0, 1$ respectively) as follows, see [1], [5]:

$$\alpha_n^{(s)} = (A^s r^n, r^n)/(Ar^n, A^s r^n), \;\; \beta^n = 0, \;\; p^n = r^n. \tag{14}$$

For the spd-matrices, these approaches provide the local variational properties only, i.e. for just one iteration, but minimization of the functional $\Phi_{n,m}^{(s)} = (A^{s-1} r^{n+m}, r^{n+m})$ in the Krylov subspaces $\mathcal{K}_{n,m}(r^n, A)$ can be achieved by the LSM-ℓ approaches (11) or (12). Such methods will be called SD-LSM-ℓ and MR-LSM-ℓ, $\ell = 1, 2$. Of course, for SD and MR methods with local variational properties, the convergence rates of iterations are worse as compared to the previous algorithms ($n(\varepsilon) \sim condA$ only), but let us remind that it is just the way to obtain the basis vector for LSM optimization.

In all the approaches considered above, we use the least squares methods, based on the direction vectors p^n with weak orthogonal, or variational, properties. Instead of this, we can construct the basis vectors by application of the some spectral iterative process. If the matrix A has real positive eigenvalues $\lambda \in [0 < \lambda_1, \lambda_N]$, then the optimal convergence rate of iterations is provided by the

Chebyshev acceleration [1], [5], [8]. Such approaches can be implemented in different forms, and we use the two-terms recurrent representation, which consists of the following relations:

$$\begin{aligned} &p^0 = r^0 = f - Au^0, \\ &u^n = u^{n-1} + \alpha_{n-1} p^{n-1}, \\ &r^n = r^{n-1} - \alpha_{n-1} A p^{n-1}, \\ &p^n = r^n + \beta_n p^{n-1}. \end{aligned} \tag{15}$$

Here we use the restarted procedures which also suppose applying the LSM approaches by (11) or (12) after each m iteration. The coefficients in (15) are defined via three terms description of the Chebyshev acceleration presented in [8]:

$$\begin{aligned} &u^1 = u^0 + \tau\, r^0, \;\; \tau = 2/(\lambda_1 + \lambda_N), \;\; r^n = f - Au^n, \\ &u^{n+1} = u^n + \tau_n \tau\, r^n + (\tau_n - 1)(u^n - u^{n-1}), \;\; \tau_0 = 2, \\ &\tau_n = 4(4 - \tau_{n-1}\gamma)^2) - 1, \;\; \gamma = (1-c)/(1+c), \;\; c = \lambda_1/\lambda_N. \end{aligned} \tag{16}$$

The values of α_n, β_n from (15) provide the equivalence to reccurences (16) by the formulas

$$\alpha_0 = \tau, \;\; \alpha_n = \tau_n \tau, \;\; \beta_n = (\tau_n - 1)\alpha_{n-1}/\alpha_n. \tag{17}$$

After each m iterations by formulas (15)–(17) we can apply the acceleration procedures according to (10)–(13). The corresponding algorithms we will call the Chebyshev least squares methods (CHEB-LSM-1 and CHEB-LSM-2). We conclude this section with the following two remarks. First, it is easy to check that from theoretical viewpoint, LSM-1 and LSM-2 coincide because, in exact arithmetic, by solving equations (11) and (12) one obtains one and the same vector $\bar{\gamma}_{n,m}$. Second, an approach similar to the one considered above was applied by P.L.Montgomery in [9] (see [10] also) in solving special systems of linear algebraic equations over a finite field and was referred to as the block Lanczos method.

3 Properties of parallel implementation

As is seen, the implementation of the optimal SCR method includes at each iteration the following main stages:

- one matrix-vector multiplications (MV-operations);
- $2m + 3$ vector-vector (VV) operations, i.e. linear combinations of the vectors;
- computing the $m + 2$ inner vector products.

It is important that all these operations are fulfilled successively. The idea of parallel implementation of the methods proposed with LSM-2 approaches consists in the simultaneous computation of the entries of the matrix

$$B_{n,m} = \{b_{k,\ell}^{(n,m)} = (w_k, w_\ell);\; k, \ell = n, ..., n-m\}.$$

And for $m \ll N$, we can neglect the costs for solving SLAEs (12) and compute the vector $\bar{\gamma}_{n,m}$ by formula (13) on the all processor units simultaneously.

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 (1), and the block rows $A_k = \{A_{k,\ell},\ \ell = 1, .., P\} \in \mathcal{R}_{N_k,N},\ N_k \cong N/P,\ N_1 + ... + N_p = N$ 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 [11]). 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 $p^n, p^{n-1}, ..., p^{n-m}$ and also the current vectors u^n and r^n are partitioned into subvectors of lengths N_k, each being stored in the corresponding k-th 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 k-th MPI process using a multicore processor, "inner" parallelization (of the second level) can be effected based on multi-thread 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 $w_k,\ k = 1, ..., m$. We assume that in all the algorithms the standard double-precision computer arithmetics 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 simple model of the computation process:

$$T_P = T_P^a + T_P^c \approx \tau_a N_a + (\tau_0 + \tau_c V_c) N_c. \tag{18}$$

Here, T_P^a and T_P^c are the times for performing arithmetic and communication operations, respectivery; 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.

It is easy to check that in CG-LSM-2 or CR-LSM-2 for $n \neq m$ we need to compute by formulas (2), (3) just 2 inner products and 3 VV-operations. And if we use SD or MR approaches by (2), (14) with local variational properties, then we must perform 2 inner products and 2 vector linear combination, i.e. the difference is not significant as compared with CG or CR methods.

Let us now consider the combination of the Chebyshev acceleration (15)-(17) and the LSM approach. These algorithms do possess orthogonal or variational properties, but have the same optimal estimation of $n(\varepsilon)$. And what is important: the specrtal itarations do not need computation of inner products!

The last circumstance is highly valuable in terms of the implementation of the iterative process at the MPS, because these operations obviously need data communications. But this approach demands the knowledge of the spectrum boundaries of the matrix. Of course, this is too strong requirement, but in many practical problems the necessary estimations can be obtained.

It should be remarked that the implementation of the LSM with different preliminary iterative approaches does not need the computation of the vectors u^n, because at the end of any algorithms considered, the resulting vector is realized by (13). Of course, this operation can also be parallelized efficiently.

4 Discussion of numerical experiments

Let us consider the Dirichlet problem [8] for the convection-diffusion equation

$$\begin{aligned} &-\frac{\partial^2 u}{\partial x^2}-\frac{\partial^2 u}{\partial y^2}+p\frac{\partial u}{\partial x}+q\frac{\partial u}{\partial y}=f(x,y),\ \ (x,y)\in\Omega,\\ &u|_\Gamma=g(x,y), \end{aligned} \tag{19}$$

in a square computational domain $\Omega=(0,1)^2$ with the boundary Γ and the convection coefficients p,q, which for simplicity are assumed to be constant. This boundary value problem is approximated on a square grid with the step size $h=1/(L+1)$ and the total number of interior nodes $N=L^2$,

$$x_i=i_h,\ \ y_j=j_h,\ \ i,j=0,1,...,L+1, \tag{20}$$

using the five-point finite-volume monotone approximations of exponential type [12]

$$(Au)_l=a_{l,l}u_l+a_{l,l-1}u_{l-1}+a_{l,l+1}u_{l+1}+a_{l,l-L}u_{l-L}+a_{l,l+L}u_{l+L}=f_l, \tag{21}$$

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 equations (20) may be different, and we use the following ones:

$$\begin{aligned} &a_{l,l\pm1}=e^{\pm ph/2}/h,\ \ a_{l,l\pm L}=e^{\pm qh/2}/h,\\ &a_{l,l}=a_{l,l-1}+a_{l,l-L}+a_{l,l+1}+a_{l,l+L}. \end{aligned} \tag{22}$$

Equations (21) 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 $D=$ diag $\{a_\ell,\ell\}$:

$$\begin{aligned} &D^{-1/2}AD^{-1/2}D^{1/2}u=D^{-1/2}f,\\ &\bar{A}\bar{u}=\bar{f},\ \ \bar{A}=D^{-1/2}AD^{-1/2},\ \ \bar{u}=D^{1/2}u,\ \ \bar{f}=D^{-1/2}f. \end{aligned} \tag{23}$$

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 problem (19). 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 the restart parameter $m=8,16,32,64$, and 128. In the tables below, we present the results obtained in solving problem (19) with the convection coefficients $p=q=0$ and $p=q=4$ on the grids with $N=7^2,15^2,31^2,63^2,127^2$ nodes 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 (to be exact, the systems obtained in LSM-1 and LSM-2 have been solved using the SVD program (the singular value decomposition algorithm) from LAPACK, included into the program library MKL Intel [13]). Let us remark that the matrix $B_{n,m}$ from SLAE (12), which corresponds to LSM-2, has a bigger condition number, as compared to the matrix $W_{n,m}$ from (11). So, LSM-1 is more preferable,

from the stability point of view. But in our experiments, the resulting errors are approximately equal as for LSM-1 and LSM-2. So, in the following tables we present the numerical results for LSM-2 only.

The main goal of our experimental research consists not in demonstration of the high performance of algorithms for very large SLAEs, but in study of the stability and convergence rate of LSM approaches with preliminary cheap iterative processes. All the calculations have been carried out on the Siberian Super Computing Center cluster (http://www2.sscc.ru).

In the each cell of the following tables we present two values: the upper is the number of iterations, and the lower is the resulting maximal error $\delta = \max\limits_{i,j}\{|1 - u^n_{i,j}|\}$. In our experiments the results are approximately the same for different initial guesses, and we present data for $u^0 = x^2 + y^2$ only.

In the Tables 1, 2 we give the results for CHEB-LSM-2 algorithm for symmetric and non-symmetric SLAEs. In both cases the boundaries λ_1, λ_N of matrix spectrum in formylas (16), (17) were taken for $p = q = 0$, but the presented results are close to each other enough. The columns with $m = \infty$ correspond to "pure" Chebyshev acceleration without LSM. It is evident from these tables, that in all cases considered there is an optimal value m.

Table 1: CHEB-LSM-2, $p = q = 0$, $u^0 = x^2 + y^2$

$N \setminus m$	8	16	32	64	128	∞
7^2	34 $2.4 \cdot 10^{-7}$	29 $7.8 \cdot 10^{-8}$	32 $9.9 \cdot 10^{-16}$	41 $1.3 \cdot 10^{-7}$	41 $1.3 \cdot 10^{-7}$	41 $1.3 \cdot 10^{-7}$
15^2	90 $1.2 \cdot 10^{-6}$	75 $5.3 \cdot 10^{-7}$	63 $5.3 \cdot 10^{-8}$	64 $5.9 \cdot 10^{-9}$	82 $2.0 \cdot 10^{-7}$	82 $2.0 \cdot 10^{-7}$
31^2	281 $3.6 \cdot 10^{-6}$	197 $3.5 \cdot 10^{-6}$	140 $1.3 \cdot 10^{-6}$	127 $1.6 \cdot 10^{-7}$	128 $3.1 \cdot 10^{-8}$	163 $3.0 \cdot 10^{-7}$
63^2	960 $1.0 \cdot 10^{-5}$	586 $1.0 \cdot 10^{-5}$	390 $9.6 \cdot 10^{-6}$	267 $6.8 \cdot 10^{-6}$	251 $2.3 \cdot 10^{-6}$	327 $3.1 \cdot 10^{-7}$
127^2	3429 $2.9 \cdot 10^{-5}$	1991 $2.9 \cdot 10^{-5}$	1148 $2.9 \cdot 10^{-5}$	734 $2.7 \cdot 10^{-5}$	528 $2.2 \cdot 10^{-5}$	653 $3.5 \cdot 10^{-7}$

Table 2: CHEB-LSM-2, $p = q = 4$, $u^0 = x^2 + y^2$

$N \setminus m$	8	16	32	64	128	∞
7^2	34 $7.5 \cdot 10^{-8}$	31 $2.6 \cdot 10^{-8}$	32 $4.6 \cdot 10^{-15}$	45 $8.2 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$	45 $8.2 \cdot 10^{-8}$
15^2	67 $5.0 \cdot 10^{-7}$	75 $2.6 \cdot 10^{-7}$	71 $3.4 \cdot 10^{-7}$	64 $9.8 \cdot 10^{-9}$	91 $1.6 \cdot 10^{-7}$	91 $1.6 \cdot 10^{-7}$
31^2	210 $2.9 \cdot 10^{-6}$	158 $3.4 \cdot 10^{-7}$	142 $1.3 \cdot 10^{-6}$	149 $8.6 \cdot 10^{-7}$	128 $4.3 \cdot 10^{-8}$	184 $2.2 \cdot 10^{-7}$
63^2	740 $7.9 \cdot 10^{-6}$	421 $6.6 \cdot 10^{-6}$	348 $3.7 \cdot 10^{-6}$	285 $3.6 \cdot 10^{-6}$	271 $2.7 \cdot 10^{-6}$	363 $1.8 \cdot 10^{-7}$
127^2	2654 $2.4 \cdot 10^{-5}$	1531 $2.3 \cdot 10^{-5}$	884 $2.1 \cdot 10^{-5}$	662 $1.8 \cdot 10^{-6}$	543 $7.6 \cdot 10^{-6}$	719 $1.7 \cdot 10^{-7}$

The Tables 3, 4 demonstrate the similar results for CR-LSM-2 algorithm. The symmetric case

($p = q = 0$) show that conjugate residual is optimal for such SLAEs, and least squares approach is not resonable here. But for non-symmetric algebraic systems the application of LSM gives the considerable improvement of the iterative process. Let us remark, that the resulting numbers of iteration and errors δ are approximatelly the same in CR and CHEB.

Table 3: CR-LSM-2, $p = q = 4$

$N \setminus m$	8	16	32	64	128
7^2	34 $1.3 \cdot 10^{-7}$	31 $7.9 \cdot 10^{-8}$	63 $1.6 \cdot 10^{-9}$	127 $4.6 \cdot 10^{-12}$	255 $5.1 \cdot 10^{-13}$
15^2	74 $8.8 \cdot 10^{-7}$	64 $9.7 \cdot 10^{-7}$	94 $2.2 \cdot 10^{-7}$	127 $3.1 \cdot 10^{-9}$	255 $3.9 \cdot 10^{-12}$
31^2	236 $2.9 \cdot 10^{-6}$	149 $2.0 \cdot 10^{-6}$	129 $8.1 \cdot 10^{-7}$	190 $3.0 \cdot 10^{-8}$	255 $7.3 \cdot 10^{-7}$
63^2	592 $8.1 \cdot 10^{-6}$	472 $8.0 \cdot 10^{-6}$	305 $4.3 \cdot 10^{-6}$	331 $4.6 \cdot 10^{-6}$	382 $1.9 \cdot 10^{-7}$
127^2	2612 $2.4 \cdot 10^{-5}$	1347 $2.3 \cdot 10^{-5}$	897 $2.1 \cdot 10^{-5}$	539 $1.3 \cdot 10^{-5}$	659 $1.2 \cdot 10^{-5}$

Table 4: CR-LSM-2, $p = q = 0$, $u^0 = x^2 + y^2$

$N \setminus m$	8	16	32	64	128
7^2	37 $2.7 \cdot 10^{-7}$	20 $4.2 \cdot 10^{-8}$	20 $5.4 \cdot 10^{-9}$	20 $5.4 \cdot 10^{-9}$	20 $5.4 \cdot 10^{-9}$
15^2	99 $6.9 \cdot 10^{-7}$	75 $8.3 \cdot 10^{-7}$	42 $3.1 \cdot 10^{-7}$	40 $8.8 \cdot 10^{-8}$	40 $8.8 \cdot 10^{-8}$
31^2	314 $3.6 \cdot 10^{-6}$	199 $304 \cdot 10^{-6}$	145 $2.6 \cdot 10^{-6}$	83 $1.2 \cdot 10^{-6}$	83 $2.4 \cdot 10^{-7}$
63^2	1084 $1.0 \cdot 10^{-5}$	626 $1.0 \cdot 10^{-5}$	390 $9.2 \cdot 10^{-6}$	283 $8.4 \cdot 10^{-6}$	160 $2.6 \cdot 10^{-6}$
127^2	3860 $2.9 \cdot 10^{-5}$	2119 $2.9 \cdot 10^{-5}$	1185 $2.8 \cdot 10^{-5}$	746 $2.8 \cdot 10^{-5}$	538 $2.1 \cdot 10^{-5}$

In the Tables 5, 6, we present the results for CG-LSM, which confirm that the efficiency of conjugate residual method, in combination with the least squares approach is approximately the same that of CR algorithm.

Table 5: CG-LSM-2, $p = q = 0$, $u^0 = x^2 + y^2$

$N \setminus m$	8	16	32	64	128
7^2	38 $1.8 \cdot 10^{-7}$	20 $4.7 \cdot 10^{-8}$	20 $5.4 \cdot 10^{-9}$	20 $5.4 \cdot 10^{-9}$	20 $5.4 \cdot 10^{-9}$
15^2	99 $5.4 \cdot 10^{-7}$	76 $5.5 \cdot 10^{-7}$	43 $1.5 \cdot 10^{-7}$	41 $2.6 \cdot 10^{-8}$	41 $2.6 \cdot 10^{-8}$
31^2	316 $3.0 \cdot 10^{-6}$	211 $1.3 \cdot 10^{-6}$	156 $9.9 \cdot 10^{-7}$	86 $5.7 \cdot 10^{-7}$	81 $1.7 \cdot 10^{-7}$
63^2	1086 $9.8 \cdot 10^{-6}$	631 $8.6 \cdot 10^{-6}$	404 $5.1 \cdot 10^{-6}$	316 $2.3 \cdot 10^{-6}$	167 $1.2 \cdot 10^{-6}$
127^2	3865 $2.9 \cdot 10^{-5}$	2131 $2.7 \cdot 10^{-5}$	1210 $2.1 \cdot 10^{-5}$	757 $2.2 \cdot 10^{-5}$	614 $3.6 \cdot 10^{-6}$

Table 6: CG-LSM-2, $p = q = 4$, $u^0 = x^2 + y^2$

$N \setminus m$	8	16	32	64	128
7^2	34 $1.5 \cdot 10^{-7}$	31 $1.6 \cdot 10^{-8}$	63 $2.5 \cdot 10^{-10}$	127 $3.8 \cdot 10^{-12}$	255 $2.1 \cdot 10^{-13}$
15^2	78 $1.3 \cdot 10^{-7}$	69 $2.2 \cdot 10^{-7}$	94 $2.2 \cdot 10^{-8}$	127 $1.1 \cdot 10^{-9}$	455 $6.6 \cdot 10^{-11}$
31^2	239 $2.1 \cdot 10^{-6}$	151 $1.5 \cdot 10^{-6}$	156 $7.5 \cdot 10^{-7}$	190 $1.9 \cdot 10^{-8}$	255 $5.2 \cdot 10^{-8}$
63^2	596 $7.8 \cdot 10^{-6}$	481 $5.8 \cdot 10^{-6}$	311 $1.7 \cdot 10^{-6}$	337 $7.1 \cdot 10^{-7}$	382 $9.9 \cdot 10^{-8}$
127^2	2612 $2.4 \cdot 10^{-5}$	1351 $1.9 \cdot 10^{-5}$	900 $1.9 \cdot 10^{-5}$	568 $2.9 \cdot 10^{-6}$	736 $3.9 \cdot 10^{-6}$

Table 7: MR-LSM-2, $p = q = 0, u^0 = x^2 + y^2$

$N \setminus m$	8	16	32	64	128	∞
7^2	37 $2.7 \cdot 10^{-7}$	21 $5.5 \cdot 10^{-8}$	32 $7.0 \cdot 10^{-9}$	64 $2.8 \cdot 10^{-9}$	128 $2.8 \cdot 10^{-9}$	185 $4.8 \cdot 10^{-7}$
15^2	99 $5.4 \cdot 10^{-7}$	76 $5.5 \cdot 10^{-7}$	67 $4.3 \cdot 10^{-7}$	82 $2.1 \cdot 10^{-7}$	128 $2.6 \cdot 10^{-8}$	703 $1.3 \cdot 10^{-6}$
31^2	316 $3.0 \cdot 10^{-6}$	202 $3.1 \cdot 10^{-6}$	187 $1.3 \cdot 10^{-6}$	253 $1.9 \cdot 10^{-7}$	267 $1.4 \cdot 10^{-6}$	2614 $3.7 \cdot 10^{-6}$
63^2	1086 $9.8 \cdot 10^{-6}$	631 $8.6 \cdot 10^{-6}$	559 $6.1 \cdot 10^{-6}$	505 $2.1 \cdot 10^{-6}$	636 $4.8 \cdot 10^{-6}$	9622 $1.0 \cdot 10^{-5}$
127^2	3860 $2.9 \cdot 10^{-5}$	2123 $2.9 \cdot 10^{-5}$	1427 $2.6 \cdot 10^{-5}$	1702 $2.2 \cdot 10^{-5}$	1906 $1.7 \cdot 10^{-5}$	35050 $2.9 \cdot 10^{-5}$

At last, in the Table 7 we give the similar results for the minimal residual method with local variational properties. This approach presents a big disadvantage in efficiency, as compared to the previous algorithms, even with application of the least squares methods. The close effect is demonstrated for steepest decent (SD) method, both for symmetric and non-symmetric matrices.

5 Conclusion

We consider the generalization of Anderson acceleration, for parallel solving non-symmetric large SLAEs with sparse matrices, on the base of least squares methods applied to some preliminary "cheap" iterative process, which is used just for computing basis vectors for implicit, or block, implementation of the Krylov type algorithms with periodicaly minimization of the residual vector before restarts. The comparative experimental analysis of the variational conjugate gradient and conjugate residual methods, as well as spectral Chebyshev acceleration demonstrates reasonable stability and convergence rate of the iterations the methods proposed. The idea of increasing parallelism consists in the simultaneous computations of big number of inner products, in contrast to successive computations in the conventional Krykov algorithms. The perfomance of the proposed approaches at the real multiprocessor systems with distributed and hierarchical shared memory is the topic of futher research.

This work was supported by the Russian Science Foundation (project N 14-11-00485) and the Russian Foundation for Basic Research (project N 16-29-15122).

References

1. Saad Y. Iterative Methods for Sparse Linear Systems. N.Y.:PWS Publ. (2002)

2. Il'in V.P. Problems of parallel solution of large systems of linear algebraic equations. J. of Mathem. Sci, v. 216, N 6, 795-804 (2016)

3. Pratara P.P., Suryanarayana P., Pask J.E. 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)

4. Anderson D.G. Itaerative Procedures for Nonlinear Integral Equations. – J.Assoc. Comput. Mach., 12, 547-560 (1965)

5. Il'in V.P. Methods and Technologies of Finite Elements (in Russian), ICM&MG SB RAS Publ. Novosibirsk (2007)

6. Il'in V.P. Methods of semiconjugate directions. Russ. J. Numer. Anal.Math. Modelling, v.23 N 4, 369-387 (2008)

7. Lawson C.L., Hanson R.Z. Solving Least Squares Problems. Prentice-Hall, Inc., Englewood Cliffs, N.J., (1974)

8. Samarskii A.A., Nikolaev E.S. Mehtod for Solving Grid Equations (in Russian), Nauka, Moscow (1978)

9. Montgomery P.L. A Block Lonczos Algorithm for Finding Dependences over GF(2).– Proceed. EUROCRYPT 95 Confer., LNCS, v. 921, 106-120 (1995)

10. Zamarashkin N., Zheltkov D. Block Lanczos-Montgomery method with reduced data exchange. In: Voevodin V., Sobolev S. (eds). RuSCDays 20166. Communications in Computer Science, v. 687, Springer, Cham, (2016)

11. Gurieva Y.L., Il'in V.P. Parallel approaches and technologies of domain decomposition methods. – Journal of Mathematical Sciences, v. 207, N 5, 724-735 (2015)

12. Il'in V.P. On exponential finite volume approximations. – Rus.J. Num. Anal. and Math. Modell, v. 18, N 6, 479-506 (2003)

13. Intel®Mathematical Kernel Library from Intel. http://www.software.intel.com/en-us/intel-mkl.