Comparative Analysis of Parallel Methods for Solving SLAEs in Three-Dimensional Initial-Boundary Value Problems

V. S. Gladkikh[1], V. P. Ilin[1], and M. S. Pekhterev[1,2(✉)]

[1] Institute of Computational Mathematics and Mathematical Geophysics SB RAS, Novosibirsk, Russia
[2] Novosibirsk State University, Novosibirsk, Russia
maxim-pekhterev@mail.ru

Abstract. Iterative methods for solving systems of linear algebraic equations with high-order sparse matrices that arise in absolutely stable implicit finite-volume approximations of three-dimensional initial-boundary value problems for the heat and mass transfer equation on unstructured grids in computational domains with a complex configuration of multiply connected piecewise smooth boundary surfaces and contrasting material properties are considered. At each time step, algebraic systems are solved using parallel preconditioned algorithms for conjugate directions in Krylov subspaces. To speed up the iterative processes, variational methods for choosing initial approximations are applied using numerical solutions from previous time steps. It is discussed how the proposed approaches can be more general formulations of problems, as well as how to increase the productivity of computational methods and technologies in the multiple solution of algebraic systems with sequentially determined different right-hand sides and with the scalable parallelization of algorithms based on the additive methods of domain decomposition. The efficiency of the proposed approaches is investigated for the implicit Euler and Crank–Nicholson schemes based on the results of numerical experiments on a representative series of methodological problems.

Keywords: initial-boundary value problem · implicit schemes · iterative processes · Krylov subspaces · least squares method · numerical experiments

1 Introduction

The numerical solution of multidimensional initial-boundary value problems for partial differential equations of parabolic type is an urgent practical problem in the mathematical modeling of processes and phenomena in many applications, including interdisciplinary ones [7]. A typical example is non-isothermal multiphase filtration [11] in porous media with different-scale geometric and material characteristics. Modern numerical algorithms and technologies for solving the

L. Sokolinsky and M. Zymbler (Eds.): PCT 2022, CCIS 1618, pp. 59–72, 2022.
https://doi.org/10.1007/978-3-031-11623-0_5

considered computational problems of thermal conductivity are presented, for example, in [1,3,10,13–15]. Implicit approximations of the original statements on adaptive unstructured grids, necessary to ensure the absolute stability of numerical integration in time and high-resolution calculations, require the construction of high-performance algorithms for multiprocessor computing systems (MCS). The most resource-intensive stage here is the solution of systems of linear algebraic equations (SLAEs), which take up to 80 % and more computer time when implementing non-stationary and nonlinear models, since at this stage the volume of arithmetic operations performed grows nonlinearly with an increase in the number of degrees of freedom. An existing technological feature of algorithms for solving problems with real data is the storage of matrices of large algebraic systems (with orders $10^8 - 10^{10}$ and higher) in sparse compressed formats.

The purpose of this work is to analyze the features of the application of parallel preconditioned iterative methods in Krylov subspaces in relation to a three-dimensional linear initial-boundary value problem with mixed-type boundary conditions for a non-stationary heat equation. It should be noted that in classical iterative processes, algorithms that converge regardless of the nature of the initial approximation are studied. The main emphasis in our studies is placed on the choice of initial approximations, which make it possible to significantly reduce the number of iterations when solving the SLAE at each time level, due to the use of the results obtained at the previous steps. In contrast to the approximation approaches in common methods of predictor-corrector type [8], we propose a variational algebraic principle based on minimizing the initial residual. Incomplete factorization methods in Krylov subspaces are used as iterative solvers. Solvable algebraic systems are formed using barycentric finite volume methods on a tetrahedral grid, described in [3]. For approximation in time, parametrized two-layer schemes are used, with an emphasis on the implicit Euler and Crank–Nicholson schemes (for further research, the discontinuous Galerkin methods of various orders of accuracy in space and time proposed in [2] are of considerable interest). We also discuss a possible generalization of the results to the solution of initial-boundary value problems in the presence of convection and/or nonlinearity, as well as the possibility of accelerating computations when repeatedly solving SLAEs with different right-hand sides and using the parallelization of algorithms based on the additive method of domain decomposition, see [4,6,9].

The present work is structured as follows. Section 2 describes the features of the continuous and discrete formulations of the problems under consideration, including the study of the stability and additional error of the grid solution due to the approximate nature of the iterative implementation of implicit schemes. Section 3 is devoted to the presentation of the proposed parallel iterative algorithms with the analysis of different approaches to the choice of initial approximations at different time steps. Section 4 presents the results of experimental studies of the effectiveness of algorithms based on the results of calculations for a series of methodological problems. In conclusion, the results obtained and plans for further research are discussed.

2 Continuous and Discrete Problem Setting

Consider the formulation of the three-dimensional initial-boundary value problem of heat conduction in the computational domain

$$(x, y, z, t) \in \Omega \times [0, T], \quad \Omega \in R^3, \quad T \in R^1,$$

with a piecewise smooth boundary Γ, in general, multiply connected, and a closure $\overline{\Omega} = \Omega \cup \Gamma$. The heat conduction equation can be written as

$$c\frac{\partial u}{\partial t} = div(\lambda \cdot gradu) + f(x, y, z, t), \tag{1}$$

where λ is the thermal conductivity coefficient, c is the heat capacity coefficient, and $f(x, y, z, t)$ is the continuous sufficiently smooth source function. On different sections of the boundary Γ_D and Γ_N, $\Gamma_D \cup \Gamma_N = \Gamma$, the Dirichlet and Neumann boundary conditions are imposed on the sought solution, respectively:

$$u|_{\Gamma_D} = u_D(x, y, z, t), \quad \lambda \left.\frac{\partial u}{\partial n}\right|_{\Gamma_N} = \sigma_N(x, y, z, t). \tag{2}$$

Here u_D and σ_N are the given functions of temperature and heat flux distribution. Relations (1), (2) are supplemented by the initial conditions for $(x, y, z) \in \Omega$:

$$u(x, y, z, 0) = u^0(x, y, z). \tag{3}$$

We assume that the initial data of problem (1)–(3) have properties that ensure the existence, uniqueness, and sufficient smoothness of the solution necessary to justify the approximation, stability, and convergence of the approximate methods used below for solving the initial-boundary value problem.

Relations (1)–(3) are approximated on the space-time grid $\Omega^h \times \Omega^\tau$, where the time steps are generally different, i.e.,

$$\Omega^\tau = \{t_{n+1} = t_n + \tau_n, \quad n = 0, 1, \cdots, N_t\},$$

and the spatial mesh with the number of nodes N_h is adaptive and unstructured. For simplicity, we consider it static, i.e., not changing over time. The process of discretizing the initial continuous statement is carried out in two stages. First, using the barycentric finite volume method [3], we approximate the partial differential equation and boundary conditions, as a result of which we obtain a system of ordinary differential equations (ODEs)

$$\begin{gathered} C_h\frac{d(u)^h}{dt} + A_h(u)^h = g^h + \psi^h, \\ u^h, g^h, \psi^h \in \Re^{N_h}, \quad C_h, A_h \in \Re^{N_h, N_h} \end{gathered} \tag{4}$$

where $(u)^h = \{u_k(t)\}$ is the vector of values of the desired solution at grid nodes, ψ^h is the spatial approximation error, and C_h and A_h are some independent from

time to time symmetric matrices, traditionally called mass and stiffness matrices, respectively, see [7]. At the second stage, we approximate the ODE system using the parameterized one-step scheme

$$C_h((u)^{n+1}-(u)^n) = \tau_n[\theta(g^{n+1}-A_h(u)^{n+1})+(1-\theta)(g^n-A_h(u)^n)+\psi_n^h+\psi_n^\tau]. \quad (5)$$

Here $(u)^n = \{u_k(t_n)\}$ are the vectors of exact values of the sought solution at the nodes of the space-time grid, $\theta \in [0,1]$ is the parameter of the approximating scheme, and $\psi^\tau = O(\tau^\gamma)$ is the time approximation error vector ($\gamma = 2$ for $\theta = \frac{1}{2}$ and $\gamma = 1$ for the rest cases). Note that the algorithms defined by relation (4) for $\theta = \frac{1}{2}, 0, 1$ are called Crank–Nicholson, explicit Euler and "strictly implicit" Euler schemes, respectively. Discarding the approximation terms ψ_n^h and ψ_n^τ in (4), we arrive at a system of linear algebraic equations for the vectors of approximate grid solutions $u^n = \{u_k^n\}$:

$$\begin{gathered}(C_h + \theta\tau_n A_h)u^{n+1} = (C_h - (1-\theta)\tau_n A_h)u^n + \tau_n g^{n+\theta}, \\ g^{n+\theta} = \tau_n(\theta g^{n+1} + (1-\theta)g^n).\end{gathered} \quad (6)$$

It is noteworthy that if at each n-th time step SLAE (3) is solved approximately using some iterative process, then u^{n+1} is replaced by the iterative approximation $\tilde{u}^{n+1}$, for which the residual vector is determined

$$r^{n+1} = g^{n+\theta} + \tau_n^{-1}(C_h - (1-\theta)\tau_n A_h)\tilde{u}^n - (C_h + \theta\tau_n A_h)\tilde{u}^{n+1} \quad (7)$$

As shown in [10], for $\theta \geq 1/2$ and a sufficiently small residual norm $\|r^{n+1}\|$ for the implicit schemes under consideration, the absolute stability of numerical integration with respect to time follows.

In this paper, we restrict ourselves to considering the simplest approximations in time, which can be called one-stage Runge-Kutta (R-K) methods. More accurate approximations can be built using multistage R-K algorithms, both explicit and implicit, see [8].

3 Methods for Solving SLAEs in Implicit Schemes

System of equations (6), solved at each time step, can be rewritten as

$$\begin{gathered}\overline{A}u^{n+1} = \overline{f}^{n+1}, \quad n = 0, 1, \cdots, N_t, \\ \overline{A} = C_h + \theta\tau_n A_h, \quad \overline{f}^{n+1} = \tau_n g^{n+1} + (C_h - (1-\theta)\tau_n A_h)u^n.\end{gathered} \quad (8)$$

It is natural to solve SLAE (8) for $\theta > 0$ using iterative algorithms for two reasons. The first is related to the spectral properties of the matrix $\overline{A}$. Since for the most common spatial approximations considered by us, the mass matrix C_h has eigenvalues $\nu = O(1)$, i.e., independent of the characteristic mesh steps tau, h, and the eigenvalues of the stiffness matrix A_h lie in the interval $\lambda \in [\lambda_1, \lambda_N]$, $\lambda_1 = O(1)$, $\lambda_N = O(h^{-2})$, see [7], for the eigenvalues of the matrix $\overline{A}$ we obtain the following relations:

$$\mu(\overline{A}) \in [\mu_1, \mu_N], \quad \mu_1 = O(1), \quad \mu_N = O(1 + \theta\tau h^{-2}). \quad (9)$$

Hence, for the corresponding condition number, we have
$cond(\overline{A}) = \frac{\max_k\{\mu_k\}}{\min_k\{\mu_k\}} = O(\theta\tau h^{-2})$, which for small values of τ means a sufficiently fast convergence of iterations.

The second feature of the problems under consideration is that when solving the SLAE at the current time step, the previous solutions are already known, which can be used to find a good initial approximation and reduce the number of iterations. Let us describe some possible approaches here.

3.1 Choice of an Initial Approximation for Solving a SLAE

a) The simplest trick is to choose an arbitrary initial approximation, for example $u^{n+1,0} = 0$. However, this means that the specifics of the initial-boundary value problem being solved is not taken into account in any way.

b) The most natural way is to put $u^{n+1,0} = u^n$, which formally means the use of zero-order interpolation (we denote this approach by I^0). This approximation principle can be generalized to higher orders if one remembers the solutions $u^{n-1}, u^{n-2}, \cdots$ from the previous time steps. For example, using linear extrapolation gives

$$u^{n+1,0} = u^n + (u^n - u^{n-1})\tau_n/\tau_{n-1}. \tag{10}$$

c) The predictor-corrector method [8], which is implemented in two stages, is widespread, especially when solving ordinary differential equations. At the first stage, a preliminary (prognostic) approximation is calculated, for which, in fact, the explicit scheme obtained from (6) is used for $\theta = 0$:

$$\hat{u}^{n+1} = C_h^{-1}[(C_h - \tau_n A_h)u^n + \tau_n g^n]. \tag{11}$$

At the second stage, the calculated value is corrected with the determination of the initial approximation by the formula

$$u^{n+1,1} = C_h^{-1}[(C_h - (1-\theta)\tau_n A_h)u^n + \tau_n g^{n+\frac{1}{2}} - \theta\tau_n A_h\hat{u}^{n+1}]. \tag{12}$$

Obviously, procedure (14) can be interpreted as the application of a simple iteration algorithm. This correction can be repeated any given number of m times, resulting in a method denoted as PC^m (PC for $m = 1$):

$$\begin{gathered}\hat{u}^{n+1,k} = C_h^{-1}[(C_h - (1-\theta)\tau_n A_h)u^n + \tau_n g^{n+1} - \theta\tau_n A_h\hat{u}^{n+1,k-1}], \\ k = 1, \cdots, m; \quad \hat{u}^{n+1,0} = \hat{u}^{n+1}; \quad u^{n+1,0} = \hat{u}^{n+1,m}.\end{gathered} \tag{13}$$

Obviously, for sufficiently small τ_n, this iterative process converges, but slowly. If we formally restrict ourselves here to the case $m = 0$, i.e., no correction, and set $u^{n+1,0} = \hat{u}^{n+1}$, then we denote this method as P. Note also that the PC^m methods can be applied both for the Crank–Nicholson schemes, and for the implicit Euler method ($\theta = \frac{1}{2}, 1$, respectively), but these, naturally, will be different algorithms. It is important to bear in mind that predictor-corrector methods are traditionally used without iterative refinement.

d) Reducing the number of iterations at each time step can be ensured if fast preconditioned methods in Krylov subspaces are used to solve SLAE (5), and the initial approximation $u^{n+1,0}$ is determined not from approximation, but from optimization algebraic approaches. For example, in the PC method, instead of correction stage (14), one can use a linear combination of the vectors $u^n, \hat{u}^{n+1}$ according to the condition of minimizing the initial discrepancy $r^{n+1,0} = \overline{f}^{n+1} - \overline{A}u^{n+1,0}$, determined from equation (10):

$$\begin{aligned} u^{n+1,0} &= u^n + cv^n, \quad v^n = \hat{u}^{n+1} - u^n, \\ r^{n+1,0} &= r^n - c\overline{A}v^n, \quad c = \frac{(r^n, \overline{A}v^n)}{(\overline{A}v^n, \overline{A}v^n)}. \end{aligned} \tag{14}$$

Note that in this case, the condition $\overline{A}$, the orthogonalization of the vectors $r^{n+1,0}$ and v^n, is satisfied, i.e., $(r^{n+1,0}, \overline{A}v^n) = 0$. Since formulas (16) implement the simplest version of the least squares method, the corresponding algorithm is further denoted as P-LSM1. It can be generalized in an obvious way if in formulas (16) the vector $\hat{u}^{n+1}$ is replaced by $\hat{u}^{n+1,m}$ from (15), obtained after m corrections, which formally defines the PC ^{m} -LSM1 method.

e) The natural development of the considered least squares method is an increase in the number of solutions stored and used to select $u^{n+1,0}$ from the previous time steps $u^{n-1}, \cdots, u^{n-s}$. We describe this algorithm without using a predictor, denoting it as LSMs:

$$\begin{aligned} u^{n+1,0} &= u^n + c_1 v_1^n + \cdots + c_s v_s^n = u^n + V_n \vec{c}, \\ \vec{c} &= (c_1, \ldots, c_s)^T, \quad V_n = (v_1^n, \ldots, v_s^n) \in \Re^{N,s}, \\ r^{n+1,0} &= r^n - W_n \vec{c}, \quad W_n = \overline{A}V_n. \end{aligned} \tag{15}$$

Here the vectors $v_k, k = 1, \ldots, n$, can be defined in different ways as the differences of the already calculated approximations. For example, in the P-LSM2 method considered further in Sect. 4, we define $v^1 = \hat{u}^{n+1} - u^n$, $v^2 = u^{n-1} - u^n$. Hence, we obtain that the minimization of the norms $\|r^{n+1,0}\|_2$ is equivalent to the orthogonality relation $W_n^T r^{n+1,0} = 0$. This formally leads to the problem of calculating the normal solution of the overdetermined joint algebraic system $W_n \vec{c} = r^n$:

$$B\vec{c} \equiv W_n^T W_n \vec{c} = W_n^T r^n, \quad \vec{c} = (W_n^T W_n)^+ W_n^T r^n, \quad B \in \Re^{s,s}. \tag{16}$$

Here B^+ means the generalized inverse matrix [12], which in this case coincides with the inverse matrix B^{-1}, if W_n has full rank, which means the linear independence of the vectors $v_1^n, \ldots, v_s^n$. It should be noted that instead of (18), to determine the vector $\vec{c}$, one can use the relations obtained from the orthogonality condition
$V_n^T r^{n+1,0} = 0$:

$$\tilde{A}\vec{c} \equiv V_n^T \overline{A} V_n \vec{c} = V_n^T r^n, \quad \vec{c} = \tilde{A}^+ V_n^T r^n. \tag{17}$$

Moreover, the matrix $\tilde{A} \in \Re^{s,s}$ is called a low-rank approximation to $\overline{A}$. After calculating the vector $\vec{c}$ by formulas (18) or (19) (the question of their preference is still open), the initial approximation is determined from (17).

3.2 Iterative Algorithms for Implicit Schemes

We represent the matrix of system of equations (8) as the sum $\overline{A} = D + L + U$, where D, L and U are the diagonal (or block-diagonal), lower and upper triangular matrices, respectively. Following the method of symmetric sequential upper relaxation SSOR (or its block version BSSOR, see [5]), or incomplete factorization, to speed up iterations at each time step, we define preconditioning matrices:

$$B = \check{B}\hat{B}, \quad \check{B}^{-1} = \check{G}(G+L)^{-1}, \quad \hat{B}^{-1} = (G+U)^{-1}\check{G}, \quad G = \check{G}\hat{G}. \tag{18}$$

Here $G, \check{G}, \hat{G}$ are the easily invertible matrices selected for the optimization of the algorithm. In the simplest case, when D is a diagonal positive definite matrix, it is assumed

$$\hat{G}^{\frac{1}{2}} = \check{G}^{\frac{1}{2}} = G = \omega^{-1}D, \tag{19}$$

where $\omega \in [1, 2)$ is the upper relaxation parameter.

Consider Aizenshtat's modification for two-way SSOR preconditioning in the following form:

$$\tilde{A}\tilde{u} \equiv \check{B}^{-1}\check{G}A\check{G}^{-1}\hat{B}^{-1}\hat{B}^{-1}\tilde{u}\hat{B} = \tilde{f} \equiv \check{B}^{-1}f, \quad \tilde{u} = \hat{B}u. \tag{20}$$

Then the matrix of the preconditioned SLAE is written as

$$\begin{gathered}\tilde{A} = (I+\overline{L})^{-1} + (I+\tilde{U})^{-1} + (I+\overline{L})^{-1}(\tilde{D}-2I)(I+\tilde{U})^{-1}, \\ \tilde{D} = \hat{G}^{-1}D\check{G}^{-1}, \quad \overline{L} = \hat{G}^{-1}L\check{G}^{-1}, \quad \tilde{U} = \hat{G}^{-1}U\check{G}^{-1}.\end{gathered} \tag{21}$$

Here, for the parameter ω from (21) selected from the condition of approximate minimization of the condition number $cond(\tilde{A})$, the following formula demonstrates good practical results, confirmed in the simplest cases by theoretical estimates:

$$\omega = b - \frac{\sqrt{b^2 - 4ab}}{2a}, \quad a = (LD^{-1}Ue, e), \quad b = (De, e), \tag{22}$$

where $e = (1, \ldots, 1)^T$ is the vector with unit components. The corresponding approach, according to [5], will be called the incomplete Aizenshtat factorization IFE. Its distinguishing feature is the efficiency of implementing each iteration, since the multiplication of a vector by the matrix $\tilde{A}$ by the formula

$$\tilde{A}v = (I+\tilde{L})^{-1}[v + (D-2I)w] + w, \quad w = (I+\tilde{U})^{-1}v \tag{23}$$

requires almost as many arithmetic operations as multiplying by the original matrix A.

To solve preconditioned algebraic system (22), consider the iterative process of conjugate directions (see [5])

$$\begin{aligned} p^0 = r^0 = \tilde{f} - \tilde{A}u^0, \quad n = 0, 1, \cdots : \\ u^{n+1} = u^n + a_n p^n, \quad r^{n+1} = r^n - a_n \tilde{A} p^n, \end{aligned} \tag{24}$$

where u^0 is the arbitrary initial approximation, r^n is the residual vector, and p^n are the direction vectors with respect to which we assume that the following orthogonalization conditions are satisfied:

$$(A^\gamma p^k, Ap^n) = \rho_n \delta_{k,n}, \quad \rho_n = (A^\gamma p^n, Ap^n), \tag{25}$$

where $\delta_{k,n}$ is the Kronecker symbol and $\gamma = 0, 1$ for the conjugate gradient and conjugate residual methods, respectively. It is easy to check that when determining in (26) the iterative parameters by the formula

$$a_k = (r^0, A^\gamma p^k)/\rho_n, \quad k = 0, 1, \ldots, n, \tag{26}$$

residual functionals $\psi_\gamma^n = (\overline{A}^{\gamma-1} r^{n+1}, r^{n+1})$ are minimized in Krylov subspaces

$$\mathfrak{K}_{n+1}(p^0, \tilde{A}) = Span\{p^0, \tilde{A}p^0, \ldots, \tilde{A}^n p^0\}. \tag{27}$$

Due to the symmetry of the matrix $\tilde{A}$, orthogonality conditions (27) are satisfied if the direction vectors are determined using the two-term recursion

$$p^{n+1} = r^{n+1} + \beta_n p^n = \sigma_{n+1}/\sigma_n, \quad \sigma_n = (A^\gamma r^n, r^n), \tag{28}$$

in this case, it is expedient to calculate the iterative parameters α_n instead of (26) by the formula $\alpha_n = \sigma_n/\rho_n$. The criterion for the termination of iterations is the fulfillment of the condition

$$\|r^{n+1}\|_2^2 = (r^{n+1}, r^{n+1}) \le \varepsilon^2 (\tilde{f}, \tilde{f}), \tag{29}$$

where $\varepsilon << 1$ is the priori given value, the optimal definition of which, strictly speaking, requires a special analysis of the final error of the numerical solution in accordance with formula (9). The number of iterations $n(\varepsilon)$ required to satisfy condition (31) is determined by the inequality

$$n(\varepsilon) \le \frac{1}{2} |ln \frac{\varepsilon}{2}| cond(\tilde{A}^{\frac{1}{2}}) + 1. \tag{30}$$

3.3 Some Questions of Generalization of the Considered Approaches and Speed-up of Computations

Above, we presented the main directions for improving the efficiency of absolutely stable implicit grid approximations of resource-intensive multidimensional initial-boundary value problems based on the application of the universal least squares method. The above algorithms for the classical heat conduction equation

can naturally be transferred to more general formulations: diffusion-convective processes, nonlinear problems with phase transitions, interdisciplinary problems, an example of which is nonisothermal filtration in porous media. Generally speaking, the question of the optimal choice of initial approximations in the iterative implementation of implicit schemes is relevant when modeling any non-stationary processes and phenomena.

The second side of the issue is to apply the proposed approaches to grid equations of a higher order of accuracy, in relation to both spatial and temporal approximations. Here, in particular, promising discontinuous Galerkin algorithms [2,7] are actively developing. Such methods will lead to more complex calculations at each step in time, but also to a reduction in their total number and, as a consequence, to a decrease in communication losses, which is highly important in the light of the evolution of computer platforms.

Another potential opportunity to improve performance in the considered computational models is the use of known technologies for the multiple solution of SLAEs with different sequentially determined right-hand sides. Here, similarly, one can successfully apply the least squares method, using previously stored information to speed up iterations in Krylov subspaces (see the review on deflation algorithms in [4]).

Finally, we emphasize that all the approaches outlined above are based on vector operations that allow scalable parallelization by means of hybrid programming on various computer architectures with distributed and/or hierarchical shared memory. When solving large sparse SLAEs with orders of $10^{10} - 10^{11}$ and higher, additional calculations due to least squares methods are parallelized almost ideally with linear acceleration. This is achieved by means of either MPI message passing (here, the additive methods of the decomposition of areas are natural [9]), or multithreading (OPEN MP), or the vectorization of operations (command systems of AVX type). Results for specific applications here require special experimental applications, and general principles can be found in [6,9].

4 Examples of Numerical Experiments

We investigate the efficiency of the above algorithms experimentally using the results of the numerical solution of three-dimensional initial-boundary value problems for Eq. (1) with constant coefficients c, λ and with Dirichlet boundary conditions. The main goal in this case is to carry out a comparative analysis of the efficiency of the iterative algorithms described in clauses 3.1 and 3.2 for various methods of choosing the initial approximations. All calculations were carried out for a cubic computational domain $\Omega = [0,1]^3$ on a cubic grid with the number of steps along each coordinate $N_x = 16.32.64$. The time steps τ were also chosen constant, and their values and quantities were selected from the conditions of visual representations of the characteristics of the algorithms. All arithmetic operations in the experiments were performed with standard double precision. We do not dwell on the issues of the performance of software implementations and the execution time of the algorithms, since the main goal of

research in this case is the mathematical characteristics of the methods, i.e., accuracy and asymptotic estimates of their resource intensity.

The studies were carried out on two test examples with well-known analytical solutions, which were used to determine the initial and boundary conditions. In the first test, the exact solution is

$$u(x, y, z, t) = x(1 - x) + y(1 - y) + z(1 - z) + \frac{t^2}{2},$$

for which the spatial approximation error is zero, and the right-hand side of Eq. (1) is written as

$$f(x, y, z, t) = 2t + 6.$$

For the second example, the sought solution and the corresponding right-hand side are described by the following formulas:

$$u(x, y, z, t) = \sin(\pi x) \sin(\pi y) \sin(\pi z) t(T - t),$$
$$f(x, y, z, t) = (2(T - 2t) + 1.5\pi^2 t(T - t)) \sin(\pi x) \sin(\pi y) \sin(\pi z).$$

The tables below show the results of applying the preconditioned iterative conjugate gradient method described in Sect. 3.2 ($\gamma = 0$ in formulas (26)–(30)).

Table 1 shows the results of calculations for the 1st test problem for nine different space-time grids: $N_x = 16, 32, 64$; $N_t = 10, 20, 40$, using the iteration end criterion in (31) $\varepsilon = 10^{-3}, 10^{-5}$. We consider five ways of choosing the initial iterative approximations described in subparagraphs (a)–(e) from Sect. 3.1: O corresponds to $u^{n+1,0} = 0$, I^0 is the extrapolation of the form $u^{n+1,0} = u^n$, see item b), PI is the predictor with the definition $u^{n+1,0} = \hat{u}^{n+1}$, P-LSM1, P-LSM2, each of which in this case applies to the Crank–Nicholson scheme. The values n_1 and n_2 indicated in the cells of the table are the number of iterations averaged over time steps for $\varepsilon = 10^{-3}$ and $\varepsilon = 10^{-5}$, respectively, and $delta = \max_n \|u(t_{n+1}) - u^{n+1}\|_\infty$ is the uniform norm (maximum vector component) of the numerical solution error for $\varepsilon = 10^{-3}$. The content of Table 2 is similar, but for the second test problem.

The content of Tables 3 and 4 repeats Tables 1 and 2, but only for the implicit Euler scheme.

The analysis of the above results allows us to draw the following preliminary conclusions:

- the use of the least squares method LSM1 and even more so LSM2 can significantly reduce the number of iterations at time steps, with the fundamental possibility of constructing implicit non-iterative approximations;
- the effect obtained from the variational (algebraic) choice of initial approximations has approximately the same character for different values of the steps of the space-time grid and for different types of implicit schemes.

Table 1. Calculation results for the first test, $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-5}$, Crank–Nicholson scheme

Methods \ N_h		16			32			64		
		δ	n_1	n_2	δ	n_1	n_2	δ	n_1	n_2
O	Nt=10	8,45E-07	10	16	2,35E-06	15	24	4,62E-06	23	36
	Nt=20	4,44E-07	9	14	1,32E-06	13	21	2,66E-06	20	33
	Nt=40	2,25E-07	8	12	3,32E-07	12	18	3,01E-06	17	28
I^0	Nt=10	4,99E-07	9	15	1,42E-06	13	22	2,70E-06	19	33
	Nt=20	1,07E-06	7	13	4,99E-07	11	19	4,17E-06	16	29
	Nt=40	1,43E-06	6	10	1,71E-06	9	15	3,22E-06	13	23
PI	Nt=10	8,52E-07	7	13	1,88E-06	11	20	3,11E-06	16	30
	Nt=20	1,75E-06	6	11	1,38E-06	8	16	4,38E-06	12	25
	Nt=40	6,58E-07	4	9	1,06E-06	6	13	6,97E-06	9	19
P-LSM1	Nt=10	1,35E-06	3	3	7,97E-06	5	5	5,63E-06	10	8
	Nt=20	1,27E-06	2	2	1,44E-06	4	4	3,79E-06	8	8
	Nt=40	1,50E-06	2	2	3,56E-06	3	3	6,63E-06	6	6
P-LSM2	Nt=10	9,07E-07	1	2	6,04E-06	1	2	6,30E-06	2	5
	Nt=20	1,13E-06	0	2	2,54E-06	1	2	5,00E-06	1	2
	Nt=40	1,42E-06	0	1	1,46E-06	1	1	4,97E-06	1	1

Table 2. Calculation results for the second test, $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-5}$, Crank–Nicholson scheme

Methods \ N_h		16			32			64		
		δ	n_1	n_2	δ	n_1	n_2	δ	n_1	n_2
O	Nt=10	7,46E-04	8	13	1,86E-04	12	20	4,65E-05	18	30
	Nt=20	7,48E-04	7	11	1,87E-04	10	17	4,67E-05	15	26
	Nt=40	7,49E-04	6	9	1,87E-04	8	14	4,68E-05	12	20
I^0	Nt=10	7,46E-04	8	13	1,86E-04	11	19	4,65E-05	17	29
	Nt=20	7,48E-04	6	10	1,87E-04	9	16	4,67E-05	13	24
	Nt=40	7,50E-04	5	8	1,87E-04	6	12	4,68E-05	9	18
PI	Nt=10	7,46E-04	7	12	1,86E-04	10	18	4,65E-05	15	28
	Nt=20	7,48E-04	5	9	1,87E-04	7	14	4,67E-05	11	21
	Nt=40	7,50E-04	3	7	1,87E-04	5	10	4,68E-05	7	15
P-LSM1	Nt=10	7,46E-04	2	6	1,86E-04	4	9	4,66E-05	9	15
	Nt=20	7,48E-04	2	4	1,87E-04	3	6	4,67E-05	6	9
	Nt=40	7,49E-04	1	3	1,87E-04	2	4	4,68E-05	5	6
P-LSM2	Nt=10	7,46E-04	1	5	1,86E-04	2	7	4,65E-05	3	12
	Nt=20	7,48E-04	0	3	1,87E-04	1	5	4,67E-05	1	8
	Nt=40	7,49E-04	0	2	1,87E-04	0	3	4,68E-05	0	5

A further increase in the performance of algorithms for solving SLAEs in multidimensional initial-boundary value problems can be developed in various directions. The first is the optimization of the considered initial approximations $u^{n+1,0}$. The second is to use the SLAE solution with different right-hand sides, see the overview in [4]. The third is the parallelization of algorithms by

Table 3. Calculation results for the first test, $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-5}$, Euler's scheme

Methods \ N_h		16			32			64		
		δ	n_1	n_2	δ	n_1	n_2	δ	n_1	n_2
O	Nt=10	1,88E-06	11	17	1,56E-06	17	25	5,05E-06	26	38
	Nt=20	1,61E-06	10	16	4,46E-06	15	24	8,78E-06	23	36
	Nt=40	8,86E-07	9	14	2,50E-06	13	21	5,17E-06	20	33
I^0	Nt=10	1,56E-06	10	16	4,63E-06	14	24	7,64E-06	21	36
	Nt=20	1,10E-06	8	14	2,61E-06	12	21	3,55E-06	18	32
	Nt=40	9,78E-07	7	12	2,85E-06	10	18	6,67E-06	15	28
PI	Nt=10	3,89E-07	8	14	2,19E-06	12	21	7,78E-06	17	32
	Nt=20	9,29E-07	6	12	1,95E-06	9	19	4,98E-06	13	28
	Nt=40	3,57E-06	5	10	1,81E-06	7	15	5,83E-06	10	22
P-LSM1	Nt=10	8,00E-07	3	4	2,91E-06	5	6	6,38E-06	9	9
	Nt=20	1,75E-06	2	2	2,76E-06	4	4	5,74E-06	7	6
	Nt=40	1,84E-06	1	1	3,01E-06	3	3	4,34E-06	5	5
P-LSM2	Nt=10	2,98E-06	1	3	9,8E-06	1	4	1,07E-05	2	8
	Nt=20	2,03E-06	0	2	3,14E-06	0	3	2,05E-05	0	4
	Nt=40	2,55E-06	0	1	6,25E-06	0	2	1,80E-05	0	2

Table 4. Calculation results for the second test, $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-5}$, Euler's scheme

Methods \ N_h		16			32			64		
		δ	n_1	n_2	δ	n_1	n_2	δ	n_1	n_2
O	Nt=10	0,032	10	15	3,18E-02	14	22	3,18E-02	21	33
	Nt=20	1,74E-02	8	13	1,71E-02	12	20	1,71E-02	18	30
	Nt=40	9,15E-03	7	11	8,90E-03	10	17	8,84E-03	15	26
I^0	Nt=10	0,032	9	14	3,18E-02	12	21	3,18E-02	18	31
	Nt=20	1,74E-02	7	12	1,71E-02	10	18	1,71E-02	15	28
	Nt=40	9,15E-03	6	10	8,90E-03	8	15	8,84E-03	12	23
PI	Nt=10	0,032	8	14	3,18E-02	11	20	3,18E-02	17	30
	Nt=20	1,74E-02	6	11	1,71E-02	8	17	1,71E-02	13	26
	Nt=40	9,15E-03	4	9	8,90E-03	6	13	8,84E-03	9	20
P-LSM1	Nt=10	0,032	4	10	3,18E-02	6	16	3,18E-02	12	26
	Nt=20	1,74E-02	2	7	1,71E-02	4	12	1,71E-02	7	20
	Nt=40	9,15E-03	1	5	8,90E-03	2	8	8,84E-03	4	13
P-LSM2	Nt=10	0,032	3	9	3,18E-02	5	13	3,18E-02	10	20
	Nt=20	1,74E-02	2	6	1,71E-02	3	9	1,71E-02	7	14
	Nt=40	9,15E-03	1	4	8,90E-03	1	6	8,84E-03	3	9

using additive methods for decomposing domains. It is also noteworthy that these directions of development (possibly in combination with approximation approaches) can be carried over to more general formulations of problems: in the presence of convection, nonlinear effects, etc.

5 Conclusion

The performed comparative analysis of iterative methods for solving algebraic systems with sparse matrices of high orders that arise in the implementation of implicit approximations of multidimensional initial-boundary value problems demonstrates the high efficiency of the algorithms while maintaining the absolute stability of numerical integration over time. The number of iterations at each time step is significantly reduced by choosing an initial approximation with the sequential use of already calculated solutions using the least squares method. The efficiency of the proposed approaches is illustrated by the results of experimental studies on a representative series of methodological problems using preconditioned iterative incomplete factorization processes in Krylov subspaces. The presented test results on accelerating computations of the proposed algorithms indicate their efficiency for supercomputers with distributed and hierarchical shared memory. The issues of transferring the proposed approaches to more general problem statements are discussed. A further increase in the performance of the algorithms is possible when using scalable parallelization based on additive methods for the decomposition of regions in Krylov subspaces. At the same time, an additional significant increase in the speed of computations can be achieved due to parallel methods of the decomposition of domains, as well as through the use of techniques for the multiple solution of SLAEs with sequentially determined right-hand sides.

References

1. Bychin, I.V., Gavrilenko, T.V., Galkin, V.A., et al.: Numerical modeling of 3D heat conduction problems with phase transitions on computing systems with distributed memory. Cybern. Herald **3**(19), 84–93 (2015)
2. Gander, M.J., Neumuller, M.: Analysis of a new space-time parallel multigrid algorithm for parabolic problems. SIAM J. Sci. Comput. **38**, A2173–A2208 (2016)
3. Gladkikh, V., Ilin, V.P., Petukhov, A.V., et al.: Numerical modeling of nonstationary heat problems in a two-phase medium. J. Phys. Conf. Ser. **1715**, 012002 (2020)
4. Gurieva, Y.L., Ilin, V.P.: Conjugate direction methods for multiple solution of slaes. J. Math. Sci. **255**, 231–241 (2021)
5. Ilin, V.P.: Finite element methods and technologies. ICMiMG SB RAS (2007)
6. Ilin, V.P.: Problems of parallel solution of large systems of linear algebraic equations. J. Math. Sci. **216**, 795–804 (2016)
7. Ilin, V.P.: Math modeling. Part 1 continuous and discrete models. Publisher SB RAS, p. 428 (2017)
8. Ilin, V.P.: Methods for solving ordinary differential equations. NSU Publishing House (2017)
9. Ilin, V.P.: Multi-preconditioned domain decomposition methods in the Krylov subspaces. In: Dimov, I., Faragó, I., Vulkov, L. (eds.) Numerical Analysis and Its Applications. LNCS, vol. 10187, pp. 95–106. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57099-0_9

10. Ilin, V.: High-performance computation of initial boundary value problems. In: Sokolinsky, L., Zymbler, M. (eds.) PCT 2018. CCIS, vol. 910, pp. 186–199. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99673-8_14
11. Roy, T., Jönsthövel, T.B., Lemon, C., et al.: A constrained pressure-temperature residual (CPTR) method for non-isothermal multiphase flow in porous media. SIAM J. Sci. Comput. **42**, B1014–B1040 (2020)
12. Saad, Y.: Iterative Methods for Sparse Linear Systems, 2nd edn. SIAM (2003). https://doi.org/10.1137/1.9780898718003
13. Vabishchevich, P.N., Samarsky, A.A.: Computational heat transfer. Editorial URSS (2003)
14. Vaganova, N.A., Filimonov, M.Y.: Simulation of cooling devices and effect for thermal stabilization of soil in a cryolithozone with anthropogenic impact. In: Dimov, I., Faragó, I., Vulkov, L. (eds.) FDM 2018. LNCS, vol. 11386, pp. 580–587. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-11539-5_68
15. Vasiliev, V.I., Vasilieva, M.V., Grigoriev, A.V., et al.: Numerical modelling of heat and mass transfer processes in the permafrost zone. NEFU Publishing House, Yakutsk (2019)