

ВЫЧИСЛИТЕЛЬНАЯ МАТЕМАТИКА COMPUTATIONAL MATHEMATICS



УДК 519.688

Научная статья

<https://doi.org/10.23947/2587-8999-2023-7-2-19-30>


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

В.Н. Литвинов^{1,2}  , А.М. Атаян¹ , Н.Н. Грачева^{1,2} , Н.Б. Руденко^{1,2} , Н.Ю. Богданова¹¹Донской государственный технический университет, Российская Федерация, г. Ростов-на-Дону, пл. Гагарина, 1²Азово-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>

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 Atayan¹ , Natalya N Gracheva^{1,2} , Nelly B Rudenko^{1,2} , Natalya Yu Bogdanova¹

¹Don State Technical University, 1, Gagarina Sq., Rostov-on-Don, Russian Federation

²Azov-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>

Введение. Для предсказания состояния мелководных водоемов в экстренных ситуациях, вызванных деятельностью человека или природными и климатическими катаклизмами, используется математическое моделирование. При этом необходимо учитывать такие особенности каждого конкретного водного объекта, как геометрия водоема и его прибрежной зоны, климатические условия и гидродинамические режимы. Такие задачи актуализируют совершенствование методов решения систем сеточных уравнений высокой размерности в случае несамосопряженного оператора. Из-за большого объема данных и сложности вычислений необходимо использовать многопроцессорные вычислительные системы и видеоадаптеры для увеличения скорости получения решения.

Моделирование многих гидрофизических и гидробиологических задач сводится к необходимости решения уравнения диффузии-конвекции-реакции с несамосопряжённым оператором. Обзор актуальных численных

методов решения выполнен в работе П. Вабищевича [1], где сформулирован ряд теорем, позволяющих определить численные параметры и границы применимости исследуемых методов решения сеточных уравнений. Активно развиваются итерационные методы решения таких задач. В работе Geiser, Hueso, Martinez [2] проанализированы различные типы методов расщепления, предложены модификации методов SLIS и SQIS, на основе которых построены эффективные адаптивные алгоритмы, позволяющие увеличить шаг по времени без снижения точности вычислений.

За последние несколько лет значительно увеличилось количество исследований, направленных на разработку эффективных по скорости вычисления алгоритмов, предназначенных для решения систем сеточных уравнений высокой размерности. Российские и зарубежные учёные занимаются разработкой параллельных алгоритмов для гетерогенных вычислительных сред, изучают производительность кластерных вычислительных систем для различных методов дискретизации различных дифференциальных уравнений. Так, например, в работах [3, 4] Subbaian G. и Reddy Sathi произведён анализ производительности нескольких итерационных методов решения уравнения Навье-Стокса с ускорением вычислений на графическом процессоре (GPU) с помощью технологии CUDA. Учёные Lakshmiranganatha S., Muknahallipatna S., Paliwal M., Chilla R., Prasanth N., Goundar S. и Raja S.P. сравнили производительность различных параллельных алгоритмов для нахождения решения обыкновенных дифференциальных уравнений, зависящих от времени, на CPU и графическом процессоре (GPU) с использованием трёх технологий распараллеливания: OpenMP, OpenACC и CUDA. В результате исследования установлено, что технология CUDA является наиболее эффективным ускорителем для решения указанных уравнений [5, 6]. Российские и казахские учёные разработали параллельные алгоритмы для нахождения решений систем линейных алгебраических уравнений. Алгоритмы были реализованы на многоядерных процессорах с использованием технологии OpenMP [7, 8]. Проверка эффективности параллельных алгоритмов для решения задачи одномерной теплопроводности для трёх методов конечно-разностной аппроксимации выполнялась на центральном и графическом процессорах на языках программирования C(CPU) и CUDA C (GPU). Ускорение вычислений на GPU увеличивалось до 60 раз [9, 10]. В работе [11] рассматривается построение параллельных алгоритмов, основанных на функциональной декомпозиции метода встречных прогонок, для решения сеточных уравнений трехдиагонального вида. Д. Б. Волков-Богородский, Г. Б. Сушко и С. А. Харченко в работе [12] описывают гибридные параллельные алгоритмы аппроксимации решений нестационарного уравнения теплопроводности с фазовыми переходами на основе аналитического метода блоков, а именно технологию MPI+threads.

В данном исследовании необходимо разработать параллельный вариант алгоритма, который сократит время решения СЛАУ за счёт использования алгоритма обработки наложения сегментов геометрии и распараллеливания процесса вычислений.

Материалы и методы

1. Постановка задачи. Математическая модель гидродинамики мелководных водоемов включает [13]:

– уравнения движения Навье-Стокса:

$$u'_t + uu'_x + vv'_y + ww'_z = -\frac{1}{\rho} P'_x + (\mu u'_x)'_x + (\mu v'_y)'_y + (\mu w'_z)'_z + 2\Omega(v \sin \vartheta - w \cos \vartheta), \quad (1)$$

$$v'_t + uv'_x + vv'_y + ww'_z = -\frac{1}{\rho} P'_y + (\mu v'_x)'_x + (\mu v'_y)'_y + (\mu w'_z)'_z + 2\Omega u \sin \vartheta, \quad (2)$$

$$w'_t + uw'_x + vv'_y + ww'_z = -\frac{1}{\rho} P'_z + (\mu w'_x)'_x + (\mu w'_y)'_y + (\mu w'_z)'_z + 2\Omega u \cos \vartheta + g; \quad (3)$$

– уравнение неразрывности в случае переменной плотности:

$$\rho'_t + (\rho u)'_x + (\rho v)'_y + (\rho w)'_z = 0, \quad (4)$$

где $V = \{u, v, w\}$ — компоненты вектора скорости; P — полное гидродинамическое давление; ρ — плотность водной среды; μ, ν — горизонтальная и вертикальная составляющие коэффициента турбулентного обмена; $\Omega = \Omega \cdot (\cos \vartheta \cdot j + \sin \vartheta \cdot k)$ — угловая скорость вращения Земли определяется выражением; ϑ — широта места; g — ускорение свободного падения; f_T, f_s — источники тепла и соли (находится на границе области).

Исходная модель гидродинамики (1–4) разбивается на несколько подзадач [14, 15]. Первая подзадача представлена уравнением диффузии-конвекции-реакции, при помощи которого вычисляются компоненты поля вектора скорости на промежуточном слое по времени:

$$\begin{aligned} \frac{\tilde{u} - u}{\tau} + u\tilde{u}'_x + v\tilde{u}'_y + w\tilde{u}'_z &= (\mu\tilde{u}'_x)'_x + (\mu\tilde{u}'_y)'_y + (\mu\tilde{u}'_z)'_z + 2\Omega(v \sin \theta - w \cos \theta), \\ \frac{\tilde{v} - v}{\tau} + u\tilde{v}'_x + v\tilde{v}'_y + w\tilde{v}'_z &= (\mu\tilde{v}'_x)'_x + (\mu\tilde{v}'_y)'_y + (\mu\tilde{v}'_z)'_z - 2\Omega u \sin \theta, \\ \frac{\tilde{w} - w}{\tau} + u\tilde{w}'_x + v\tilde{w}'_y + w\tilde{w}'_z &= (\mu\tilde{w}'_x)'_x + (\mu\tilde{w}'_y)'_y + (\mu\tilde{w}'_z)'_z + 2\Omega u \cos \theta + g \left(\frac{\rho_0}{\rho} - 1 \right). \end{aligned} \quad (5)$$

Для аппроксимации уравнения диффузии-конвекции-реакции (5) по времени использована схема Кранка-Николсона. Здесь $u = \sigma \tilde{u} + (1 - \sigma)u$, $\sigma \in [0, 1]$ — вес схемы.

2. Методы решения сеточных уравнений. Модифицированный попеременно-треугольный итерационный метод. Введём трёхмерную равномерную расчетную сетку [14]:

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

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

где τ — временной шаг; h_x, h_y, h_z — размер шагов по координатным направлениям; n_t — количество временных слоев; T — верхняя граница по временной координате; n_1, n_2, n_3 — количество узлов по пространственным координатам; l_x, l_y, l_z — пространственные размеры расчетной области.

При построении дискретной модели получим систему сеточных уравнений. Каждое уравнение системы может быть представлено в канонической форме, при этом будем использовать семиточечный шаблон:

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

где $m_0(x_i, y_j, z_k)$ — центр шаблона; $M'(P) = \{m_1(x_{i+1}, y_j, z_k), m_2(x_{i-1}, y_j, z_k), m_3(x_i, y_{j+1}, z_k), m_4(x_i, y_{j-1}, z_k), m_5(x_i, y_j, z_{k+1}), m_6(x_i, y_j, z_{k-1})\}$ — окрестность центра; $c_0 \equiv c(m_0)$ — коэффициент центра шаблона; $c_i \equiv c(m_0, m_i)$ — коэффициенты окрестности центра шаблона; F — вектор правых частей; u — рассчитываемый вектор.

Алгоритм МПТМ состоит из четырёх этапов:

- 1) расчет вектора невязки r^m ;
- 2) расчет вектора поправки w^m ;
- 3) расчет скалярных произведений на основе итерационных параметров τ_{m+1}, ω_{m+1} ;
- 4) переход на новый итерационный слой.

Условием окончания итерационного процесса является достижение нормой вектора невязки r^m заданной точности. При этом наиболее трудоемкая часть алгоритма — расчет w^m , который сводится к решению СЛАУ с нижне-треугольной и верхне-треугольной матрицами.

3. Метод решения сеточных уравнений с трехдиагональным предобуславливателем. В случае если шаги по одной из пространственных координат существенно меньше размеров шагов по другим (например, при решении задач тепломассопереноса в мелководных водоемах), размеры расчетной области по вертикальному направлению могут быть на несколько порядков меньше горизонтальных размеров. Для решения задачи (1) на основе разностных схем с относительно небольшими трудозатратами на переход между временными слоями, по сравнению с явной схемой (в 1,5–2 раза большими), с большими шагами по времени (около 30 раз больше) воспользуемся схемами расщепления на двумерную и одномерную задачи [16–17]:

$$\frac{c^{n+1/2} - c^n}{\tau} + u(c^n)'_x + v(c^n)'_y = \left(\mu(c^n)'_x \right)'_x + \left(\mu(c^n)'_y \right)'_y, \quad (6)$$

$$\frac{c^{n+1} - c^{n+1/2}}{\tau} + w(c^{n+(\sigma+1)/2})'_z = \left(v(c^{n+(\sigma+1)/2})'_z \right)'_z + f^{n+(\sigma+1)/2}, \quad (7)$$

здесь $c^{n+(\sigma+1)/2} = \sigma c^{n+1} + (1 - \sigma)c^{n+1/2}$; σ — вес схемы [7].

Для численной реализации дискретной математической модели поставленной задачи вводится пространственная сетка [18]:

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

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

Для аппроксимации однородного уравнения (2) будем использовать схемы расщепления по пространственным координатным направлениям:

$$\begin{aligned} \frac{c^{n+1/4} - c^n}{\tau} + u(c^n)'_x &= \left(\mu(c^n)'_x \right)'_x, \\ \frac{c^{n+1/2} - c^{n+1/4}}{\tau} + v(c^{n+1/4})'_y &= \left(\mu(c^{n+1/4})'_y \right)'_y. \end{aligned} \quad (8)$$

Для решения реальных задач гидрофизики мелководных водоемов применены трехслойные разностные схемы, построенные на основе линейной комбинации разностных схем «кабаре» и «крест» с весовыми коэффициен-

тами 2/3 и 1/3 соответственно. Для увеличения точности расчетов используется схема, учитывающая заполненность расчетных ячеек [19–21]:

– разностная схема для уравнения, описывающего перенос вдоль направления O_x :

$$\begin{aligned} & \frac{2q_{2,i,j} + q_{0,i,j}}{3} \frac{c_{i,j}^{n+1/4} - c_{i,j}^n}{\tau} + 5u_{i-1/2,j} q_{2,i,j} \frac{c_{i,j}^n - c_{i-1,j}^n}{3h_x} + u_{i+1/2,j} \min(q_{1,i,j}, q_{2,i,j}) \frac{c_{i+1,j}^n - c_{i,j}^n}{3h_x} + \\ & + \frac{2\Delta_x c_{i-1,j}^n q_{2,i,j} + \Delta_x c_{i,j}^n q_{0,i,j}}{3} = 2\mu_{i+1/2,j} q_{1,i,j} \frac{c_{i+1,j}^n - c_{i,j}^n}{h_x^2} - 2\mu_{i-1/2,j} q_{2,i,j} \frac{c_{i,j}^n - c_{i-1,j}^n}{h_x^2} - \\ & - |q_{1,i,j} - q_{2,i,j}| \mu_{i,j} \frac{\alpha_x c_{i,j}^n + \beta_x}{h_x}, \quad u_{i,j} \geq 0, \end{aligned}$$

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

– разностная схема для уравнения (4), описывающего перенос вдоль направления O_y :

$$\begin{aligned} & \frac{2q_{4,i,j} + q_{0,i,j}}{3} \frac{c_{i,j}^{n+1/2} - c_{i,j}^{n+1/4}}{\tau} + 5v_{i,j-1/2} q_{4,i,j} \frac{c_{i,j}^{n+1/4} - c_{i,j-1}^{n+1/4}}{3h_y} + v_{i,j+1/2} \min(q_{3,i,j}, q_{4,i,j}) \frac{c_{i,j+1}^{n+1/4} - c_{i,j}^{n+1/4}}{3h_y} + \\ & + \frac{2\Delta_y c_{i,j-1}^{n+1/4} q_{4,i,j} + \Delta_y c_{i,j}^{n+1/4} q_{0,i,j}}{3} = 2\mu_{i,j+1/2} q_{3,i,j} \frac{c_{i,j+1}^{n+1/4} - c_{i,j}^{n+1/4}}{h_y^2} - 2\mu_{i,j-1/2} q_{4,i,j} \frac{c_{i,j}^{n+1/4} - c_{i,j-1}^{n+1/4}}{h_y^2} - \\ & - |q_{3,i,j} - q_{4,i,j}| \mu_{i,j} \frac{\alpha_y c_{i,j}^{n+1/4} + \beta_y}{h_y}, \quad v_{i,j} \geq 0, \end{aligned}$$

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

Здесь q_0, q_1, q_2, q_3, q_4 — степени заполненности контрольных областей.

Для получения разностных схем, аппроксимирующих систему уравнений (4), при $u_{i,j} < 0$ и $v_{i,j} < 0$ из представленных аппроксимаций, необходимо направить соответствующие координатные оси O_x и O_y в противоположные стороны. Уравнение (3) решается методом прогонки.

Для решения нестационарных задач преимуществом обладает схема расщепления на двумерную и одномерную задачи. При этом двумерная задача решается на основе явных схем, а одномерная аппроксимируется схемами с весами и решается методом прогонки. При решении стационарных задач используются схемы с весами. Использование данного подхода позволяет свести исходную задачу к решению сеточных уравнений итерационными методами [22].

4. Алгоритм наложения сегментов геометрии. При составлении СЛАУ необходимо учитывать сложную геометрию водоёма, формируемую комбинацией донной поверхности и прибрежных инженерных сооружений. Программно предложен и реализован подход, позволяющий моделировать геометрию исследуемого объекта как множество геометрических примитивов. Особенностью подхода является поддержка наложения примитивов друг на друга. В библиотеке классов, разработанной при программной реализации, все геометрические примитивы наследуются от абстрактного класса Geometry2DPrimitive (рис. 1), который содержит такие данные, как координаты смещения _dS0, тип примитива _primitiveType и логическое свойство, характеризующее «вырез» (_isCavity).

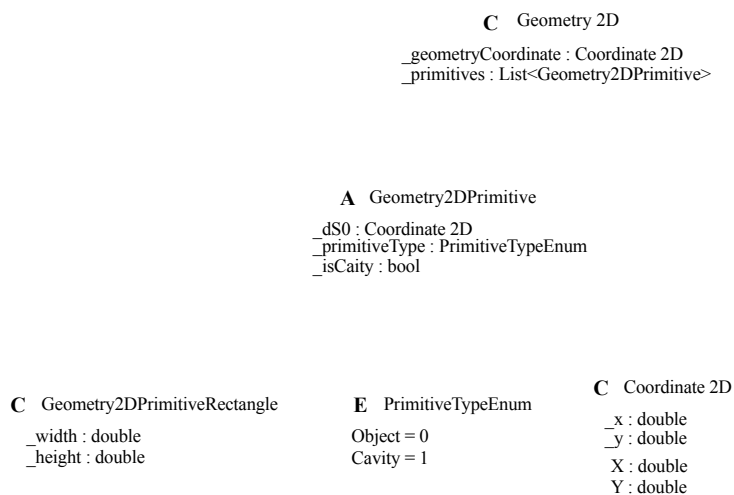


Рис. 1. Геометрия исследуемого объекта. Диаграмма классов

Объектно-ориентированное моделирование сегмента геометрии выполняется при помощи типизированного класса `GeometryPrimitiveSegment<T>`, в котором тип T является классом, используемым для хранения данных о координатах начала и конца сегмента.

Обозначим координаты начала и конца соответственно первого и второго сегментов $c_{11}, c_{12}, c_{21}, c_{22}$.

Введём логические переменные: $A = c_{11} < c_{21}$, $B = c_{11} = c_{21}$, $C = c_{11} > c_{21}$, $D = c_{12} < c_{22}$, $E = c_{12} = c_{22}$, $F = c_{12} > c_{22}$, $G = c_{11} = c_{22}$, $H = c_{12} = c_{21}$, $I = c_{12} < c_{21}$, $J = c_{11} > c_{22}$. K, L — типы первого и второго сегментов; M, N — заполненность пространства под первым и вторым сегментами; V — флаг, означающий, что второй сегмент является «вырезом».

Все возможные комбинации наложения сегментов геометрии сведены в таблицу 1.

Учет различных комбинаций геометрических примитивов является основой исходного линейного алгоритма.

Для повышения производительности введен ряд модификаций на основе условных конструкций:

1. Инициализация: $c_{11}, c_{12}, c_{21}, c_{22}$.
2. Вычисление: $A, B, C, D, E, F, G, H, I, J, K, L, M, N, V$.
3. Проверка условия корректности: $L \wedge \bar{N} \vee K \wedge \bar{M} \neq \text{true}$.
4. Определение типа наложения.
5. Дальнейшие действия выполняются для найденного типа наложения. Для примера описан вариант № 1 (таблица 1). Для других типов действия те же.
6. Если первый сегмент не является граничным ($K = \text{true}$), то переход к шагу 12.
7. Если второй сегмент не является граничным ($L = \text{true}$), то переход к шагу 10.
8. Вычисление выражений $\bar{M} \wedge \bar{N} \wedge V$, $\bar{M} \wedge \bar{N} \wedge \bar{V}$, $\bar{M} \wedge N \wedge V$, $\bar{M} \wedge N \wedge \bar{V}$, $M \wedge \bar{N} \wedge V$, $M \wedge \bar{N} \wedge \bar{V}$, $M \wedge N \wedge V$, $M \wedge N \wedge \bar{V}$ и создание результирующих сегментов.
9. Переход к шагу 17.
10. Вычисление $\bar{M} \wedge N \wedge V$, $M \wedge N \wedge \bar{V}$, $M \wedge N \wedge V$, $\bar{M} \wedge N \wedge \bar{V}$ и создание результирующих сегментов.
11. Переход к шагу 17.
12. Если второй сегмент граничный ($L = \text{false}$), то переход к шагу 15, иначе переход к шагу 17.
13. Проверка условий $M \wedge N \wedge V$, $M \wedge \bar{N} \wedge V$, $M \wedge \bar{N} \wedge \bar{V}$, $M \wedge N \wedge \bar{V}$.
14. Создание и возврат результирующих сегментов.
15. Переход к шагу 17.
16. Вычисление $M \wedge N \wedge V$, $M \wedge N \wedge \bar{V}$ и создание результирующих сегментов.
17. Конец.

Таблица 1

Варианты наложения сегментов геометрии

Вариант наложения	Графическая интерпретация	Логическое выражение
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		$C \wedge F$

Окончание таблицы 1

Вариант наложения	Графическая интерпретация	Логическое выражение
10		$C \wedge F$
11		$A \wedge D \wedge \bar{I}$
12		I
13		J

5. Параллельная реализация. Для численной реализации МПТМ, применимого к СЛАУ высокой размерности, разработан конвейерный параллельный алгоритм, позволяющий задействовать все доступные вычислительные ресурсы. При этом каждым вычислителем (ядром CPU или вычислительным блоком GPU) обрабатываются только закреплённые за ним фрагменты расчетной области.

Связи между фрагментами и организация процесса параллельно-конвейерных вычислений описаны графовой моделью (рис. 2), где узлы представляют собой фрагменты расчетной области. Вычислительный процесс организовывается в соответствии со значениями счетчика этапов вычислений $s = k \cdot i + j$.

Разработанная графовая модель используется в алгоритме решения СЛАУ с нижнетреугольной матрицей (рис. 3). Входными параметрами алгоритма являются коэффициенты сеточных уравнений c_0, c_2, c_4, c_6 и константа ω . Результатом — вектор скорости водного потока r . При запуске программной реализации алгоритма на языке CUDA C необходимо задать значения размерностей вычислительных блоков `blockDim.x`, `blockDim.z`. Параллельно-конвейерный процесс вычислений организован в виде цикла (строка 6).

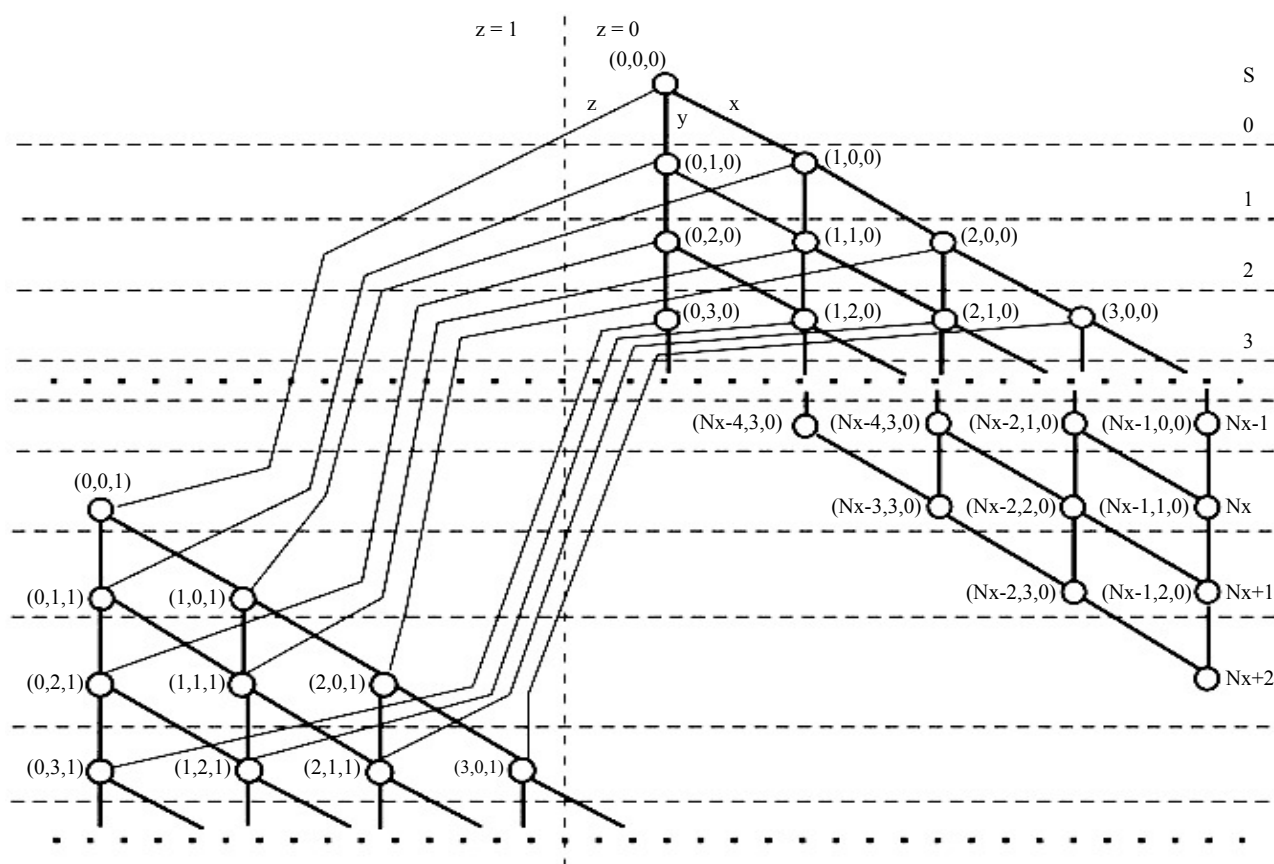


Рис. 2. Графовая модель параллельно-конвейерного процесса вычислений

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

```

1:  $threadX \leftarrow blockDim.x \cdot blockIdx.x + threadIdx.x$ ;
2:  $threadZ \leftarrow blockDim.z \cdot blockIdx.z + threadIdx.z$ ;
3:  $i \leftarrow threadX + 1$ ;
4:  $j \leftarrow 1$ ;
5:  $k \leftarrow threadZ + 1$ ;
6: for  $s \in [3; n_1 + n_2 + n_3 - 3]$  do
7:   if  $(i + j + k = s) \wedge (s < i + n_2 + k)$  then
8:      $m_0 \leftarrow i + (blockDim.x + 1) \cdot j + n_1 \cdot n_2 \cdot k$ ;
9:     if  $c0[m_0] > 0$  then
10:       $m_2 \leftarrow m_0 - 1$ ;
11:       $m_4 \leftarrow m_0 - n_1$ ;
12:       $m_6 \leftarrow m_0 - n_1 \cdot n_2$ ;
13:       $rm4 \leftarrow 0$ ;
14:      if  $(s > 3 + threadX + threadZ)$  then
15:         $rm4 \leftarrow cache[threadX][threadZ]$ ;
16:      else
17:         $rm4 \leftarrow r[m_4]$ ;
18:       $rm2 \leftarrow 0$ ;
19:      if  $(threadX \neq 0) \wedge (s > 3 + threadX + threadZ)$  then
20:         $rm2 \leftarrow cache[threadX - 1][threadZ]$ ;
21:      else
22:         $rm2 \leftarrow r[m_2]$ ;
23:       $rm6 \leftarrow 0$ ;
24:      if  $(threadZ \neq 0) \wedge (s > 3 + threadX + threadZ)$  then
25:         $rm6 \leftarrow cache[threadX][threadZ - 1]$ ;
26:      else
27:         $rm6 \leftarrow r[m_6]$ ;
28:       $rm0 \leftarrow (\omega \cdot (c2[m_0] \cdot rm2 + c4[m_0] \cdot rm4 + c6[m_0] \cdot rm6) + r[m_0]) / ((0.5 \cdot \omega + 1) \cdot c0[m_0])$ ;
29:       $cache[threadX][threadZ] \leftarrow rm0$ ;
30:       $r[m_0] \leftarrow rm0$ ;
31:       $j \leftarrow j + 1$ ;

```

Рис. 3. Алгоритм решения системы уравнений с нижнетреугольной матрицей

Для уменьшения количества считываний из глобальной видеопамати введён двумерный массив *cache*, размещённый в разделяемой (shared) памяти GPU. В нём сохраняются промежуточные результаты вычислений на текущем слое по оси O_y , что ускоряет процесс вычислений на 30 %.

Результаты исследования. Вычислительный эксперимент по сравнению производительности базового и модифицированного алгоритмов проводился на вычислительной системе с процессором Intel Core i5 3.3 ГГц и оперативной памятью DDR4 32 Гб (таблица 2). В модифицированном алгоритме зафиксировано снижение времени расчетов на величину до 27 %.

Таблица 2

Результаты сравнения производительности базового и модифицированного алгоритмов объединения сегментов геометрии

Количество объединений, $\times 10^6$	1	2	3	4	5	6	7	8	9	10
Базовый алгоритм, с	0,53	0,75	0,13	0,16	0,19	0,23	0,26	0,29	0,35	0,38
Модифицированный алгоритм, с	0,41	0,55	0,97	0,12	0,15	0,20	0,22	0,25	0,29	0,31

Проведен численный эксперимент по определению количества потоков GPU по осям O_x и O_z расчетной сетки (X, Z соответственно) при фиксированном значении узлов сетки по оси O_y , равным 10000, позволяющему уменьшить время расчёта одного шага МПТМ (T_{GPU}) на GPU. Уровни варьирования факторов X и Z и результаты численного эксперимента показаны в таблице 3.

В ходе проведённого эксперимента установлено, что время расчёта одного шага МПТМ на GPU обратно пропорционально количеству узлов расчетной сетки по оси O_z . Наименьшее значение целевой функции получено при X и Z , равных 16 и 64 соответственно.

Таблица 3

Результаты эксперимента

№ п/п	X	Z	$T_{\text{GPU}}, \text{с}$
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

Таблица 4

Сравнение ускорения параллельных алгоритмов

P	Якоби		Зейдель		МПТМ		МРСУ с трехдиагональным предобуславливателем	
	Ускорение	Эффектив- ность	Ускорение	Эффектив- ность	Ускорение	Эффектив- ность	Ускорение	Эффектив- ность
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

В таблице 4 представлено сравнение ускорения параллельных алгоритмов для методов Зейделя, Якоби, модифицированного попеременно-треугольного метода и метода решения сеточных уравнений с трехдиагональным предобуславливателем от количества вычислительных узлов. Расчеты производились на сетке, насчитывающей один миллион расчетных ячеек. Запуски проводились последовательно, начиная от запуска на одном вычислительном узле и заканчивая подключением всех доступных узлов.

Обсуждение и заключения. Предложены алгоритмы решения СЛАУ, получаемых при дискретизации задачи гидродинамики мелководного водоема, МПТМ с использованием технологии NVIDIA CUDA. Предложенные метод декомпозиции расчетной сетки и графовая модель позволяют эффективно организовать параллельно-конвейерные вычисления на вычислительных системах различной конфигурации.

Проведены численные эксперименты по определению наилучшей двумерной конфигурации потоков в вычислительном блоке, минимизирующей время одного шага МПТМ на GPU, $-X = 16$ и $Z = 64$.

Наибольшее ускорение показал метод решения сеточных уравнений для задач гидродинамики в плоских областях, который основывается на явно-неявной схеме. МПТМ, по сравнению с методами Якоби и Зейделя, для сходимости требует существенно меньшего числа итераций. При хорошей оптимизации параллельного алгоритма МПТМ ускорение отличается не более чем на 10 % на числе вычислительных узлов до 24 по сравнению с ускорением параллельного алгоритма метода Якоби.

Разработанный программный инструментариум позволяет более эффективно задействовать вычислительные ресурсы GPU, используемой для решения вычислительно-трудоемких пространственно-трехмерных задач гидрофизики.

Объединение сегментов геометрии исследуемого объекта приводит к уменьшению количества вычислительных операций, что позволяет увеличить производительность вычислений.

Список литературы

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., Urmashiev B., 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., Goundar S., Raja P. 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. Акимова Е.Н., Султанов М.А., Мисилов В.Е. и др. Parallel sweep algorithm for solving direct and inverse problems for time-fractional diffusion equation. *Numerical Methods and Programming (Vychislitel'nye Metody i Programirovanie)*. 2022;23(4):275–287. <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. Головашкин Д.Л. Параллельные алгоритмы решения сеточных уравнений трехдиагонального вида, основанные на методе встречных прогонок. *Математическое моделирование*. 2005;17(11):118–128.
12. Волков-Богородский Д.Б., Сушко Г.Б., Харченко С.А. Комбинированная MPI+threads параллельная реализация метода блоков для моделирования тепловых процессов в структурно-неоднородных средах. *Вычислительные методы и программирование*. 2010;11(1):127–136.
13. Munk D.J., Kipourous T., Vio G.A. 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 A.I., Chistyakov A.E., Shishenya A.V., 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. Коновалов А.Н. Метод скорейшего спуска с адаптивным попеременно-треугольным переобусловливателем. *Дифференциальные уравнения*. 2004;40(7):953–963.
16. Самарский А.А., Вабищевич П.Н. Численные методы решения задач конвекции-диффузии, Изд. стереотип. Москва: Книжный дом «ЛИБРОКОМ»; 2015. 248 с.
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 N. I., Petrov I. B. 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 A.I., Belova Yu.V., Chistyakov A.E. Solution of the matter transport problem at high Peclet numbers. *Numerical methods and programming*. 2017;18(4):371–380.
20. Sukhinov A.I., Chistyakov A.E., Protsenko E.A., 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 A.I., Chistyakov A.E., Protsenko E.A. Upwind and Standard Leapfrog Difference Schemes. *Numerical methods and programming*. 2019;20(2):170–181. <https://www.doi.org/10.26089/NumMet.v20r216>; Sukhinov A.I., Chistyakov A.E., Kuznetsova I.Y., 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 A.I., Chistyakov A.E. Adaptive analog-SSOR iterative method for solving grid equations with non-selfadjoint operators. *Mathematical Models and Computer Simulations*. 2012;4(4):398–409.

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 Programirovanie)*. 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, Kipourous 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/10.26089/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.

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.

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.