COMPUTATIONAL MATHEMATICS
ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА

UDC 519.688
https://doi.org/10.23947/2587-8999-2023-7-2-19-30

Original article

Numerical Realization of Shallow Water Bodies' Hydrodynamics Grid Equations using Tridiagonal Preconditioner in Areas of Complex Shape

Vladimir N Litvinov[1,2] ✉, **Asya M Atayan[1]**, **Natalya N Gracheva[1,2]**,
Nelly B Rudenko[1,2], **Natalya Yu Bogdanova[1]**

[1]Don State Technical University, 1, Gagarina Sq., Rostov-on-Don, Russian Federation
[2]Azov-Black Sea Engineering Institute of Don State Agrarian University, 21, Lenina St., Zernograd, Russian Federation
✉ LitvinovVN@rambler.ru

Abstract
Introduction. Mathematical modeling of hydrodynamic processes in shallow reservoirs of complex geometry in the presence of coastal engineering systems requires an integrated approach in the development of algorithms for constructing computational grids and methods for solving grid equations. The work is devoted to the description of algorithms that allow to reduce the time for solving SLAE by using an algorithm for processing overlapping geometry segments and organizing parallel pipeline calculations. The aim of the work is to compare the acceleration of parallel algorithms for the methods of Seidel, Jacobi, modified alternately triangular method and the method of solving grid equations with tridiagonal preconditioner depending on the number of computational nodes.
Materials and Methods. The numerical implementation of the modified alternating-triangular iterative method for solving grid equations (MATM) of high dimension is based on parallel algorithms based on a conveyor computing process. The decomposition of the computational domain for the organization of the pipeline calculation process has been performed. A graph model is introduced that allows to fix the connections between neighboring fragments of the computational grid. To describe the complex geometry of a reservoir, including coastal structures, an algorithm for overlapping geometry segments is proposed.
Results. It was found that the efficiency of implementing one step of the MATM on the GPU depends only on the number of threads along the O_z axis, and the step execution time is inversely proportional to the number of nodes of the computational grid along the O_z axis. Therefore, it is recommended to decompose the computational domain into parallelepipeds in such a way that the size along the O_z axis is maximum, and the size along the O_x axis is minimal. Thanks to the algorithm for combining geometry segments, it was possible to speed up the calculation by 14–27 %.
Discussion and Conclusions. An algorithm has been developed and numerically implemented for solving a system of large-dimensional grid equations arising during the discretization of the shallow water bodies' hydrodynamics problem by MATM, adapted for heterogeneous computing systems. The graph model of a parallel-pipeline computing process is proposed. The connection of water body's geometry segments allowed to reduce the number of computational operations and increase the speed of calculations. The efficiency of parallel algorithms for the methods of Seidel, Jacobi, modified alternately triangular method and the method of solving grid equations for problems of hydrodynamics in flat areas, depending on the number of computational nodes, is compared.

Keywords: mathematical modeling, computational domain geometry, parallel programming, graphics accelerator.

Funding information. The study was supported by the Russian Science Foundation no. 21-71-20050. https://rscf.ru/project/21-71-20050/

For citation. Litvinov VN, Atayan AM, Gracheva NN, et al. Numerical realization of shallow water bodies' hydrodynamics grid equations using tridiagonal preconditioner in areas of complex shape. *Computational Mathematics and Information Technologies*. 2023;7(2):19–30. https://doi.org/10.23947/2587-8999-2023-7-2-19-30

Научная статья

Численная реализация сеточных уравнений гидродинамики мелководных водоёмов с использованием трехдиагонального предобуславливателя в областях сложной формы

В.Н. Литвинов[1,2] ✉, **А.М. Атаян**[1], **Н.Н. Грачева**[1,2], **Н.Б. Руденко**[1,2], **Н.Ю. Богданова**[1]

[1]Донской государственный технический университет, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1

[2]Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ, Российская Федерация, г. Зерноград, ул. Ленина, 21

✉ LitvinovVN@rambler.ru

Аннотация

Введение. Математическое моделирование гидродинамических процессов в мелководных водоёмах сложной геометрии при наличии прибрежных инженерных систем требует комплексного подхода при разработке алгоритмов построения расчетных сеток и методов решения сеточных уравнений. Работа посвящена описанию алгоритмов, позволяющих уменьшить время решения СЛАУ за счёт использования алгоритма обработки наложения сегментов геометрии и организации параллельно-конвейерных вычислений. Целью работы является сравнение ускорения параллельных алгоритмов для методов Зейделя, Якоби, модифицированного попеременно-треугольного метода и метода решения сеточных уравнений с трехдиагональным предобуславливателем в зависимости от количества вычислительных узлов.

Материалы и методы. Численная реализация модифицированного попеременно-треугольного итерационного метода решения сеточных уравнений (МПТМ) высокой размерности основана на параллельных алгоритмах, построенных на основе конвейерного вычислительного процесса. Произведена декомпозиция расчётной области для организации процесса конвейерного вычисления. Введена графовая модель, позволяющая зафиксировать связи между соседними фрагментами расчетной сетки. Для описания сложной геометрии водоёма, включающей прибрежные сооружения, предложен алгоритм наложения сегментов геометрии.

Результаты исследования. В ходе исследований было установлено, что время расчета одного шага МПТМ на GPU зависит от количества потоков по оси O_z и обратно пропорционально количеству узлов расчетной сетки по данной оси. Поэтому рекомендуется декомпозировать расчетную область на параллелепипеды таким образом, чтобы их размер по оси O_x был наименьшим, а по O_z — наибольшим. Предложенный алгоритм объединения сегментов геометрии позволил уменьшить время вычислений на величину от 14 до 27 %.

Обсуждение и заключения. Разработан и численно реализован алгоритм решения системы сеточных уравнений большой размерности, возникающих при дискретизации задачи гидродинамики мелководного водоема методом МПТМ, адаптированный для гетерогенных вычислительных систем. Предложена графовая модель параллельно-конвейерного вычислительного процесса. Соединение сегментов геометрии водного объекта позволило сократить количество вычислительных операций и увеличить скорость расчетов. Проведено сравнение эффективности параллельных алгоритмов для методов Зейделя, Якоби, модифицированного попеременно-треугольного метода и метода решения сеточных уравнений для задач гидродинамики в плоских областях в зависимости от количества вычислительных узлов.

Ключевые слова: математическое моделирование, геометрия расчетной области, параллельное программирование, графический ускоритель.

Финансирование. Исследование выполнено за счет гранта Российского научного фонда № 21-71-20050. https://rscf.ru/project/21-71-20050/

Для цитирования. Литвинов В.Н., Атаян А.М., Грачева Н.Н. и др. Численная реализация сеточных уравнений гидродинамики мелководных водоёмов с использованием трехдиагонального предобуславливателя в областях сложной формы. *Computational Mathematics and Information Technologies*. 2023;7(2):19–30. https://doi.org/10.23947/2587-8999-2023-7-2-19-30

Introduction. Mathematical modeling is used to predict the state of shallow reservoirs in emergency situations caused by human activity or natural and climatic disasters. It is necessary to take into account such features of each specific water body as the geometry of the reservoir and its coastal zone, climatic conditions and hydrodynamic regimes. Such problems actualize the improvement of methods for solving systems of grid equations of high dimension in the case of a non-self-adjoint operator. It is necessary to use multiprocessor computing systems and video adapters to increase the speed of obtaining a solution, due to the large amount of data and the complexity of calculations.

Modeling of many hydrophysical and hydrobiological problems reduces to the need to solve the diffusion-convection-reaction equation with a non-self-adjoint operator. The review of actual numerical methods of solution is carried out in the work of P. Vabishevich [1], where a number of theorems are formulated that allow determining the numerical parameters and the limits of applicability of the studied methods for solving grid equations. Iterative methods for solving such problems are actively developing. In the work of Geiser, Hueso, Martinez [2], various types of splitting methods are analyzed, modifications of SLIS and SQIS methods are proposed, on the basis of which effective adaptive algorithms are built that allow increasing the time step without reducing the accuracy of calculations.

There has been a significant increase in the number of studies aimed at developing algorithms that are efficient in computing speed and designed to solve systems of high-dimensional grid equations over the past few years. Russian and foreign scientists are developing parallel algorithms for heterogeneous computing environments, studying the performance of cluster computing systems for various methods of discretization of various differential equations. For example, Subbaian G. and Reddy Sathi [3, 4] analyzed the performance of several iterative methods for solving the Navier-Stokes equation with accelerated computing on a graphics processor (GPU) using CUDA technology. Scientists Lakshmiranganatha S., Muknahallipatna S., Paliwal M., Chilla R., Prasanth N., Goundar S. and Raja S.P. compared the performance of various parallel algorithms for finding solutions to time-dependent ordinary differential equations on CPU and GPU using three parallelization technologies: OpenMP, OpenACC and CUDA. It was found that CUDA technology is the most effective accelerator for solving these equations as a result of the study [5, 6]. Russian and Kazakh scientists have developed parallel algorithms for finding solutions to systems of linear algebraic equations. The algorithms were implemented on multicore processors using OpenMP technology [7, 8]. The efficiency of parallel algorithms for solving the one-dimensional thermal conductivity problem for three finite-difference approximation methods was tested on central and graphics processors in the programming languages C (CPU) and CUDA C (GPU). GPU computing acceleration increased up to 60 times [9, 10]. In [11], the construction of parallel algorithms based on the functional decomposition of the counter-run method for solving tridiagonal grid equations is considered. D. B. Volkov-Bogorodsky, G. B. Sushko and S. A. Kharchenko in their work [12] describe hybrid parallel algorithms for approximating solutions of the nonstationary thermal conductivity equation with phase transitions based on the analytical method blocks, namely MPI+threads technology.

It is necessary to develop a parallel version of the algorithm, which will reduce the time of solving SLAE by using an algorithm for processing the overlapping of geometry segments and parallelizing the calculation process, in this study,

Materials and Methods

1. Problem statement. Shallow water bodies' hydrodynamics mathematical model includes [13]:

– Navier-Stokes equations:

$$u_t' + uu_x' + vu_y' + wu_z' = -\frac{1}{\rho}P_x' + \left(\mu u_x'\right)_x' + \left(\mu u_y'\right)_y' + \left(\mu u_z'\right)_z' + 2\Omega\left(v\sin\vartheta - w\cos\vartheta\right), \tag{1}$$

$$v_t' + uv_x' + vv_y' + wv_z' = -\frac{1}{\rho}P_y' + \left(\mu v_x'\right)_x' + \left(\mu v_y'\right)_y' + \left(v v_z'\right)_z' + 2\Omega u\sin\vartheta, \tag{2}$$

$$w_t' + uw_x' + vw_y' + ww_z' = -\frac{1}{\rho}P_z' + \left(\mu w_x'\right)_x' + \left(\mu w_y'\right)_y' + \left(v w_z'\right)_z' + 2\Omega u\cos\vartheta + g; \tag{3}$$

– continuity equation in the case of variable density:

$$\rho_t' + \left(\rho u\right)_x' + \left(\rho v\right)_y' + \left(\rho w\right)_z' = 0, \tag{4}$$

where $V = \{u, v, w\}$ are the velocity vector components; P is the total hydrodynamic pressure; ρ is the aqueous medium density; μ, v are the horizontal and vertical turbulent exchange coefficient components; $\Omega = \Omega \cdot (\cos\vartheta \cdot j + \sin\vartheta \cdot k)$ is the angular velocity of the Earth›s rotation; ϑ is the latitude of the place; g is the acceleration of gravity; f_T, f_s are the sources of heat and salt (located on the region border).

The initial hydrodynamics model (1–4) is divided into several subtasks [14, 15]. The first subtask is represented by the diffusion-convection-reaction equation, which is used to calculate the components of the velocity vector field on the intermediate layer in time:

$$\frac{\tilde{u}-u}{\tau}+u\bar{u}'_x+v\bar{u}'_y+w\bar{u}'_z=\left(\mu\bar{u}'_x\right)'_x+\left(\mu\bar{u}'_y\right)'_y+\left(\nu\bar{u}'_z\right)'_z+2\Omega(v\sin\theta-w\cos\theta),$$

$$\frac{\tilde{v}-v}{\tau}+u\bar{v}'_x+v\bar{v}'_y+w\bar{v}'_z=\left(\mu\bar{v}'_x\right)'_x+\left(\mu\bar{v}'_y\right)'_y+\left(\nu\bar{v}'_z\right)'_z-2\Omega u\sin\theta, \tag{5}$$

$$\frac{\tilde{w}-w}{\tau}+u\bar{w}'_x+v\bar{w}'_y+w\bar{w}'_z=\left(\mu\bar{w}'_x\right)'_x+\left(\mu\bar{w}'_y\right)'_y+\left(\nu\bar{w}'_z\right)'_z+2\Omega u\cos\theta+g\left(\frac{\rho_0}{\rho}-1\right).$$

The Krank-Nicholson scheme was used to approximate the diffusion-convection-reaction equation (5) in time. Here, $\bar{u}=\sigma\tilde{u}+(1-\sigma)u,\ \ \sigma\in[0,1]$ is the diagram weight.

2. Methods for solving grid equations. Modified alternately triangular iterative method. Let's introduce three-dimensional uniform computational grid [14]:

$$\bar{w}_h=\{t^n=n\tau, x_i=ih_x,\ y_j=jh_y,\ z_k=kh_z;\ n=\overline{0,n_t-1},\ i=\overline{0,n_1-1},$$

$$j=\overline{0,n_2-1},\ k=\overline{0,n_3-1};\ (n_t-1)\tau=T,\ (n_1-1)h_x=l_x,(n_2-1)h_y=l_y,(n_3-1)h_z=l_z\},$$

where τ is the time step; h_x, h_y, h_z is the size of the steps along the coordinate directions; n_t is the number of time layers; T is the upper bound by time coordinate; n_1, n_2, n_3 are the number of nodes by spatial coordinates; l_x, l_y, l_z are the spatial dimensions of the calculated area.

Let's get a system of grid equations when constructing a discrete model. Each equation of the system can be presented in canonical form, and we will use a seven-point template:

$$c(m_0)u(m_0)-\sum_{i=1}^{6}c(m_0,m_i)u(m_i)=F(m_0),$$

where $m_0(x_i,y_j,z_k)$ is the template center; $M'(P)=\{m_1(x_{i+1},y_j,z_k),\ \ m_2(x_{i-1},y_j,z_k),\ \ m_3(x_i,y_{j+1},z_k),\ \ m_4(x_i,y_{j-1},z_k),$ $m_5(x_i,y_j,z_{k+1}),\ \ m_6(x_i,y_j,z_{k-1})\}$ is the neighborhood of the center; $c_0\equiv c(m_0)$ is the coefficient of the template center; $c_i\equiv c(m_0,m_i)$ are the coefficients of the neighborhood of the template center; F is the vector of the right parts; u is the calculated vector.

The MATM algorithm consists of four stages:

1) calculation of the residual vector r^m;

2) calculation of the correction vector w^m;

3) calculation of scalar products based on iterative parameters τ_{m+1},ω_{m+1};

4) transition to a new iterative layer.

The condition for the end of the iterative process is that the norm of the residual vector r^m reaches the specified accuracy. At the same time, the most time-consuming part of the algorithm is the calculation w^m, which boils down to solving SLAE with lower-triangular and upper-triangular matrices.

3. Method for solving grid equations with tridiagonal preconditioner. If the steps along one of the spatial coordinates are significantly smaller than the steps along the others (for example, when solving problems of heat and mass transfer in shallow reservoirs), the dimensions of the calculated area in the vertical direction can be hundreds to thousands of times smaller than the horizontal dimensions. To solve problem (1) on the basis of difference schemes with relatively small labor costs for the transition between time layers, compared with the explicit scheme (1.5–2 times larger), with large time steps (about 30 times more), we will use splitting schemes into two-dimensional and one-dimensional problems [16–17]:

$$\frac{c^{n+1/2}-c^{n}}{\tau}+u\left(c^{n}\right)'_{x}+v\left(c^{n}\right)'_{y}=\left(\mu\left(c^{n}\right)'_{x}\right)'_{x}+\left(\mu\left(c^{n}\right)'_{y}\right)'_{y}, \tag{6}$$

$$\frac{c^{n+1}-c^{n+1/2}}{\tau}+w\left(c^{n+(\sigma+1)/2}\right)'_{z}=\left(\nu\left(c^{n+(\sigma+1)/2}\right)'_{z}\right)'_{z}+f^{n+(\sigma+1)/2}, \tag{7}$$

where $c^{n+(\sigma+1)/2}=\sigma c^{n+1}+(1-\sigma)c^{n+1/2}$; σ is the weight of the scheme [7].

The spatial grid is introduced for the numerical implementation of a discrete mathematical model of the problem [18]:

$$\overline{w}_h=\{t^n=n\tau, x_i=ih_x, y_j=jh_y; n=\overline{0,n_t-1}, i=\overline{0,n_1-1},$$

$$j=\overline{0,n_2-1};\ (n_t-1)\tau=T,(n_1-1)h_x=l_x,(n_2-1)h_y=l_y\}.$$

To approximate the homogeneous equation (2), splitting schemes in spatial coordinate directions will be used:

$$\begin{gathered}\frac{c^{n+1/4}-c^{n}}{\tau}+u\left(c^{n}\right)'_{x}=\left(\mu\left(c^{n}\right)'_{x}\right)'_{x},\\ \frac{c^{n+1/2}-c^{n+1/4}}{\tau}+v\left(c^{n+1/4}\right)'_{y}=\left(\mu\left(c^{n+1/4}\right)'_{y}\right)'_{y}.\end{gathered} \tag{8}$$

To solve real problems of hydrophysics of shallow water bodies, three-layer difference schemes based on a linear combination of the Upwind Leapfrog and Standart Leapfrog difference schemes with weight coefficients 2/3 and 1/3, respectively, are used. To increase the accuracy of calculations, a scheme is used that takes into account the fullness of the calculation cells [19–21]:

– difference scheme for the equation describing the transfer along the direction O_x:

$$\frac{2q_{2,i,j}+q_{0,i,j}}{3}\frac{c_{i,j}^{n+1/4}-c_{i,j}^{n}}{\tau}+5u_{i-1/2,j}q_{2,i,j}\frac{c_{i,j}^{n}-c_{i-1,j}^{n}}{3h_x}+u_{i+1/2,j}\min\left(q_{1,i,j},q_{2,i,j}\right)\frac{c_{i+1,j}^{n}-c_{i,j}^{n}}{3h_x}+$$

$$+\frac{2\Delta_x c_{i-1,j}^{n}q_{2,i,j}+\Delta_x c_{i,j}^{n}q_{0,i,j}}{3}=2\mu_{i+1/2,j}q_{1,i,j}\frac{c_{i+1,j}^{n}-c_{i,j}^{n}}{h_x^2}-2\mu_{i-1/2,j}q_{2,i,j}\frac{c_{i,j}^{n}-c_{i-1,j}^{n}}{h_x^2}-$$

$$-\left|q_{1,i,j}-q_{2,i,j}\right|\mu_{i,j}\frac{\alpha_x c_{i,j}^{n}+\beta_x}{h_x},\ u_{i,j}\ge 0,$$

where $\Delta_x c_{i,j}^{n}=\dfrac{c_{i,j}^{n-3/4}-c_{i,j}^{n-1}}{\tau}$;

– difference scheme for equation (4) describing the transfer along the direction O_y:

$$\frac{2q_{4,i,j}+q_{0,i,j}}{3}\frac{c_{i,j}^{n+1/2}-c_{i,j}^{n+1/4}}{\tau}+5v_{i,j-1/2}q_{4,i,j}\frac{c_{i,j}^{n+1/4}-c_{i,j-1}^{n+1/4}}{3h_y}+v_{i,j+1/2}\min\left(q_{3,i,j},q_{4,i,j}\right)\frac{c_{i,j+1}^{n+1/4}-c_{i,j}^{n+1/4}}{3h_y}+$$

$$+\frac{2\Delta_y c_{i,j-1}^{n+1/4}q_{4,i,j}+\Delta_y c_{i,j}^{n+1/4}q_{0,i,j}}{3}=2\mu_{i,j+1/2}q_{3,i,j}\frac{c_{i,j+1}^{n+1/4}-c_{i,j}^{n+1/4}}{h_y^2}-2\mu_{i,j-1/2}q_{4,i,j}\frac{c_{i,j}^{n+1/4}-c_{i,j-1}^{n+1/4}}{h_y^2}-$$

$$-\left|q_{3,i,j}-q_{4,i,j}\right|\mu_{i,j}\frac{\alpha_y c_{i,j}^{n+1/4}+\beta_y}{h_y},\ v_{i,j}\ge 0,$$

where $\Delta_y c_{i,j}^{n+1/4}=\dfrac{c_{i,j}^{n-1/2}-c_{i,j}^{n-3/4}}{\tau}$.

Where q_0, q_1, q_2, q_3, q_4 are the degrees of control areas occupancy.

To obtain difference schemes approximating the system of equations (4), with $u_{i,j}<0$ and $v_{i,j}<0$ from the approximations presented, it is necessary to direct the corresponding coordinate axes O_x and O_y in opposite directions. Equation (3) is solved by the run-through method.

The splitting scheme into two-dimensional and one-dimensional problems has an advantage for solving non-stationary problems. The two-dimensional problem is solved on the basis of explicit schemes, and the one-dimensional one is approximated by schemes with weights and solved by the run-through method in this case. Schemes with weights are used, when solving stationary problems. Using this approach allows to reduce the initial problem to solving grid equations by iterative methods [22].

4. Geometry Segment overlay algorithm. It is necessary to take into account the complex geometry of the reservoir formed by a combination of the bottom surface and coastal engineering structures when drawing up SLAE. Programmatically, an approach is proposed and implemented that allows modeling the geometry of the object under study as a set of geometric primitives. A feature of the approach is the support for superimposing primitives on each other. In the class library developed with the software implementation, all geometric primitives are inherited from the abstract Geometry2DPrimitive class (Fig. 1), which contains data such as the _dS0 offset coordinates, the _primitiveType primitive type, and a logical property characterizing the "cutout" (_isCavity).

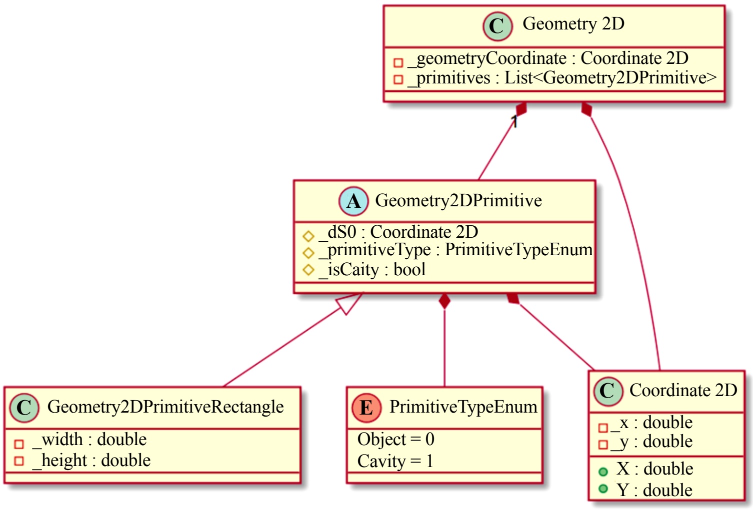

Fig. 1. Research object geometry. Class diagram

Object-oriented modeling of a geometry segment is performed using a typed class GeometryPrimitiveSegment<T>, in which type T is a class used to store data about the coordinates of the beginning and end of the segment.

Let's denote the coordinates of the beginning and end of the first and second segments c_{11}, c_{12} ,c_{21}, c_{22}.

Let's introduce logical variables: $A = c_{11} < c_{21}$, $B = c_{11} == c_{21}$, $C = c_{11} > c_{21}$, $D = c_{12} < c_{22}$, $E = c_{12} == c_{22}$, $F = c_{12} > c_{22}$, $G = c_{11} == c_{22}$, $H = c_{12} == c_{21}$, $I = c_{12} < c_{21}$, $J = c_{11} > c_{22}$. K, L are the types of the first and second segments; M, N are the fullness of the space under the first and second segments; V is the flag indicating that the second segment is a "cutout".

All possible combinations of overlapping geometry segments are summarized in Table 1.

The basis of the original linear algorithm is to take into account various combinations of geometric primitives. To increase productivity, a number of modifications based on conditional constructions have been introduced:

1. Initialization: c_{11}, c_{12} ,c_{21}, c_{22}.

2. Calculation: A, B, C, D, E, F, G, H, I, J, K, L, M, N, V.

3. Checking the correctness condition: $L \wedge \overline{N} \vee K \wedge \overline{M} \neq \text{true}$.

4. Definition of the overlay type.

5. Further actions are performed for the found overlay type. For example, option No. 1 is described (Table 1). For other types, the actions are the same.

6. If the first segment is not a boundary $(K = \text{true})$, then go to step 12.

7. If the second segment is not a boundary $(L = \text{true})$, then go to step 10.

8. Calculating expressions $\overline{M} \wedge \overline{N} \wedge \overline{V}$, $\overline{M} \wedge \overline{N} \wedge V$, $\overline{M} \wedge N \wedge \overline{V}$, $\overline{M} \wedge N \wedge V$, $M \wedge \overline{N} \wedge \overline{V}$, $M \wedge \overline{N} \wedge V$ $M \wedge N \wedge \overline{V}$, $M \wedge N \wedge V$ and creating the resulting segments.

9. Go to step 17.
10. Calculating $\overline{M} \wedge N \wedge V$, $M \wedge N \wedge \overline{V}$, $M \wedge N \wedge V$, $\overline{M} \wedge N \wedge \overline{V}$ and creating the resulting segments.
11. Go to step 17.
12. If the second segment is boundary $(L = \text{ false})$, then transition to step 15, otherwise transition to step 17.
13. Checking the conditions $M \wedge N \wedge V,\ M \wedge \overline{N} \wedge V,\ M \wedge \overline{N} \wedge \overline{V},\ M \wedge N \wedge \overline{V}.$
14. Creating and returning the resulting segments.
15. Go to step 17.
16. Calculation $M \wedge N \wedge V,\ M \wedge N \wedge \overline{V}$ and creation of the resulting segments.
17. The end.

Table 1

Options for overlapping geometry segments

Overlay option	Graphical interpretation	Logical expression
1		H
2		G
3		$B \wedge E$
4		$A \wedge F$
5		$C \wedge D$
6		$B \wedge F$
7		$B \wedge D$
8		$A \wedge E$
9		$C \wedge E$
10		$C \wedge F$
11		$A \wedge D \wedge \bar{I}$
12		I
13		J

5. Parallel implementation. The pipeline parallel algorithm has been developed that allows using all available computing resources for the numerical implementation of the MATM applicable to a high-dimensional SLAE. At the same time, each computer (CPU core or GPU computing unit) processes only the fragments of the computational domain assigned to it.

The connections between fragments and the organization of the parallel-pipeline computing process are described by a graph model, where nodes represent fragments of the computational domain. The computational process is organized according to the values of the counter of the calculation stages $s = k \cdot i + j$.

The developed graph model is used in the algorithm for solving SLAE with a lower-triangular matrix (Fig. 3). The input parameters of the algorithm are the coefficients of grid equations c_0, c_2, c_4, c_6 and the constant ω. The result is the velocity vector of the water flow r. When starting the software implementation of the algorithm in the CUDA C language, it is necessary to set the values of the dimensions of the CUDA computing blocks *blockDim.x, blockDim.z* . The parallel-pipelined computing process is organized as a cycle (line 6).

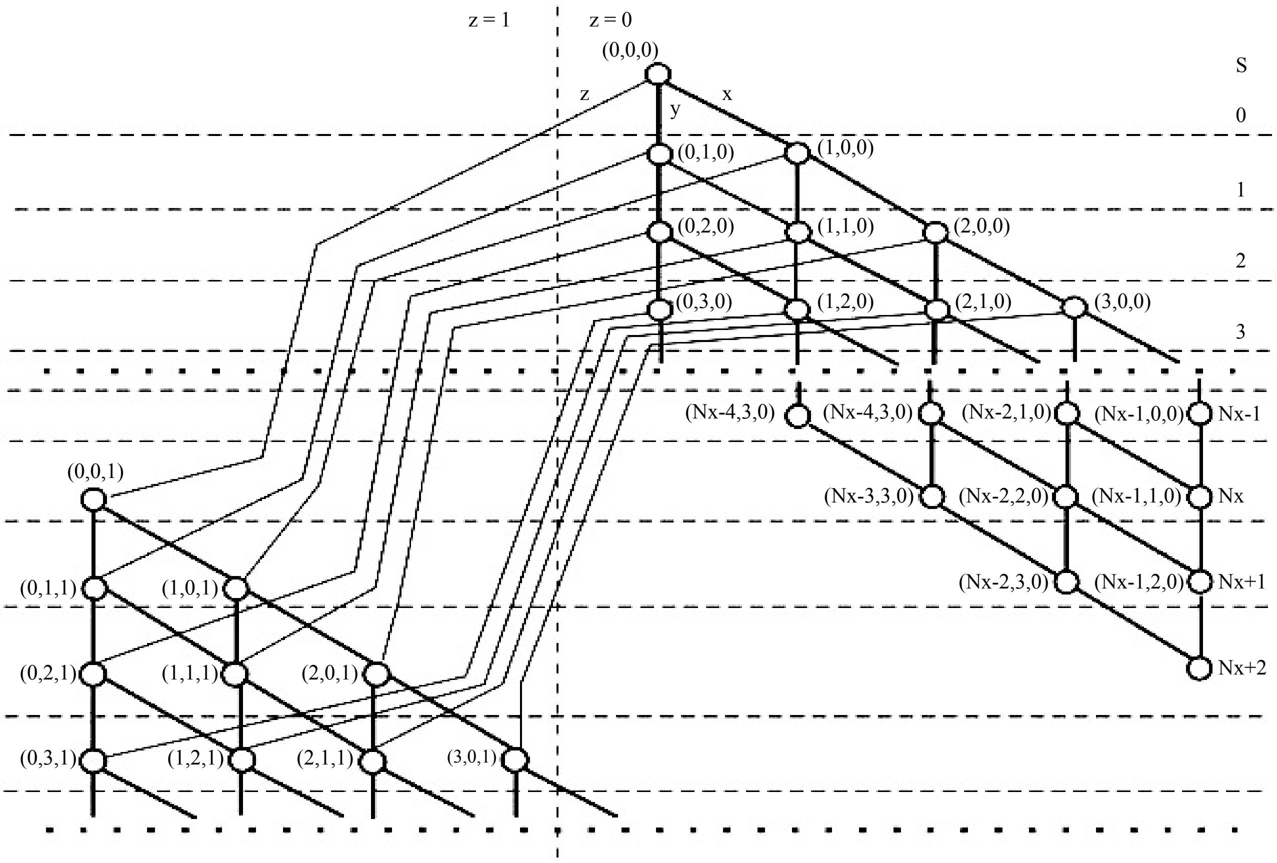

Fig. 2. Graph model of parallel-pipeline computing process

Algorithm 1 ptmKernel3(IN: $c_0, c_2, c_4, c_6, \omega$ IN/OUT: r;)

```
threadX ← blockDim.x · blockIdx.x + threadIdx.x;
threadZ ← blockDim.z · blockIdx.z + threadIdx.z;
i ← threadX + 1;
j ← 1;
k ← threadZ + 1;
for s ∈ [3; n1 + n2 + n3 − 3] do
    if (i + j + k = s) ∧ (s < i + n2 + k) then
        m0 ← i + (blockDim.x + 1) · j + n1 · n2 · k;
        if c0[m0] > 0 then
            m2 ← m0 − 1;
            m4 ← m0 − n1;
            m6 ← m0 − n1 · n2;
            rm4 ← 0;
            if (s > 3 + threadX + threadZ) then
                rm4 ← cache[threadX][threadZ];
            else
                rm4 ← r[m4];
            rm2 ← 0;
            if (threadX ≠ 0) ∧ (s > 3 + threadX + threadZ) then
                rm2 ← cache[threadX − 1][threadZ];
            else
                rm2 ← r[m2];
            rm6 ← 0;
            if (threadZ ≠ 0) ∧ (s > 3 + threadX + threadZ) then
                rm6 ← cache[threadX][threadZ − 1];
            else
                rm6 ← r[m6];
            rm0 ← (ω · (c2[m0] · rm2 + c4[m0] · rm4 + c6[m0] · rm6) + r[m0])/((0.5 · ω + 1) · c0[m0]);
            cache[threadX][threadZ] ← rm0;
            r[m0] ← rm0;
    j ← j + 1;
```

Fig. 3. Algorithm for solving a system of equations with a lower triangular matrix

Two-dimensional array *cache*, placed in shared GPU memory has been introduced to reduce the number of reads from global video memory. It stores the intermediate results of calculations on the current layer along the axis O_y, which speeds up the calculation process by 30 %.

The results of the study. Computational experiment comparing the performance of the basic and modified algorithms was conducted on a computer system with an Intel Core i5 3.3 GHz processor and 32 GB DDR4 RAM (Table 2). The modified algorithm recorded a decrease in the calculation time by up to 27 %.

Table 2

Results of comparing the performance of the basic and modified algorithms for combining geometry segments

Number of unions, $\times 10^6$	1	2	3	4	5	6	7	8	9	10
Basic algorithm, s	0.53	0.75	0.13	0.16	0.19	0.23	0.26	0.29	0.35	0.38
Modified algorithm, s	0.41	0.55	0.97	0.12	0.15	0.20	0.22	0.25	0.29	0.31

The numerical experiment was carried out to determine the number of GPU threads along the axes O_x and O_z the calculated grid (X, Z) with a fixed value of grid nodes along the axis O_y, equal to 10000, which allows to reduce the calculation time of one step of the MATM (T_{GPU}) on the GPU. The levels of variation of the factors X and Z and the results of the numerical experiment are shown in Table 3.

Table 4

Results of the experiment

№	X	Z	T_{GPU}, s
1.	16	64	0.064
2.	32	32	0.065
3.	64	16	0.081
4.	128	8	0.109
5.	256	4	0.100
6.	512	2	0.103

In the experiment, it was found that the calculation time of one MATM step on the GPU is inversely proportional to the number of nodes of the calculated grid along the axis O_z. The smallest value of the objective function is obtained at X and Z, equal to 16 and 64, respectively.

Table 5

Comparison of parallel algorithms' acceleration

P	Jacobi		Seidel		MATM		MSGE with a tridiagonal preconditioner	
	Speed-up ratio	Efficiency	Speed-up ratio	Efficiency	Speed-up ratio	Efficiency	Speed-up ratio	Efficiency
1	1.00	100.00	1.00	100.00	1.00	100.00	1.00	100.00
2	1.95	97.50	1.95	97.50	1.94	97.00	1.84	92.00
3	2.96	98.67	2.92	97.33	2.82	94.00	2.97	99.00
4	3.98	99.50	3.75	93.75	3.82	95.50	3.32	83.00
8	7.36	92.00	7.02	87.75	7.31	91.38	8.03	100.38
16	13.29	83.06	12.92	80.75	12.78	79.88	15.80	98.75
24	16.93	70.54	16.49	68.71	17.03	70.96	19.53	81.38

Table 4 presents a comparison of the speed-up ratio of parallel algorithms for the Seidel, Jacobi methods, the modified alternately triangular method and the method for solving grid equations with a tridiagonal preconditioner on the number of computational nodes. Calculations were made on a grid of one million calculation cells. The launches were carried out sequentially, starting from the launch on one computing node and ending with the connection of all available nodes.

Discussion and Conclusions. Algorithms for solving SLAE obtained by discretizing the problem of hydrodynamics of a shallow reservoir, MATM using NVIDIA CUDA technology are proposed. The proposed method of decomposition of the computational grid and the graph model make it possible to efficiently organize parallel pipeline calculations on computing systems of various configurations.

Numerical experiments have been carried out to determine the best two-dimensional configuration of threads in the computing unit, minimizing the time of one step of the MATM on the GPU, $-X = 16$ and $Z = 64$.

The maximum speed-up ratio was shown by the method of solving grid equations for hydrodynamic problems in flat areas, which is based on an explicit-implicit scheme. MATM, in comparison with the methods of Jacobi and Seidel, requires significantly fewer iterations for convergence. With a good optimization of the parallel MATM algorithm, the speed-up ratio differs by no more than 10 % by the number of computing nodes up to 24 compared to the acceleration of the parallel algorithm of the Jacobi method.

The developed software tools make it possible to more effectively use the computing resources of the GPU used to solve computationally time-consuming spatial-three-dimensional problems of hydrophysics.

Combining segments of the geometry of the object under study leads to a reduction in the number of computational operations, which allows to increase the performance of calculations.

References

1. Vabishchevich P. Iterative Methods for Solving Convection-diffusion Problem. *Computational Methods in Applied Mathematics*. 2002;2(4):410–444. https://www.doi.org/10.2478/cmam-2002-0023

2. Geiser J, Hueso J, Martinez E. Adaptive Iterative Splitting Methods for Convection-Diffusion-Reaction Equations. *Mathematics.* 2020;8:302. https://www.doi.org/10.3390/math8030302

3. Subbaian G, Reddy S. *Performance Analysis of Different Iterative Solvers Parallelized On GPU Architecture*. 2023;2:215–220. https://www.doi.org/10.1007/978-981-19-6970-6_39

4. Lakshmiranganatha S, Muknahallipatna S. Performance Analysis of Accelerator Architectures and Programming Models for Parareal Algorithm Solutions of Ordinary Differential Equations. *Journal of Computer and Communications.* 2021;9(2):29–56. https://www.doi.org/10.4236/jcc.2021.92003

5. Temirbekov A, Baigereyev D, Temirbekov N, et al. Amantayeva A. Parallel CUDA implementation of a numerical algorithm for solving the Navier-Stokes equations using the pressure uniqueness condition. AIP Conference Proceedings; 2021;2325:020063. https://www.doi.org/10.4236/jcc.2021.9200310.1063

6. Paliwal M, Chilla R, Prasanth N, et al. Parallel implementation of solving linear equations using OpenMP. *International Journal of Information Technology*. 2022;14:1677–1687. https://www.doi.org/10.1007/s41870-022-00899-9

7. Akimova EN, Sultanov MA, Misilov VE, et al. Parallel sweep algorithm for solving direct and inverse problems for time-fractional diffusion equation. *Numerical Methods and Programming (Vychislitel'nye Metody i Programmirovanie)*. 2022;23(4):275–287. (In Russ.). https://www.doi.org/10.26089/NumMet.v23r417

8. Sultanov M, Akimova E, Misilov V, et al. Parallel Direct and Iterative Methods for Solving the Time-Fractional Diffusion Equation on Multicore Processors. *Mathematics*. 2022;10(3):323. https://www.doi.org/10.3390/math10030323

9. Sechenov P, Rybenko I. Solving the problem of one-dimensional thermal conductivity on graphics processors using CUDA technology. *Applied Mathematics and Control Sciences*. 2021;4:23–41. https://www.doi.org/10.15593/2499-9873/2021.4.02

10. Khimich A, Polyanko V, Chistyakova T. Parallel Algorithms for Solving Linear Systems on Hybrid Computers. *Cybernetics and Computer Technologies*. 2020:53–66. https://www.doi.org/10.34229/2707-451X.20.2.6

11. Golovashkin DL. Parallel algorithms for solving tridiagonal grid equations based on the method of counter runs. *Mathematical modeling*. 2005;17(11):118–128. (In Russ.).

12. Volkov-Bogorodsky DB, Sushko GB, Kharchenko SA. Combined MPI+threads parallel implementation of the block method for modeling thermal processes in structurally inhomogeneous media. *Computational methods and programming*. 2010;11(1):127–136. (In Russ.).

13. Munk DJ, Kipouros T, Vio GA. Multi-physics bi-directional evolutionary topology optimization on GPU-architecture. *Engineering with Computers*. 2019;35(4):1059–1079. https://www.doi.org/10.1007/s00366-018-0651-1

14. Sukhinov AI, Chistyakov AE, Shishenya AV, et al. Predictive Modeling of Coastal Hydrophysical Processes in Multiple-Processor Systems Based on Explicit Schemes. *Mathematical Models and Computer Simulations*. 2018;10(5):648–658. https://www.doi.org/10.1134/S2070048218050125

15. Konovalov AN. The method of rapid descent with an adaptive alternately triangular preconditioner. Differential equations. 2004;40(7):953–963. (In Russ.).

16. Samarskiy AA, Vabishevich PN. *Numerical methods for solving convection-diffusion problems, Stereotype Publishing House*. Moscow: Book House «LIBROCOM»; 2015. 248 p. (In Russ.).

17. Oyarzun G, Borrell R, Gorobets A, et al. MPI-CUDA sparse matrix–vector multiplication for the conjugate gradient method with an approximate inverse preconditioner. *Computers and Fluids*. 2014;92:244–252. https://www.doi.org/10.1016/j.compfluid.2013.10.035

18. Khokhlov NI, Petrov IB. Application of the grid-characteristic method for solving the problems of the propagation of dynamic wave disturbances in high-performance computing systems. *Proceedings of ISP RAS*. 2019;31:237–252.

19. Sukhinov AI, Belova YuV, Chistyakov AE. Solution of the matter transport problem at high Peclet numbers. *Numerical methods and programming*. 2017;18(4):371–380.

20. Sukhinov AI, Chistyakov AE, Protsenko EA, et al. Accounting method of filling cells for the hydrodynamics problems solution with complex geometry of the computational domain. *Mathematical Models and Computer Simulations*. 2019;31(8):79–100. https://www.doi.org/10.1134/S0234087919080057

21. Sukhinov AI, Chistyakov AE, Protsenko EA. Upwind and Standard Leapfrog Difference Schemes. *Numerical methods and programming*. 2019;20(2):170–181. https://www.doi.org/0.26089/NumMet.v20r216; Sukhinov AI, Chistyakov AE, Kuznetsova IY, et al. Modelling of suspended particles motion in channel. *Journal of Physics: Conference Series*. 2020;1479(1). https://www.doi.org/10.1088/1742-6596/1479/1/012082

22. Sukhinov AI, Chistyakov AE. Adaptive analog-SSOR iterative method for solving grid equations with nonselfadjoint operators. *Mathematical Models and Computer Simulations*. 2012;4(4):398–409.

About the Authors:

Vladimir N Litvinov, Associate Professor of the Mathematics and Computer Science Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), PhD. (Tech.), ScopusID, ORCID, LitvinovVN@rambler.ru

Asya M Atayan, Assistant of the Computer Engineering and Automated Systems Software Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), ScopusID, ORCID, atayan24@mail.ru

Natalia N Gracheva, PhD. (Tech.), Associate Professor of the Mathematics and Bioinformatics Department, Azov-Black Sea Engineering Institute of the Donskoy GAU (21, Lenin St., Zernograd, 347740, RF), ScopusID, ORCID, grann72@mail.ru

Nelly B Rudenko, Associate Professor of the Mathematics and Bioinformatics Department, Azov-Black Sea Engineering Institute of the Donskoy GAU (21, Lenin St., Zernograd, 347740, RF), PhD. (Tech.), ScopusID, ORCID, nelli-rud@yandex.ru

Natalia Yu Bogdanova, Lecturer of the Mathematics and Computer Science Department, Don State Technical University (1, Gagarin Sq., Rostov-on-Don, 344003, RF), nat_bogdanova07@mail.ru

Claimed contributorship:

VN Litvinov: development of mathematical models and algorithms. AM Atayan: conducting numerical experiments. NN Gracheva: statistical processing of experimental data. NB Rudenko: software implementation, preparation of illustrations. NYu Bogdanova: conducting numerical experiments.

Received 04.04.2023.
Revised 16.05.2023.
Accepted 17.05.2023.

Conflict of interest statement

The authors do not have any conflict of interest.

All authors have read and approved the final manuscript.

Об авторах:

Литвинов Владимир Николаевич, доцент кафедры математики и информатики, Донской государственный технический университет (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), кандидат технических наук, ScopusID, ORCID, LitvinovVN@rambler.ru

Атаян Ася Михайловна, ассистент кафедры программного обеспечения вычислительной техники и автоматизированных систем, Донской государственный технический университет (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), ScopusID, ORCID, atayan24@mail.ru

Грачева Наталья Николаевна, доцент кафедры математики и биоинформатики, Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ (347740, РФ, г. Зерноград, ул. Ленина, 21), кандидат технических наук, ScopusID, ORCID, grann72@mail.ru

Руденко Нелли Борисовна, доцент кафедры математики и биоинформатики, Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ (347740, РФ, г. Зерноград, ул. Ленина, 21), кандидат технических наук, ScopusID, ORCID, nelli-rud@yandex.ru

Богданова Наталья Юрьевна, старший преподаватель кафедры математики и информатики, Донской государственный технический университет (344003, РФ, г. Ростов-на-Дону, пл. Гагарина, 1), nat_bogdanova07@mail.ru

Заявленный вклад соавторов:

Литвинов В.Н. — разработка математических моделей и алгоритмов. Атаян А.М. — проведение численных экспериментов. Грачева Н.Н. — статистическая обработка экспериментальных данных. Руденко Н.Б. — программная реализация, подготовка иллюстраций. Богданова Н.Ю. — проведение численных экспериментов.

Поступила в редакцию 04.04.2023.
Поступила после рецензирования 16.05.2023.
Принята к публикации 17.05.2023.

Конфликт интересов

Авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.