

МЕЖДУНАРОДНЫЙ ЖУРНАЛ  
ТЕОРЕТИЧЕСКИХ И ПРИКЛАДНЫХ  
ВОПРОСОВ ЦИФРОВЫХ ТЕХНОЛОГИЙ

P-ISSN: 2181-3086

E-ISSN: 2181-3094

Самаркандский филиал Ташкентского университета информационных технологий имени Мухаммада ал-Хоразмий



WEB: <https://ijdt.uz/index.php/ijdt>

**ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РАСЧЕТА ПОЛЯ  
ПРИЗЕМНОЙ КОНЦЕНТРАЦИИ ЗАГРЯЗНЯЮЩЕЙ  
ПРИМЕСИ В АТМОСФЕРЕ**

*\*Дилшод Ахмедов<sup>1</sup>, Бахтиёр Боборахимов<sup>1</sup>, Малик Убайдумаев<sup>1</sup>,  
Парвиз Насрумаев<sup>1</sup>*

<sup>1</sup> Научно-исследовательский институт развития цифровых технологий и  
искусственного интеллекта, Ташкент, Узбекистан

\* [a.dilshot@mail.ru](mailto:a.dilshot@mail.ru)

**Цитирование:** Ахмедов Д.Д., Боборахимов Б.И., Убайдумаев М.Ш., Насрумаев П.А. (2024). Параллельный алгоритм расчета поля приземной концентрации загрязняющей примеси в атмосфере. Международный Журнал Теоретических и Прикладных Вопросов Цифровых Технологий, 7(3), –С. 19-32. <https://doi.org/10.62132/ijdt.v7i3.193>

Дата поступления: 06.05.2024

Дата принятия: 10.06.2024

Дата печати: 30.09. 2024

DOI: <https://doi.org/10.62132/ijdt.v7i3.193>

УДК 519.6+51-74::628.395

## ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РАСЧЕТА ПОЛЯ ПРИЗЕМНОЙ КОНЦЕНТРАЦИИ ЗАГРЯЗНЯЮЩЕЙ ПРИМЕСИ В АТМОСФЕРЕ

\*Ахмедов Д.Д.<sup>1</sup>, Боборахимов Б.И.<sup>1</sup>, Убайдуллаев М.Ш.<sup>1</sup>, Насруллаев П.А.<sup>1</sup>

<sup>1</sup> Научно-исследовательский институт развития цифровых технологий и искусственного интеллекта, Ташкент, Узбекистан  
\*a.dilshot@mail.ru

**Аннотация.** Моделирование распространения загрязняющих примесей в атмосфере является одной из важнейших задач в экологических исследованиях и охране окружающей среды. Для исследования данного процесса широко используется уравнение адвекции-диффузии. Точность численного решения задач, сформулированных на основе уравнения адвекции-диффузии, во многом определяется способом аппроксимации членов, описывающих учитываемые эффекты, включая перенос, турбулентную диффузию, поглощение примеси, свойства частиц примеси, рельеф местности и т.д. В данной работе внимание сосредоточено на ряде вопросов, касающихся построения математической модели исследуемого процесса, разработки численного алгоритма решения задачи на основе разностной схемы Кранк-Николсона, повышения эффективности алгоритма за счет параллелизации вычислительных процедур, а также анализа точности и сходимости используемого численного метода, применительно к решению задачи распространения примеси в атмосфере.

**Ключевые слова:** атмосфера, загрязняющая примесь, турбулентность, параллельный алгоритм, схема Кранка-Николсона, итерационный метод.

### I. ВВЕДЕНИЕ

Уравнение адвекции-диффузии – одна из разновидностей дифференциальных уравнений в частных производных параболического типа [1]. Непроходящий повышенный интерес многочисленных исследователей к решению задач на основе уравнения адвекции-диффузии связан с его широким прикладным применением в различных предметных областях.

Качество решения задач вычислительной гидродинамики, сформулированных на основе уравнения адвекции-диффузии, к которым относятся задачи оценки и прогнозирования распространения загрязняющих примесей в атмосфере, в значительной мере определяется тем, каким образом аппроксимируются члены, описывающие адвективный перенос и турбулентную диффузию [2, 3].

Отметим, что за последние годы разработано немало методов численного решения задач, основанных на уравнении адвекции-диффузии, разной степени эффективности. Тем не менее, построение математических моделей и поиск новых методов решения подобных задач остается актуальным научным направлением [4], как уже было сказано – в силу их широкого применения в исследовании различных динамических процессов в геофизике и гидрологии, биологии, финансовой математике, экологии, механике сплошных сред и т.д.

Среди многообразия различных методов численного решения дифференциальных уравнений, выделяется метод конечных разностей [5, 6] – простой и мощный инструмент, широко применяемой в научных и инженерных областях при моделировании динамических процессов, развивающихся в пространстве и по времени.

Математическая модель процесса распространения примеси в атмосфере, рассматриваемая в данной работе, относится к семейству эйлеровых моделей. Модели этого типа хорошо зарекомендовали себя в прикладных расчетах оценки распространения примеси от источников любого типа и в различных масштабах. Они позволяют моделировать любые поля ветра и турбулентности, а также любые комбинации источников выбросов. Количество отслеживаемых видов загрязняющих веществ может превышать сотню наименований, причем примесь может присутствовать в газовой, жидкой и твердой фазах.

Основные недостатки эйлеровых моделей связаны с наличием нескольких ограничений, которые сегодня уже не имеют принципиального характера, но все еще актуальны [7]. К ним относится разрешение пространственно-временной дискретизации вычислительной сетки, сопряженное с нехваткой желаемого уровня детализации полей метеорологических переменных. Данное ограничение постепенно

снимается с развитием автоматических метеорологических сетей, информационных веб-сервисов и улучшением качества баз данных по выбросам, загрязняющим веществам, субъектам экологического мониторинга, цифровому описанию земной поверхности.

Другое ограничение – это ошибки аппроксимации, возникающие при использовании стандартных конечно-разностных схем. Что, собственно, и решается путем разработки более совершенных численных методов.

Еще один важный аспект – это высокие требования к вычислительной мощности в случае решения задач в многомерных постановках с граничными условиями различных типов. Однако, этот узкий момент постепенно нивелируется возрастающей производительностью современных компьютеров и появлением широкого разнообразия средств для реализации па-

раллельных и распределенных вычислительных алгоритмов, а также динамичное развитие и удешевление высокопроизводительных вычислительных систем.

В этой связи, целью настоящей работы является формулировка математической модели процесса адвективного переноса и турбулентной диффузии примеси в атмосфере, а также разработка параллельного алгоритма численного решения задачи оценки распределения концентрации примеси на основе конечно-разностной схемы Кранка-Николсона.

## II. ПОСТАНОВКА ЗАДАЧИ

Основное уравнение адвективного переноса и турбулентной диффузии загрязняющей примеси в атмосфере с учетом поглощения имеет следующий вид [8-11]:

$$\frac{\partial \theta}{\partial t} + u \frac{\partial \theta}{\partial x} + v \frac{\partial \theta}{\partial y} + (w - w_g) \frac{\partial \theta}{\partial z} = D_x \frac{\partial^2 \theta}{\partial x^2} + D_y \frac{\partial^2 \theta}{\partial y^2} + D_z \frac{\partial^2 \theta}{\partial z^2} - \sigma \theta + Q(x, y, z). \quad (1)$$

Уравнение (1) сопровождаем начальным

$$\theta|_{t=0} = \theta^0, \quad (2)$$

и граничными условиями

$$D_x \frac{\partial \theta}{\partial x} \Big|_{x=0} = 0; \quad D_x \frac{\partial \theta}{\partial x} \Big|_{x=L} = 0; \quad (3)$$

$$D_y \frac{\partial \theta}{\partial y} \Big|_{y=0} = 0; \quad D_y \frac{\partial \theta}{\partial y} \Big|_{y=M} = 0; \quad (4)$$

$$-D_z \frac{\partial \theta}{\partial z} \Big|_{z=0} = \beta \theta; \quad D_z \frac{\partial \theta}{\partial z} \Big|_{z=N} = 0. \quad (5)$$

Здесь  $\theta(x, y, z, t)$  – концентрация примеси ( $\text{г}/\text{м}^3$ );  $\theta^0$  – начальная концентрация примеси ( $\text{г}/\text{м}^3$ );  $u, v, w$  – составляющие скорости ветра ( $\text{м}/\text{с}$ );  $w_g$  – скорость осаждения примеси ( $\text{м}/\text{с}$ );

$D_x, D_y, D_z$  – коэффициенты диффузии по направлениям  $x, y, z$ , соответственно ( $\text{м}^2/\text{с}$ );  $\sigma$  – коэффициент распада (поглощения) примеси ( $\text{с}^{-1}$ );  $Q(x, y, z)$  – источник выброса загрязняющей примеси;  $\beta$  – коэффициент взаимодействия с нижней границей, регулирующий отражение и прилипание частиц примеси ( $0 \leq \beta \leq 1$ );  $L, M, N$  – размеры области по направлениям  $x, y, z$ , соответственно ( $\text{м}$ );  $t \in [0; T]$  – время ( $\text{с}$ ).

## III. МЕТОД РЕШЕНИЯ

Для численного решения поставленной задачи, прежде всего, область непрерывного изменения искомых  $(0, L) \times (0, M) \times (0, N) \times (0, T)$  переменных, покроем равномерной сеткой:

$$\Omega_{xyzt} = \left\{ (x_i = i\Delta x, y_j = j\Delta y, z_k = k\Delta z, t_n = n\Delta t); \right. \\ \left. i = \overline{0, Nx-1}; j = \overline{0, Ny-1}; k = \overline{0, Nz-1}; n = \overline{0, Nt} \right\}, \quad (6)$$

с шагами

$$\Delta x = \frac{L}{Nx-1}, \quad \Delta y = \frac{M}{Ny-1}, \\ \Delta z = \frac{N}{Nz-1}, \quad \Delta t = \frac{T}{Nt}.$$

Площадный источник выброса примеси  $Q(x, y, z)$  размером  $L_S \times M_S$  с координатами центра  $x_S$  и  $y_S$ , действующий на некоторой заданной высоте  $z = z_S$ , например  $Q(x, y, 0)$  – на поверхности земли, дискретизируется следующим образом:

$$Q_{i,j,0} = Q_S \cdot I(i, j), \quad (7) \quad \text{где } Q_S \text{ – мощность выброса (г/м}^2\cdot\text{с); } I(i, j) \text{ –}$$

индикаторная функция, определяемая как:

$$I(i, j) = \begin{cases} 1 & \text{если } x_S - \frac{L_S/\Delta x}{2} \leq i \leq x_S + \frac{L_S/\Delta x}{2}, y_S - \frac{M_S/\Delta y}{2} \leq j \leq y_S + \frac{M_S/\Delta y}{2}, k = 0; \\ 0. & \end{cases}$$

К основному уравнению (1) применяем смешанную схему Кранка-Николсона с весом 0,5 для текущего и следующего временных шагов,

позволяющую повысить точность и стабильность решения по сравнению с явными и неявными схемами [12, 13]:

$$\begin{aligned} \frac{\theta_{i,j,k}^{n+1} - \theta_{i,j,k}^n}{\Delta t} = & \frac{1}{2} \left[ -u \left( \frac{\theta_{i+1,j,k}^{n+1} - \theta_{i-1,j,k}^{n+1}}{2\Delta x} + \frac{\theta_{i+1,j,k}^n - \theta_{i-1,j,k}^n}{2\Delta x} \right) - \right. \\ & -v \left( \frac{\theta_{i,j,k+1}^{n+1} - \theta_{i,j,k-1}^{n+1}}{2\Delta y} + \frac{\theta_{i,j,k+1}^n - \theta_{i,j,k-1}^n}{2\Delta y} \right) - (w-w_g) \left( \frac{\theta_{i,j,k+1}^{n+1} - \theta_{i,j,k-1}^{n+1}}{2\Delta z} + \frac{\theta_{i,j,k+1}^n - \theta_{i,j,k-1}^n}{2\Delta z} \right) + \\ & + D_x \left( \frac{\theta_{i+1,j,k}^{n+1} - 2\theta_{i,j,k}^{n+1} + \theta_{i-1,j,k}^{n+1}}{\Delta x^2} + \frac{\theta_{i+1,j,k}^n - 2\theta_{i,j,k}^n + \theta_{i-1,j,k}^n}{\Delta x^2} \right) + \\ & + D_y \left( \frac{\theta_{i,j,k+1}^{n+1} - 2\theta_{i,j,k}^{n+1} + \theta_{i,j,k-1}^{n+1}}{\Delta y^2} + \frac{\theta_{i,j,k+1}^n - 2\theta_{i,j,k}^n + \theta_{i,j,k-1}^n}{\Delta y^2} \right) + \\ & + D_z \left( \frac{\theta_{i,j,k+1}^{n+1} - 2\theta_{i,j,k}^{n+1} + \theta_{i,j,k-1}^{n+1}}{\Delta z^2} + \frac{\theta_{i,j,k+1}^n - 2\theta_{i,j,k}^n + \theta_{i,j,k-1}^n}{\Delta z^2} \right) - \\ & \left. - \sigma \left( \frac{\theta_{i,j,k}^{n+1} - \theta_{i,j,k}^n}{2} \right) + \frac{Q_{i,j,k}^{n+1} - Q_{i,j,k}^n}{2} \right]. \end{aligned} \quad (8)$$

Уравнение (8) приведем к более удобному виду для решения на каждом временном шаге:

$$\begin{aligned} \theta_{i,j,k}^{n+1} = & \theta_{i,j,k}^n + \frac{\Delta t}{2} \left[ -u \frac{\theta_{i+1,j,k}^n - \theta_{i-1,j,k}^n}{2\Delta x} - v \frac{\theta_{i,j,k+1}^n - \theta_{i,j,k-1}^n}{2\Delta y} - \right. \\ & - (w-w_g) \frac{\theta_{i,j,k+1}^n - \theta_{i,j,k-1}^n}{2\Delta z} + D_x \frac{\theta_{i+1,j,k}^n - 2\theta_{i,j,k}^n + \theta_{i-1,j,k}^n}{\Delta x^2} + \\ & + D_y \frac{\theta_{i,j,k+1}^n - 2\theta_{i,j,k}^n + \theta_{i,j,k-1}^n}{\Delta y^2} + D_z \frac{\theta_{i,j,k+1}^n - 2\theta_{i,j,k}^n + \theta_{i,j,k-1}^n}{\Delta z^2} - \sigma \theta_{i,j,k}^n + Q_{i,j,k} \left. \right] + \\ & + \frac{\Delta t}{2} \left[ -u \frac{\theta_{i+1,j,k}^{n+1} - \theta_{i-1,j,k}^{n+1}}{2\Delta x} - v \frac{\theta_{i,j,k+1}^{n+1} - \theta_{i,j,k-1}^{n+1}}{2\Delta y} - (w-w_g) \frac{\theta_{i,j,k+1}^{n+1} - \theta_{i,j,k-1}^{n+1}}{2\Delta z} + \right. \\ & + D_x \frac{\theta_{i+1,j,k}^{n+1} - 2\theta_{i,j,k}^{n+1} + \theta_{i-1,j,k}^{n+1}}{\Delta x^2} + D_y \frac{\theta_{i,j,k+1}^{n+1} - 2\theta_{i,j,k}^{n+1} + \theta_{i,j,k-1}^{n+1}}{\Delta y^2} + \\ & \left. + D_z \frac{\theta_{i,j,k+1}^{n+1} - 2\theta_{i,j,k}^{n+1} + \theta_{i,j,k-1}^{n+1}}{\Delta z^2} - \sigma \theta_{i,j,k}^{n+1} + Q_{i,j,k} \right]. \end{aligned} \quad (9)$$

Аппроксимируем граничные условия с использованием центральной разности:

$$-D_x \frac{\theta_{1,j,k} - \theta_{0,j,k}}{2\Delta x} = 0; \quad D_x \frac{\theta_{N_x-1,j,k} - \theta_{N_x-2,j,k}}{2\Delta x} = 0; \quad (10)$$

$$-D_y \frac{\theta_{i,1,k} - \theta_{i,0,k}}{2\Delta y} = 0; \quad D_y \frac{\theta_{i,Ny-1,k} - \theta_{i,Ny-2,k}}{2\Delta y} = 0; \quad (11)$$

$$-D_z \frac{\theta_{i,j,1} - \theta_{i,j,0}}{2\Delta z} = \beta\theta_{i,j,0}; \quad D_z \frac{\theta_{i,j,Nz-1} - \theta_{i,j,Nz-2}}{2\Delta z} = 0. \quad (12)$$

Соответственно, для первых и последних точек имеем:

$$\begin{aligned} \theta_{0,j,k} &= \theta_{i,j,k}; & \theta_{Nx-1,j,k} &= \theta_{Nx-2,j,k}; \\ \theta_{i,0,k} &= \theta_{i,1,k}; & \theta_{i,Ny-1,k} &= \theta_{i,Ny-2,k}; \\ \theta_{i,j,0} &= \frac{-D_z \theta_{i,j,1}}{2\Delta z \beta - D_z}; & \theta_{i,j,Nz-1} &= \theta_{i,j,Nz-2}. \end{aligned} \quad (13)$$

На каждом временном шаге для каждой точки сетки  $(i, j, k)$  возникает система линейных уравнений вида:

$$Ax = b. \quad (14)$$

Матрица коэффициентов  $A$ , включающая все разностные операторы для  $\theta^{n+1}$ , формируется из коэффициентов дискретизированного уравнения переноса и диффузии примеси в атмосфере (9):

- в узлах  $(i, j, k)$  на главной диагонали:

$$A_{i,j,k;i,j,k} = 1 + \frac{\Delta t}{2} \left( -\sigma + \frac{D_x}{\Delta x^2} + \frac{D_y}{\Delta y^2} + \frac{D_z}{\Delta z^2} \right);$$

- для узлов  $(i+1, j, k)$  и  $(i-1, j, k)$  в соседних точках по  $x$ :

$$\begin{aligned} A_{i,j,k;i+1,j,k} &= \frac{\Delta t}{4} \left( -u \frac{1}{\Delta x} + D_x \frac{1}{\Delta x^2} \right); \\ A_{i,j,k;i-1,j,k} &= \frac{\Delta t}{4} \left( u \frac{1}{\Delta x} + D_x \frac{1}{\Delta x^2} \right); \end{aligned}$$

$$\begin{aligned} b_{i,j,k} &= \theta_{i,j,k}^n + \frac{\Delta t}{2} \left[ -u \frac{\theta_{i+1,j,k}^n - \theta_{i-1,j,k}^n}{2\Delta x} - v \frac{\theta_{i,j+1,k}^n - \theta_{i,j-1,k}^n}{2\Delta y} - \right. \\ &\quad \left. -(w - w_g) \frac{\theta_{i,j,k+1}^n - \theta_{i,j,k-1}^n}{2\Delta z} + D_x \frac{\theta_{i+1,j,k}^n - 2\theta_{i,j,k}^n + \theta_{i-1,j,k}^n}{\Delta x^2} + \right. \\ &\quad \left. + D_y \frac{\theta_{i,j+1,k}^n - 2\theta_{i,j,k}^n + \theta_{i,j-1,k}^n}{\Delta y^2} + D_z \frac{\theta_{i,j,k+1}^n - 2\theta_{i,j,k}^n + \theta_{i,j,k-1}^n}{\Delta z^2} - \sigma \theta_{i,j,k}^n + Q_{i,j,k} \right]. \end{aligned} \quad (15)$$

Для решения системы линейных уравнений (14), используем итерационный метод сопряженных градиентов, особенно эффективный для разреженных, симметричных, положительно определенных матриц. Метод сопря-

- для узлов  $(i, j+1, k)$  и  $(i, j-1, k)$  в соседних точках по  $y$ :

$$\begin{aligned} A_{i,j,k;i,j+1,k} &= \frac{\Delta t}{4} \left( -v \frac{1}{\Delta y} + D_y \frac{1}{\Delta y^2} \right); \\ A_{i,j,k;i,j-1,k} &= \frac{\Delta t}{4} \left( v \frac{1}{\Delta y} + D_y \frac{1}{\Delta y^2} \right); \end{aligned}$$

- для узлов  $(i, j, k+1)$  и  $(i, j, k-1)$  в соседних точках по  $z$ :

$$\begin{aligned} A_{i,j,k;i,j,k+1} &= \frac{\Delta t}{4} \left( -(w - w_g) \frac{1}{\Delta z} + D_z \frac{1}{\Delta z^2} \right); \\ A_{i,j,k;i,j,k-1} &= \frac{\Delta t}{4} \left( (w - w_g) \frac{1}{\Delta z} + D_z \frac{1}{\Delta z^2} \right). \end{aligned}$$

Вектор правой части  $b$  формируется из всех уже известных значений концентрации  $\theta^n$  с включением влияния источника выброса примеси:

женных градиентов – это частный случай метода сопряженных направлений [14, 15]. Суть этого метода состоит в том, чтобы найти такое приближение для решения  $x^*$  при котором разность (невязка)  $r^* = b - Ax^*$  станет настолько

малой, чтобы считать текущее приближение близким к истинному решению.

Для начала итерационного процесса выбирается произвольное начальное приближение  $x_0$  (обычно нулевой вектор) – в нашем случае начальная концентрация для которого вычисляется невязка  $r_0$  и вектор направления  $p_0$ :

$$x_0 = \theta^0, \quad r_0 = b - Ax_0 = b, \quad p_0 = r_0 = b.$$

Затем, на каждом шаге итерации  $k = 0, 1, 2, \dots$ , до тех пор, пока не сойдется итерационный процесс, выполняем следующие действия:

1) вычисляем длину шага в направлении движения поиска – коэффициент  $\alpha_k$ :

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k};$$

2) обновляем приближение решения на очередной итерации:

$$x_{k+1} = x_k + \alpha_k p_k;$$

3) обновляем значение невязки:

$$r_{k+1} = r_k - \alpha_k A p_k;$$

4) проверяем условие сходимости:

$$\|r_{k+1}\|_2 = \|b - Ax_{k+1}\|_2 < \varepsilon;$$

5) если условие выполнено – итерации прекращаются, иначе вычисляем коэффициент  $\beta_k$ :

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k};$$

6) обновляем направление движения поиска:

$$p_{k+1} = r_{k+1} + \beta_k p_k;$$

7) увеличиваем счетчик итераций:

$$k = k + 1;$$

8) возвращаемся к действию 1.

Критерий сходимости итерационного процесса выбирается в зависимости от требований к точности решения. Для задач со средней точностью значение  $\varepsilon = 10^{-6}$  будет оптимальным.

Рассмотренная конечно-разностная схема имеет второй порядок точности по времени и

пространству  $O(\tau^2) + O(h^2)$ , а общая сложность вычислительного алгоритма составляет  $O(I \cdot k)$ , где  $I$  – общее число узлов сетки, а  $k$  – число итераций.

#### IV. ПАРАЛЛЕЛИЗАЦИЯ ВЫЧИСЛИТЕЛЬНОГО АЛГОРИТМА

Выше рассмотренный вычислительный алгоритм при всей своей точности и относительной простоте программной реализации, в смысле эффективности чрезвычайно зависит от пространственного разрешения сетки. Вычислительная сложность алгоритма значительно возрастает с увеличением размеров сетки  $N_x, N_y, N_z$  и количества временных шагов  $Nt$ . Тем самым, для комфортного и многократного проведения вычислительных экспериментов становится целесообразным максимально возможное повышение его производительности.

На практике данная проблема, чаще всего, решается путем «распараллеливания» наиболее ресурсоемких частей вычислительного алгоритма [16, 17]. Отметим, что в нашем случае программная реализация была выполнена на языке python, который имеет встроенные модули многопоточной обработки данных. Но для по-настоящему сложных вычислительных задач, когда требуется параллельная обработка данных как между локальными процессорными ядрами, так и между процессорами в сети, используются сторонние специализированные библиотеки и фреймворки для параллелизации, такие как «multiprocessing», «joblib», «numba», «scipy.pyamg», «mpi4py» и др. Выбор конкретной библиотеки зависит от доступных вычислительных ресурсов и требований к производительности.

Подавляющую часть времени работы рассмотренного численного алгоритма занимают операции над матрицами: умножение матрицы на вектор, вычисление скалярного произведения векторов и обновление вектора. Все эти операции достаточно хорошо поддаются распараллеливанию. В этой связи, с учетом доступной вычислительной мощности и требований была выбрана библиотека «numba», позволяющая значительно повысить эффективность (производительность) алгоритма за счет оптимизации большого количества операций над массивами [18].

Распараллеливание алгоритма реализовано на отдельных участках программного кода, инкапсулированных в рамках четырех функций:

«time\_step», «apply\_boundary\_conditions», «cg\_solver» и «matvec» (рис. 1).

Первая в иерархии вложенности функция «time\_step» вызывается в основном цикле по времени. За один шаг  $\Delta t$  функция выполняет обновление значений концентрации  $\theta$  в узлах сетки  $\Omega$ , обновление вектора правой части  $b$ , применение граничных условий и решение системы линейных уравнений (рис. 1).

Следующая «apply\_boundary\_conditions» функция обновляет элементы вектора  $b$  с учетом условия взаимодействия на нижней гра-

нице. Вектор  $b$  здесь представляется одномерным массивом длиной  $N_x \times N_y \times N_z$ . Для каждого элемента  $b_{i,j,0}$ , соответствующего нижней границе (плоскость  $z = 0$ ), выполняется операция:

$$b_{i+j \cdot N_x} = b_{i,j,0} + \frac{Dz}{\Delta z} \frac{\theta_{i,j,1}}{1 + \beta \frac{\Delta z}{Dz}},$$

где  $b_{i+j \cdot N_x}$  – это элемент вектора  $b$ , соответствующий координатам  $(i, j, 0)$  в одномерном представлении.

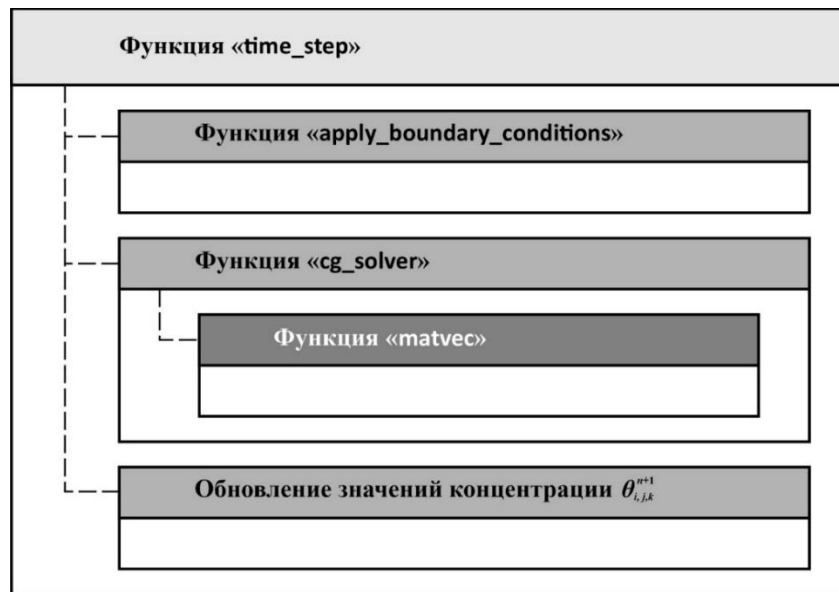


Рис. 1. Наиболее ресурсоемкие функции в программном коде

Распределение данных в функции «apply\_boundary\_conditions» осуществляется следующим образом – каждый из доступных процессоров  $P$  получает определенный диа-

пазон индексов массива для обработки. Предположим, что у нас четыре процессора и  $N_y = 40$ ,  $N_x = 40$ , тогда распределение будет таким (таб. 1):

Таблица 1. Распределение элементов вектора по процессорам

№ процессора	Цикл по $j$	Цикл по $i$
1	от 0 до 9	от 0 до 39
2	от 10 до 19	от 0 до 39
3	от 20 до 29	от 0 до 39
4	от 30 до 39	от 0 до 39

Здесь каждый элемент  $b_{i+j \cdot N_x}$  обновляется независимо, соответственно отсутствует необходимость в синхронизации между процессорами. Иными словами отсутствуют блокировки или ожидания других процессоров, что минимизирует накладные расходы на параллелизацию. После завершения параллельных циклов,

обновленный вектор  $b$  становится доступным для дальнейших вычислений.

Наиболее ресурсоемкая функция «cg\_solver» реализует метод сопряженных градиентов для решения разреженной системы линейных уравнений  $Ax = b$ . Каждый процессор обрабатывает выделенный ему диапазон строк

матрицы  $A$  и частей векторов  $x$ ,  $r$ ,  $p$ , соответствующих выделенным строкам матрицы (рис. 2).

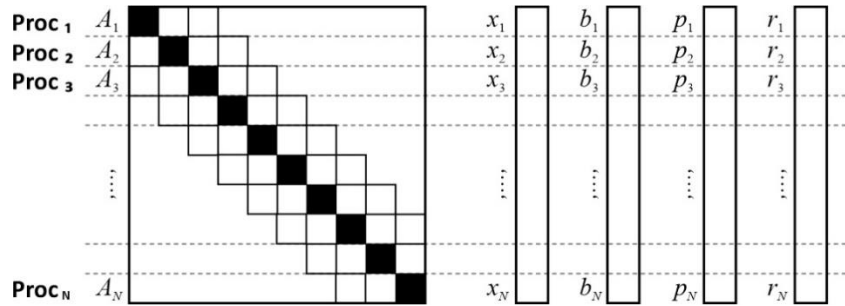


Рис. 2. Построчное разделение по процессорам

То есть, если имеем  $P$  процессоров и матрица  $A$  имеет  $N$  строк, то каждый процессор получает примерно  $N/P$  строк и такое же количество соответствующих элементов векторов  $x$ ,  $r$ ,  $p$ . Основное взаимодействие (синхронизация) между процессорами происходит на этапе вычисления скалярных произведений, где необходимо объединение частичных сумм.

Внутри функции «cg\_solver» вызывается функция «matvec», которая выполняет умножение разреженной матрицы  $A$  на вектор  $p_k$ , а на выходе возвращает результирующий вектор  $Ap_k$ . Каждая итерация цикла, отвечающая за вычисление одного элемента результирующего вектора  $Ap_{k_i}$ , будет выполняться параллельно.

То есть для каждого  $i$ :

$$Ap_{k_i} = A_{ii}p_{k_i} + \sum_{j \neq i} A_{ij}p_{k_j},$$

Таблица 2. Распределение строк матрицы по процессорам

№ процессора	Вычисление элементов $Ap_k$ для каждого $i$
1	от 0 до 249
2	от 250 до 499
3	от 500 до 749
4	от 750 до 999

Далее, функция «cg\_solver» выполняет вычисление скалярных произведений  $r_k^T r_k$ ,  $p_k^T Ap_k$ ,  $r_{k+1}^T r_{k+1}$  и  $\alpha_k$ ,  $\beta_k$ . Скалярные произведения векторов вычисляются как суммы произведений их элементов. Каждый процессор обрабатывает свой диапазон индексов для вычисления частичных сумм. Частичные суммы объединяются путем синхронизации. Обновления векторов  $x$ ,  $r$ ,  $p$  выполняются локально на каждом процессоре для выделенной ему ча-

где  $A_{ii}$  – диагональный элемент матрицы  $A$ ;  $p_{k_i}$  – соответствующий элемент вектора  $p_k$ ;  $A_{ij}$  – ненулевой элемент матрицы  $A$ ;  $p_{k_j}$  – элемент вектора  $p_k$ , соответствующий  $j$ -му ненулевому элементу  $i$ -ой строки матрицы  $A$ .

Распределение данных осуществляется следующим образом – каждый процессор получает диапазон значений индекса  $i$  для обработки. Например, если имеется 4 процессора и длина  $p_k$  составляет 1000, то каждый процессор получит по 250 итераций цикла, причем синхронизация между ними не потребуется. После завершения всех итераций, результаты вычисления значений элементов вектора  $Ap_k$  объединяются. Предположим, что у нас 4 процессора и длина вектора  $p_k$  равна 1000 (таб. 2):

сти данных. После завершения каждой итерации  $k$ , функция «cg\_solver» проверяет условие сходимости и решает, продолжать ли вычисления.

В результате работы «cg\_solver» получаем решение  $\theta$  на текущей итерации цикла по времени, после чего функция «time\_step» обновляет значение концентрации  $\theta^{n+1}$ , а затем выполняется переход к очередному шагу итерации цикла по времени.

Распараллеливание функции «time\_step» позволяет значительно ускорить вычисления за

счет распределения операций между доступными ядрами процессора. В частности, функция «`apply_boundary_conditions`» эффективно использует параллелизацию с минимальными накладными расходами на синхронизацию, а функции «`cg_solver`» и «`matvec`» значительно ускоряют выполнение всех матричных операций. Тем самым, скорость работы численного

алгоритма многократно возрастает при запуске на многопроцессорных системах.

Для проверки производительности последовательного и параллельного вариантов вычислительного алгоритма было проведено несколько экспериментов. Тестирование скорости выполнения всех вычислительных операций осуществлялось на ПК со следующими характеристиками (таб. 3):

Таблица 3. Характеристики вычислительной системы

ЦПУ	Тактовая частота (ГГц)	Кол-во физических ядер	Кол-во логических процессоров	ОЗУ (ГБ)
1 x Intel(R) Core(TM) i7-10700	2,90	8	16	16

В таб. 4 приведены результаты тестирования времени работы последовательного и па-

раллельного алгоритмов при различном разрешении дискретной сетки и связанном объеме вычислительной нагрузки.

Таблица 4. Вычислительная нагрузка и производительность обычной и параллельной версий численного алгоритма

$N_x$	$N_y$	$N_z$	$N_t$	Количество узлов сетки	Количество вычислительных операций	Время работы последовательного алгоритма (сек.)	Время работы параллельного алгоритма (сек.)
10	10	10	10	1 000	$10^7$	0,60	6,67
20	20	20	10	8 000	$8 \cdot 10^7$	78,00	6,19
30	30	30	10	27 000	$2,7 \cdot 10^8$	1156,10	6,22
40	40	40	10	64 000	$6,4 \cdot 10^8$	2749,70	6,24
100	100	100	10	1 000 000	$10^{13}$	42802,90	8,05

Таким образом, общая сложность параллельного вычислительного алгоритма составляет:

$$O\left(I \frac{n^2}{P}\right),$$

где  $n$  – количество неизвестных (размер матрицы коэффициентов);  $I$  – количество итераций;  $P$  – количество используемых процессоров.

## V. АНАЛИЗ РЕЗУЛЬТАТОВ ВЫЧИСЛИТЕЛЬНЫХ ЭКСПЕРИМЕНТОВ

Адекватность разработанной компьютерной модели проверялась путем проведения серии вычислительных экспериментов. В ходе расчетов во всех случаях использовалась параллельная версия вычислительного алгоритма.

Для моделирования исследуемого процесса переноса и диффузии примеси в приземном слое атмосферы были приняты следующие

условия и ограничения. В качестве области решения задачи рассматривался параллелепипед с размерами  $L = M = N = 100$  м, а в качестве источника выброса загрязняющей примеси – участок подстилающей поверхности площадью  $S = 4$  м<sup>2</sup>. Размер вычислительной сетки во всех экспериментах принимался одинаковым:  $N_x = N_y = N_z = 100$  с шагами  $\Delta x, \Delta y, \Delta z, \Delta t = 1$  (рис. 3).

Вычислительные эксперименты проводились при различных значениях компонент скорости ветра  $u, v, w$ ; коэффициентов турбулентной диффузии  $D_x, D_y, D_z$ ; коэффициента поглощения  $\sigma$ , коэффициента  $\beta$  и мощности источника  $Q$ .

Отметим, что мощность площадного источника  $Q$  определяется как произведение интенсивности выброса с единицы площади на общую площадь источника, например,

$$Q = Q_S \cdot S = 1 \text{ г/м}^2\text{с} \times 4 \text{ м}^2 = 4 \text{ г/с}.$$

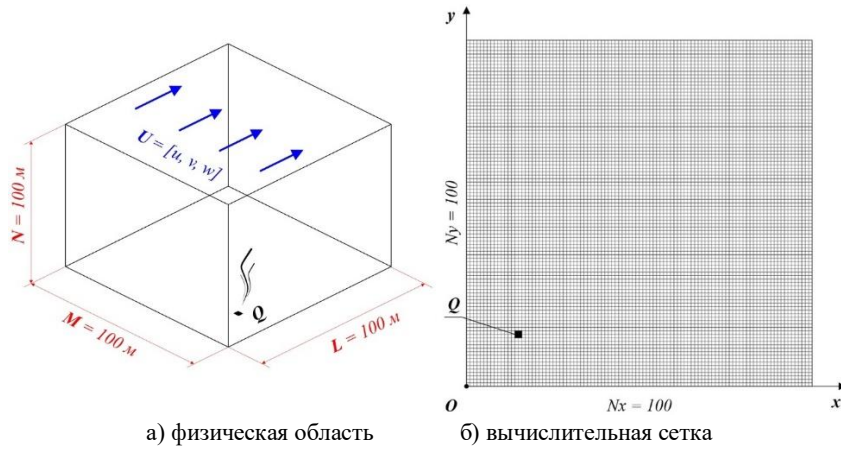


Рис. 3. Схематичное представление расчётной области

Полное количество (массу) примеси в области  $\Omega$  за время  $[0, T]$ , определяли с помощью функционала, определенного на множестве значений  $\theta$ :

$$P = \int_0^T \int_{\Omega} \theta(x, y, z, t) \Delta\Omega \Delta t,$$

или в дискретной форме

$$P \approx \sum_{n=0}^{N_t-1} \sum_{i=0}^{N_x-1} \sum_{j=0}^{N_y-1} \sum_{k=0}^{N_z-1} \theta_{i,j,k}^n \Delta x \Delta y \Delta z \Delta t.$$

На рис. 4-6 приводится визуализация результатов расчета поля концентрации загрязняющей примеси на различных высотах при следующих параметрах процесса:

- горизонтальная скорость ветра  $U = 2,4776$  м/с; направление ветра  $45^\circ$ ;
- $u = 1,7525, v = 1,7525, w = 0,0,$
- $w_g = 0,01$  м/с;
- $D_x = 0,001247, D_y = 0,001247,$
- $D_z = 0,05247$  м<sup>2</sup>/с;
- $\sigma = 0,0$  с<sup>-1</sup>;  $\beta = 0$ ;  $Q_s = 0,25$  г/м<sup>2</sup>·с;
- $\theta^0 = 0$  г/м<sup>3</sup>;  $t = 1.$

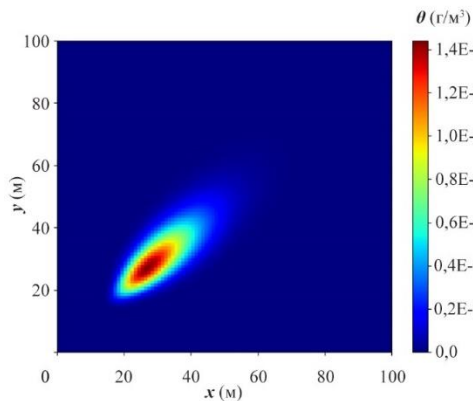


Рис. 4. Распределение концентрации примеси на высоте  $z = 5$  м

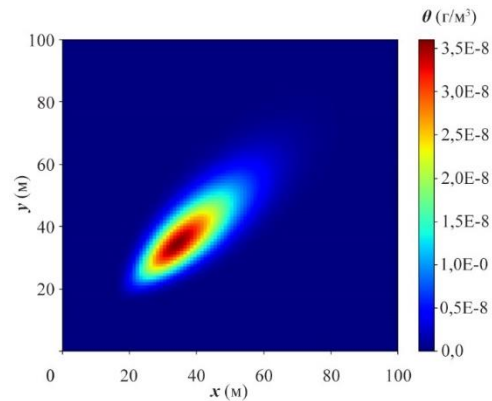


Рис. 5. Распределение концентрации примеси на высоте  $z = 7$  м

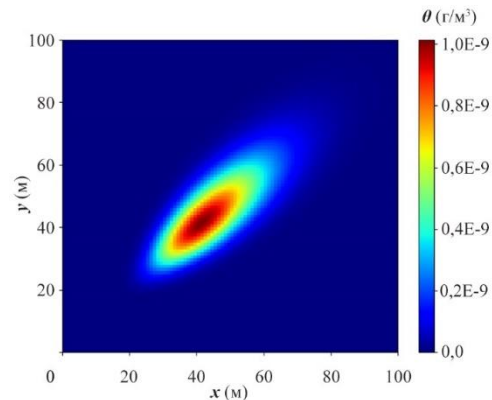


Рис. 6. Распределение концентрации примеси на высоте  $z = 9$  м

Как можно видеть из рис. 4-6, с удалением от источника шлейф выброса расширяется, а концентрация примеси снижается. Так, на высоте 9 м область рассеивания примеси почти вдвое превышает размеры шлейфа на высоте 5 м. Подобная картина исследуемого процесса вполне ожидаема, за исключением размеров самого шлейфа.

Другой аспект, на который нужно обратить внимание это то, что общая масса примеси

$P \approx 0,9999784485753571$  г, накопленная в рассматриваемой области за время  $t=1$  близка к точному значению. Однако, уже при  $t=10$  и прежних значениях остальных параметров, масса выброшенной примеси составляет  $P \approx 14,07$ , что значительно отличается от ожидаемого точного значения в 10 г.

Вполне очевидно, что данные проблемы связаны с выбором подходящего разрешения сетки по пространственным переменным и времени [19]. Тем не менее, для заданных  $u$ ,  $v$ ,  $w$  выбор значений  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$ ,  $\Delta t=1$  являлся оптимальным с точки зрения баланса между точностью и вычислительной эффективностью численного алгоритма даже с учетом его параллельной реализации.

На рис. 7-9 иллюстрируются влияние параметра поглощения примеси воздушной массой атмосферы  $\sigma$ , например в следствие повышенной влажности, а также параметра поглощения примеси на нижней границе  $\beta$ , например за счет прилипания к влажным участкам подстилающей поверхности земли. На рис. 9 дополнительно показано влияние возросшей скорости осаднения примеси  $w_g$ .

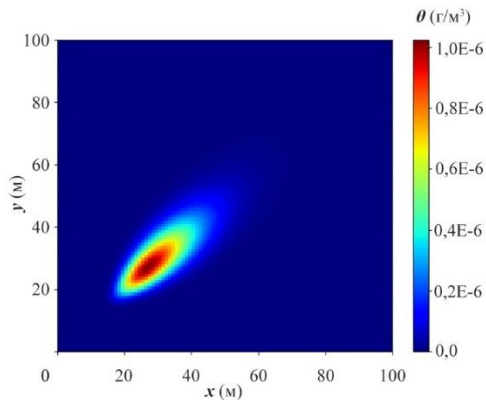


Рис. 7. Распределение концентрации примеси при  $\sigma=0,5$ ,  $\beta=0,5$ .  $P \approx 0,7116814481489455$  г.

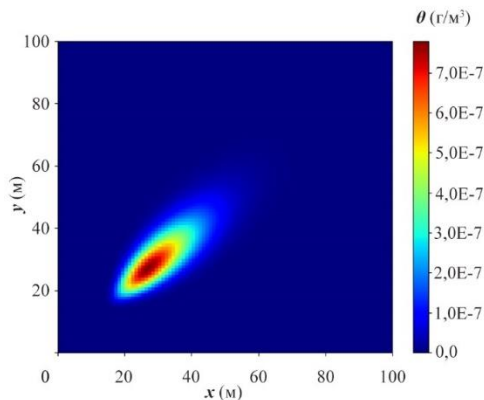


Рис. 8. Распределение концентрации примеси при  $\sigma=1,0$ ,  $\beta=1,0$ .  $P \approx 0,5413553701658289$  г.

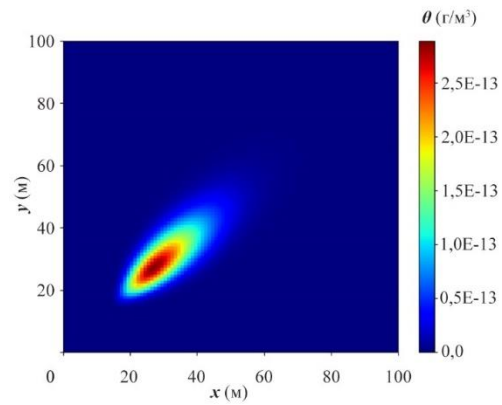


Рис. 9. Распределение концентрации примеси при  $\sigma=1,0$ ,  $\beta=1,0$  и  $w_g=0,1$ .

$$P \approx 0,3730115344110097 \text{ г.}$$

Распределение концентрации примеси на рис. 7-9 приведено на высоте  $z=5$  м. Значения остальных параметров аналогичны приведенным на рис. 4-6.

Из рис. 7 следует, что при вдвое больших значениях параметров  $\sigma$  и  $\beta$ , примерно на 30% снижается концентрация примеси на рассматриваемой высоте  $z=5$  м и в целом в области. В случае крайних верхних значений  $\sigma=1,0$  и  $\beta=1,0$  (рис. 8), масса примеси сокращается вплоть до 45%. Если же дополнительно увеличить скорость осаднения (рис. 9), то наблюдается более чем 60-ти процентное сокращение за счет «прилипания» к нижней границе большего количества примеси.

На рис. 10 приводится визуализация результатов расчета поля концентрации загрязняющей примеси на высоте  $z=5$  м при следующих параметрах процесса:  $U=2,5$  м/с; угол  $34,5^\circ$ ;  $u=2,05955$ ,  $v=1,41293$ ,  $w=0,0$ ,  $w_g=0,005$  м/с;  $D_x=0,000235$ ,  $D_y=0,002357$   $D_z=0,05351$  м<sup>2</sup>/с;  $\sigma=0,0$  с<sup>-1</sup>;  $\beta=0$ ;  $Q_S=1,0$  г/м<sup>2</sup>·с;  $\theta^0=0$  г/м<sup>3</sup> и разных значениях времени моделирования  $t$ .

Эксперименты на рис. 10 и 11 проводились при  $Q_S=1,0$ , то есть общая мощность источника составляла  $Q=4,0$  г/с. Небольшие изменения значений скорости ветра и коэффициентов турбулентной диффузии в целом не повлияли на характер развития процесса распространения примеси.

Но общее увеличение выброса примеси, в отличие от экспериментов на рис. 4-9, теперь привело к расхождению баланса массы в сторону повышения. Причем на каждой итерации по времени данная ошибка возрастает. Если при  $t=1$  (рис. 10) значение  $P$  можно считать

допустимым, то при  $t = 5$  (рис. 11) значение  $P$  уже слишком сильно отличается от ожидаемого значения в 20 г.

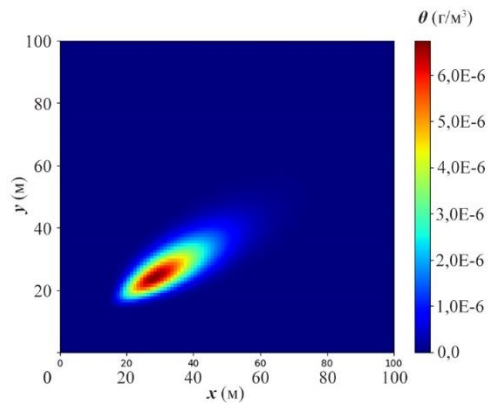


Рис. 10. Распределение концентрации примеси на высоте  $z = 5$  м при  $t = 1$ .  $P \approx 4,000017894176836$  г.

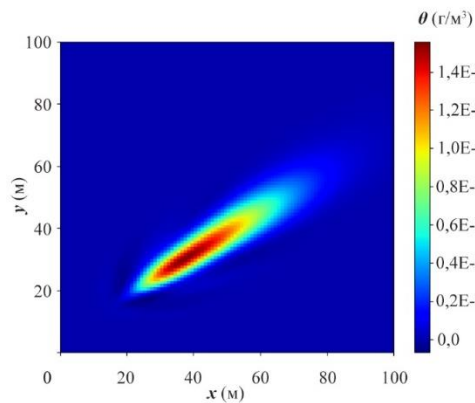


Рис. 11. Распределение концентрации примеси на высоте  $z = 5$  м при  $t = 5$ .  $P \approx 28,93248897006549$  г.

Данная проблема вновь отсылает к необходимости выбора оптимальных значений  $\Delta x$ ,  $\Delta y$ ,  $\Delta z$  и  $\Delta t$ , напрямую влияющих на точность и стабильность численного решения.

Для оценки сходимости численных алгоритмов используются разнообразные методы, в том числе: метод эталонного решения, если известно аналитическое решение задачи; анализ сеточной сходимости, заключающийся в

наблюдении за решениями при изменениях сетки разными пространственными шагами; анализ временной сходимости – аналогичный метод, но с использованием различных значений временного шага.

Так как, получить аналитическое решение задачи (1)-(5) практически не представляется возможным, а анализ сеточной сходимости чрезвычайно ресурсоемкий для имеющейся вычислительной системы (таб. 3), то для экспериментов в данном исследовании был выбран метод анализа временной сходимости.

Анализ временной сходимости позволяет убедиться, что численный метод, реализованный в виде программного алгоритма, стабилен и точно аппроксимирует решение при уменьшении временного шага  $\Delta t$ . Суть анализ временной сходимости заключается в сравнении решений с различных временными шагами и оценке ошибок.

В первую очередь требуется получить эталонное решение при достаточно малом временном шаге. В данном случае эталонное решение было получено при  $\Delta t_{\text{ref}} = 0,01$  и ранее указанных исходных параметрах процесса:  $L = M = N = 100$  м;  $N_x = N_y = N_z = 100$ ;  $\Delta x, \Delta y, \Delta z, \Delta t = 1$ ;  $u = 1,7525$ ,  $v = 1,7525$ ,  $w = 0,0$ ,  $w_g = 0,01$  м/с;  $D_x = 0,001247$ ,  $D_y = 0,001247$ ,  $D_z = 0,05247$  м<sup>2</sup>/с;  $\sigma = 0,0$  с<sup>-1</sup>;  $\beta = 0$ ;  $Q_S = 0,25$  г/м<sup>2</sup>·с;  $S = 4$  м<sup>2</sup>;  $\theta^0 = 0$  г/м<sup>3</sup>;  $T = 1$ .

Далее, решение задачи при тех же параметрах было последовательно получено для временных шагов:  $\Delta t_1 = 1,0$ ,  $\Delta t_2 = 0,5$ ,  $\Delta t_3 = 0,25$ ,  $\Delta t_4 = 0,1$ ,  $\Delta t_5 = 0,05$ . После чего, для каждого временного шага вычислялась ошибка по сравнению с эталонным решением. Расчет ошибки выполнялся с использованием  $L_2$ -нормы [20]:

$$E(\Delta t_n) = \sqrt{\frac{1}{N_x \cdot N_y \cdot N_z} \sum_{i,j,k} (\theta_{i,j,k}(\Delta t_n) - \theta_{i,j,k}(\Delta t_{\text{ref}}))^2},$$

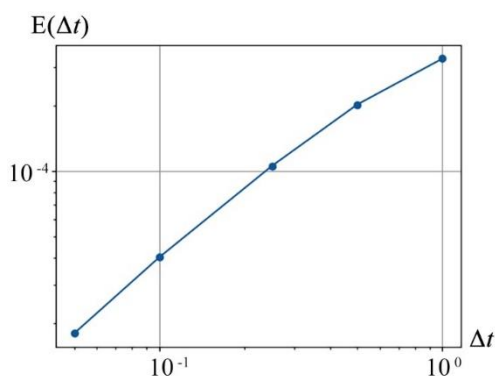
где  $E(\Delta t_n)$  – среднеквадратичная ошибка между решением для шага  $\Delta t_n$  и эталонным решением для шага  $\Delta t_{\text{ref}}$ .

Полученные результаты приведены в таб. 5 и рис. 12. Ошибка для эталонного решения  $\theta_{\text{ref}}$  при  $\Delta t_{\text{ref}} = 0,01$  принимается равной нулю.

Как следует из полученных результатов (таб. 5, рис. 12), ошибка уменьшается с уменьшением  $\Delta t$ , то есть можно считать, что численный алгоритм становится более точным, а схема сходимая. Так, при  $\Delta t_4 = 0,1$ ,  $\Delta t_5 = 0,05$  ошибки значительно меньше, чем для больших временных шагов, что показывает хорошее асимптотическое поведение схемы при уменьшении временного шага.

Таблица 5. Ошибки решения для различных  $\Delta t$ 

Ошибка	Значение
$E(\Delta t_1)$	0,00032996525660511887
$E(\Delta t_2)$	0,00020284153281318896
$E(\Delta t_3)$	0,00010592688787832563
$E(\Delta t_4)$	4,0258128144193756E-05
$E(\Delta t_5)$	1,7958847115007180E-05

Рис. 12. График ошибок решения для различных  $\Delta t$ .

Наконец, для оценки порядка временной сходимости для приведённых значений  $\Delta t$  и ошибок  $E(\Delta t)$ , используем метод Ричардсона [20]:

$$p \approx \frac{\log(E(\Delta t)/E(\Delta t/2))}{\log(2)} \quad \text{или}$$

$$p \approx \frac{\log(E(\Delta t_i)/E(\Delta t_{i+1}))}{\log(\Delta t_i/\Delta t_{i+1})}.$$

Полученные оценки порядка сходимости  $p$  для каждой пары  $\Delta t$  приведены на рис. 13.

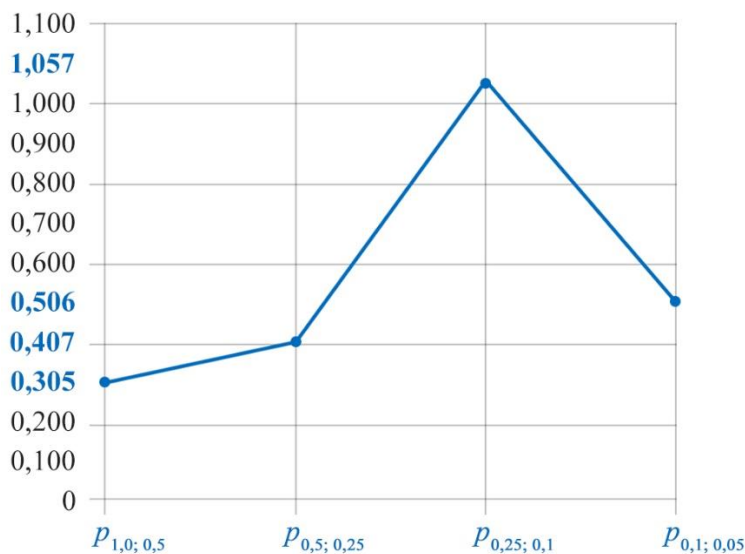


Рис. 13. Оценка порядка сходимости

Величина оценки порядка сходимости должна быть как можно больше. Высокий порядок сходимости означает, что решение численной схемы быстрее приближается к точному решению при уменьшении шага по времени.

Однако, график на рис. 13 указывает на то, что рассмотренная конечно-разностная схема не соответствует равномерному порядку сходимости, а его колебания вероятно связаны с какими-то особенностями программной реализации алгоритма либо самой схемы.

## VI. ЗАКЛЮЧЕНИЕ

В качестве заключения по проведенной работе можно отметить следующие основные выводы.

Построена математическая модель процесса адвективного переноса и турбулентной диффузии загрязняющей примеси в атмосфере с учетом поглощения частиц примеси воздушной массой и характера взаимодействия частиц примеси с подстилающей поверхностью назем-

ного покрова. Для приближенного решения поставленной задачи предложен численный алгоритм на основе методов конечных разностей и сопряженных градиентов с применением смешанной схемы Кранк-Николсона, позволяющей повысить точность и стабильность решения по сравнению с обычными явными и неявными схемами.

Программно реализованный численный алгоритм позволяет эффективно решать задачу распространения примеси в атмосфере с учетом граничных условий и влияния основных факторов. Значительное ускорение работы алгоритма достигается за счет распараллеливания с помощью python библиотеки numba наиболее ресурсоёмких участков кода, содержащих вычислительные процедуры решения систем линейных уравнений.

Анализ результатов временной сходимости показал уменьшение ошибки при уменьшении временного шага, что подтверждает временную сходимость примененной схемы. При соблюдении баланса между точностью, стабильностью и вычислительными затратами, для получения наиболее точных решений, следует выбирать значения  $\Delta t$  в диапазоне  $0,05 < \Delta t < 0,1$ .

Полученные оценки порядка сходимости для каждой пары временных шагов указывают на то, что предложенная разностная схема не демонстрирует равномерный порядок сходимости, а его колебания могут быть связаны с особенностями временного шага и программной реализации численного алгоритма.

В целом, анализ результатов вычислительных экспериментов позволяет считать, что разработанная модель достаточно адекватно описывает процесс переноса и диффузии загрязняющей примеси в атмосфере и не противоречит его физической природе.

#### ЛИТЕРАТУРА

- [1] *Szymkiewicz R.* Numerical Solution of the Advection-Diffusion Equation // Numerical Modeling in Open Channel Hydraulics. – Dordrecht: Springer, 2010. – P. 263-300. – (Water Science and Technology Library; Vol. 83). – DOI: 10.1007/978-90-481-3674-2\_7.
- [2] *Wang S.T., McMillan A.F., Chen B.H.* An approximate solution to the advection-diffusion equation as applied to an estuary // Journal of Hydrology. – 1980. – Vol. 48, Issue 3-4. – P. 251-268. 10.1016/0022-1694(80)90118-3.
- [3] Уравнение адвекции и численные методы его решения // Динамические модели в биологии / Московский государственный Университет им. М.В.Ломоносова; Г.Ю. Ризниченко и др. – 2001. – Режим доступа: <https://shorturl.at/OWZND>.
- [4] *Adeyemo O.D., Motsepa T., Khalique C.M.* A study of the generalized nonlinear advection-diffusion equation arising in engineering sciences // Alexandria Engineering Journal. – 2022. – Vol. 61, Issue 1. – P. 85-194. – DOI: 10.1016/j.aej.2021.04.066.
- [5] *Rapp B.E.* Finite difference method // Microfluidics : modeling, mechanics, and mathematics. – 2d ed. – Amsterdam: Elsevier, 2023. – P. 667-676. – (Micro and Nano Technologies). – DOI: 10.1016/B978-0-12-824022-9.00050-4.
- [6] *Самарский А.А.* Теория разностных схем. – 3-е изд., испр. – М.: Наука, 1977. – 616 с.
- [7] *Вараксин А.Ю.* Гидрогазодинамика и теплофизика двухфазных потоков: проблемы и достижения (обзор) // Теплофизика высоких температур. – 2013. – Т. 51, № 3. – С. 421-455.
- [8] *Ravshanov N., Akhmedov D., Roziyeva G.* GIS based estimation of the vertical wind profile effect on air pollutants disperse in the atmosphere // AIP Conference Proceedings. – 2023. – Vol. 2781, Issue 1. – DOI: 10.1063/5.0144801.
- [9] *Sharipov D., Akhmedov D., Boborakhimov B., Sharipov Kh.* Modelling of fine aerosols diffusive transport in the atmosphere // International Conference on Information Science And Communications Technologies: Applications, Trends And Opportunities. – Tashkent, 2022. – P. 1-6. – DOI: 10.1109/ICISCT55600.2022.10146840.
- [10] *Sharipov D., Akhmedov D.* Aggregation of Meteorological and Spatial Data for Air Pollution Modeling // International Conference on Information Science And Communications Technologies: Applications, Trends And Opportunities. – Tashkent, 2021. – P. 1-4. DOI: 10.1109/ICISCT52966.2021.9670325.
- [11] *Ravshanov N., Akhmedov D., Kravets O.Ja.* Atmospheric dispersion modeling in ecological engineering problems // IOP Conf. Series: Materials Science and Engineering. – 2020. – Vol. 862. – DOI: 10.1088/1757-899X/862/6/062017.
- [12] *Clain S., Machado G.J., Malheiro M.T.* Compact schemes in time with applications to partial differential equations // Computers & Mathematics with Applications. – 2023. –

- Vol. 140. – P. 107-125. – DOI: 10.1016/j.camwa.2023.03.011.
- [13] Jena S.R., Senapati A. One-dimensional heat and advection-diffusion equation based on improvised cubic B-spline collocation, finite element method and Crank-Nicolson technique // International Communications in Heat and Mass Transfer. – 2023. – Vol. 147. 10.1016/j.icheatmasstransfer.2023. 106958.
- [14] Shewchuk J.R. An Introduction to the Conjugate Gradient Method Without the Agonizing Pain : Technical Report. – Pittsburgh: Carnegie Mellon University, 1994. – 58 p. – DOI: 10.5555/865018.
- [15] Максимов Ю.Я. Алгоритмы линейного и дискретного программирования. – Москва : МИФИ, 1980. – 72 с.
- [16] Meier U., Eigenmann R. Parallelization and performance of Conjugate Gradient algorithms on the Cedar hierarchical-memory multiprocessor // ACM SIGPLAN Notices. – 1991. – Vol. 26, Issue 7. – P. 178-188. – DOI: 10.1145/109626.109644.
- [17] Ping H. et al. DCG: An efficient Distributed Conjugate Gradient algorithm for solving linear equations in multi-agent networks // Results in Control and Optimization. – 2023. – Vol. 10. – DOI: 10.1016/j.rico.2023.100213.
- [18] Numba: A High Performance Python Compiler / T. Oliphant; Continuum Analytics; Community project. – 2012. . – Режим доступа: <https://numba.pydata.org>.
- [19] Равианов Н., Боборахимов Б.И., Журабовева О.С. Численное моделирование турбулентного потока и распространения примеси в условиях уличного каньона // Проблемы вычислительной и прикладной математики. – 2024. – №1(55). – С. 8-25.
- [20] Arnold D.N. Stability, Consistency, and Convergence of Numerical Discretizations // Encyclopedia of Applied and Computational Mathematics / ed. B. Engquist. – Berlin: Springer, 2015. – P. 1358-1364. – DOI: 10.1007/978-3-540-70529-1.

Поступила в редакцию 06.05.2024

**Цитирование:** Ахмедов Д.Д., Боборахимов Б.И., Убайдуллаев М.Ш., Насруллаев П.А. (2024). Параллельный алгоритм расчета поля приземной концентрации загрязняющей примеси в атмосфере. *Международный Журнал Теоретических и Прикладных Вопросов Цифровых Технологий*, 7(3), –С. 19-32. <https://doi.org/10.62132/ijdt.v7i3.193>

## A PARALLEL ALGORITHM FOR CALCULATING THE FIELD OF SURFACE CONCENTRATION OF POLLUTANTS IN THE ATMOSPHERE

\*Akhmedov D.D.<sup>1</sup>, Boborakhimov B.I.<sup>1</sup>, Ubaydullaev M.Sh.<sup>1</sup>, Nasrullaev P.A.<sup>1</sup>

<sup>1</sup> Digital technologies and artificial intelligence research institute, Tashkent, Uzbekistan  
\*a.dilshot@mail.ru

**Abstract.** Modeling the distribution of pollutants in the atmosphere is one of the most important tasks in environmental research and environmental protection. The advection-diffusion equation is widely used to study this process. The accuracy of the numerical solution of problems formulated on the basis of the advection-diffusion equation is largely determined by the method of approximating terms that describe the effects taken into account, including transport, turbulent diffusion, absorption of impurities, properties of impurity particles, terrain, etc. In this work, attention is focused on a number of issues related to the construction of a mathematical model of the process under study, the development of a numerical algorithm for solving the problem based on the Crank-Nicholson difference scheme, increasing the efficiency of the algorithm by parallelizing computational procedures, as well as analyzing the accuracy and convergence of the numerical method used in relation to solving the problem of pollutant propagation in the atmosphere.

**Keywords:** atmosphere, pollutant, turbulence, parallel algorithm, Crank-Nicholson scheme, iterative method.