

A Method for Solving Grid Equations for Hydrodynamic Problems in Flat Areas

A. I. Sukhinov^{a,*}, A. E. Chistyakov^{a,**}, A. V. Nikitina^{a,b,***},
A. M. Atayan^{a,****}, and V. N. Litvinov^{a,*****}

^a Don State Technical University, Rostov-on-Don, Russia

^b Southern Federal University, Rostov-on-Don, Russia

*e-mail: sukhinov@gmail.com

**e-mail: cheese_05@mail.ru

***e-mail: nikitina.vm@gmail.com

****e-mail: atayan24@mail.ru

*****e-mail: litvinovvn@rambler.ru

Received November 14, 2022; revised November 14, 2022; accepted December 12, 2022

Abstract—This paper discusses the numerical implementation of the mathematical model of the hydrodynamic process in the computational domain with an “extended geometry,” when its characteristic dimensions in the horizontal direction significantly exceed the vertical dimension. This is the typical property of a shallow water body or coastal system, which requires the development of specialized methods for solving the problems that arise in the process of discretizing grid equations. The explicit-implicit scheme has proved itself effective in solving the problem of transport in a shallow water body. The transition between time layers can be considered as an iterative process for solving the problem of diffusion-convection to settle. This idea forms the base for the formation of a preconditioner in the proposed method for solving grid equations obtained by approximating hydrodynamic problems in areas with extended geometry. A numerical experiment is carried out with the developed software module, which makes it possible to estimate the norm of the residual vector obtained by solving the grid equations of the pressure calculation problem based on the modified alternating triangular method (MATM) and the method for solving grid equations with a tridiagonal preconditioner, taking into account the hydrostatic approximation. According to the specifics of the developed method, it is effective in solving problems of aquatic ecology in the case of the computational domain, when its horizontal dimensions significantly exceed the vertical dimensions.

Keywords: mathematical modeling of hydrodynamic processes, computational domain with extended geometry, grid equations, explicit-implicit scheme, modified alternating triangular method, method for solving grid equations with a tridiagonal preconditioner

DOI: 10.1134/S2070048223050113

INTRODUCTION

Numerical implementation of spatial-three-dimensional mathematical models, including differential equations in partial derivatives, can be performed by the finite difference method. We single out its two main stages: at the first stage, a difference approximation of the system of differential equations of mathematical physics on a grid is carried out—the development of a difference scheme with specified properties (stability, accuracy)—of a discrete analog of the original continuous mathematical model; the second stage consists of solving systems of linear algebraic equations (SLAEs) of a special type (with high-order matrices having a band structure and ill-conditioned) using the most modern computer technology, including supercomputers, MCS, and graphics accelerators. The use of classical methods of higher algebra for solving such SLAEs is often impractical due to the fact that these methods require a large amount of computational work and necessitate information storage. In recent decades, for the numerical implementation of computationally labor-intensive problems of continuum mechanics, aerodynamics, hydrodynamics, and biological kinetics, methods have been actively developed that take into account the specifics of the problem posed, which make it possible to reduce the number of operations compared to the

standard methods. In [1, 2], the iterative method is treated as an operator-difference scheme with an operator in the Hilbert space of grid functions.

Selection of the optimal parameters for which the rate of convergence of the iterative method used to solve the emerging SLAE is the highest is a separate problem. Various iterative methods used to solve grid equations are compared using the canonical form of iterative schemes, which allows singling out the operators responsible for the convergence of the computational process. An iterative method for solving a problem is selected based on studying the nature of computational stability and estimating the rate of convergence [3]. Iterative methods of the variational type, including the methods of conjugate gradients, steepest descent, minimum residuals, and minimum corrections, are widely used in the implementation of mathematical models of hydrophysics.

Specialized libraries and application packages are being developed for the implementation of iterative methods for a nonself-adjoint, degenerate, and indefinite SLAE operator, triangular methods with the inversion of a triangular matrix at the subsequent iteration, including the upper relaxation method, the alternating direction method, and the method for solving nonlinear difference equations using curvilinear systems coordinates. The highly efficient alternating triangular method proposed and developed in [3, 4] is universal and is used to solve the Dirichlet problem for the Poisson equation with a strongly varying coefficient.

In [5], a nonstandard approach is used to increase the stability margin of explicit schemes in the mathematical modeling of multiphase flows in an underground space. The core of the approach lies in the use of regularized schemes oriented on supercomputers. Part of the research discusses the development of an original kinetically based approach to model filtration processes, including a mathematical model built by analogy with a quasi-gasdynamic system of equations and allowing implementation by explicit numerical methods.

In [6], for the multiple solution of SLAEs, the methods of conjugate gradients and conjugate residuals are considered. The SLAEs under consideration have the same matrices, but different successively determined right-hand sides. When solving the second and subsequent SLAEs, deflation algorithms are used to speed up the iterative processes. In these algorithms, the vectors obtained during the solution of the first system are used as basis direction vectors.

When discretizing the problems of aero- and hydrodynamics in the case of a complex shape of the computational domain, optimal computational grids need to be constructed. In [7], an algorithm for constructing a tetrahedral mesh is proposed. The difference between the algorithm and other currently existing tetrahedral algorithms is that the mesh is guaranteed to be built at the given time. When describing the main stages of the algorithm, an estimate of its computational complexity is given.

The work [8] is devoted to the development of new applications of matrix methods. Particular attention is paid to methods of separating variables based on a special decomposition of matrices and tensors, the development of implementing algorithms and their applications in solving multidimensional problems of computational mathematics, data analysis, and machine learning.

The work [9] considers the numerical implementation of mathematical models in medicine. An analysis of the solutions for the personalization of mathematical models in problems of clinical cardiology, including a virtual assessment of the hemodynamic significance of coronary artery stenoses, an assessment of changes in the systemic blood flow after the hemodynamic correction of complex heart defects, and the calculation of the coaptation characteristics of the reconstructed aortic valve, is given.

In [10], the preconditioner of the block incomplete inverse triangular expansion of the first order “by value” BIIC-IC1 is considered for solving an SLAE with a symmetric positive definite matrix. The BIIC-IC1 preconditioner is constructed and inverted using the MPI+OpenMP technology, while the number of blocks in the preconditioner is a multiple of the number of used processors and used threads. The works [11, 12] study the creation of specialized methods for the implementation of mathematical models.

The complex shape of the computational domain and the need to develop difference schemes that adapt to the solution of the problem significantly complicate the computation process. The “extended geometry”—the characteristic dimensions of the computational domain in the horizontal direction significantly exceed its vertical dimension—is the typical property of a shallow water body or a coastal system. In the numerical implementation of mathematical models of the hydrodynamics of domains of the described shape, grid equations of a certain type appear in the process of their discretization. As a result, there is a need to develop specialized methods for their solution. Classical methods for solving such problems are not suitable. In [13], to solve the problem of transport in a shallow water body, it was proposed to use an explicit-implicit scheme (an explicit scheme in the horizontal directions and a symmetric scheme with weights in the vertical direction), which makes it possible to numerically solve the problem

5–30 times faster than the explicit scheme. The transition between time layers can be considered as an iterative process for solving the problem of diffusion-convection to settle. This idea formed the base for the formation of a preconditioner in the proposed method for solving grid equations obtained by approximating hydrodynamic problems in areas with an extended geometry.

1. PROBLEM STATEMENT

The system of Navier–Stokes equations and the continuity equation with the corresponding initial and boundary conditions are used to describe hydrodynamic problems [14, 15]. The problem of hydrodynamics with respect to the time variable can be approximated based on splitting schemes with respect to physical processes [16]. In this case, the most computationally time-consuming part will be the calculation of the pressure field based on the Poisson equation. The use of regularizers [5] in the continuity equation entails a change in the method of calculating the pressure field, which will be calculated based on the wave equation of the form [17]:

$$\frac{1}{c^2} P'' - \Delta P = f, \quad (1)$$

where Δ is the Laplace operator or Laplacian, c is the speed of sound, and f is the given function.

It follows from this approach that pressure cannot propagate faster than the speed of the shock front (in the linear approximation of the speed of sound). If we do not take into account the time of the collision between molecules in the continuity equation, then we obtain the Poisson equation [18]. Applying the Gauss–Ostrogradsky theorem for this equation, we obtain an instantaneous propagation of the pressure field from the sources of the field to the sinks (which is not physical). This approach is less laborious from a computational point of view, since the discrete analog of the wave equation has diagonal dominance, in contrast to the discrete analog of the Poisson equation, as a result of which the conditionality number of the operator, which affects the rate of convergence of the iterative method, decreases. It should also be noted that when calculating the pressure based on the Poisson equation in the case of boundary conditions in the Neumann form, the determinant of the system will be zero, and as a result, the problem of the existence of a solution obtained in the process of discretization of the grid equation arises. When using regularizers [5], there is no such problem.

For Eq. (1), we set the initial conditions in the following form:

$$P|_{t=0} = P_0, \quad P_t|_{t=0} = 0. \quad (2)$$

If the pressure field is known, then we will use the boundary conditions of the first kind:

$$P|_{(x,y,z) \in \gamma} = P_1.$$

If the flow through the boundary is known, then we will use the boundary conditions of the second (for $\alpha_n = 0$) or the third kind (when $\alpha_n > 0$):

$$P_n|_{(x,y,z) \in \gamma} = \alpha_n P + \beta_n, \quad (3)$$

where \mathbf{n} is the normal directed inside the computational domain, γ is the boundary of the computational domain, and $\alpha = \{\alpha_x, \alpha_y, \alpha_z\}$, $\beta = \{\beta_x, \beta_y, \beta_z\}$ are the given vectors.

Task (1)–(3) is considered in region Γ , the linear dimensions of which along the vertical are significantly smaller than the dimensions along the horizontal coordinate directions.

We will assume that the computational domain Γ is inscribed in a parallelepiped, which we will cover with a uniform computational grid:

$$\begin{aligned} \bar{w}_i &= \{t^n = n\tau, \quad x_i = ih_x, \quad y_j = jh_y, \quad z_k = kh_z; \\ n &= \overline{0, N_t}, \quad i = \overline{0, N_x - 1}, \quad j = \overline{0, N_y - 1}, \quad k = \overline{0, N_z - 1}; \\ N_t \tau &= T, \quad (N_x - 1)h_x = l_x, \quad (N_y - 1)h_y = l_y, \quad (N_z - 1)h_z = l_z, \end{aligned}$$

where τ is the time step; h_x , h_y , and h_z are steps in space; N_t is the number of time layers; T is the upper time limit; N_x , N_y , and N_z are number of nodes in space; and l_x , l_y , and l_z are boundaries in space.

To describe the geometry of the computational domain in the discrete case, the parameter $\sigma_{i,j,k} = V_{\Omega_{i,j,k}} / V_{D_{i,j,k}}$ ($V_{\Omega_{i,j,k}}$ is the volume of the cell filled with the medium and $V_{D_{i,j,k}} = h_x h_y h_z$ is the total

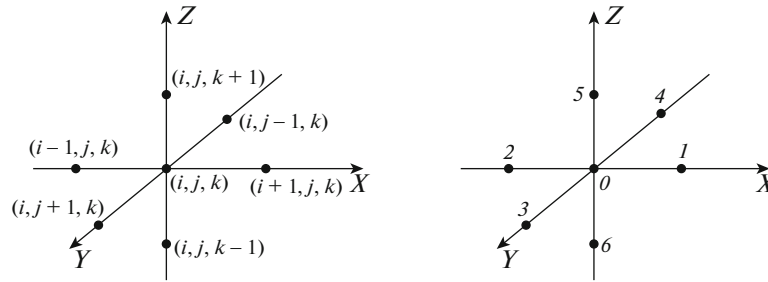


Fig. 1. Seven-point stencil for solving grid equations.

volume of the cell), which describes the fullness of the cell (i, j, k) , is introduced. To construct a difference scheme, we also need the coefficients $q_i, i = \overline{0, 6}$, describing the degree to which the control areas are filled [19]. The difference scheme for Eq. (1) with boundary conditions (3) can be written as

$$\begin{aligned}
 & q_{0,i,j,k} \frac{P_{i,j,k}^{n+1} - 2P_{i,j,k}^n + P_{i,j,k}^{n-1}}{(c\tau)^2} - q_{1,i,j,k} \frac{P_{i+1,j,k}^{n+1} - P_{i,j,k}^{n+1}}{h_x^2} + q_{2,i,j,k} \frac{P_{i,j,k}^{n+1} - P_{i-1,j,k}^{n+1}}{h_x^2} \\
 & + |q_{1,i,j,k} - q_{2,i,j,k}| \frac{\alpha_x P_{i,j,k}^{n+1} + \beta_x}{h_x} - q_{3,i,j,k} \frac{P_{i,j+1,k}^{n+1} - P_{i,j,k}^{n+1}}{h_y^2} + q_{4,i,j,k} \frac{P_{i,j,k}^{n+1} - P_{i,j-1,k}^{n+1}}{h_y^2} \\
 & + |q_{3,i,j,k} - q_{4,i,j,k}| \frac{\alpha_y P_{i,j,k}^{n+1} + \beta_y}{h_y} - q_{5,i,j,k} \frac{P_{i,j,k+1}^{n+1} - P_{i,j,k}^{n+1}}{h_z^2} + q_{6,i,j,k} \frac{P_{i,j,k}^{n+1} - P_{i,j,k-1}^{n+1}}{h_z^2} \\
 & + |q_{5,i,j,k} - q_{6,i,j,k}| \frac{\alpha_z P_{i,j,k}^{n+1} + \beta_z}{h_z} = q_{0,i,j,k} f_{i,j,k}.
 \end{aligned} \tag{4}$$

The difference scheme (4) approximates the wave equation (1) and stores information about the boundary conditions, which can be different for each of the spatial directions, and about the geometry of the computational domain, which can have a complex dynamically tunable shape.

When discretizing models of mathematical physics, especially hydrodynamics, we obtain a system of equations. Each equation of the system can be represented in canonical form, and we will use a seven-point template (Fig. 1):

$$A(m_0)P(m_0) - \sum_{r=1}^6 B(m_0, m_r)P(m_r) = F(m_0),$$

where $m_0(x_i, y_j, z_k)$ is the center of the template, $M'(P) = \{m_1(x_{i+1}, y_j, z_k), m_2(x_{i-1}, y_j, z_k), m_3(x_i, y_{j+1}, z_k), m_4(x_i, y_{j-1}, z_k), m_5(x_i, y_j, z_{k+1}), m_6(x_i, y_j, z_{k-1})\}$ is the neighborhood of the center, $A \equiv A(m_0)$ is the coefficient of the center of the template, $B_r \equiv B(m_0, m_r), r \in \overline{1, 6}$, are the coefficients of the neighborhood of the center of the template, F is the vector of right parts, and P is the calculated vector.

In this formulation of the problem, the coefficients of the grid equations and the right-hand side take the form

$$\begin{aligned}
 & B_{1,i,j,k} = q_{1,i,j,k}/h_x^2, \quad B_{2,i,j,k} = q_{2,i,j,k}/h_x^2, \quad B_{3,i,j,k} = q_{3,i,j,k}/h_y^2, \\
 & B_{4,i,j,k} = q_{4,i,j,k}/h_y^2, \quad B_{5,i,j,k} = q_{5,i,j,k}/h_z^2, \quad B_{6,i,j,k} = q_{6,i,j,k}/h_z^2, \\
 & A_{i,j,k} = \frac{q_{0,i,j,k}}{(c\tau)^2} + \sum_{r=1}^6 B_{r,i,j,k} + \frac{|q_{1,i,j,k} - q_{2,i,j,k}| \alpha_x}{h_x} + \frac{|q_{3,i,j,k} - q_{4,i,j,k}| \alpha_y}{h_y} + \frac{|q_{5,i,j,k} - q_{6,i,j,k}| \alpha_z}{h_z}, \\
 & F_{i,j,k} = q_{0,i,j,k} \left(f_{i,j,k} + \frac{2P_{i,j,k}^n - P_{i,j,k}^{n-1}}{(c\tau)^2} \right) - \frac{|q_{1,i,j,k} - q_{2,i,j,k}| \beta_x}{h_x} - \frac{|q_{3,i,j,k} - q_{4,i,j,k}| \beta_y}{h_y} - \frac{|q_{5,i,j,k} - q_{6,i,j,k}| \beta_z}{h_z}.
 \end{aligned}$$

The transition from a 3D mesh node representation (i, j, k) to a one-dimensional record (node number) is carried out according to the following formula:

$$m_0 = k + jN_z + iN_zN_y.$$

The numbers of nodes located in the vicinity of the center of the template $m_i, i = \overline{1,6}$, is calculated by the formulas

$$\begin{aligned} m_1 &= m_0 + N_zN_y, & m_2 &= m_0 - N_zN_y, & m_3 &= m_0 + N_z, \\ m_4 &= m_0 - N_z, & m_5 &= m_0 + 1, & m_6 &= m_0 - 1. \end{aligned}$$

2. METHOD FOR SOLVING GRID EQUATIONS

We write the grid equation (4) in the matrix form

$$Ax = f, \quad (5)$$

where A is a linear, self-adjoint, positive definite operator ($A = A^* > 0$). Imagine the operator A as

$$A = D + \Lambda_x + \Lambda_y + \Lambda_z, \quad (6)$$

where D is the diagonal operator, $\Lambda_x = (1/h_x^2)\Lambda_{D,x} \otimes E_y \otimes E_z$; $\Lambda_y = (1/h_y^2)E_x \otimes \Lambda_{D,y} \otimes E_z$; $\Lambda_z = (1/h_z^2)E_x \otimes E_y \otimes \Lambda_{D,z}$; \otimes is the symbol denoting the Kronecker product; E_x, E_y, E_z are identity matrices with dimensions $N_x \times N_x, N_y \times N_y$, and $N_z \times N_z$, respectively; and $\Lambda_{D,x}, \Lambda_{D,y}, \Lambda_{D,z}$ are square tridiagonal matrices containing the coefficients of the grid equations for the terms describing the finite differences for second-order partial derivatives along the corresponding axes.

Operator $\Lambda_{D,x}$ in the case of boundary conditions of the first kind will be written in the form

$$(\Lambda_{D,x})_{i,j} = \begin{cases} 2, & \text{if } i = j; \\ -1, & \text{if } |i - j| = 1; \\ 0, & \text{if } |i - j| > 1; \end{cases} \quad \Lambda_{D,x} = \begin{pmatrix} 2 & -1 & 0 & \dots \\ -1 & 2 & -1 & \dots \\ 0 & -1 & 2 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}. \quad (7)$$

Operator $\Lambda_{D,x}$ in the case of boundary conditions of the second kind ($\alpha_x = 0$) and third kind ($\alpha_x > 0$) is written in the following form:

$$(\Lambda_{D,x})_{i,j} = \begin{cases} 1 + \sigma_x, & \text{if } (i + j = 2) \cup (i + j = 2N); \\ 2, & \text{if } (i = j) \cap (2 < i + j < 2N); \\ -1, & \text{if } |i - j| = 1; \\ 0, & \text{if } |i - j| > 1; \end{cases} \quad \Lambda_{D,x} = \begin{pmatrix} 1 + \sigma_x & -1 & 0 & \dots \\ -1 & 2 & -1 & \dots \\ 0 & -1 & 2 & \dots \\ \dots & \dots & \dots & \dots \end{pmatrix}. \quad (8)$$

Here, $\sigma_x = \alpha_x h_x$. Operators $\Lambda_{D,y}, \Lambda_{D,z}$ will be written analogously.

To find a solution to problem (7), we will use the implicit iterative process

$$B(x^{n+1} - x^n)/\tau_{n+1} + Ax^n = f. \quad (9)$$

In Eq. (9) n is the iteration number, $\tau_{n+1} > 0$ is the iterative parameter, B is a preconditioner, whose conversion in (9) should be much simpler than the direct conversion of the original operator A in (5). In the case of a stationary ($\tau_{n+1} = \tau$) iterative process, (9) can be represented as

$$x^{n+1} = x^n - \tau B^{-1}(Ax^n - f). \quad (10)$$

At each iteration, the resulting solution x^n will differ from the exact solution x by the amount of the error z^n : $x^n = x + z^n$. Then Eq. (10) with respect to the error, taking into account (5), will be written as

$$z^{n+1} = z^n - \tau B^{-1}Az^n.$$

We replace the variables $z^n = B^{-1/2}y^n$ and multiply both sides of the last expression by $B^{1/2}$. As a result, we get $y^{n+1} = y^n - \tau B^{-1/2}AB^{-1/2}y^n$. We assume $C = B^{-1/2}AB^{-1/2}$, $C = C^* > 0$; then

$$y^{n+1} = (E - \tau C)y^n. \tag{11}$$

The iterative process converges at [3]

$$-\rho E \leq E - \tau C \leq \rho E, \quad \rho < 1, \tag{12}$$

where ρ is the parameter describing the rate of convergence of the iterative method.

Inequality (12) can be represented as

$$((1 - \rho)/\tau)E \leq C \leq ((1 + \rho)/\tau)E.$$

Parameters ρ and τ are found from the conditions $(1 - \rho)/\tau = \lambda_{\min}$, $(1 + \rho)/\tau = \lambda_{\max}$:

$$\tau = 2/(\lambda_{\min} + \lambda_{\max}), \quad \rho = (v - 1)/(v + 1), \quad v = \lambda_{\max}/\lambda_{\min}. \tag{13}$$

Here λ_{\min} and λ_{\max} are the minimum and maximum values of the operator's eigenvalues C , and v is the conditionality number of the operator C .

We will form the preconditioner in the following form:

$$B = D + \omega D_{xy} + \Lambda_z, \tag{14}$$

where D_{xy} is the operator composed of the diagonal part of the operator $\Lambda_{xy} = \Lambda_x + \Lambda_y$ and ω is the determined parameter ($\omega > 0$). Let us estimate the maximum eigenvalue of operator C taking into account $C = B^{-1/2}AB^{-1/2}$ and $B^{-1/2}x = y$:

$$\lambda_{\max}(C) = \max_{|y|>0} \frac{(Cx, x)}{(x, x)} = \max_{|y|>0} \frac{(B^{-1/2}Ay, B^{1/2}y)}{(B^{1/2}y, B^{1/2}y)} = \max_{|y|>0} \frac{(Ay, y)}{(By, y)}.$$

We write the resulting expression, taking into account expressions (6) and (14):

$$\lambda_{\max}(C) = \max_{|y|>0} \frac{((D + \Lambda_{xy} + \Lambda_z)y, y)}{((D + \omega D_{xy} + \Lambda_z)y, y)} = 1 + \max_{|y|>0} \left(\frac{((\Lambda_{xy} - \omega D_{xy})y, y)}{((D + \omega D_{xy} + \Lambda_z)y, y)} \right).$$

An estimate of the maximum eigenvalue follows from the inequalities $\delta D_{xy} \leq \Lambda_{xy} \leq \Delta D_{xy}$, $\delta \geq 0$, $\Delta \leq 2$:

$$\begin{aligned} \lambda_{\max}(C) &\leq \begin{cases} 1 + \frac{(\Delta - \omega)}{\lambda_{\max}(D_{xy}^{-1/2}BD_{xy}^{-1/2})}, & \omega \geq \Delta, \\ 1 + \frac{(\Delta - \omega)}{\lambda_{\min}(D_{xy}^{-1/2}BD_{xy}^{-1/2})}, & \omega < \Delta, \end{cases} \\ &= \begin{cases} \frac{\Delta + \lambda_{\max}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}{\omega + \lambda_{\max}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}, & \omega \geq \Delta, \\ \frac{\Delta + \lambda_{\min}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}{\omega + \lambda_{\min}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}, & \omega < \Delta. \end{cases} \end{aligned} \tag{15}$$

We estimate the minimum eigenvalue

$$\lambda_{\min}(C) \geq \begin{cases} \frac{\delta + \lambda_{\min}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}{\omega + \lambda_{\min}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}, & \omega \geq \delta, \\ \frac{\delta + \lambda_{\max}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}{\omega + \lambda_{\max}(D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_zD_{xy}^{-1/2})}, & \omega < \delta. \end{cases} \tag{16}$$

We designate $S = D_{xy}^{-1}D + D_{xy}^{-1/2}\Lambda_z D_{xy}^{-1/2}$, then an estimate of the conditionality number of operator C is written:

$$\nu(C) \leq \begin{cases} \frac{\Delta + \lambda_{\max}(S)}{\delta + \lambda_{\min}(S)} \left(1 - \frac{\lambda_{\max}(S) - \lambda_{\min}(S)}{\omega + \lambda_{\max}(S)} \right), & \Delta < \omega, \\ \frac{\Delta + \lambda_{\min}(S)}{\delta + \lambda_{\min}(S)}, & \delta \leq \omega \leq \Delta, \\ \frac{\Delta + \lambda_{\min}(S)}{\delta + \lambda_{\max}(S)} \left(1 + \frac{\lambda_{\max}(S) - \lambda_{\min}(S)}{\omega + \lambda_{\min}(S)} \right), & \omega < \delta. \end{cases} \quad (17)$$

It follows from the resulting estimate that the conditionality number of operator C is minimal at $\delta \leq \omega \leq \Delta$. Next parameter ω will be taken equal to one. The operation of stationary iterative methods requires a priori information about the values of the maximum and minimum eigenvalues of the operators. In the case of choosing a preconditioner in form (14) for an arbitrary geometry of the computational domain, the complexity is caused by estimating the minimum eigenvalues of the operator.

3. EIGENVALUES AND EIGENVECTORS OF THE OPERATOR Λ_D

In [3], in order to find the eigenvalues of the operator of the second difference derivative Λ_D , boundary value problems are formulated, represented by the grid equation

$$y_{xx} + \lambda y = 0, \quad y_{xx,i} = (y_{i+1} - 2y_i + y_{i-1})/h^2; \quad (18)$$

in the case of boundary conditions of the first kind,

$$y_0 = y_N = 0; \quad (19)$$

in the case of boundary conditions of the second kind,

$$(2/h)y_{x,0} + \lambda y_0 = -(2/h)y_{x,N} + \lambda y_N = 0; \quad (20)$$

and in a mixed boundary value problem (of the first and second kind),

$$y_0 = -(2/h)y_{x,N} + \lambda y_N = 0. \quad (21)$$

When solving applied problems (for example, hydrodynamics), it becomes necessary to find the eigenvalues of the second-order difference derivative operator in the case of boundary conditions of the third kind.

3.1. The First Boundary Value Problem for Eigenvalues

It is known that trigonometric functions will be eigenfunctions for the operator of the second difference derivative. There are N grid trigonometric functions that satisfy the boundary conditions of the first kind. Vectors composed of these functions have the form

$$\mathbf{X}_j = \sin(\pi(i+1)(j+1)/(N+1)), \quad i = \overline{0, N-1}, \quad j = \overline{0, N-1}. \quad (22)$$

We multiply the i th line of operator (7) by vector \mathbf{X}_j :

$$\begin{aligned} & -\sin(\pi(i+1)(j+1)/(N+1)) + 2\sin(\pi(i+1)(j+1)/(N+1)) - \sin(\pi(i+2)(j+1)/(N+1)) \\ & = 2(1 - \cos(\pi(j+1)/(N+1)))\sin(\pi(i+1)(j+1)/(N+1)). \end{aligned}$$

Therefore, the eigenvectors for operator (7) are vectors (22) and the eigenvalues are

$$\lambda_j = 2(1 - \cos(\pi(j+1)/(N+1))) = 4\sin^2(\pi(j+1)/(2(N+1))). \quad (23)$$

The minimum and maximum eigenvalues and the conditionality number of operator (7) are expressed as follows:

$$\lambda_{\min} = 4\sin^2\left(\frac{\pi/2}{N+1}\right), \quad \lambda_{\max} = 4\cos^2\left(\frac{\pi/2}{N+1}\right), \quad \nu = \cot^2\left(\frac{\pi/2}{N+1}\right). \quad (24)$$

3.2. The Second Boundary Value Problem for Eigenvalues

In the case of boundary conditions of the second kind, we introduce fictitious nodes -1 and N and require the fulfillment of the equalities on the boundaries

$$i = 0 : \frac{-y_{i+1} + 2y_i - y_{i-1}}{h^2} = \frac{-y_{i+1} + y_i}{h^2} \quad \text{and} \quad i = N - 1 : \frac{-y_{i+1} + 2y_i - y_{i-1}}{h^2} = \frac{y_i - y_{i-1}}{h^2}.$$

Therefore, the trigonometric functions must satisfy the properties

$$y_0 = y_{-1}, \quad y_N = y_{N-1}. \tag{25}$$

Vectors composed of trigonometric functions satisfying (25) have the form

$$X_j = \cos(\pi(i + 1/2)j/N), \quad i = \overline{0, N - 1}, \quad j = \overline{0, N - 1}. \tag{26}$$

We multiply the i th line of operator (8) when $\sigma = 0$ by vector X_j :

$$\begin{aligned} &\cos(\pi(i - 1/2)j/N) + 2\cos(\pi(i + 1/2)j/N) - \cos(\pi(i + 3/2)j/N) \\ &= 2(1 - \cos(\pi j/N))\cos(\pi(i + 1/2)j/N). \end{aligned}$$

Consequently, the eigenvectors for operator (8) for $\sigma = 0$ are vectors (26) and the eigenvalues are

$$\lambda_j = 2(1 - \cos(\pi j/N)) = 4\sin^2(\pi j/(2N)), \quad j = \overline{0, N - 1}. \tag{27}$$

3.3. The Third Boundary Value Problem for Eigenvalues

When forming the eigenvectors of operator (8) in the case of boundary conditions of the third kind ($\sigma > 0$), we will use the eigenvectors obtained in the case of boundary conditions of the first (22) and second (26) kind, and also the fact that the operator Λ_D in the case of boundary conditions of the third kind (8) at $\sigma = 1$ corresponds to operator (7), obtained in the case of boundary conditions of the first kind.

We consider the system of vectors

$$X_j = \cos\left(\frac{\pi(2i + a_j + 1)(j + a_j)}{2(N + a_j)} - \frac{\pi a_j}{2}\right), \quad i = \overline{0, N - 1}, \quad j = \overline{0, N - 1}. \tag{28}$$

System (28) at $a_j = 1$ coincides with (22), and at $a_j = 0$, with (26). We introduce the fictitious nodes -1 and N and we require the following equalities to hold on the boundaries:

$$\begin{aligned} i = 0 : & (-y_{i+1} + 2y_i - y_{i-1})/h^2 = (-y_{i+1} + (1 + \sigma)y_i)/h^2 \quad \text{and} \\ i = N - 1 : & (-y_{i+1} + 2y_i - y_{i-1})/h^2 = ((1 + \sigma)y_i - y_{i-1})/h^2. \end{aligned}$$

Therefore, the trigonometric functions (27) must satisfy the properties

$$(1 - \sigma)y_0 = y_{-1}, \quad y_N = (1 - \sigma)y_{N-1}. \tag{29}$$

We express σ through a_j from (28) and (29) on the left boundary and obtain the dependence a_j from j , which is implicitly expressed from the equation

$$f(a_j) = 1 - \cos\left(\pi \frac{j + a_j}{N + a_j}\right) - \tan\left(\frac{\pi(a_j + 1)(j + a_j)}{2(N + a_j)} - \frac{\pi a_j}{2}\right) \sin\left(\pi \frac{j + a_j}{N + a_j}\right) = \sigma. \tag{30}$$

Figure 2 shows dependence f on a_j for various j at $N = 8$. On the graph, the curves are arranged from left to right in descending order of j .

Parameters a_j through $\sigma = \text{const}$ can be expressed numerically. The eigenvalues of operator (8) corresponding to eigenvectors (28) are, respectively,

$$\lambda_j = 2\left(1 - \cos\left(\pi \frac{j + a_j}{N + a_j}\right)\right) = 4\sin^2\left(\frac{\pi j + a_j}{2N + a_j}\right), \quad a_j \in [0, 1], \quad j = \overline{0, N - 1}. \tag{31}$$

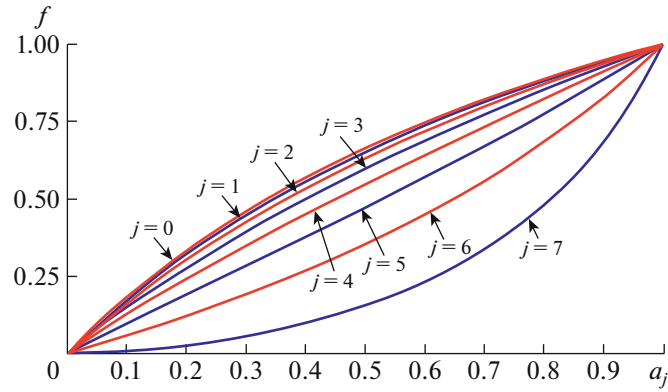


Fig. 2. Dependence f on a_j for different values of j at $N = 8$.

4. VARIATIONAL OPTIMIZATION OF ITERATIVE METHODS

To calculate the parameter τ_{n+1} , we use the steepest descent method [3]. In the nonstationary case, we will use the convergence criterion $(Az^{n+1}, z^{n+1}) \rightarrow \min$ together with $z^{n+1} = z^n - \tau_{n+1}B^{-1}Az^n$:

$$(Az^{n+1}, z^{n+1}) = (r^n, z^n) - 2\tau_{n+1}(r^n, w^n) + \tau_{n+1}^2(Aw^n, w^n), \quad (32)$$

where $r^n = Az^n$ is the residual vector and $w^n = B^{-1}r^n$ is the correction vector.

The minimum value τ_{n+1} is achieved at

$$\tau_{n+1} = (r^n, w^n) / (Aw^n, w^n). \quad (33)$$

Expression (32), taking into account (33), takes the form

$$\|z^{n+1}\|_A^2 = \rho \|z^n\|_A^2, \quad \rho^2 = 1 - (r^n, w^n)^2 / ((r^n, z^n)(Aw^n, w^n)). \quad (34)$$

To estimate the convergence rate of the steepest descent method, we use the Kantorovich inequality [20]: $4xy \leq (ax + y/a)^2$:

$$\rho^2 \leq 1 - 4 \left(a \frac{(r^n, z^n)}{(r^n, w^n)} + \frac{(Aw^n, w^n)}{(r^n, w^n)a} \right)^{-2}. \quad (35)$$

Taking into account $A^{1/2}z^n = y^n$ and $B^{-1/2}y = z^n$, expressions in denominator (35) will take the form

$$\frac{(r^n, z^n)}{(r^n, w^n)} = \frac{(y^n, y^n)}{(y^n, Cy^n)} = \frac{(Bz^n, z^n)}{(Az^n, z^n)} \leq (\lambda_{\min}(C))^{-1}, \quad \frac{(Aw^n, w^n)}{(Bw^n, w^n)} \leq \lambda_{\max}(C). \quad (36)$$

Inequality (35), taking (36) and $a = \lambda_{\min}$ into account, transform to the form

$$\rho \leq \sqrt{1 - \frac{4}{(1 + \lambda_{\max}/\lambda_{\min})^2}} = \frac{\lambda_{\max}/\lambda_{\min} - 1}{\lambda_{\max}/\lambda_{\min} + 1} = \frac{v - 1}{v + 1}, \quad v \leq \frac{\Delta + \lambda_{\min}(S)}{\delta + \lambda_{\min}(S)}. \quad (37)$$

We describe an algorithm for solving the grid equation (4) based on method (9), (14), and (33).

Calculation of the residual vector from the equation $r = Ax^n - F$.

Finding the correction vector w^n by the sweep method from the equation

$$A_{i,j,k}w_{i,j,k}^n - B_{5,i,j,k}w_{i,j,k+1}^n - B_{6,i,j,k}w_{i,j,k-1}^n = r_{i,j,k}^n.$$

Iterative parameter calculation: $\tau_n = (r^n, w^n) / (Aw^n, w^n)$.

Transition to the next iteration: $x^{n+1} = x^n - \tau_n w^n$.

5. SOLVING TEST PROBLEMS BASED ON THE DEVELOPED METHOD FOR SOLVING GRID EQUATIONS WITH A TRIDIAGONAL PRECONDITIONER

In this section, we consider the solution of three test difference problems by the method of solving grid equations with a tridiagonal preconditioner. The first one is the Dirichlet problem for the Poisson equation in a rectangle. The second one is the Poisson difference equation with boundary conditions of the second and third kind in different directions. The choice of boundary conditions is due to the fact that in order to calculate pressure in problems of hydrodynamics, it becomes necessary to solve the Poisson equation with a boundary condition of the third kind, which is specified on the surface of a reservoir. On solid walls, boundary conditions are specified in the Neumann form. The third task is to solve the grid equations that arise when calculating the pressure for the three-dimensional problem of hydrodynamics of the Sea of Azov. Based on the solution of the first two problems, the convergence of the proposed method for solving grid equations was studied theoretically and practically (empirically). The convergence of the solution of the third problem was studied only practically.

5.1. The Dirichlet Problem for the Poisson Equation in a Rectangle

We illustrate the proposed method on the example of solving the Dirichlet difference problem for the Poisson equation in a rectangle $\bar{G} = \{0 \leq x \leq l_x, 0 \leq z \leq l_z\}$:

$$\begin{aligned} u_{xx}^- + u_{zz}^- &= -\varphi(x, z), & (x, z) \in \omega, \\ u(x, z) &= g(x, z), & (x, z) \in \gamma \end{aligned} \tag{38}$$

on the grid $\bar{\omega} = \{x_{ij} = (ih_x, jh_z) \in \bar{G}, 0 \leq i < N_x, 0 \leq j < N_z, h_x(N_x - 1) = l_x, h_z(N_z - 1) = l_z\}$ with border γ .

We estimate the number of iterations n to solve problem (38) by the method of solving grid equations with a tridiagonal preconditioner (14). To get an estimate of $\delta D_x \leq \Lambda_x \leq \Delta D_x$, the eigenvalues of the operator $D_x^{-1/2} \Lambda_x D_x^{-1/2}$ need to be analyzed based on the solution of the boundary value problem represented by the grid equation $y_{\bar{x}\bar{x}} + \lambda Dy = 0$ in the case of boundary conditions of the first kind, $\delta \leq \lambda \leq \Delta$. Values δ, Δ will be written as follows:

$$\delta = 2 \sin^2(\pi/(2N_x + 2)), \quad \Delta = 2 \cos^2(\pi/(2N_x + 2)). \tag{39}$$

The value of the eigenvalue of the operator $S = D_x^{-1/2} \Lambda_z D_x^{-1/2}$ follows from (24):

$$\lambda_{\min}(S) = \frac{1}{2} (h_x/h_z)^2 \sin^2(\pi/(2N_z + 2)). \tag{40}$$

An estimate of the conditionality number of operator $C = B^{-1/2} AB^{-1/2}$ is written:

$$v(C) \leq (\Delta + \lambda_{\min}(S))/(\delta + \lambda_{\min}(S)). \tag{41}$$

For parameter ρ describing the rate of convergence of the iterative method, we have the estimate

$$\rho \leq (v - 1)/(v + 1). \tag{42}$$

An estimate of the number of iterations n to solve problem (38) takes the form

$$n \geq (\ln \|z_n\| - \ln \|z_0\|)/\ln(1/\rho). \tag{43}$$

Table 1 shows the results of a numerical experiment on the calculation of n , theoretical, and n_{pr} , practical, values of the number of iterations for solving problem (38) by the method of solving grid equations with a tridiagonal preconditioner (14) on a grid with dimensions $N_x = N_z = 100$ in the case of a right side function: $\varphi(x, z) = \sin(\pi x/l_x) \sin(\pi z/l_z)$. In this case, the minimum and maximum eigenvalues of the operator $D_x^{-1/2} \Lambda_x D_x^{-1/2}$ are $\delta = 5.035 \times 10^{-4}$ and $\Delta = 1.999$. As a result of calculating the grid equation, the error rate for solving the problem decreased from $\|z_0\| = 1$ to $\|z_n\| = 10^{-8}$. When performing a numerical experiment, the ratio of the steps of the computational grid in spatial coordinates Ox and Oz , respectively, $k = h_x/h_z$ and $h_z = 1$, was changed. Table 1 also gives theoretical estimates of the minimum eigenvalue of the operator $S = D_x^{-1/2} \Lambda_z D_x^{-1/2}$, number of conditionality $v(C)$ of operator C , parameter ρ describing the rate of convergence of the iterative method, and the number of iterations n .

Table 1. Theoretical and practical value of the number of iterations in solving the Dirichlet problem for the Poisson equation in a rectangle by the proposed method

k	1	2	5	10	100	1000
n_{pr}	15358	5862	1044	256	8	4
$\lambda_{\min}(\mathcal{S})$	1.259×10^{-4}	5.035×10^{-4}	3.147×10^{-3}	0.013	1.259	125.864
$\nu(C)$	3.177×10^3	1.986×10^3	548.659	153.713	2.588	1.016
ρ	0.999	0.999	0.996	0.987	0.443	7.878×10^{-3}
n	29266	18295	5054	1415	23	4

Table 2. The number of iterations in solving the Dirichlet problem for the Poisson equation on a sequence of refining grids

k	1	2	5	10	100	1000
100×100	15358	5862	1044	256	8	4
200×200	$>10^5$	22136	3920	946	16	4
500×500	$>10^5$	$>10^5$	22612	5422	59	5
1000×1000	$>10^5$	$>10^5$	$>10^5$	20340	203	7

Table 2 shows the number of iterations calculated practically when solving the Dirichlet problem for the Poisson equation (38) in a rectangle using the proposed method for solving grid equations. When solving this problem, a sequence of refining grids with different ratios of the spatial steps of the computational grid was used.

From the data given in Tables 1 and 2, it can be seen that when using the proposed method, an increase in k leads to a significant reduction in the number of iterations.

5.2. Solution of the Difference Poisson Equation with Boundary Conditions of the Second and Third Kind in Different Directions

We assume that on grid $\bar{\omega}$ it is required to find the solution of the Poisson equation (1) in the rectangle $\bar{G} = \{0 \leq x \leq l_x, 0 \leq z \leq l_z\}$,

$$u''_{xx} + u''_{zz} = -\varphi(x, z), \quad (x, z) \in G, \quad (44)$$

satisfying the Neumann boundary conditions on sides $x = 0$ and $x = l_x$ at $0 \leq z \leq l_z$;

$$u'_x = 0, \quad (45)$$

on sides $z = 0$ and $z = l_z$ at $0 \leq x \leq l_x$ by the boundary conditions of the third kind

$$u'_z|_{z=0} = \alpha_z u, \quad -u'_z|_{z=l_z} = \alpha_z u. \quad (46)$$

On the grid $\bar{\omega} = \{x_{ij} = (ih_x, jh_z) \in \bar{G}, 0 \leq i < N_x, 0 \leq j < N_z, h_x(N_x - 1) = l_x, h_z(N_z - 1) = l_z\}$, problem (44)–(46) corresponds to the difference scheme

$$\Lambda_x u + \Lambda_z u = \varphi(x, z),$$

$$\Lambda_x u = \begin{cases} -2u_x/h_x, & x = 0, \\ -u_{\bar{x}\bar{x}}, & h_x \leq x \leq l_x - h_x, \\ 2u_{\bar{x}}/h_x, & x = l_x, \end{cases} \quad \Lambda_z u = \begin{cases} 2(-u_z + \alpha_z)/h_z, & z = 0, \\ -u_{\bar{z}\bar{z}}, & h_z \leq z \leq l_z - h_z, \\ 2(u_z + \alpha_z)/h_z, & z = l_z. \end{cases} \quad (47)$$

We estimate the number of iterations n to solve problem (47) by the method of solving grid equations with a tridiagonal preconditioner (14). For the Neumann problem, the minimum and maximum eigen-

Table 3. Number of iterations calculated analytically

h_x/h_z	1	2	5	10	100	1000
$\sigma = 1$	152336	38091	6103	1533	24	4
$\sigma = 0.1$	210096	52531	8413	2110	30	5
$\sigma = 0.01$	859638	214917	34395	8606	95	6
$\sigma = 0.001$	7487831	1871965	299522	74888	758	15

Table 4. Number of iterations calculated experimentally

h_x/h_z	1	2	5	10	100	1000
$\sigma = 1$	32065	8790	1226	318	12	9
$\sigma = 0.1$	44077	11306	1288	438	18	11
$\sigma = 0.01$	$>10^5$	45470	6522	2133	41	25
$\sigma = 0.001$	$>10^5$	$>10^5$	60474	16625	272	–

values of the operator $D_x^{-1/2} \Lambda_x D_x^{-1/2}$ are $\delta = 0$ and $\Delta = 2$. The value of the eigenvalue of operator S follows from (31):

$$\lambda_{\min}(S) = \frac{1}{2} \left(\frac{h_x}{h_z} \right)^2 \sin^2 \left(\frac{\pi a_0}{2 N_z + a_0} \right). \tag{48}$$

Parameter a_0 is found implicitly from the equation

$$1 - \cos \left(\frac{\pi a_0}{N_z + a_0} \right) - \tan \left(\frac{\pi a_0(1 + a_0)}{2(N_z + a_0)} - \frac{\pi a_0}{2} \right) \sin \left(\frac{\pi a_0}{N_z + a_0} \right) = \sigma. \tag{49}$$

The parameters are calculated according to formulas (41)–(43): estimate of the conditionality number of operator C , the rate of convergence of the iterative method ρ , and a theoretical estimate of the number of iterations n . Tables 3 and 4, respectively, show the results of the numerical experiment on the calculation of n , the theoretical value, and n_{pr} , the practical value, of the number of iterations for solving problem (47) by the method with the tridiagonal preconditioner (14) on a grid with dimensions $N_x = N_z = 100$ in the case of the function of the right side $\varphi(x, z) = \sin(\pi x/l_x) \sin(\pi z/l_z)$. Parameters k and σ were changed. As a result of solving the grid equation, the error rate for solving the problem decreased from $\|z_0\| = 1$ to $\|z_n\| = 10^{-8}$.

From the data in Tables 3 and 4, it can be seen that the number of iterations increases with decreasing parameter σ .

5.3. Solution of Grid Equations Arising in the Calculation of Pressure for a Three-Dimensional Problem of Hydrodynamics of the Sea of Azov

The proposed method for solving grid equations is integrated into the previously developed software package for modeling three-dimensional hydrophysical processes in shallow water bodies and is called AZOV3D. The developed software module is used to solve the grid equations obtained by approximating a three-dimensional pressure calculation problem based on the wave equation. The AZOV3D software package is based on a mathematical model of the hydrodynamics of shallow water bodies, which includes three equations of motion (Navier–Stokes) and a continuity equation with regularizers [21, 22] in the case of variable density.

The area of the calculations coincides with the real dimensions of the studied water area of the Sea of Azov, which are 355 km long and 233 km wide. The maximum depth of the sea is 13.2 m. In the developed software package, 4 m above the surface is allocated for the rise in the water level. The spatial step is chosen to be 1000 m horizontally and 50 cm vertically. Thus, the ratio of horizontal steps to vertical steps is 2000. The dimensions of the calculation grid are $255 \times 350 \times 35$ nodes. To calculate the pressure on the surface

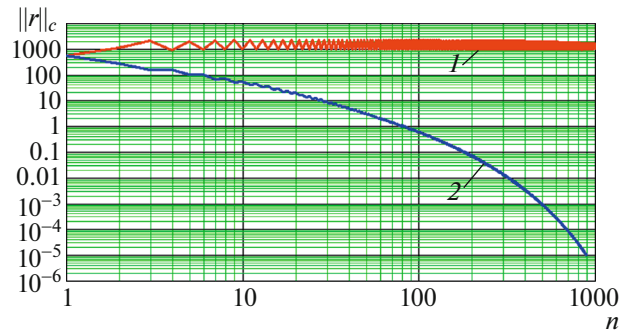


Fig. 3. Dependence of the values of the uniform norm of the residual vector on the number of iterations: (1) calculation based on MATM; (2) based on the method of solving grid equations with a tridiagonal preconditioner.

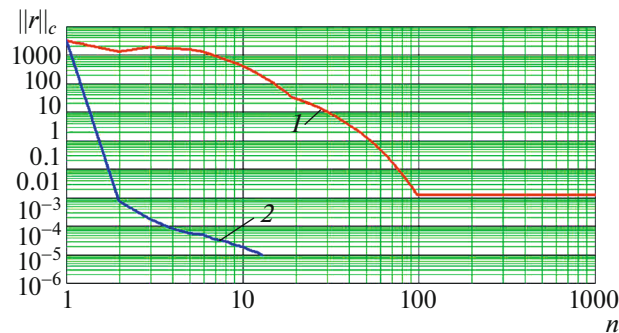


Fig. 4. Dependence of the uniform norm of the residual vector on the number of iterations: (1) calculation based on MATM; (2) based on the method of solving grid equations with a tridiagonal preconditioner.

of the Sea of Azov, boundary conditions of the third kind are set; and on all other parts of the boundary, of the second kind. Figure 3 shows plots of the norm of the residual vector obtained by solving grid equations for the pressure calculation problem based on the modified alternating triangular method (MATM) [3] and the method for solving grid equations with a tridiagonal preconditioner. On the Ox axis, n is the iteration number; and on the Oy axis, the norm of the residual vector ($\|r\|_c$).

It can be seen from Fig. 3 that when solving grid equations of form (8) with the help of the MATM, the norm of the vector barely decreases. When using the proposed method for solving grid equations with a tridiagonal preconditioner, the iterative process converged in less than 1000 iterations. When solving the problem of the pressure calculation, the pressure calculated in the hydrostatic approximation can be used as the initial approximation. The pressure based on the hydrostatic approximation is calculated based on a two-dimensional problem, which is much less complex than calculating the pressure based on a three-dimensional problem. Figure 4 shows the plots of the norm of the residual vector, obtained by solving the grid equations of the pressure calculation problem based on the MATM and the method for solving grid equations with a tridiagonal preconditioner, taking into account the hydrostatic approximation.

Experimentally, it was found that when using the hydrostatic approximation as the initial pressure values, the methods for solving grid equations converge faster for a model of hydrodynamics of a shallow reservoir, which includes three equations of motion. The developed method for solving grid equations with a tridiagonal preconditioner converges in 10–15 iterations. For the convergence of the MATM in solving the test problem under consideration, about 100 iterations are required.

CONCLUSIONS

For the numerical implementation of a spatial three-dimensional mathematical model of the hydrodynamics of shallow water bodies, including the equations of motion (Navier–Stokes) and the continuity equation with regularizers [5] in the case of variable density, a method for solving grid equations is proposed. The developed method is included in the previously created AZOV3D software package designed

for modeling three-dimensional hydrophysical processes in shallow water bodies. A software module has been developed that can be used to solve grid equations obtained as a result of approximating a three-dimensional pressure calculation problem based on the wave equation.

When solving grid equations using the MATM, the vector norm barely decreases, and when using the proposed method for solving grid equations with a tridiagonal preconditioner, the iterative process converges in less than 1000 iterations. When solving the problem of calculating the pressure, the pressure calculated in the hydrostatic approximation based on a two-dimensional problem was used as the initial approximation. The hydrostatic pressure is a good initial approximation for the pressure obtained from the application of splitting schemes by physical processes for a hydrodynamic model that includes three equations of motion. This has a significant effect on the convergence of the applied method.

The numerical experiment carried out with the developed software module made it possible to estimate the residual vector norm obtained by solving the grid equations of the pressure calculation problem based on the MATM and the method for solving grid equations with a tridiagonal preconditioner, taking into account the hydrostatic approximation. It has been experimentally established that when using the hydrostatic approximation as the initial pressure values, the methods for solving grid equations converge faster. For the convergence of the MATM when solving the test problem under consideration, it took about 100 iterations. The proposed method for solving grid equations with a tridiagonal preconditioner converges in 10–15 iterations. According to the specifics of the developed method, it is effective in solving problems of aquatic ecology in the case of a computational domain with an extended geometry.

FUNDING

This study was supported by the Russian Science Foundation (grant no. 21-71-20050).

CONFLICT OF INTEREST

The authors declare that they have no conflicts of interest.

REFERENCES

1. A. A. Samarskii and P. N. Vabishchevich, “Finite-difference approximations to the transport equation. II,” *Differ. Equations* **36**, 1069–1077 (2000).
<https://doi.org/10.1007/BF02754509>
2. A. A. Samarskii and P. N. Vabishchevich, *Numerical Methods for Solving Convection-Diffusion Problems* (URSS, Moscow, 2009) [in Russian].
3. A. A. Samarskii and E. S. Nikolaev, *Methods for Solving Grid Equations* (Nauka, Moscow, 1978) [in Russian].
4. A. N. Konovalov, “To the theory of the alternating triangle iteration method,” *Sib. Math. J.* **43**, 439–457 (2022).
<https://doi.org/10.1023/A:1015455317080>
5. B. N. Chetverushkin, “Kinetic models for solving continuum mechanic problems on supercomputers,” *Math. Models Comput. Simul.* **7**, 531–539 (2015).
<https://doi.org/10.1134/S2070048215060034>
6. Y. L. Gurieva and V P. Il'in, “Conjugate direction methods for multiple solution of SLAEs,” in *Computational Methods and Algorithms, Part XXXIII*, Zap. Nauchn. Semin. POMI, Vol. 496, 26–42 (2020).
7. M. V. Yakobovskiy and S. K. Grigorjev, “Projection-based guaranteed mesh generation algorithm,” KIAM Preprint No. 109 (Keldysh Inst. Appl. Math. RAS, Moscow, 2018) [in Russian].
<https://doi.org/10.20948/prepr-2018-109>
8. N. L. Zamarashkin, I. V. Oseledets, and E. E. Tyrtshnikov, “New applications of matrix methods,” *Comput. Math. Math. Phys.* **61**, 669–673 (2021).
<https://doi.org/10.1134/S0965542521050183>
9. Yu. V. Vassilevski, S. S. Simakov et al., “Personalization of mathematical models in cardiology: obstacles and perspectives,” *Comput. Res. Model.* **14**, 911–930 (2022).
<https://doi.org/10.20537/2076-7633-2022-14-4-911-930>
10. O. Yu. Milyukova, “MPI+OpenMP implementation of conjugate gradients method with preconditioner of the block incomplete inverse triangular decomposition of the first order,” *Vychisl. Metody Program.* **23**, 191–206 (2022).
<https://doi.org/10.26089/NumMet.v23r312>
11. J. H. Bramble and J. E. Pasciak, “Analysis of a Cartesian PML approximation to the three dimensional electromagnetic wave scattering problem,” *Int. J. Numer. Anal. Model.* **9**, 543–561 (2012).

12. A. Kleefeld and T.-C. Lin, “Boundary element collocation method for solving the exterior Neumann problem for Helmholtz’s equation in three dimensions,” *Electron. Trans. Numer. Anal.* **39**, 113–143 (2012).
13. A. I. Sukhinov, A. E. Chistyakov, V. V. Sidoryakina, and E. A. Protsenko, “Economic explicit-implicit schemes for solving multidimensional diffusion-convection problems,” *Vychisl. Mekh. Sploshnykh Sred* **12**, 435–445 (2019).
<https://doi.org/10.7242/1999-6691/2019.12.4.37>
14. A. I. Sukhinov, A. E. Chistyakov, and E. V. Alekseenko, “Numerical realization of the three-dimensional model of hydrodynamics for shallow water basins on a high-performance system,” *Math. Models Comput. Simul.* **3**, 562–574 (2011).
<https://doi.org/10.1134/S2070048211050115>
15. E. Alekseenko, B. Roux, D. Fougere, and P. G. Chen, “The effect of wind induced bottom shear stress and salinity on *Zostera noltii* replanting in a Mediterranean coastal lagoon,” *Estuarine, Coastal Shelf Sci.* **187**, 293–305 (2017).
<https://doi.org/10.1016/j.ecss.2017.01.010>
16. O. M. Belotserkovskii, V. A. Gushchin, and V. V. Shchennikov, “Use of the splitting method to solve problems of the dynamics of a viscous incompressible fluid,” *USSR Comput. Math. Math. Phys.* **15**, 190–200 (1975).
[https://doi.org/10.1016/0041-5553\(75\)90146-9](https://doi.org/10.1016/0041-5553(75)90146-9)
17. A. I. Sukhinov, A. E. Chistyakov, I. Y. Kuznetsova, A. M. Atayan, and A. V. Nikitina, “Regularized difference scheme for solving hydrodynamic problems,” *Math. Models Comput. Simul.* **14**, 745–754 (2022).
<https://doi.org/10.1134/S2070048222050155>
18. A. I. Sukhinov, A. V. Nikitina, A. M. Atayan, V. N. Litvinov, Yu. V. Belova, and A. E. Chistyakov, “Supercomputer simulation of hydrobiological processes of coastal systems,” *Math. Models Comput. Simul.* **14**, 677–690 (2022).
<https://doi.org/10.1134/S2070048222040123>
19. A. I. Sukhinov, A. E. Chistyakov, E. A. Protsenko, V. V. Sidoryakina, and S. V. Protsenko, “Accounting method of filling cells for the solution of hydrodynamics problems with a complex geometry of the computational domain,” *Math. Models Comput. Simul.* **12**, 232–245 (2020).
<https://doi.org/10.1134/S2070048220020155>
20. L. V. Kantorovich, “Functional analysis and applied mathematics,” *Usp. Mat. Nauk* **3** (6), 89–185 (1948).
21. B. N. Chetverushkin, M. V. Yakobovskiy, M. A. Kornilina, and A. V. Semenova, “Numerical algorithms for HPC systems and fault tolerance,” in *Parallel Computational Technologies, PCT 2019*, Ed. by L. Sokolinsky and M. Zymbler, Communications in Computer and Information Science, vol. 1063 (Springer, Cham, 2019), pp. 34–44.
https://doi.org/10.1007/978-3-030-28163-2_3
22. N. D’Ascenzo, V. I. Saveliev, and B. N. Chetverushkin, “On an algorithm for solving parabolic and elliptic equations,” *Comput. Math. Math. Phys.* **55**, 1290–1297 (2015).
<https://doi.org/10.1134/S0965542515080035>