A software package for solving grid equations in areas with geometry "stretched" along one of the spatial directions

Vladimir Litvinov[1,2*], *Nelli* Rudenko[1,2], *Natalya* Gracheva[1,2] and *Alexander* Chistyakov[1]

[1]Don State Technical University, Rostov-on-Don, 344000, Russia
[2]Azov-Black Sea Engineering Institute of Don State Agrarian University, Zernograd, 347740, Russia

Abstract. A method for solving grid equations in areas with geometry "stretched" along one of the spatial directions is proposed. Method proposed for model includes pollutants whose concentration in the Azov Sea, according to the analysis of scientific publications and environmental databases, exceeds the maximum permissible concentration (MPC). The mechanism of the preconditioner formation is described. The estimation of the minimum and maximum eigenvalues is performed. Using object-oriented analysis and design, a class library has been implemented to solve the problem of impurity propagation in a shallow reservoir. The architecture of an information system to support scientific research capable of functioning in conditions of using heterogeneous computing environments is described. The system is based on the requirement of the maximum possible use of free software in order to minimize capital and operational costs. The system uses classes that allow you to read and write information about the geometry of the object under study and the parameters of the calculated grid. Numerical experiments were carried out using the developed software package, including the proposed mathematical model, focused on the multiprocessor computing system (MCS). The dependences of the SLAE solution time on the number of diagonals and the order of the matrix are obtained. The number of iterations for solving grid equations has been experimentally determined depending on the ratio of the number of nodes in spatial coordinates.

1 Introduction

Among the works devoted to mathematical modeling of problems of hydrodynamics and hydrobiology, in particular the spread of pollutants in coastal systems, are the works of Russian scientists Matishov G.G., Muraveyko V.M., Berdnikov S.V., Ilyin G.V., Zuev A.N., Ilyichev V.G., Kravchenko V.V., etc. Leading foreign research centers and organizations, such as: Sweden's Meteorelogical and Hydrological Institute (SMHI, Sweden); Center for Water Research (CWR, Australia); National Oceanic and Atmospheric Administration (NOAA, USA); Centre for Ecology and Hydrology (CEH, Wallingford, United Kingdom); Korea Research Institute of Ships and Ocean Engineering (KRISO, Korea) are engaged in the

* Corresponding author: litvinovvn@rambler.ru

development of information systems and software systems designed for predictive modeling and monitoring of aquatic ecosystems, which allow to obtain a detailed description of the objects under study [1-2]. At the same time, the urgent task is to develop effective information systems focused on use in high-performance heterogeneous computing environments that allow real-time forecasts of the development of various emergencies [3-7].

The paper considers algorithms and software modules that implement the solution of grid equations in areas with geometry "elongated" along one of the spatial directions" arising from the numerical solution of problems of hydrodynamics and hydrobiology [8].

2 Material and methods

2.1 Problem statement

Let's consider the complete 3D system of equations of the pollutant distribution, including the oil and its refined products, in the computational domain – Azov Sea – with the side surface σ, the water undisturbed surface Σ_0, the bottom Σ_H, the variable depth H in the Cartesian coordinate system for the axes Oz, Oy, Ox, directed vertically downwards, to the north and east, respectively. The model is based on the researches [9, 10] and has the following form:

$$\frac{\partial S_i}{\partial t}+u\frac{\partial S_i}{\partial x}+v\frac{\partial S_i}{\partial y}+(w-w_{gi})\frac{\partial S_i}{\partial z}+\sigma S_i=\mu_i\left(\frac{\partial^2 S_i}{\partial x^2}+\frac{\partial^2 S_i}{\partial y^2}\right)+\frac{\partial}{\partial z}\left(\upsilon_i(\mathrm{z})\frac{\partial S_i}{\partial z}\right)+f(\mathrm{x,y,z,t}), \quad (1)$$

where u, v, w are components of the water flow velocity vector; w_{gi} is the gravitational deposition rate of *i*-th component, if it is in a suspended state; σ_i is the decomposition coefficient of *i*-th impurity; f is a chemical and biological source (the runoff); μ_i, υ_i are diffusion coefficients in the horizontal and vertical directions; S_i is the concentration of *i*-th impurity, $i=\overline{1,6}$: 1 is the mercury (Hg) ; 2 is the plumbum (Pb) ; 3 is the manganese (Mn) ; 4 is the iron (Fe^{+2}) ; 5 is the phytoplankton (*Sceletonema costatum* diatoms); 6 are oil and petroleum products.

The model includes pollutants whose concentration in the Azov Sea, according to the analysis of scientific publications and environmental databases, exceeds the maximum permissible concentration (MPC) [11-14]. This model takes into account the water flow movement; microturbulent diffusion; interaction and gravitational subsidence of pollutants and plankton; biogenic, temperature and oxygen regimes; the influence of salinity.

Let's define the initial conditions:

$$S_i(\mathrm{x,y,z},0)=\mathrm{S}_{0i}(\mathrm{x,y,z}),\ i=\overline{1,6}. \quad (2)$$

The computational domain G it is a closed area bounded by the undisturbed water surface Σ_0, the bottom $\Sigma_H=\Sigma_H(x,y)$, and the cylindrical surface σ for $0<t\le T_0$. $\sum=\sum_0\cup\sum_H\cup\sigma$, Σ is a piecewise smooth boundary of the domain G.

Let $\mathbf{U_n}$ is the component of the water flow velocity vector, normal to the boundary Σ, $\mathbf{n}$ is the external normal vector to the Σ. Then the boundary conditions for the model (1) will have the form:

$$S_i = 0 \ on \sigma, \ if \ U_n < 0; \ \frac{\partial S_i}{\partial \mathbf{n}} = 0 \ on \ \sigma, \ if \ U_n \geq 0; \tag{3}$$

$$\frac{\partial S_i}{\partial z} = \varphi\left(S_i\right) on \ \Sigma_0; \ \frac{\partial S_i}{\partial z} = -\varepsilon_i S_i \ on \ \Sigma_H, \tag{4}$$

where ε_i is the of *i*-th component by the bottom material; φ is the given function.

We took into account the water exchange with the Black Sea, the flow of the Don River, the complex shape of the coastline, and the bottom relief at setting the boundary conditions.

2.2 Method for solving grid equations

As a result of the discretization of equations (1) – (4), we obtain a system of grid equations in matrix form:

$$Ax = f, \tag{5}$$

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

Imagine A in the form:

$$A = \Lambda_x + \Lambda_y + \Lambda_z, \tag{6}$$

where $\Lambda_x = \frac{1}{h_x^2}\Lambda_{D,x} \otimes E_y \otimes E_z$; $\Lambda_y = \frac{1}{h_y^2} E_x \otimes \Lambda_{D,y} \otimes E_z$; $\Lambda_z = \frac{1}{h_z^2} E_x \otimes E_y \otimes \Lambda_{D,z}$; $\otimes$ is the symbol denoting the Kronecker product; E_x, E_y, E_z are unit matrices with dimensions $N_x \times N_x$, $N_y \times N_y$ и $N_z \times N_z$ accordingly; Λ_D is the square tridiagonal matrix containing coefficients of grid equations for terms describing finite differences for second-order partial derivatives along all axes.

The operator Λ_D in the case of boundary conditions of the 2nd and 3rd kind ($\delta \geq 0$) will be written as:

$$\left(\Lambda_D\right)_{i,j} = \begin{cases} 1+\delta, если \ \left(i+j=2\right) \cup \left(i+j=2N\right); \\ 2, если \ \left(i=j\right) \cap \left(2<i+j<2N\right); \\ -1, если \ \left|i-j\right|=1; \\ 0, если \ \left|i-j\right|>1; \end{cases} \quad \Lambda_D = \begin{pmatrix} 1+\delta & -1 & 0 & ... \\ -1 & 2 & -1 & ... \\ 0 & -1 & 2 & ... \\ ... & ... & ... & ... \end{pmatrix}. \tag{7}$$

The method of solving problem (5) will be based on an implicit iterative process:

$$B\frac{x^{n+1}-x^n}{\tau_{n+1}} + Ax^n = f, \tag{8}$$

where B is the preconditioner, whose handling is significantly simpler than the original operator A in (5); n is the iteration number, $\tau_{n+1} > 0$ is the iterative parameter.

Equation (8) in the case of a stationary iterative process is represented as:

$$x^{n+1} = x^n - \tau B^{-1}\left(Ax^n - f\right). \tag{9}$$

At each iteration, the resulting solution x^n will differ from the exact solution x by the amount of error z^n: $x^n = x + z^n$. Then equation (9) with respect to the error will be written as follows:

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

Let's introduce variable substitution $z^n = B^{-1/2} y^n$ multiply both parts of the last expression by $B^{1/2}$, as a result we get:

$$y^{n+1} = y^n - \tau B^{-1/2} A B^{-1/2} \cdot y^n.$$

Let's introduce the replacement of variables: $C = B^{-1/2} A B^{-1/2}$, $C = C^* > 0$, as a result, we get:

$$y^{n+1} = (E - \tau C) y^n. \tag{10}$$

The iterative process converges when the conditions are met:

$$-\rho E \le E - \tau C \le \rho E, \quad \rho < 1, \tag{11}$$

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

In equality (11) is represented as:

$$\frac{1-\rho}{\tau} E \le C \le \frac{1+\rho}{\tau} E.$$

Parameters ρ and τ are from conditions $\frac{1-\rho}{\tau} = \lambda_{\min}, \frac{1+\rho}{\tau} = \lambda_{\max}$:

$$\tau = \frac{2}{\lambda_{\min} + \lambda_{\max}}, \ \rho = \frac{\nu - 1}{\nu + 1}, \ \nu = \frac{\lambda_{\max}}{\lambda_{\min}}. \tag{12}$$

Here $\lambda_{\min}$, $\lambda_{\max}$ are minimum and maximum values of the eigenvalues of the operator C, ν is the number of conditionality of the operator C.

The preconditioner will be formed as follows:

$$B = \Lambda_z. \tag{13}$$

Let's estimate the maximum eigenvalue taking into account the equalities $C = B^{-1/2} A B^{-1/2}$, $B^{-1/2} x = y$:

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

Taking into account expressions (12), (13), the estimate of the maximum eigenvalue will have the form:

$$\lambda_{\max}(C) = \max_{|y|>0} \frac{\left(\left(\Lambda_x + \Lambda_y + \Lambda_z\right)y, y\right)}{\left(\Lambda_z y, y\right)} = \\ = 1 + \max_{|y|>0} \left(\frac{\left(\left(\Lambda_x + \Lambda_y\right)y, y\right)}{\left(\Lambda_z y, y\right)} \right) \le 1 + \frac{\lambda_{\max}\left(\Lambda_x + \Lambda_y\right)}{\lambda_{\min}\left(\Lambda_z\right)}. \tag{14}$$

Let 's estimate the minimum eigenvalue:

$$\lambda_{\min}(C) = \min_{|y|>0} \frac{(Ay, y)}{(By, y)} = \min_{|y|>0} \frac{\left(\left(\Lambda_x + \Lambda_y + \Lambda_z\right)y, y\right)}{\left(\Lambda_z y, y\right)} = \\ = 1 + \min_{|y|>0} \left(\frac{\left(\left(\Lambda_x + \Lambda_y\right)y, y\right)}{\left(\Lambda_z y, y\right)} \right) \ge 1. \tag{15}$$

The operation of stationary iterative methods requires a priori information about the values of the maximum and minimum eigenvalues of operators. In the case of choosing a preconditioner in the form of (15) for an arbitrary geometry of the computational domain, the complexity is caused by estimating the minimum proper number Λ_z.

2.3 Architecture of the software module

Computer modeling of the assigned tasks, while meeting the requirements for accuracy and calculation speed, requires large computing resources that are not always available in production conditions. To increase the efficiency of using available technical means, the architecture of an information system for supporting scientific research is proposed, capable of functioning in conditions of using heterogeneous computing environments.

The system is based on the requirement of the maximum possible use of free software in order to minimize capital and operating costs. As the main operating system, it is proposed to use Ubuntu Server 22.04 with the nginx http server installed, the MariaDB database management system and the platform.NET Core 6. The application software of the system is developed in the form of a server part implemented as an ASP .NET Core MVC web application, and the client part, which is a number of services installed on all involved calculators. The interaction between these components of the system is provided through the use of the SignalR library.

The server part provides the solution of the following tasks:

- management of user accounts of the system, including the assignment of roles;
- configuration management of available computing resources;
- managing the process of uploading simulation tasks to the server in the form of rpm packages;
- managing the task monitoring service.

The system and application software development process is managed using the Git version control system. Developers are working with the system's code base on their workstations, where local repositories of system component projects are deployed, synchronized with a remote repository accessible via a web service based on the Git version control system.

The system deployment, configuration and operation processes are managed by the administrator using any available SSH client, for example, ssh for Linux console terminal or Bitvise SSH Client for Windows.

The developed system is built using the principles of object-oriented design (Figure 1). The system is based on classes that allow reading and writing information about the geometry of the object under study and the parameters of the computational grid. These classes should implement the IGeometrySerializer and IgridSerializer interfaces, respectively. The initialized objects of the specified types are used to create Geometry Geometry objects and the Grid calculation grid, on the basis of which an object of the GridGeometryPreCalculated class is created, which is a geometry object with a superimposed calculation grid and pre-calculated parameters, such as, for example, node characteristics: internal, boundary or fictitious.

In order to provide the possibility of intermediate data storage for continuing calculations in case of emergency situations or transferring the task for calculation to other computing nodes, the gridgeometryprecalculated object is connected by an aggregation relationship with the IgridGeometryPreCalculat-edSerializer interface. Classes implementing the IgridGeometryPreCalculatedSerializer interface solve the tasks of serializing data in various formats (xml, json, storing data in relational databases, etc.).

The configured GridGeometryPreCalculated object is used when creating the SolverTask calculation task object, during which the user sets a number of modeling characteristics, the most important of which are the model parameters and the derivative calculation template.

Directly modeling is performed by the Solver class, which uses objects of the SolverTask class as tasks. The features of the Solver class are the possibility of dynamic control of computing devices by means of an aggregation relationship with an object of the IDeviceManager interface and the provision of work with an object of a system of linear algebraic equations by means of a composition relationship with an object of the SLAE class.

A distinctive feature of the SAE class is working with the Matrix object Matrix. The implementation of this interface in the work is the SparseMatrixRepSeq class, which provides efficient data storage with the elimination of duplication.

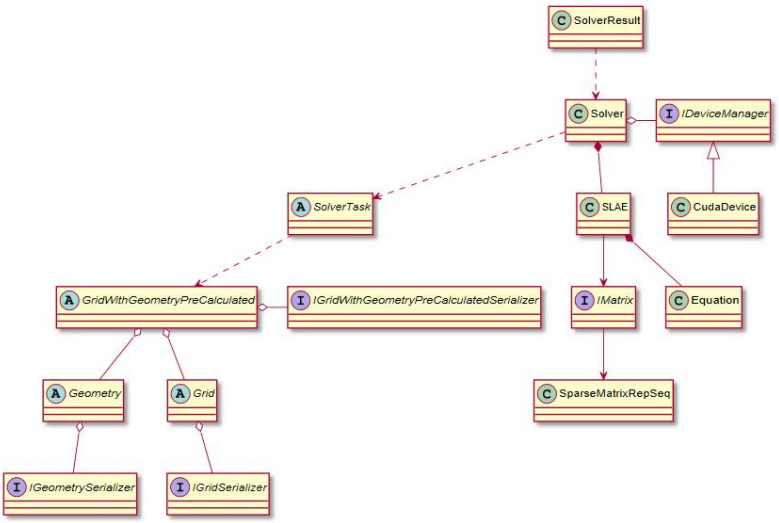

Fig. 1. Generalized class diagram of the developed software package

3 Results

The software implementation of the computational model in C# is based on the developed class library, which allows working with coordinates, geometry of the object under study (Figures 2, 3), consisting of a set of geometric primitives and computational grids

The system allows you to set coordinates for one-dimensional (Coordinate1D class), two-dimensional (Coordinate2D class) and three-dimensional (Coordinate3D class) coordinate systems. Coordinate classes are inherited from the abstract Coordinate class, which is directly used in computational algorithms.

The Geometry 1D, Geometry2D, and Geometry3D classes for describing the geometric shape of the objects under study are inherited from the Geometry abstract class and define one-dimensional, two-dimensional, and three-dimensional geometries, respectively.

The Geometry abstract class implements the operations of writing and reading data in XML format using the methods ExportToXml and ImportFromXml.

The most important element of the Geometry Abstract Class is the Geometry Elements property of the List<GeometryElement> type, which stores a set of geometry elements.

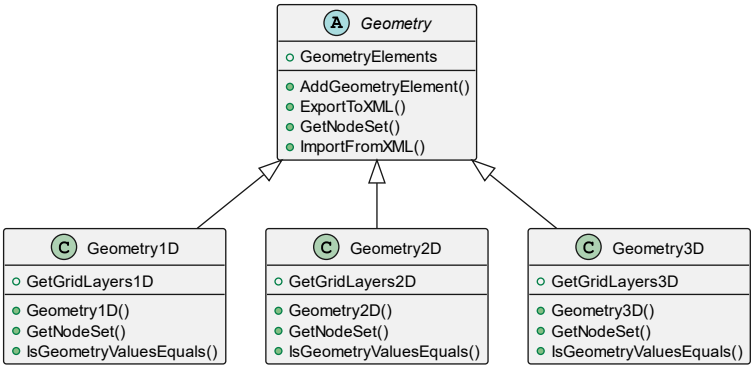

Fig. 2. Hierarchy of geometry classes

Geometry elements are also implemented for one-dimensional, two-dimensional and three-dimensional cases in the form of successor classes of the abstract GeometryElement class, respectively GeometryElement1D, GeometryElement2D and GeometryElement3D.

The GeometryElement abstract class allows you to save the coordinates of the location of a geometry element inside a geometry object by initializing the CoordinateLocation property, work with the characteristics of materials through the MaterialCharacteristic property, and also saves a set of geometric primitives that make up this geometric element through the GeometryPrimitives property of the List<GeometryPrimitive> type.

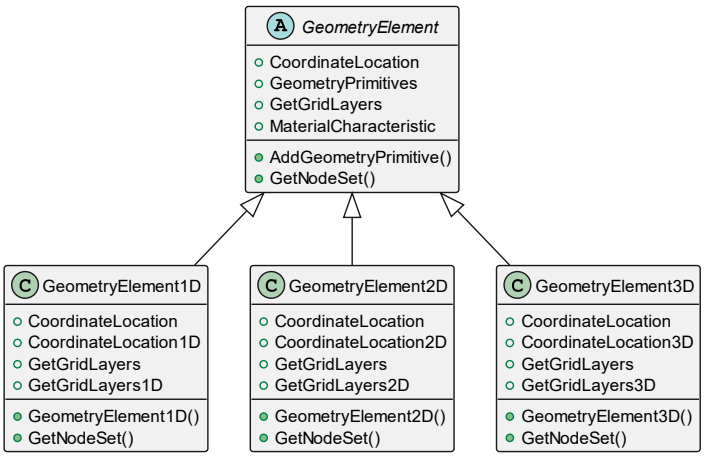

Fig. 3. Hierarchy of classes of geometry elements

The class library allows you to work with one-dimensional, two-dimensional and three-dimensional primitives by creating and initializing class objects co-responsibly GeometryPrimitive1D, GeometryPrimitive2D and GeometryPrimitive3D. These classes are inheritors of the Geometry Primitive abstract class. A one-dimensional primitive is a segment defined by the GeometryPrimitive1DLineSegment class, which is the successor of the Geome-tryPrimitive1D class. Two-dimensional primitives include the Genome-tryprimitive2dsquare rectangle, the GeometryPrimitive2DEllipse ellipse, and a free two-dimensional shape defined by an array of coordinates of the boundary points of the GeometryPrimitive2DManualByCoordinates shape. Three-dimensional primitives include objects of the GeometryPrimitive3DParallelepiped and GeometryPrimitive3DCube classes, modeling a parallelepiped and a cube, respectively.

The geometric primitive abstract class contains the coordinates in element property of the coordinate type, which stores the coordinates of the location of the primitive relative to the coordinate of the geometric element. The IsCavity property of the bool type determines whether the primitive is a "cut-out" of the space.

Work with computational grids is provided by means of classes-heirs of the abstract Grid class:

Grid1D – class for working with a one-dimensional computational grid;

Grid2D – a class for working with a two-dimensional computational grid;

Grid3D – a class for working with a three-dimensional computational grid.

A distinctive feature of these classes is the ability to set an uneven calculation grid by filling in the StepTransitions lists for Grid1D, StepTransitions1 and StepTransitions2 for Grid2D, StepTransitionsX, StepTransitionsY and StepTransitionsZ for Grid3D. The Grid3D class has a mechanism for inserting "layers" of the grid using the methods InsertLayerX, InsertLayerY and In-sertLayerZ.

4 Discussion

A numerical experiment was carried out using the developed software package, including the proposed mathematical model, focused on the MCS. The algorithm of preliminary estimation of the time spent on the task execution based on the data of preliminary testing of the performance of the hardware platform is used. At the time of launching the control module of the computer system, the presence of the file is checked performance.xml. In the absence of a file, performance tests are performed, representing a number of tasks for solving SLOWS with different parameters, and their results are recorded in a file. For the Windows 10 (x64) OS test system, CUDA Toolkit v10.0.130, Intel Core i5-6600 3.3 GHz processor, DDR4 32 GB RAM, NVIDIA GeForce GTX 750 Ti 2GB video adapter using NVIDIA CUDA technology, the results shown in Figure 4 are obtained.

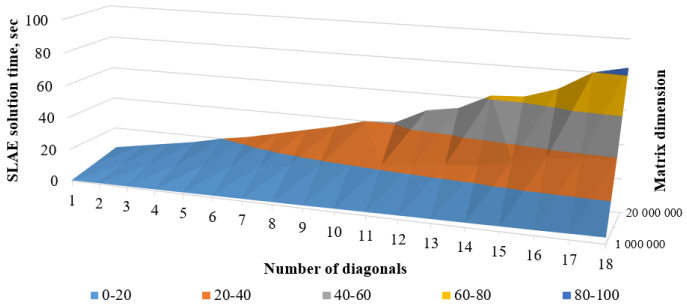

Fig. 4. Dependence of the SLAE solution time on the number of diagonals and the order of the matrix

The obtained data on the performance parameters are then used by the control modules of the computer system in the process of controlling the process of queuing the calculation problem based on the algorithm shown in Figure 5.

A numerical experiment was carried out to measure the number of iterations depending on the ratio of the number of nodes by spatial coordinates. Table 1 shows that the number of iterations required to solve the SLAE decreases from 256 to 36 with an increase in the *Nx/Nz* ratio.

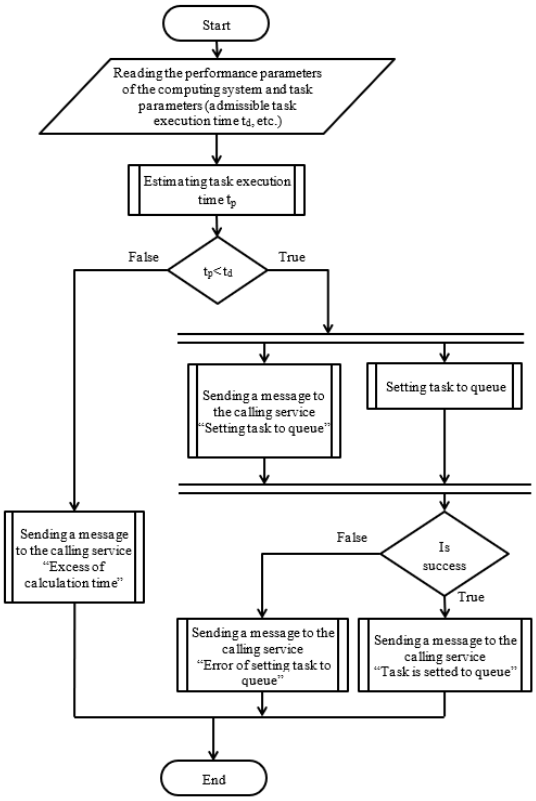

Fig. 5. Algorithm for controlling the process of queuing a calculation problem

Table 1. The results of a numerical experiment to measure the number of iterations depending on the ratio of the number of nodes in spatial coordinates.

Grid dimensions, N_xxN_z	N_x/N_z	Number of iterations
102x102	1.00	256
153x68	2.25	121
204x51	4.00	72
255x41	6.22	48
306x34	9.00	36

5 Conclusions

A software implementation of the proposed method for solving grid equations for solving problems in areas with geometry "stretched" along one of the spatial directions has been

developed. The architecture of the class library information system for solving the problem of impurity propagation in a shallow reservoir is described. The system is based on the requirement of the maximum possible use of free software in order to minimize capital and operating costs. Numerical experiments were carried out using the developed software package, including the proposed mathematical model, focused on the MVS. The dependences of the SLAE solution time on the number of diagonals and the order of the matrix are obtained. The number of iterations for solving grid equations has been experimentally determined depending on the ratio of the number of nodes in spatial coordinates. The number of iterations required to solve the SLA decreases from 256 to 36 with an increase in the *Nx/Nz* ratio, which indicates the high efficiency of the proposed method.

Acknowledgments

The study was carried out with the financial support of the Russian Science Foundation within the framework of the scientific project No. 21-71-20050.

References

1. A.A. Samarskiy, Classes of stable schemes. Zh. calculation. matem. and math. phys., **7(5)**, 1096-1133 (1967), U.S.S.R. Comput. Math. Math. Phys., **7(5)**, 171–223 (1967)
2. M.E. Ladonkina, O.A. Neklyudova, V.F. Tishkin, Matem. modeling, **26(1)**, 17-32 (2014)
3. E. Alekseenko, B. Roux, A. Sukhinov, R. Kotarba, D. Fougere, Nonlinear Processes in Geophysics, **20(2)**, 189-198 (2013). DOI: 10.1016/j.compfluid.2013.02.003.
4. E. Alekseenko, B. Roux, A. Sukhinov, R. Kotarba, D. Fougere, Computational Mathematics and Mathematical Physics, **57(6)**, 978-994 (2017). DOI: 10.5194/npg-20-189-2013.
5. X. Liu, S. Qi, Y. Huang, Y. Chen, P. Du, International Journal of Sediment Research, **30(3)**, 250–255 (2015)
6. N. Ascenzo, V.I. Saveliev, B.N. Chetverushkin, Zhurnal Vychislitel'noi Matematiki i Matematicheskoi Fiziki, **55(8)**, 1320–1328 (2015). DOI:10.1134/S0965542515080035.
7. A. Sukhinov, A. Chistyakov, V. Sidoryakina, MATEC Web of Conferences Volume 132, XIII International Scientific-Technical Conference "Dynamic of Technical Systems" (DTS-2017), Rostov-on-Don, Russian Federation, September 13-15, 2017.DOI: https://doi.org/10.1051/matecconf/201713204003
8. A.I. Sukhinov, Yu.V. Belova, A.E. Chistyakov, Computational Methods and Programming, 18(4), 371-380 (2017).
9. V.M. Goloviznin, A.A. Samarskiy, Matemat. modeling, **10(1)**, 86-100 (1998)
10. V.A. Gushchin, Matemat. modeling, **28(2)**, 6-18 (2016)
11. O.M. Belotserkovsky, V.A. Gushchin, V.N. Konshin, Splitting method for studying the flows of a stratified liquid with a free surface. Zh. calculation. matem. and math. phys., **27(4)**, 594-609 (1987)
12. A.I. Sukhinov, A.E. Chistyakov, A.V. Shishenya, Mathematical Models and Computer Simulations, **6(3)**, 324-331 (2014).
13. A.I., Sukhinov, A.E. Chistyakov, M.V. Yakobovsky, Vestn. SUSU. Ser. Vych. matem. inform., **5(1)**, 47-62 (2016)

14. R.P. Fedorenko, Application of high-precision difference schemes for numerical solution of hyperbolic equations. Zh. calculation. matem. and math. phys., **2(6)**, 1122-1128 (1962), USSR prog. Mathematics. Mathematics. Phys., **2(6)**, 1355-1365 (1963)