Two-Level Iterative Methods for Solving the Saddle Point Problems

Two-Level Iterative Methods for Solving the Saddle Point Problems

V P Il'in[1,2]

[1] Institute of Computational Mathematics and Mathematical Geophysics SB RAS Novosibirsk, Russia
[2] Novosibirsk State University, Novosibirsk, Russia

E-mail: ilin@sscc.ru

Abstract. Iterative processes in the Krylov subspaces for solving large ill conditioned saddle-type SLAEs with sparse matrices arising in finite difference, finite volume, and finite element approximations of multidimensional boundary value problems with complex geometric and functional properties of the initial data, characteristic of many relevant applications are studied. Combined two-level iterative algorithms using efficient Chebyshev acceleration and variational the conjugate directions methods, as well as the Golub-Kahan bi-diagonalization algorithms in the Krylov subspaces are considered. Examples of two-dimensional and three-dimensional filtration problems are used to study the resource consumption and computational performance of the proposed algorithms, as well as their scalable parallization on the multiprocessor systems with distributed and hierarchical shared memory.

1. Introduction

We consider a system of linear algebraic equations (SLAEs) with a saddle-type matrix

$$A\bar{u} \equiv A \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \equiv f, \quad A = \begin{bmatrix} D & C^\top \\ C & 0 \end{bmatrix}, \tag{1}$$

where $A \in \mathcal{R}^{N,N}$, $\bar{u}, \bar{f} \in \mathcal{R}^{N}$, $D \in \mathcal{R}^{N_1,N_1}$, $C \in \mathcal{R}^{N_2,N_1}$, $N = N_1 + N_2$, $u, f \in \mathcal{R}^{N_1}$ and $p, g \in \mathcal{R}^{N_2}$.

The matrix D is assumed to be symmetric and positive semidefinite. Moreover, a necessary and sufficient condition for the non-degeneracy of A is $ker(D) \cap ker(C) = \{0\}$, and a sufficient condition is the positive definiteness of the matrix D on $kerC$, see [1], and in what follows we will assume it to be satisfied.

Problems with a saddle point represent a widespread form of mathematical statements in modeling problems, including initial-boundary mixed formulations for differential classical or generalized equations, optimization problems, and computational algebra, see [2] - [3] and the references therein. We focus on methods for solving the saddle SLAEs with large sparse matrices arising from grid approximations of multidimensional boundary value problems that arise in many topical applications: electromagnetism, hydro-gasdynamics, elastic plasticity, multiphase filtration in porous media, optimization problems, and so on.

Due to the peculiarities of the block structure of the saddle algebraic systems, a considerable number of works has been devoted to methods for solving them: reviews by G. Golub with

colleagues [4], P. Vasilevski [5], and Y. Notay [6], the monograph by Yu. V. Bychenkov and E.V. Chizhokov [7], a series of papers by C. Greif and co-authors (including those for asymmetric SLAEs of the saddle type, [8] - [11]) and M. Arioli et al. [12] - [15], see also papers [16] - [19].

Note that, without loss of generality, we can consider the saddle SLAE in the form

$$\begin{bmatrix} D & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}. \tag{2}$$

Indeed, if we take some particular solution of the system $C\hat{u} = g$, then the vector $u = \check{u} + \hat{u}$, which is the solution of SLAE (1) will satisfy the system

$$\begin{bmatrix} D & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} \check{u} \\ p \end{bmatrix} = \begin{bmatrix} f - D\hat{u} \\ 0 \end{bmatrix}.$$

We also note that any solutions of SLAE (2) simultaneously satisfy the system

$$\tilde{A}v = \tilde{A} \begin{bmatrix} u \\ p \end{bmatrix} \equiv \begin{bmatrix} \tilde{D} & C^\top \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \equiv \tilde{f}, \quad \tilde{D} = D + S, \quad S = C^\top K_0^{-1} C, \tag{3}$$

where $v, \tilde{f} \in \mathcal{R}^N$, and $K_0 \in \mathcal{R}^{N_1, N_1}$ is an arbitrary non-singular matrix. Since the latter system is formally a regularization, or generalization, of SLAE (2), in the future we will dwell on algorithms for solving precisely equation (3).

The main objective of our research is the construction and comparative analysis of cost-effective and efficiently parallelizable algorithms for solving ill-conditioned SLAEs that arise in finite difference, finite volume, or finite element approximations of two-dimensional and three-dimensional problems with complex geometric and contractual material properties of the input data. In particular, we consider the application of the methods under study to the solution of the Darcy grid equations in a mixed formulation with contrasting coefficients, when the matrix block D in (3) is symmetric positive definite (s.p.d.), and the matrix A itself can be positive semidefinite.

The further content is constructed as follows. Section 2 deals with the analysis of various block preconditioned grid saddle problems. Section 3 considers the spectral properties of a family of preconditioned SLAEs of the saddle type. In Section 4, two-level iterative processes in the Krylov subspaces are described and investigated. Section 5 is devoted to the paralellization and performance issues of iterative methods. In conclusion, the results obtained and plans for the further development of the algorithms are discussed.

2. Block Preconditioning of the Saddle - Type SLAEs

Using the Shur complement

$$S = -C\tilde{D}^{-1}C^T,$$

the matrix of system (3) is factorized as

$$\tilde{A} = \begin{bmatrix} \tilde{D} & 0 \\ C & S \end{bmatrix} \begin{bmatrix} I & \tilde{D}^{-1}C^T \\ 0 & I \end{bmatrix}.$$

If the matrices $\bar{D}, S$ in (3) are replaced by their approximations (preconditioners) B_d and B_s, then we obtain the preconditioner of the matrix B in the form of an incomplete block factorization

$$B_1 = \begin{bmatrix} B_d & 0 \\ C & B_s \end{bmatrix} \begin{bmatrix} I & B_d^{-1}C^T \\ 0 & I \end{bmatrix}. \tag{4}$$

A somewhat rougher approximation, when only one (left or right) factor is used in (4) leads to incomplete block triangular preconditioning with the matrix

$$B_2 = \begin{bmatrix} B_d & C^T \\ 0 & B_s \end{bmatrix} \tag{5}$$

or to the incomplete Uzawa preconditioner

$$B_3 = \begin{bmatrix} B_d & 0 \\ C & B_s \end{bmatrix}. \tag{6}$$

The implementation of each step of the corresponding iterative processes can be represented by several stages, in which only one block component of the considered solution is recomputed (therefore, these methods are sometimes called segregation). In a somewhat generalized form, such a stationary algorithm (without Krylov acceleration) can be represented in the following three stages, see [6]:

$$\begin{aligned} \hat{u}_d^{n+1} &= u_d^n + Q_d^{(1)}(f_d - \bar{D}u_d^n - C^T u_c^n), \\ u_s^{n+1} &= u_s^n - B_s^{-1}(f_s - C\hat{u}_d^{-1}), \\ u_d^{n+1} &= \hat{u}_d^{n+1} + Q_d^{(2)}(f_d - \bar{D}A\hat{u}_d^{n+1} - C^T u_s^{n+1}). \end{aligned} \tag{7}$$

Here $Q_d^{(1)}, Q_d^{(2)}$ are some approximations of a matrix inverse or generalized inverse to the preconditioner B_d. In particular, if $Q_d^{(1)} = B_d^{-1}$, $Q_d^{(2)} = 0$ or $Q_d^{(1)} = 0$, $Q_d^{(2)} = B_d^{-1}$, then from (7) we get either the Uzawa algorithm with the preconditioner B_3 from (6) (in this case, the third stage is omitted, i.e. $u_d^{n+1} = \hat{u}_d^{n+1}$, or the incomplete block triangular preconditioning with the matrix B_2 from (5), while the first stage in (7) is omitted and $u_d^{n+1} = u_d^n$.

If the matrix $Q_d = Q_d^{(1)} + Q_d^{(2)} - Q_d^{(2)}\bar{D}Q_d^{(1)}$ is non-singular, then iterative process (7) corresponds to the preconditioner

$$B_4 = \begin{pmatrix} I & 0 \\ CQ_d^{(1)} & I \end{pmatrix} \begin{pmatrix} Q_d^{-1} & 0 \\ 0 & -B_s \end{pmatrix} \begin{pmatrix} I & Q_d^{(2)}C^T \\ 0 & I \end{pmatrix}. \tag{8}$$

In this case for $Q_d^{(1)} = Q_d^{(2)} = B_d^{-1}$ from (8) follows the so-called symmetric Uzawa method with preconditioning matrix

$$B_5 = \begin{pmatrix} I & 0 \\ CB_d^{-1} & I \end{pmatrix} \begin{pmatrix} B_d(2B_d - \bar{D})^{-1}B_d & 0 \\ 0 & B_s \end{pmatrix} \begin{pmatrix} I & B_d^{-1}C^T \\ 0 & I \end{pmatrix}. \tag{9}$$

One of the promising block-diagonal preconditioners for solving SLAE (3) is the following :

$$B_6 = \begin{bmatrix} \tilde{D} + C^\top K_1^{-1} C & 0 \\ 0 & K_2 \end{bmatrix}, \tag{10}$$

where K_1 and K_2 are some symmetric non-singular matrices. Such approaches have been proposed by C. Greif et al. in [8] - [11].

3. Spectral Analysis of the Preconditioned SLAEs

The eigenvalues and vectors of the preconditioned matrix $\bar{A} = B_6^{-1}\tilde{A}$, from the " perturbed " SLAE (3) are determined by the generalized spectral problem

$$\lambda B_6 = \tilde{A}z, \quad z = (z_1^\top, z_2^\top)^T, \quad z_k \in \mathcal{R}^{N_k}, \quad k = 1, 2,$$

having the following representation in the component form:

$$(\tilde{D} + C^\top K_0^{-1} C) z_1 + C^\top z_2 = (\lambda(D + C^\top K_1^{-1} C) z_1, \tag{11}$$

$$C z_1 = \lambda K_2 z_2.$$

An analysis of this spectral problem allows one to obtain interesting results for various special cases. In [9], [10] it was shown, in particular, that the unique spectral properties of the matrix $\bar{A}$ are obtained for the high singularity of the block $\tilde{D}$ relevant to algorithms for solving systems of the Maxwell equations. For example, the following results are valid.

Proposition 1. Let B_6 be s.p.d. matrix, and $\{z_i,\ i = 1, ..., N_1 - N_2\}$ is the basis of the kernel of the matrix C. Then the preconditioned matrix $B_6^{-1}\tilde{A}$ has $N_1 - N_2$ linearly independent vectors $(z_i^T\ 0) \in \mathcal{R}^{\mathcal{N}}$, which correspond to $N_1 - N_2$ multiple eigenvalues $\lambda = 1$.

Proposition 2. Let $K_1 = K_2 = K$, and $\tilde{D}$ be a symmetric positive semidefinite matrix with kernel dimension equal to r. Then $\lambda = 1$ is the eigenvalue of the matrix $B_6^{-1}\tilde{A}$ with multiplicity N_1, and $\lambda = -1$ is the eigenvalue of the multiplicity r. The remaining $N_2 - r$ eigenvalues belong to the interval $\lambda \in (-1, 0)$ and satisfy the relation

$$\lambda = -\mu/(\mu + 1), \tag{12}$$

where μ is $N_2 - r$ positive generalized eigenvalues

$$\mu \tilde{D} z = C^T K^{-1} C z. \tag{13}$$

Let $\{z_i,\ i = 1, ..., N_1 - N_2\}$ be the basis of the kernel C, $\{x_i,\ i = 1, ..., r\}$ be the basis of the kernel $\tilde{D}$, and $\{y_i,\ i = 1, ..., N_1 - N_2\}$ is a collection of linearly independent vectors complementing $ker(\tilde{D}) \cup \ker(C)$ based on $\mathcal{R}^N$. Then $N_1 - N_2$ vectors $(z_i, 0)$, r vectors $(x_i, K^{-1}Cx_i)$ and $N_2 - r$ vectors $(y_i, K^{-1}y_i)$ are linearly independent eigenvectors corresponding to the eigenvalues $\lambda = 1$, and r the vectors $(x_i,\ K^{-1}x_i)$ are the eigenvectors corresponding to $\lambda = -1$. In general, the presence of different matrices K_0, K_1, K_2 in the preconditioner B_6 will provide ample opportunities for constructing effective algorithms in specific cases.

4. Golub - Kahan Bidiagonalization Iterative Methods

We consider a family of iterative methods for solving the saddle symmetric SLAEs with the matrix $\tilde{A}$ from (3), based on the efficient $GK-$ bidiagonalization (Golub - Kahan) approach, which was originally proposed for the singular decomposition of rectangular matrices, but then by M. Saunders, M. Arioli, C. Greif and other authors has been successfully used to solve algebraic systems, taking into account the block saddle structure. Without loss of generality, we write down the investigated SLAE in the form

$$\tilde{A}\begin{bmatrix} u \\ p \end{bmatrix} \equiv \begin{bmatrix} \tilde{D} & C^\top \\ C & 0 \end{bmatrix}\begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} 0 \\ g \end{bmatrix}, \quad \tilde{D} = D + C^\top K^{-1} C_0. \tag{14}$$

It is easy to verify that if in (3), the function u is replaced by $u + \tilde{D}^{-1} f$, then this system will take the form (14) with the right-hand side $g = -C\tilde{D}^{-1} f$. It is assumed that in (14) $\tilde{D}$ and K are s.p.d. matrices, and the inequality $N_1 \geq N_2$ also holds. The method of $G - K$ bidiagonalization is based on the construction of $\tilde{D}$ -orthogonal vectors v_k and K – orthogonal vectors q_k that satisfy the conditions

$$\begin{aligned} C^\top Q = \tilde{D} V [B^\top 0]^\top, \quad V^\top \tilde{D} V = I_{N_1}, \\ C V = K Q [B^\top 0], \quad Q^\top K Q = I_{N_2}, \end{aligned} \tag{15}$$

where $V = [v_1, ..., v_{N_1}] \in \mathcal{R}^{N_1,N_2}$, $Q = [q_1, ..., q_{N_1}]$, and $B \in \mathcal{R}^{N_2,N_2}$ there is a bi-diagonal matrix

$$B = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \dots & 0 \\ 0 & \alpha_2 & \beta_3 & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & \alpha_{N_2-1} & \beta_{N_2-1} \\ 0 & \ddots & 0 & 0 & \alpha_{N_2} \end{bmatrix}.$$

Introducing new unknown functions

$$u = Vz, \quad p = Qy \tag{16}$$

and multiplying system (14) by the block-diagonal matrix $block\text{-}diag(V^\top, Q^\top)$, we obtain

$$\begin{bmatrix} I_{N_2} & 0 & B \\ 0 & I_{N_1-N_2} & 0 \\ B^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ Q^\top g \end{bmatrix}, \quad z_1 \in \mathcal{R}^{N_2}, \quad z_2 \in \mathcal{R}^{N_1,\cdot,N_2}. \tag{17}$$

It follows from (17) that the vector u depends on N_2 columns of the matrix V, since $z_2 = 0$. Thus, SLAE (17) is reduced to

$$\begin{bmatrix} I_{N_2} & B \\ B^\top & 0 \end{bmatrix} \begin{bmatrix} z_1 \\ y \end{bmatrix} = \begin{bmatrix} 0 \\ Q^\top g \end{bmatrix}. \tag{18}$$

From here, defining

$$q_1 = K^{-1}g/||g||_{K^{-1}}, \quad \alpha_1 \tilde{D} v_1 = C^\top q_1, \quad ||g||_{K^{-1}} = (g, K^{-1}g)^{1/2},$$

we find the initial vector v_1 :

$$v_1 = w/\alpha_1, \quad \alpha_1 = \sqrt{w^\top C^\top q_1}, \quad w = \tilde{D} C^\top q_1.$$

The further vectors v_i, q_i and the entries α_i, β_i of the matrix B are calculated from the recurrence relations $(n = 1, ...,)$

$$s = K^{-1}(Cv_n - \alpha_i K q_n), \quad \beta_{n+1} = \sqrt{s^\top K s},$$
$$q_{n+1} = s/\beta_{n+1}, \quad w = \tilde{D}^{-1}(C^\top q_{n+1} - \beta_{n+1}\tilde{D}v_n), \ \alpha_{n+1} = (w^\top \tilde{D} w)^{1/2}, \quad v_{n+1} = w/\alpha_{n+1}.$$

The consecutive approximations of u^n, according to (16), are determined by the first n columns of the matrix V, i.e.

$$u^n = V_n z_1 = \sum_{j=1}^{n} z_1^{(j)} v_j, \quad z_1 = (z_1^{(1)}, z_1^{(2)}, ..., z_1^{(j)}),$$

where $z_1^{(j-1)}$ are the components of the vector z_1 from (18) calculated by the formulas

$$z_1^{(1)} = ||g||_{N^{-1}}/\alpha_1, \quad z_1^{(j+1)} = -\beta_{j+1} z_1^{(j)}/\alpha_{j+1}, \quad j = 1, ..., N_2.$$

Omitting the details of the derivation of formulas (see [12] for details), we obtain the resulting recurrence relations for the iterative solution:

$$u^1 = z_1^{(1)} v_1, \quad u^{n+1} = u^n + z_1^{(n+1)} v_{n+1}, \quad n = 1, ..., N_2 - 1, \quad d_1 = q_1/\alpha_1,$$
$$p^1 = -z_1^{(1)}\alpha_1, \quad p^{n+1} = p^n - z_1^{(n+1)} d_{n+1}, \quad d_{n+1} = q_{n+1} - \beta_{n+1} d_n/\alpha_{n+1}, \tag{19}$$

where d_n is the n - th column of $D = QB^{-1}$. Note that the described algorithm has the following optimization properties: at each n - th step, the error of the iterative approximation reaches its minimum, i. e.

$$\begin{array}{c} \min \\ u_n \in \mathcal{U}_n \\ Cu^n - g \perp \mathcal{Q}_n \end{array} \quad ||u - u^n||_{\tilde{D}}, \tag{20}$$

$$\mathcal{U}_n = Span\{v_1, ..., v_n\}, \quad \mathcal{Q}_n = Span\{q_1, ..., q_n\}.$$

As was noted in [12], this method has a high convergence-rate when solving SLAEs arising from finite element approximations of continuous multi-dimensional saddle-type boundary value problems, i.e. in mixed statement. Moreover, an essential factor is that at each iteration it is required to solve the arising algebraic subsystems with the matrices $\tilde{D}$ and K, which, in a sense, play the role of preconditioners. Their approximate inversion leads, in fact, to two-level iterative processes in certain subspaces [20]. The implementation of the internal iterations can be done, for example, by means of the efficient Chebyshev acceleration or conjugate direction methods in Krylov subspaces (CA and CD, respectively). It is important, that in many cases, the eigenvalue bounds and condition number of the matrix $\tilde{D}$ can be estimated.

If we denote by $\hat{A}$ and $\hat{B}$ the matrix and the preconditoner for SLAEs $\hat{A}u = \hat{f}$, to be solved at the internal iterations, the CA and CD algorithms can be written in the unified form with two-terms coupled recurrences, see [21]:

$$\begin{gathered} r^0 = \hat{f} - \hat{A}u^0, \quad p^0 = \hat{B}^{-1}r^n, \quad n = 0, 1, ... : \\ u^{n+1} = u^n + \alpha_n p^n, \quad r^{n+1} = r^n - \alpha_n \hat{A}p^n, \\ p^{n+1} = \hat{B}^{-1}r^{n+1} + \beta_n p^n. \end{gathered} \tag{21}$$

For the Chebyshev approach, the iterative coefficients are defined by the formulas

$$\alpha_0 = \tau, \quad \alpha_n = \gamma_n \tau, \quad \beta_n = (\gamma_n - 1)\alpha_{n-1}/\alpha_n,$$

where the parameters of τ, τ_n, and γ depend on the minimum and maximum eigenvalues m, M of the matrix $\hat{B}^{-1}\hat{A}$:

$$\tau = 2/(M + m), \quad \gamma_n = 4/(4 - \gamma_{n-1}\gamma^2), \ \ \gamma_0 = 2, \quad \gamma = (C - 1)/(1 + C), \quad C = M/m.$$

For the preconditioned conjugate direction methods, the iterative formulas can be derived from two-side preconditioning of the original SLAE with the matrix $\hat{A}$ and right hand vector $\hat{f}$:

$$\begin{gathered} \check{A}\check{u} \equiv L^{-1}\hat{A}U^{-1}Uu = L^{-1}\hat{f} = \check{f}, \quad \hat{B} = LU, \quad L = U^\top, \\ \check{u} = Uu, \quad \check{A} = L^{-1}\hat{A}U^{-1} = \check{A}^\top. \end{gathered} \tag{22}$$

To solve this SLAE, the generalized CD method can be written in the form

$$\begin{gathered} \check{u}^0 = Uu^0, \quad \check{p}^0 = \check{r}^0 = \check{f} - \check{A}\check{u}^0 = L^{-1}r^0 = L^{-1}(\hat{f} - \hat{A}u^0), \quad \rho_n = (\check{A}^\gamma \check{p}^n, \check{p}^n), \\ \check{u}^{n+1} = \check{u}^n + \alpha_n \check{p}^n, \quad \check{r}^{n+1} = \check{r}^n - \alpha_n \check{A}\check{p}^n, \quad \alpha_n = \sigma_n/\rho_n, \\ \check{p}^{n+1} = \check{r}^n + \beta_n \check{p}^n, \quad \beta_n = \sigma_{n+1}/\sigma_n, \quad \sigma_n = (\check{A}^{\gamma-1}\check{r}^n, \check{r}^n), \end{gathered} \tag{23}$$

where the direction vectors $\check{p}^n$ satisfy to the following orthogonal properties ($\gamma = 0, 1, 2$) :

$$(\check{p}^n, \check{p}^k)_\gamma = (\check{A}^\gamma \check{p}^n, \check{p}^k) = \rho_n^{(\gamma)}\delta_{k,n}, \quad \rho_n^{(\gamma)} = ||\check{p}_n||_\gamma^2, \quad \gamma = 0, 1, 2,$$

which provide the minimization of the "preconditioned" residual functionals

$$\Psi_n^{(\gamma)} = (\check{r}^n, \check{r}^n)_{\gamma-2} \equiv (\check{A}^{\gamma-2}\check{r}^n, \check{r}^n).$$

Here $\delta_{k,n}$ is Kronecker symbol, and $\gamma = 2, 1, 0$ corresponds to conjugate residual, conjugate gradient and minimal iteration methods, see [21] and references therein. For $\gamma = 1$ (CG - conjugate gradient method), for example, the parameters α_n, β_n are defined as follows :

$$\alpha_n = \sigma_n/\rho_n, \quad \beta_n = \sigma_{n+1}/\sigma_n,$$

$$\sigma_n = (\hat{B}^{-1}r^n, r^n), \quad \rho_n = (\hat{A}p^n, p^n),$$

and iterative process is implemented by means of (21).

If $\gamma = 2$ (CR - conjugate residual algorithms) the coefficients σ_n, ρ_n, as well as solution, residual and direction vectors, are computed, instead of (21), (24), by the formulas

$$\begin{aligned} \hat{p}^0 = \hat{r}^0 = \hat{B}^{-1}p^0 = \hat{B}^{-1}r^0 = \hat{B}^{-1}(\hat{f} - \hat{A}u^0), \\ u^{n+1} = u^n + \alpha_n\hat{p}^n, \quad \hat{r}^{n+1} = \hat{r}^n - \alpha_n\hat{B}^{-1}\hat{A}p^n, \\ \hat{p}^{n+1} = \hat{r}^{n+1} + \beta_n\hat{p}^n, \quad \alpha_n = \sigma_n/\rho_n, \quad \beta_n = \sigma_{n+1}/\sigma_n, \\ \sigma_n = (\hat{A}\hat{r}^n, r^n), \quad \rho_n = (\hat{B}^{-1}\hat{A}\hat{p}^n, \hat{A}\hat{p}^n). \end{aligned} \tag{24}$$

Let us remark, that implementation the preconditioned conjugate gradient and conjugate residual methods, by means of (21), (24) , and (25) for mulas respectively require matrix-vector product by $\hat{A}$ and $\hat{B}^{-1}$ only and, in fact, we do not need LU – factorization of the matrix $\hat{B}$.

5. Parallelization and Performance Issues of Iterative Methods

Modern understanding of the quality of the algorithm includes the two main characteristics: mathematical efficiency and performance of its implementation on a specific supercomputer architecture. The first aspect includes the design and optimization of iterative methods with a high convergence rate, as well as theoretical estimates of computational resource consumption (the necessary volumes of arithmetic operations and memory). The second aspect is purely practical and is characterized by the real time of the algorithm execution for a certain class of tasks, which depends on the scalability of its parallelization, as well as on programming technologies on a particular computer platform.

The SLAEs that are of most interest to us are of high orders $(10^8 - 10^{11})$ and sparse matrices with large condition numbers $(10^{12} - 10^{15})$ and an irregular structure. This does not only leads to a large number of iterations, but also forces one to work with distributed and hierarchical shared memory systems, and also significantly slows down the access to data.

The main quantitative characteristic of parallelization is the acceleration of calculations

$$S_p = T_1/T_p, \quad T_p = T_p^a + T_p^c,$$

where T_p is the time of solving the problem on p processors, which is the total of the time of information exchanges and arithmetic operations, which are described by approximate formulas

$$T_p^a = \tau_a N_a, \quad T_p^c = N_e(\tau_0 + \tau_c N_c).$$

Here τ_a and N_a are the average execution time of one arithmetic operation and their total number, N_e is the number of exchanges, τ_0 and τ_c are the waiting time and the duration of the transmission of one number, and N_c is the average volume of one communication.

Since the inequalities $\tau_0 \gg \tau_c \gg \tau_0$ are valid for machine constants, obvious recommendations follow for the constructed algorithms: we must try to minimize the amount of communications, and exchange them not in small, but in large portions, i.e. it is possible to carry out preliminary accumulation of data buffers. These conclusions are all the more true that inter-processor information transfers do not only slow down the computational process, but are also the most energy-consuming operations, and this becomes a significant factor in the cost of operating a supercomputer, see [22].

The parallelization strategy for large grid SLAEs arising from the approximation of multi-dimensional boundary value problems is based on hybrid programming tools and additive domain decomposition methods (DDM, [22]) with two-level iterations in the Krylov subspaces. Iterations of the upper level (over subdomains) are carried out using the data exchange among processes (by means of MPI library), each one performing (simultaneously) solving an algebraic subsystem in the corresponding subdomain. At each such iteration, the values of approximate solutions are exchanged on the interface boundary surfaces of the contacting subdomains. Naturally, all the matrix and vector data for subsystems are preliminary generated in a distributed form over processes. The SLAE solution in each of the subdomains is parallelized using multi-threaded computing (systems like OpenMP) on multi-core processors with shared memory. Additional acceleration here can be achieved by vectorizing operations (command systems such as AVX), see the review in [23]. Let us note, that because of the absence of an adequate model of computing on modern machines, the code optimization being an experimental research.

Substantial computational acceleration can be obtained by using a variable precision machine arithmetic. When solving large SLAEs, it is to use a double precision conventional with a representation length a floating-point real number of 64 bits, however, for extremely ill conditioned matrices, the transition to quadruple precision (128 bits) is necessary. On the contrary, at certain stages of the algorithm, it is permissible to use a simple and even a half-precision arithmetic (32 and 16 bits, respectively), which are performed much faster. Such an approach is natural at the first steps of the iterative process, when the error of the approximate solution is still relatively large. Another possibility of using a reduced accuracy exists in the DDM when solving auxiliary SLAEs in subdomains. It must be borne in mind that such solutions require a thorough analysis of the realized numerical errors of the method to ensure stable calculations in general, see [24]. In this book, for example, there is a discussion how the stopping criteria of iteration

$$||r^n|| \le \varepsilon_1 ||\bar{f}|| + \varepsilon_2 ||A|| \cdot ||\bar{u}^n||, \quad 0 < \varepsilon_1,\ \varepsilon_2 \ll 1,$$

where $r^n = \bar{f} - A\bar{u}^n$ is the residual of equation (1), provides the real accuracy of the approximate solution for the specific values of $||\bar{f}||$, $||\bar{u}^n||$ and $||A||$.

Further reserve for increasing the performance is the code optimization, which can be achieved by using high-quality computing tools (SPARSE BLAS, for example), using various compiler options and special properties of the supercomputer platform.

6. Conclusion

In the presented brief review, we have discussed the main computational and technological issues for a simple, in a sense, the algebraic saddle point problem. The generalization of the statement could be done in various directions : non-zero right lower block of the matrix A in (1), non-symmetric algebraic system, non-stationary initial-boundary value problems, as well as the interdisciplinary problems, including direct or inverse ones. In any cases, the numerical solution of the linear algebraic saddle point system of equations presents the "bottle neck" of the large-scale computer experiments, because the computational resources of this stage of mathematical modeling grow non-linearly under increasing the degree of freedom for the problem to be solved. The flexibility of the proposed approaches provides good possibilities for the further research.

Acknowledgments
The work is supported by grants RFBR N 18-01-00295 and RSF N 19-11-00048

7. References

[1] Brezzi F 2013 Stability of saddle points in finite dimensions, *Frontiers in numerical analysis* ed J F Blowey, A W Craig and T Shardlow (Berlin, Heidelberg: Universitext. Springer) pp 17–61
[2] Elman H C, Silvester D J and Wathen A J 2014 Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics Numerical Mathematics and Scientific Computation second edition (London: Oxford University Press)
[3] Ivanov M I Kremer I A and Laevsky Yu M 2019 On one upflow scheme for solving filtration problem Siberian *Electronic Mathematical Reports* **16** 757–776 [in Russian]
[4] Benzi M, Golub G H and Liesen J 2005 Numerical solution of saddle point problems *Acta Numerica* **14** 1–13
[5] Vassilevski P S 1996 Preconditioning mixed finite element saddle - point tlliptic probems *J. Numer. Liniar Algebra Applic.* **3 (1)** 1–20
[6] Notay Y 2019 Convergence of some iterative methods for symmetric saddle point linear systems *SIAM J. Matrix Anal. Appl.* **40 (1)** 122–146
[7] Bychenkov Y V and Chizhonkov E V 2010 Iterative methods for solving saddle point problems (Moscow: Binom Publ.) [in Russian]
[8] Estrin R and Greif C 2018 SPMR: A family of saddle-point minimum residual solvers *SIAM Journal on Scientific Computing* **40 (3)** 1884–1914
[9] Greif C and Schotzau D 2006 Preconditioners for Saddle Point linear systems with highly Singular (1.1) Blocks *Electron. Trans. Numer. Anal.* (Special Volume on Saddle Point Problems vol 22) 114–121
[10] Greif C and Wathen M 2019 Conjugate Gradient for nonsingular saddle-point systems with a maximally rank-deficient leading block *Journal of Computational and Applied Mathematics* **358** 1–11
[11] Estrin R and Greif C 2016 Towards an optimal condition number of certain augmented Lagrangian-type saddle-point matrices *J. Numer. Linear Algebra Appl.* **23 (4)** 693–705
[12] Arioli M 2013 Generalized Golub-Kahan diagonalization and stopping criteria *SIAM J. Matrix Anal. Appl.* **34 (2)** 57–592
[13] Arioli M, Kruse C, Rude U and Tardieu N 2018 An iterative generalized Golub-Kahan algorithm for problems in structural mechanics SIAM arhiv:1808.07677vl [cs.CE]
[14] Arioli M and Manzini M A network programming approach in solving Darcy‘s equations by mixed finite elements methods *Electronic Transactions on Numerical Analysis* **22** 41–70
[15] Arioli M and Orban D 2013 Iterative methods for symmetric quasi-definite linear systems Part I: Theory. Science &Tehnology, Facilities Council, RAL, Harwell Oxford, Technical Report RAL-TR-2013-003
[16] Mardal K and Winther R 2011 Preconditioning discretizations of systems of partial differential equations *Numer. Linear. Algebra Appl.* **18** 1–40
[17] Gould N, Orban D and Rees T 2014 Projected Krylov methods for saddle-point systems *SIAM J. Matrix Anal. Appl.* **35 (4)** 329–343
[18] Il'in V P and Kazantsev G Y 2020 Iterative solution of the saddle-point SLAEs *Journal of Mathematical Sciences* **240 (6)** 135–151
[19] Popov P E and Kalinkin A A 2008 The Method of separation of variables in a problem with a saddle - point *Russian J. Numer. Anal. Math. Model.* **23 (1)** 97–106
[20] Il'in VP 2019 Two - level least squares methods in Krylov subspace *J. Math. Sci.* **232 (6)** 892–901
[21] Gurieva Y L, Il'in V P and Petukhov A V 2020 On Multigrid Methods for Solving Two-Dimensional Boundary-Value Problems *Journal of Mathematical Sciences* **240 (6)** 13–28
[22] Dongarra J, Grigori L and Higham N J 2020 Numerical algorithms for high performance computational science *Phil. Trans. R. Soc.* A 378
[23] Il'in V P 2016 Problems of parallel solution of large systems of linear algebraic equations *J. Math. Sci.* **216 (6)** 795–804
[24] Higham N J 2002 Accuracy and stability of numerical algorithms (Philadelphia: SIAM)