



Численное моделирование переноса твёрдых частиц в атмосферном городском пограничном слое с использованием лагранжева подхода: физические задачи и параллельная реализация

^{1,2} А.И. Варенцов, ORCID: 0000-0001-7029-6773 <aivarentsov98@gmail.com>

² О.А. Имеев, ORCID: 0009-0008-5592-0759 <ochir90@yandex.ru>

³ А.В. Глазунов, ORCID: 0000-0002-8780-3513 <and.glas@gmail.com>

^{1,3} Е.В. Мортиков, ORCID: 0000-0002-9683-5701 <evgeny.mortikov@gmail.com>

^{1,2} В.М. Степаненко, ORCID: 0000-0003-3033-6712 <stepanen@srcc.msu.ru>

¹ Московский государственный университет имени М.В. Ломоносова
Научно-исследовательский вычислительный центр
119234, Россия, г. Москва, Ленинские горы, д. 1, стр. 4

² Институт физики атмосферы им. А.М. Обухова Российской академии наук
119017, Россия, Пыжевский пер., 3

³ Институт вычислительной математики им. Г.И. Марчука Российской академии наук
119333, г. Москва, ул. Губкина, 8

Аннотация. Работа представляет результаты развития численной модели лагранжева переноса частиц и применения методов параллельных вычислений для увеличения эффективности программной реализации модели. Модель реализована в виде программного комплекса, позволяющего проводить расчёты переноса и осаждения аэрозольных частиц с учётом свойств частиц и входных данных, описывающих атмосферные условия и геометрию подстилающей поверхности. Описываются динамическое ядро, физические параметризации, численная реализация и алгоритм работы модели. Изначально модель использовалась для вычислительно несложных задач. В данной работе на фоне необходимости применения модели в вычислительно сложных задачах проводится оптимизация последовательной программной реализации модели, а также создание программных реализаций модели с использованием технологий параллельных вычислений OpenMP, MPI, CUDA. Результаты тестирования различных реализаций модели на вычислительной системе с процессором Intel Xeon E5-2697 v3 2.60GHz и графическим процессором Nvidia P100 показывают, что оптимизация наиболее вычислительно сложных блоков в последовательной версии модели позволяет сократить время выполнения на 27%, в то же время использование технологий параллельных вычислений позволяет добиться ускорения на несколько порядков. Применение OpenMP в динамическом блоке модели привело к ускорению работы блока до 4 раз, применение MPI – до 8 раз, применение CUDA – до 16 раз при прочих равных условиях. Предложены рекомендации по выбору технологии параллельного вычисления в зависимости от свойств вычислительной системы.

Ключевые слова: модель переноса частиц; лагранжев подход; параллельные вычисления; оптимизация численной модели.

Для цитирования: Варенцов А.И., Имеев О.А., Глазунов А.В., Мортиков Е.В., Степаненко В.М. Численное моделирование переноса твёрдых частиц в атмосферном городском пограничном слое с использованием лагранжева подхода: физические задачи и параллельная реализация. Труды ИСП РАН, том 35, вып. 4, 2023 г., стр. 145–164. DOI: 10.15514/ISPRAS–2023–35(4)–8.

Благодарности: Работа выполнена с использованием оборудования Центра коллективного пользования сверхвысокопроизводительными вычислительными ресурсами МГУ имени М.В. Ломоносова. Работа выполнена при частичной финансовой поддержке Российского научного фонда, гранты 21-17-00249 (проведение численных экспериментов) и 21-71-30023 (разработка параллельной реализации программного кода), и проекта Федеральной Научно-Технической Программы "Исследование процессов в пограничных слоях атмосферы, океана и вод суши и их параметризации в моделях системы Земли" в рамках программы "Совершенствование глобальной модели Земной системы мирового уровня для исследовательских целей и сценарного прогнозирования климатических изменений" (постановка задачи, обзор литературы и оптимизация модели).

Numerical Simulation of Particulate Matter Transport in the Atmospheric Urban Boundary Layer Using Lagrangian Approach: Physical Problems and Parallel Implementation

^{1,2} A.I. Varentsov, ORCID: 0000-0001-7029-6773 <aivarentsov98@gmail.com>

² O.A. Imeev, ORCID: 0009-0008-5592-0759 <ochir90@yandex.ru>

³ A.V. Glazunov, ORCID: 0000-0002-8780-3513 <and.glas@gmail.com>

^{1,3} E.V. Mortikov, ORCID: 0000-0002-9683-5701 <evgeny.mortikov@gmail.com>

^{1,2} V.M. Stepanenko, ORCID: 0000-0003-3033-6712 <stepanen@srcc.msu.ru>

¹ *Lomonosov Moscow State University, Research Computing Center
1, Leninskie Gory, building 4, Moscow, 119234, Russia*

² *Obukhov Institute of Atmospheric Physics of Russian Academy of Sciences
3, Pzhyvovskiy pereulok, Moscow, 119017, Russia*

³ *Marchuk Institute of Numerical Mathematics of the Russian Academy of Sciences
8, Gubkin str., Moscow, 119333, Russia*

Abstract. This paper presents the results of the development of a numerical model of the Lagrangian particle transport and the application of parallel computation methods to increase the efficiency of the software implementation of the model. The model is a software package allowing calculations of transport and deposition of aerosol particles taking into account the properties of particles and input data describing atmospheric conditions and the underlying surface geometry. The dynamic core, physical parameterizations, numerical implementation, and algorithm of the model are described. Initially, the model has been used for computationally low-intensive problems. In this paper, given the need to use the model in computationally intensive problems, we conduct optimization of the sequential software implementation of the model, as well as creation of software implementations of the model with the use of parallel computing technologies OpenMP, MPI, CUDA. The results of testing of different implementations of the model show that optimization of the most computationally complex blocks in the sequential version of the model can reduce the execution time by 27%, at the same time the use of parallel computing technologies allows to achieve acceleration by several orders of magnitude. The use of OpenMP in dynamic block of the model resulted in acceleration of block up to 4 times, the use of MPI – up to 8 times, the use of CUDA – up to 16 times, all other conditions being equal. Recommendations on the choice of parallel computing technology depending on the properties of the computing system are proposed.

Keywords: particle transport model; Lagrangian approach; parallel computing; numerical model optimization.

For citation: Varentsov A.I., Imeev O.A., Glazunov A.V., Mortikov E.V., Stepanenko V.M. Numerical simulation of particulate matter transport in the atmospheric urban boundary layer using Lagrangian approach: physical problems and parallel implementation. *Trudy ISP RAN/Proc. ISP RAS*, vol. 35, issue 4, 2023. pp. 145-164 (in Russian). DOI: 10.15514/ISPRAS-20123-35(4)-8.

Acknowledgements. The work was carried out using the equipment of the Center for Collective Use of Ultra-High-Performance Computing Resources of Lomonosov Moscow State University. The work was supported in part by the Russian Science Foundation, grant no. 21-17-00249 (numerical experiments), grant no. 21-71-

30023 (parallel implementation of the program code), and by the Federal Scientific and Technical Program, project "Investigation of Atmospheric, Ocean and Land Water Boundary Layer Processes and Their Parameterization in Earth System Models" under the program "Improvement of the World-Level Global Earth System Model for Research Purposes and Scenario Prediction of Climate Change" (problem statement, literature review and model optimization).

1. Введение

Представленная работа посвящена развитию численной модели лагранжева переноса частиц в геофизических пограничных слоях, её применению в задачах различной вычислительной сложности, оптимизации программной реализации, а также применению методов параллельных вычислений для сокращения времени работы модели.

Актуальность разработки и развития подобной модели обусловлена необходимостью совершенствования прогноза качества воздуха для здравоохранения и экономики, а также разработки исследовательских инструментов, среди которых ключевую роль играет физико-математическое моделирование.

Существует два основных подхода к моделированию переноса частиц, на основе которых создано большинство современных численных моделей. При лагранжевом подходе рассчитываются координаты и скорость для каждой отдельной частицы, могут быть явно учтены все действующие на частицу силы [1; 2]. При эйлеровом подходе производится расчёт переноса и диффузии концентрации частиц, например, в ячейках расчётной сетки [3-5]. В данной работе для создания модели выбран лагранжев подход из-за более явного учёта сил и большей информативности.

Модели лагранжева переноса частиц имеют крайне широкий спектр применения. С одной стороны, явный учёт сил, действующих на частицу, и возможность вычисления траектории каждой частицы позволяют использовать такие модели для изучения вклада в перенос отдельных сил, явлений и эффектов с отслеживанием всего одной или небольшого числа частиц. С другой – при использовании достаточно большого числа частиц можно оценивать различные статистики на их множестве, например пространственное распределение в форме концентрации частиц в ячейках расчётной сетки. Это позволяет применять модели лагранжева переноса в задачах исследования переноса твёрдых и жидких частиц внутри городской среды, когда одновременно важно учесть влияние сложной структуры воздушных течений на движение частиц и получить обоснованные оценки пространственного распределения частиц внутри и над городской застройкой. Лагранжев подход позволяет решить эту задачу, однако для получения надёжных оценок требуется проведение расчётов с большим числом частиц. В то же время подробное описание сложной структуры воздушного течения внутри городской застройки требует представления данных о течении на расчётной сетке с достаточно большим числом ячеек. В сумме это означает, что подобный эксперимент будет иметь высокую вычислительную стоимость.

Помимо прямых задач, траекторные (лагранжевы) модели позволяют решать в наиболее общем виде обратные задачи, предполагающие расчёт переноса частиц в обратном времени для определения источника частиц; частным случаем такой задачи является идентификация футпринтов измерений [6-7]. Для получения статистически значимых оценок, особенно при сложной геометрии подстилающей поверхности, требуется вычисление обратного движения очень большого числа частиц.

Необходимость проведения экспериментов с высокой вычислительной стоимостью создаёт требования к численной модели – алгоритмы должны быть максимально эффективными и оптимизированными, а вычисления внутри модели имеет смысл распараллелить, так как практически любой современный компьютер имеет более одного вычислительного элемента и позволяет проводить параллельные вычисления. Некоторые современные модели лагранжева переноса имеют последовательную программную реализацию [7], однако модели широкого профиля применения, например HYSPLIT, обычно используют параллельные

вычисления [8]. Таким образом, целью данной статьи является развитие вычислительно эффективной модели лагранжева переноса частиц для параллельных архитектур.

2. Модель лагранжева переноса частиц

В данной работе описывается и развивается численная модель переноса взвешенных в воздухе частиц, в которой для описания частиц, их перемещения и взаимодействия со средой используется лагранжев подход. В данной главе приводится описание основных уравнений модели, алгоритма её работы и верификации рассчитанных при помощи модели результатов.

2.1 Физико-математическая модель

При использовании лагранжева подхода движение каждой отдельной частицы описывается уравнением, неизвестными переменными в котором являются скорость и координата частицы [1]:

$$dx_p = u_p dt, \quad (1)$$

где $u_p = (u_{p1}, u_{p2}, u_{p3})$ – скорость частицы, $x_p = (x_{p1}, x_{p2}, x_{p3})$ – её координата, t – время. Данное уравнение определяет смещение частицы.

В разработанной в рамках данной работы модели лагранжева переноса частиц рассчитываются изменения и скорости частицы, и её позиции, таким образом движение каждой отдельной частицы описывается системой из двух уравнений: уравнения (1) для позиции частицы и уравнения для её скорости. Уравнение для скорости основано на втором законе Ньютона и имеет следующий вид:

$$\frac{du_p}{dt} = \frac{g(\rho_p - \rho)}{\rho_p} + F_D(u - u_p), \quad (2)$$

где g – ускорение силы тяжести, ρ_p – плотность частицы (её материала), ρ – плотность окружающего воздуха, $u = (u_1, u_2, u_3)$ – скорость потока воздуха, F_D – коэффициент силы сопротивления среды. Система уравнений (1) – (2) дополняется начальными условиями: для координат они определяются координатами источника частиц, а вектор начальной скорости задаётся равным нулю.

Здесь не учитываются столкновения частиц, т.к. они при не очень больших концентрациях редки. Также в правых частях уравнений (1) – (2) могут быть учтены дополнительные силы (термодиффузионная, броуновская и др.), но предполагается, что данная модель будет использована для условий плотной воздушной среды в нижних слоях атмосферы и частиц, размер которых много больше длины свободного пробега и размера молекул. В таком случае плотность частиц много больше плотности воздуха, так что указанные в уравнении внешние силы становятся преобладающими.

Первое слагаемое в правой части (2) соответствует силе плавучести – результирующей сил тяжести и Архимеда, действующих на частицу. Второе слагаемое характеризует силу сопротивления среды и отвечает за увлечение частицы потоком. Такие параметры системы уравнений движения, как ускорение силы тяжести, плотность окружающего воздуха и плотность частицы, могут быть заданы явно в виде констант или импортированы из входных данных. Каждая частица может иметь собственную плотность, отличную от других. Скорость воздушного потока и коэффициент силы сопротивления рассчитываются при помощи реализованных в модели параметризаций, в том числе с использованием импортированных входных данных.

Параметризации, как упрощённые параметрические описания различных процессов, используются в разработанной модели для описания силы сопротивления среды, влияния турбулентности, распада частиц с ограниченным временем жизни, взаимодействия частиц с твёрдыми поверхностями, влияния высоких концентраций частиц на стратификацию

атмосферы. В большинстве случаев для одного и того же процесса или явления реализовано несколько параметризаций, из которых можно выбрать более подходящую к конкретной задаче.

Коэффициент силы сопротивления среды F_D в модели может быть задан одним из двух способов. Первый способ более универсален, учитывает сопротивление среды при движении сферических частиц в любом направлении и имеет следующий вид [9]:

$$F_D = \frac{3\mu C_D Re}{4\rho_p d_p^2}, \quad (3)$$

где μ – динамическая вязкость воздуха, Re – число Рейнольдса для частицы, d_p – диаметр частицы, C_D – безразмерный коэффициент сопротивления среды, рассчитываемый по эмпирическим формулам для сферических частиц [10]:

$$C_D = a_1 + \frac{a_2}{Re} + \frac{a_3}{Re^2}, \quad (4)$$

где значения коэффициентов a_1 , a_2 , a_3 зависят от диапазона значений числа Рейнольдса для частицы (8 диапазонов в интервале значений Re от 0 до 50000).

Число Рейнольдса для частицы рассчитывается следующим образом:

$$Re \equiv \frac{\rho_d p |u_p - u|}{\mu} \quad (5)$$

Второй способ описания силы сопротивления среды F_D имеет более простой вид и является следствием закона Стокса для крупных падающих частиц [11-12]. Это предельный случай формулы (3) при малых значениях Re и размера частиц. Он был введён в модель для согласованности формулировки задачи с известными эйлеровыми аналитическими решениями в условиях, когда рассматривается движение снежных частиц в статистически однородном по горизонтали потоке [13]. При данном подходе F_D определяется следующей формулой:

$$F_D = \frac{18\mu}{\rho_p d_p^2} \quad (6)$$

В экспериментах, описанных далее в данной работе, использовалась формула (6) для соответствия условиям движения частиц в модуле лагранжева переноса вихререзрывающей модели ИВМ РАН.

Скорость воздушного потока может сильно меняться во времени и пространстве, в связи с чем она обычно вносит основной вклад в изменение скорости частиц и определяет характер их распространения. Источником данных о скорости потока могут быть результаты измерений или расчётов гидродинамических моделей, аналитические уравнения. Однако такие данные – это аппроксимация реального потока, они либо являются результатом осреднения, либо имеют определённый шаг расчётной сетки – процессы масштабом меньше этого шага явно не воспроизводятся. Аэрозоли имеют размер значительно меньше шага сетки входных данных и могут продолжительное время находиться внутри одной ячейки, попадая под влияние подсеточных вихрей, поэтому важен учёт влияния турбулентности на движение частиц. В данной модели влияние турбулентности учитывается путём представления скорости потока в виде следующей суммы:

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}', \quad (7)$$

где \mathbf{u} – полная скорость потока, $\bar{\mathbf{u}}$ – осреднённая (например, по Рейнольдсу, пространственному фильтру или по времени) скорость потока, \mathbf{u}' – пульсационная (подсеточная) составляющая. Средняя по времени компонента считывается из входных данных о средней скорости потока или задаётся аналитически, вторая рассчитывается отдельно при помощи турбулентных параметризаций.

В качестве турбулентных параметризаций реализовано две стохастических модели: одна 0-го порядка и одна 1-го порядка.

Представителем класса параметризаций 0-го порядка является модель случайных смещений

[5, 14]. Важным достоинством её использования является наличие эквивалентных в терминах концентрации эйлеровых подходов [15-16]. Это позволяет провести верификацию траекторной модели на доступных аналитических или численных решениях эйлеровой модели. Пульсация скорости рассчитывается следующим образом:

$$u'_i = \frac{\partial K_s}{\partial x_i} + \frac{\sqrt{2K_s}\xi_i}{dt}, \quad \sigma_{\xi_i} = \sqrt{dt}, \quad (8)$$

где $\xi_i, i=1, \dots, 3$, – независимые дельта-коррелированные по времени гауссовы случайные величины, σ_{ξ_i} – среднеквадратическое отклонение, dt – приращение времени, x_i – i -ая эйлерова пространственная координата, K_s – коэффициент турбулентной диффузии.

Модель случайных смещений является стохастической моделью нулевого порядка – пульсационная компонента генерируется на каждом шаге по времени и не зависит от своих предыдущих значений, то есть турбулентное движение описывается последовательностью случайных смещений. В таком случае автокорреляция между смещениями отсутствует, хотя подсеточные вихри могут быть достаточно крупными, чтобы влиять на одну и ту же частицу в течение нескольких последовательных шагов по времени. Этот эффект учитывается более сложными стохастическими моделями первого порядка, ярким примером которых является модель Ланжевена. В ней турбулентное движение – это последовательность случайных затухающих приращений скорости, такой подход обеспечивает наличие автокорреляции смещений, а сила этой автокорреляции зависит от коэффициентов затухания.

На основе уравнения Ланжевена была реализована турбулентная параметризация, относящаяся к стохастическим моделям первого порядка. В итоге было использовано следующее уравнение [17]:

$$du'_i = -\frac{1}{2}b^2 \frac{u'_i}{\sigma_{u_i}^2} dt + b\xi_i, \quad (9)$$

где $b^2 = C_0\varepsilon$, $\sigma_{u_i}^2$ – дисперсия скорости потока, C_0 – постоянная Колмогорова, ε – скорость диссипации турбулентной кинетической энергии.

В данной параметризации, в отличие от модели случайных смещений, алгебраическим соотношением определяется не сама пульсационная компонента скорости, а её приращение, что позволяет считать параметризацию стохастической моделью первого порядка.

В модель включена возможность учёта распада частиц – это актуально для биоаэрозолей, радиоактивных частиц и некоторых химически активных соединений (в форме аэрозолей), то есть для частиц, имеющих ограниченное время жизни. В реализованном алгоритме время жизни частицы представлено в виде периода полураспада, вероятность распада частицы в течение шага по времени Δt определяется по следующей формуле:

$$P_{decay}(\Delta t) = \frac{N_t - N_{t+\Delta t}}{N_t} = 1 - 2^{-\frac{\Delta t}{\tau}}, \quad (10)$$

где N_t – число частиц в момент времени t , N_0 – начальное число частиц, τ – период полураспада частиц.

Для проведения расчётов в условиях городской застройки или идеализированной геометрии, отличной от плоской поверхности, в разработанной модели заданы два вида взаимодействия частиц с твёрдыми поверхностями: при контакте частицы с такой поверхностью происходит или её прилипание, или упругий отскок (с сохранением продольной компоненты импульса и заменой нормальной компоненты на противоположное значение) – выбор зависит от настроек для конкретной поверхности. Реализовано два вида задания твёрдых поверхностей. Первый вид позволяет задавать поверхности в виде выпуклых четырёхугольников произвольных размера и ориентации. Такой подход обеспечивает возможность задания геометрически сложных поверхностей, но требует относительно больших вычислительных затрат на проверку столкновений частиц с этими поверхностями. Второй вид задания твёрдых препятствий описывает не поверхности, а твёрдые ячейки прямоугольной расчётной

сетки, занятые зданиями или другими непроницаемыми объектами. Такой подход активно используется в гидродинамических моделях и позволяет гораздо быстрее производить расчёты в случаях простой ортогональной геометрии, в том числе аппроксимирующей реалистичную городскую застройку [18].

2.2 Численная реализация

Для получения траектории каждой частицы необходимо решить уравнения (2) и (3). Это возможно путём прямого интегрирования и получения полуаналитических решений [9]:

$$\mathbf{u}_p^{n+1} = \mathbf{u}^n + e^{-\frac{\Delta t}{\tau_D}}(\mathbf{u}_p^n - \mathbf{u}^n) - \mathbf{a}\tau_D(e^{-\Delta t/\tau_D} - 1), \quad (11)$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \Delta t(\mathbf{u}^n + \mathbf{a}\tau_D) + \tau_D(1 - e^{-\Delta t/\tau_D})(\mathbf{u}_p^n - \mathbf{u}^n - \mathbf{a}\tau_D), \quad (12)$$

$$\mathbf{a} = \frac{\mathbf{g}(\rho_p - \rho)}{\rho_p}, \quad (13)$$

$$\tau_D = \frac{1}{F_D}, \quad (14)$$

где индекс n обозначает значение переменной в положении частицы в момент времени t_n , индекс $n+1$ – в положении частицы в момент t_{n+1} , \mathbf{a} – в общем случае суммарное ускорение всех сил, кроме силы сопротивления среды (в данном случае – только силы плавучести), τ_D – характерное время силы сопротивления среды.

Полуаналитическая схема (11) – (14) становится точной при равенстве скоростей частицы и потока, когда пропадает сила трения. Но это не соблюдается при большом шаге по времени или нарушении гидродинамического равновесия частицы с воздушным потоком, что не позволяет применять данную схему в условиях сложных течений и тяжелых инертных аэрозолей, так как частицы будут постоянно выходить из состояния равновесия со средой.

Существуют конечно-разностные схемы, более применимые для решения этой проблемы. В ранних версиях модели использовался явный метод Рунге-Кутты 4-го порядка, однако на текущий момент он заменён на более вычислительно эффективную схему Кранка-Николсон для скорости и координаты частицы. Уравнения для используемой конечно-разностной схемы имеют следующий вид:

$$\mathbf{u}_p^{n+1} = \mathbf{u}_p^n + \Delta t \left(\mathbf{a} + \frac{1}{\tau_D} \left(\frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2} - \frac{\mathbf{u}_p^n + \mathbf{u}_p^{n+1}}{2} \right) \right), \quad (15)$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \frac{1}{2} \Delta t (\mathbf{u}_p^n + \mathbf{u}_p^{n+1}), \quad (16)$$

где \mathbf{a} предполагается константой, так как предполагается использование модели при условиях, когда плотность воздуха неизменна во времени и много меньше плотности частиц.

2.3 Алгоритм работы модели

В разработанной модели лагранжев подход применяется к частицам, при этом характеристики воздушной среды задаются эйлеровым методом, т.е. как трёхмерные поля на дискретной сетке. Эти поля (компоненты скорости ветра и характеристики турбулентности) и их изменение со временем задаются входными данными, которые могут поступать от различных гидродинамических моделей или быть заданы аналитически. На данный момент реализовано чтение внешних входных данных из моделей ENVI-met [19] и LES-модели НИВЦ МГУ и ИВМ РАН [20-21], планируется чтение данных в формате NetCDF.

Основной алгоритм модели реализует последовательность итераций с заданным шагом по времени, ограниченную периодом моделирования. Схематично алгоритм работы модели представлен на рис. 1 и будет более подробно разобран далее. Реализация модели осуществлена на языке программирования C++, дополнительно присутствует набор скриптов

для чтения и визуализации выходных данных модели, написанный на языке программирования MATLAB.

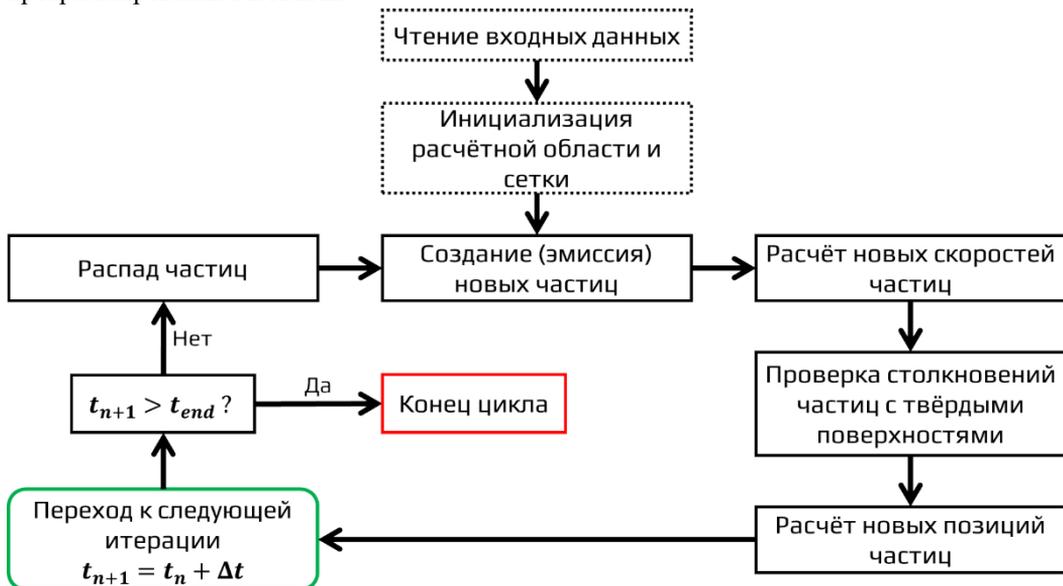


Рис. 1. Схема рабочего алгоритма модели
Fig. 1. Scheme of the model algorithm

Ключевыми частями программной реализации модели являются структуры данных, отвечающие за хранение и обработку информации о частицах, о параметрах среды (метеовеличинах) и о расчётной сетке.

Данные о частицах хранятся в объектах класса *particle* – каждый объект данного класса соответствует одной конкретной частице. Внутри класса хранятся параметры, которые могут быть индивидуальны для каждой частицы: векторы её положения в пространстве, полной скорости и пульсационной компоненты скорости, скалярные значения диаметра и плотности частицы. Также в данном классе хранятся вспомогательные параметры, необходимые для отслеживания состояния частицы или расчёта статистик: логические переменные, отвечающие за состояние частицы – активности, её осаждения, вылета за границы области, распада; скалярные переменные, отвечающие за вероятность распада частицы и момент времени её эмиссии. При необходимости в определённых задачах могут быть добавлены дополнительные переменные или же отключены некоторые из вышеописанных. Всё множество частиц, запускаемых в рамках конкретного эксперимента, описывается массивом объектов класса *particle*.

Данные о расчётной области и сетке, не считая данных о состоянии среды, хранятся в объекте класса *grid*. Данный класс включает, во-первых, множество переменных, описывающих область и сетку: координаты границ области, количество и размеры ячеек сетки, массив объектов класса *node*, отвечающих за хранение информации о состоянии среды в каждом конкретном узле (центре ячейки), и некоторые вспомогательные параметры. Во-вторых, в данном классе присутствует множество функций, описывающих взаимодействие частиц и расчётной сетки: определение ячейки, в которой находится конкретная частица, интерполяция и выдача метео данных в точке нахождения частицы, расчёт различных статистических параметров, привязанных к расчётной сетке (например, концентрация частиц в ячейках сетки).

Самым простым из трёх ключевых классов является *node* – каждый объект данного класса соответствует одной конкретной ячейке расчётной сетки и хранит в себе информацию о

пространственном положении ячейки и значении характеристик среды в данной ячейке: скорости ветра, свойствах турбулентности и др. Список характеристик среды, включённых в данный класс, может меняться в зависимости от задачи и входных данных.

Рассмотрим подробнее рабочий алгоритм модели (рис. 1). Основной подготовительный этап работы модели при запуске любого численного эксперимента – это инициализация расчётной области и сетки. На этом этапе создаётся объект класса *grid* и заполняются метеоданными ячейки сетки. Если необходимо прочитать метеоданные или характеристики самой сетки из внешнего файла, например из выходного файла гидродинамической модели, предварительно производится чтение входных данных из внешнего файла. Однако все параметры могут быть заданы и аналитически внутри кода модели, такой способ может применяться в тестовых целях или для проведения относительно простых экспериментов. После создания расчётной сетки и сохранения в её узлы исходных метеоданных могут быть рассчитаны и сохранены их производные по пространству, если это требуется. В завершение данного этапа происходит инициализация пока что пустого массива частиц (объектов класса *particle*) и вспомогательных переменных.

Далее начинается работа итерационный цикл с заданным шагом по времени. На каждой итерации, то есть на каждом шаге по времени, выполняется ряд действий над объектами класса *particle*. Первым действием является проверка распада частиц, если имеются частицы с ограниченным временем жизни. Частицы, не прошедшие проверку, то есть распавшиеся, деактивируются – далее их динамика не рассчитывается. Деактивированные частицы удаляются из массива активных частиц во избежание перерасхода памяти и проведения лишних проверок. На первом шаге по времени эта операция пропускается, так как частицы ещё не были созданы. Далее следует этап эмиссии – создания новых частиц. Если на данном шаге по времени с учётом частоты и других настроек эмиссии должны быть созданы частицы, то происходит эмиссия: создаётся определённое количество частиц, которым в соответствии с настройками эксперимента присваиваются свойства (диаметр, плотность, положение, начальная скорость и др.). В отличие от двух предыдущих, этап расчёта новых скоростей частиц обязательно происходит на каждом шаге по времени. Для каждой частицы рассчитывается новая скорость и потенциальное новое положение частицы, но они не принимаются сразу. Проводится проверка на столкновение частицы с твёрдыми поверхностями при их наличии. Если с учётом новой скорости частица столкнётся с такой поверхностью, то происходит обработка столкновения частицы и либо расчёт новых скорости и положения (в случае отскока), либо производится деактивация частицы (в случае осаждения). Если твёрдых поверхностей в эксперименте нет или частица избежала столкновения, то отдельно рассчитывается новая позиция частицы. Завершает данный этап проверка на вылет частицы за границы расчётной области и изменение параметров частицы в соответствии с граничными условиями в случае вылета. На этом основные действия заканчиваются – происходит переход к следующей итерации либо завершение выполнения модели.

Дополнительно в течение итерации могут быть рассчитаны и сохранены в памяти либо выведены во внешние файлы различные статистики и параметры, например общее количество активных частиц или концентрация частиц в каждой ячейке (мгновенная или средняя за определённый период).

Входные данные, включающие термогидродинамические характеристики воздушной среды в определённые моменты времени и геометрические параметры подстилающей поверхности, необходимы для инициализации и интегрирования уравнений модели. В качестве таких данных принимается, во-первых, описание расчётной сетки для исследуемой области, во-вторых, количество моментов времени с внешними данными и шаги по времени между ними, в-третьих, значения различных метеовеличин в каждом узле сетки. К используемым термогидродинамическим характеристикам атмосферы на данный момент относятся:

компоненты скорости ветра по трём осям координат, турбулентная кинетическая энергия, скорость диссипации турбулентной кинетической энергии, коэффициент турбулентной диффузии.

В модели используется расчётная сетка «А» по классификации Аракавы – значения метеовеличин относятся к центрам ячеек. Расчётная сетка привязывается к декартовой системе координат и точке отсчёта в соответствии со входными данными. Так как размеры ячеек исходной сетки G_1 , на которой заданы описанные выше входные эйлеровы поля, вдоль осей координат могут различаться (например, неравномерная телескопическая сетка с увеличением вертикального размера ячеек с высотой), после чтения входных данных создаётся вспомогательная равномерная сетка G_2 . Длина её ячейки вдоль каждой оси равна наибольшему общему делителю всех длин ячеек сетки G_1 вдоль данной оси (рис. 2); это предполагает, что шаги сетки вдоль любой оси относятся друг к другу, как целые числа. В итоге каждая ячейка сетки G_1 заполняется целым числом одинаковых ячеек сетки G_2 . Это позволяет быстро определять, в какой ячейке сетки G_1 в данный момент находится частица, определив номер ячейки сетки G_2 .

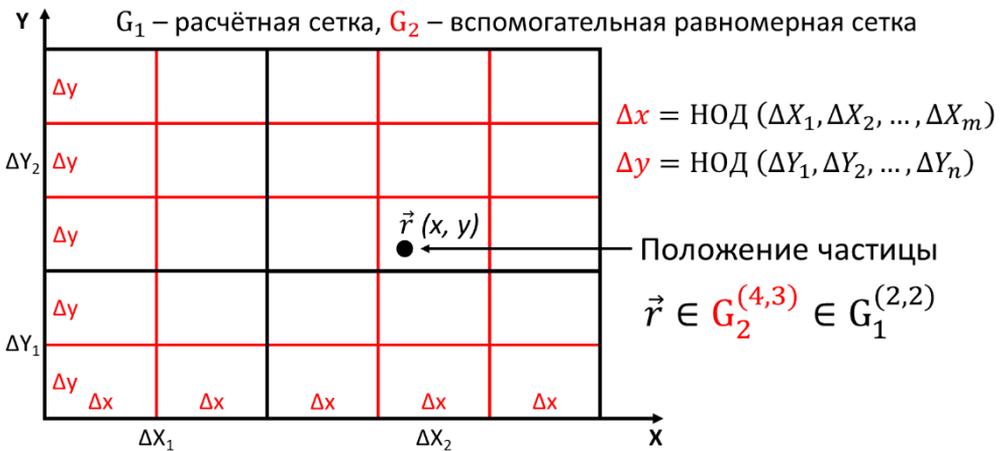


Рис. 2. Основная и вспомогательная сетки хранения данных для эйлеровых полей (НОД – наибольший общий делитель)

Fig. 2. Main and supplementary data storage grids

Значения метеовеличин для точек x_p , находящихся между узлами сетки G_1 , рассчитываются методом трилинейной интерполяции по 8-ми ближайшим узлам. Если частица находится между крайним узлом и границей области, то применяется билинейная интерполяция между 4-мя ближайшими узлами. Аналогично для граней и углов области применяется интерполяция между 2-мя ближайшими точками или использование значения из единственной ближайшей точки, соответственно. Если для метеовеличины определено граничное условие, оно будет приоритетно учтено при расчёте значения вблизи границы области.

Граничные условия для частиц в модели при отсутствии твёрдой поверхности на границе расчётной области могут иметь вид проницаемой границы, когда частицы считаются улетевшими из области (их расчёт далее не производится), вид периодической границы, когда вылетевшая наружу частица переносится и влетает в область с противоположной стороны, или вид закрытой границы, когда частица не может пересечь границу и упруго отскакивает от неё.

Основным результатом расчётов модели является вывод в отдельные файлы концентраций частиц в ячейках расчётной сетки и траекторий всех частиц. Дополнительно могут быть

выведены такие данные, как концентрация частиц, накопленных на поверхностях, параметры геометрии твёрдых поверхностей, описательная статистика.

2.4 Верификация модели

Оценку точности модели можно провести путём сравнения её результатов с эталонными данными, в роли которых обычно выступают натурно измеренные концентрации аэрозолей в воздухе или аналитические решения для пространственного распределения концентрации в идеализированных условиях.

Для верификации модели ранее было проведено сравнение модельных расчётов с эйлеровыми аналитическими решениями для пассивных частиц – маленьких лёгких частиц (обычно размером не более 2.5 мкм), для которых эффектами инерции и гравитационного оседания можно пренебречь. Были рассмотрены аналитические профили концентраций для двух идеализированных случаев: течения в нейтрально стратифицированном приземном слое и турбулентного течения Куэтта. По итогам сравнения было получено практически полное соответствие лагранжева численного и эйлерова аналитического подходов (рис. 3), что позволяет считать верификацию успешной [22].

3. Оптимизация и распараллеливание модели

3.1 Предпосылки оптимизации: вычислительно ёмкие задачи и недостатки модели

Модель изначально разрабатывалась для решения задач по расчёту переноса относительно небольшого числа частиц (от одной до нескольких десятков тысяч) в идеализированных условиях, в связи с чем эффективность и скорость работы модели не тестировались в задачах с высокой вычислительной стоимостью. Первые предпосылки необходимости оптимизации рабочего алгоритма и программной реализации модели появились при проведении экспериментов в условиях всё ещё идеализированных, но с приближенной к реальной геометрией городской застройке [22-23]. В данных экспериментах в качестве входных данных использовались результаты расчётов RANS-модели ENVI-met [19], количество ячеек сетки с метеоданными имело значение полумиллиона, а число одновременно активных частиц имело порядок тысяч. Однако количество экспериментов было небольшим (не более десяти), а время работы модели исчислялось десятками минут, что было приемлемо.

Полноценная необходимость в оптимизации модели возникла при работе с данными вихререзающих LES-моделей, в частности LES-модели ИВМ РАН [5]. В рамках задачи необходимо было сравнить особенности распространения частиц внутри и над городской застройкой при явном разрешении зданий и трёхмерного турбулентного течения в LES-модели (с внутренним модулем лагранжева переноса) и при наличии лишь осреднённых по времени вертикальных профилей метеовеличин данного течения в модели лагранжева переноса. Конфигурации городской среды, граничные условия и внешние воздействия для эйлеровой части модели были идентичны конфигурациям, описанным в работах [24-25]. Вычисления проводились в расчётных областях с размером 512 x 256 x 128 узлов равносторонней расчетной сетки с шагом 0.5 метра. Таким образом, общее количество ячеек составило уже более 16 миллионов, что на два порядка больше, чем в самых вычислительно требовательных экспериментах ранее. Возросло и количество одновременно активных частиц в модели лагранжева переноса – в зависимости от конкретного эксперимента оно достигало нескольких сотен тысяч штук. Следовательно, вычислительная стоимость экспериментов значительно возросла относительно всех ранее проводимых вычислений, что привело и к соответствующему росту времени расчёта – до порядка десятка часов на эксперимент. Такие масштабы времени вычислений уже были неприемлемы и явно показали

необходимость как в оптимизации отдельных элементов и алгоритмов модели, так и в распараллеливании её вычислений. Важной особенностью лагранжевых методов является высокий потенциал для их распараллеливания. Так как в большинстве задач не требуется рассматривать столкновения и другие взаимодействия между отдельными частицами, расчёты переноса и преобразования частиц могут быть распараллелены как по числу частиц, так и с помощью методов декомпозиции вычислительной области.

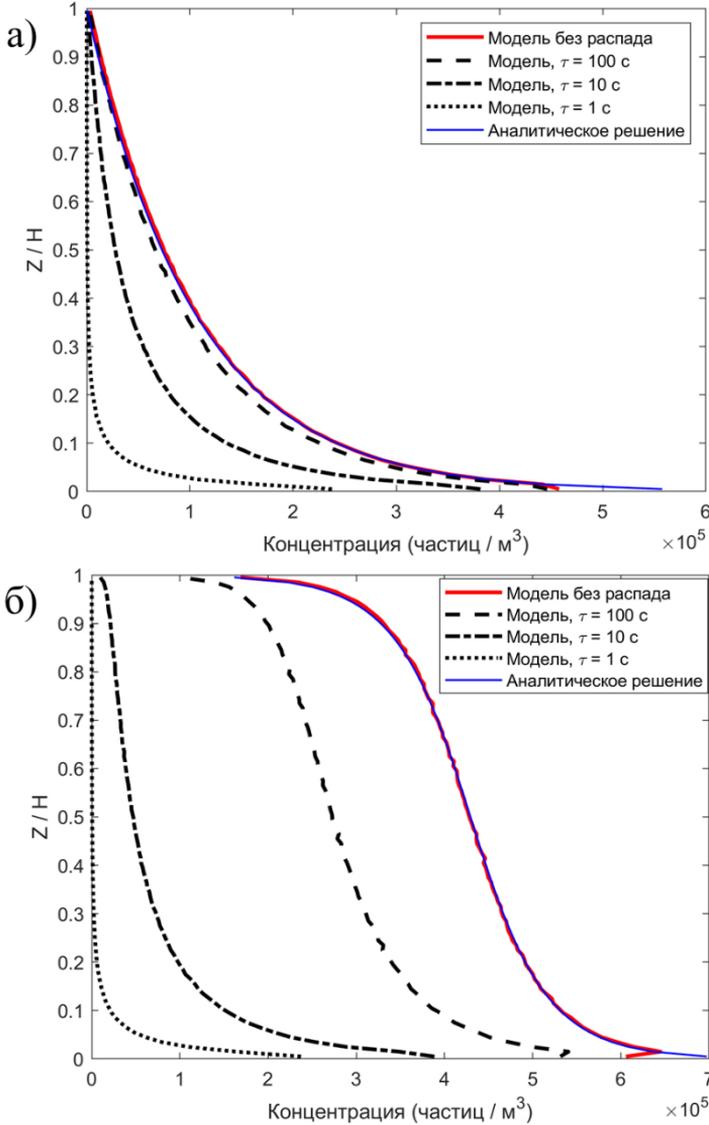


Рис. 3. Рассчитанные моделью профили концентрации частиц для случаев пограничного слоя (а) и течения Куэтта (б) без учёта конечного времени жизни частиц (красные кривые) и с его учётом (чёрные кривые), аналитические профили – синие кривые [22]

Fig. 3. Simulated particle concentration profiles for the case of a boundary layer (a) and Couette flow (b) without taking into account the finite lifetime of particles (red curve) and taking it into account (black curves), analytical profiles are blue curves [22]

3.2 Оптимизация модели и применяемые методы параллельных вычислений

Программную реализацию модели можно разделить на четыре части: этап инициализации, блок расчета динамики, обработку данных и запись результатов в файл. Блок динамики представляет собой цикл по времени, включающий создание новых частиц, обработку значений предыдущего шага, и вычисление новых значений координат и скорости частиц. В данной работе основной акцент сделан на ускорении именно блока расчёта динамики частиц с применением трёх подходов к параллельному программированию: OpenMP и MPI для расчетов на центральных процессорах и технологии CUDA для адаптации алгоритмов для архитектуры графических процессоров. Представленные далее результаты получены на суперкомпьютере Ломоносов-2. Характеристики вычислительной системы, на которой проводились эксперименты, для каждого из трёх случаев (OpenMP, MPI и CUDA) приведены в табл. 1.

Табл. 1. Характеристики вычислительных систем, использованных в экспериментах
Table 1. Characteristics of the computing systems used in the experiments

Архитектура	CPU	GPU	Число процессов
OpenMP	Intel Xeon E5-2697 v3 2.60GHz	-	14
MPI	Intel Xeon E5-2697 v3 2.60GHz	-	14
CUDA	Intel Xeon E5-2697 v3 2.60GHz	Nvidia P100	128 блоков, 64 нити

Единственной оптимизацией, проведенной для последовательной версии программного кода модели, является замена многомерных массивов на одномерные массивы длины, равной произведению всех размерностей. В программе встречается всего два таких массива: массив узлов или ячеек (объектов класса *node*) с информацией о полях метеовеличин и вспомогательный массив, использующийся при расчёте концентраций частиц в ячейках сетки. В обоих случаях изначально массивы имели четыре измерения: пространственные координаты и время. Второй массив влияет на скорость вычисления слабо на масштабе всей программы, так как обращения к нему происходят относительно редко и только при расчёте концентраций, однако его роль возрастает при увеличении интервала осреднения по времени. Зато замена первого массива с четырёхмерного на одномерный оказывает влияние на скорость работы программы на всех этапах алгоритма, в том числе и при отсутствии расчёта статистик. В итоге замена двух указанных многомерных массивов на одномерные позволила сократить время расчёта на 12-27% в зависимости от конфигурации эксперимента. Полученное ускорение достигается за счет более эффективного использования кэш-памяти и векторизации вычислений. Данная оптимизация использовалась и далее при распараллеливании вычислений модели с использованием технологий OpenMP, MPI и CUDA.

Для адаптации программы на OpenMP использовалась директива компилятора *#pragma parallel for*, примененная в функциях для расчёта координат и вектора скорости каждой частицы на следующем шаге по времени. На рис. 4 изображен график зависимости времени выполнения программного кода от модельного времени. Нелинейный характер зависимости вызван эмиссией новых частиц, которая идёт с постоянной скоростью в течение всего модельного времени. OpenMP подход ускоряет работу программы до 4 раз при использовании 14 процессов. Эффективность параллельной программы, η_r , отношение ускорения к количеству задействованных процессов, в данной задаче составляет $\eta_r \approx 2/7$. Основной причиной сравнительно небольшого ускорения является последовательная работа остальных блоков программного кода. Например, расчёты различных статистик в реализации выполняются последовательно – их распараллеливание возможно, однако в данной работе акцент был сделан на ускорении блока расчёта динамики.

На рис. 5 изображен график зависимости времени выполнения программного кода от модельного времени для OpenMP, MPI и CUDA. Использование технологии MPI позволило увеличить эффективность параллельного алгоритма – время работы относительно версии с OpenMP уменьшается примерно в 2 раза на 14 процессах. Эффективность в данном случае составляет $\eta_r \approx 4/7$.

Преимуществом реализации MPI является использование распределенной памяти, выделенной для каждого процесса, что позволяет каждому из них работать с меньшим количеством частиц и повысить эффективность обращений к памяти. Если n – количество процессов, то каждый MPI-процесс работает с массивом частиц в n раз меньшим, по сравнению с размерностью массива в OpenMP подходе. Кроме того, для OpenMP реализации необходимо отдельно обрабатывать блоки кода, которые могут привести к одновременной записи в общий, разделяемый между нитями, участок памяти. Недостатком MPI алгоритма является необходимость постоянного обмена данными между параллельными процессами.

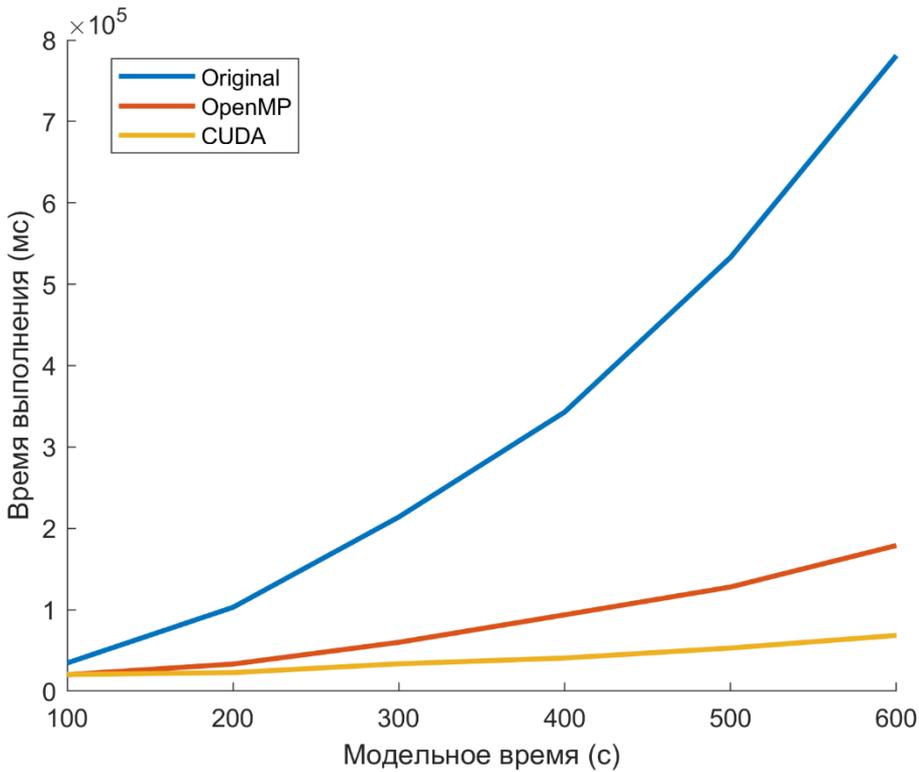


Рис. 4. Сравнение производительности последовательной версии программной реализации и параллельных версий с OpenMP и CUDA

Fig. 4. Performance comparison of the sequential version of the software implementation and parallel versions for OpenMP and CUDA

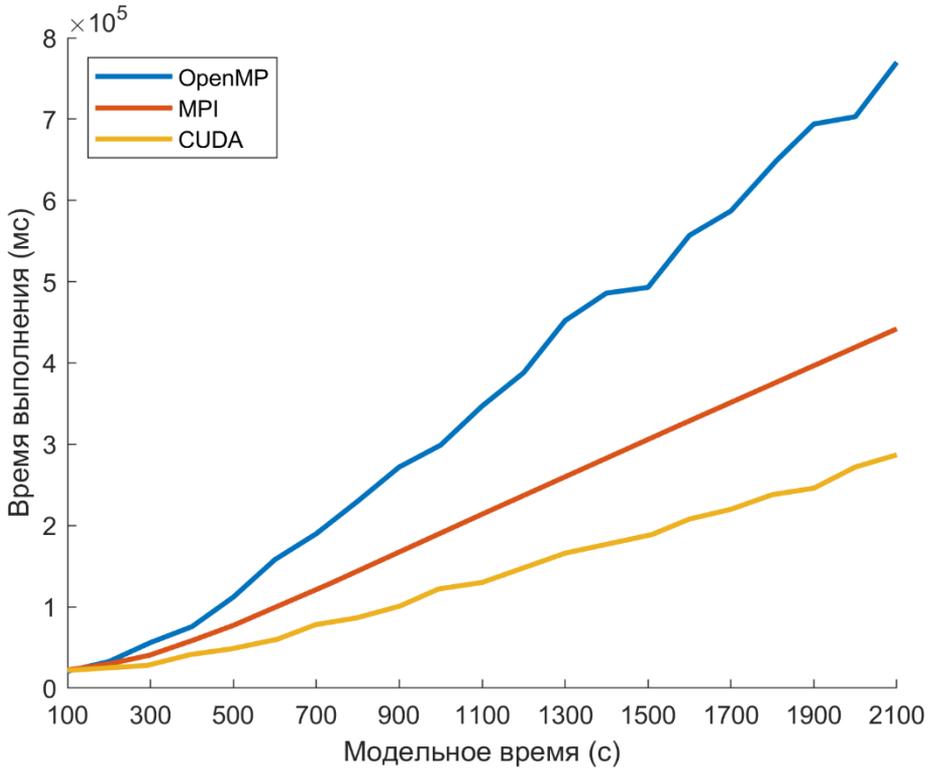


Рис. 5. Сравнение производительности программных реализаций с OpenMP, MPI и CUDA

Fig. 5. Performance comparison of software implementations with OpenMP, MPI and CUDA

Также статистическая обработка данных модели, запись результатов во внешние файлы требуют выделения дополнительной памяти. В настоящей реализации использовался синхронный алгоритм работы процессов, когда на некотором шаге все процессы, кроме первого, пересылают свои данные первому. Первый процесс их обрабатывает и при необходимости записывает результаты в файл.

Для передачи данных используются 3 рабочих массива типов `int`, `double` и `bool`, в которые записывается полное состояние системы. Работу программы можно дополнительно ускорить, если использовать асинхронный алгоритм передачи данных в блоке статистической обработки данных. Алгоритм состоит в том, чтобы на каждой итерации цикла обработку производил не один и тот же процесс, а каждый раз новый. Другими словами, на i -й итерации все процессы, кроме процессора с рангом $i \bmod n$, отдают данные для обработки и продолжают вычисления, где n – количество процессов. В тоже время, процесс $i \bmod n$ принимает и обрабатывает данные. Недостатком подобной модели является требование в n раз больше выделенной памяти для массива частиц. Текущая реализация модели с MPI требует примерно в 2 раза больше памяти по отношению к последовательной версии (первый процесс выделяет память под все частицы, в то время как остальные под $1/n$ от всех частиц), вне зависимости от количества процессов.

Работа с графическим процессором (GPU) осложнена необходимостью копирования данных между оперативной памятью и памятью GPU, а также со сложностью адаптации программного кода для подобной архитектуры. В настоящей работе для реализации на GPU была использована технология CUDA (Compute Unified Device Architecture). С помощью

данной технологии, работу всей программы удалось ускорить в 4 раза по сравнению с OpenMP или в 2 раза по сравнению с MPI (см. рис. 5). В тоже время, если анализировать только время, необходимое для работы с вычислением новых скоростей и координат частиц, включая копирование данных на GPU и обратно, ускорение достигает 8 раз по сравнению с OpenMP. Важным изменением при работе с CUDA является генерация значений псевдослучайных величин с помощью встроенной библиотеки curand. Curand позволяет создавать массивы псевдослучайных чисел большой размерности на устройстве с очень высокой скоростью. При этом время такого расчета, как правило, было ограничено лишь накладными расходами на создание структур данных и инициализацию генератора. Данная модификация позволила значительно сократить время создания вектора случайных величин ξ , использующихся при расчёте пульсационных компонент скорости частиц в формулах (8) и (9). На рис. 6 изображен график зависимости времени выполнения для блока расчета новых скоростей и координат частиц от модельного времени для OpenMP и CUDA, включая время копирования на устройство.

4. Заключение

В работе обсуждается развитие, применение и повышение эффективности численной модели лагранжева переноса частиц. Статья описывает физико-математические основы модели, её структуру и алгоритм работы. Важно, что после программной реализации модель была верифицирована и протестирована в условиях различных метеорологических условий и геометрий твёрдых поверхностей. Тем не менее, должное внимание не уделялось оптимизации модели, пока сама модель не использовалась для решения вычислительно требовательных задач. Для решения таких задач было решено провести оптимизацию структуры и алгоритмов работы модели, а также распараллеливание вычислительных процессов для ускорения расчётов.

Как показали результаты данного исследования, оптимизация наиболее вычислительно объёмных блоков модели в её последовательной реализации дало прирост в скорости вычислений, но порядок времени работы остался тем же при любом числе использованных ядер. Значительно больший рост вычислительной эффективности показало применение методов параллельных вычислений. Результаты получены при помощи вычислительной системы с процессором Intel Xeon E5-2697 v3 2.60GHz (использовалось 14 процессов) и графическим процессором Nvidia P100 (128 блоков, 64 нити), количественная оценка ускорения актуальна именно для подобной конфигурации. Использование OpenMP позволило получить ускорение в 4 раза на 14 процессах, что является хорошим результатом, но дальнейшее повышение скорости работы с данной технологией ограничено максимальным числом вычислительных элементов (процессорных ядер) в пределах одного физического процессора или узла кластера с общей памятью и не позволяет без привлечения MPI эффективно использовать доступные возможности суперкомпьютерных технологий.

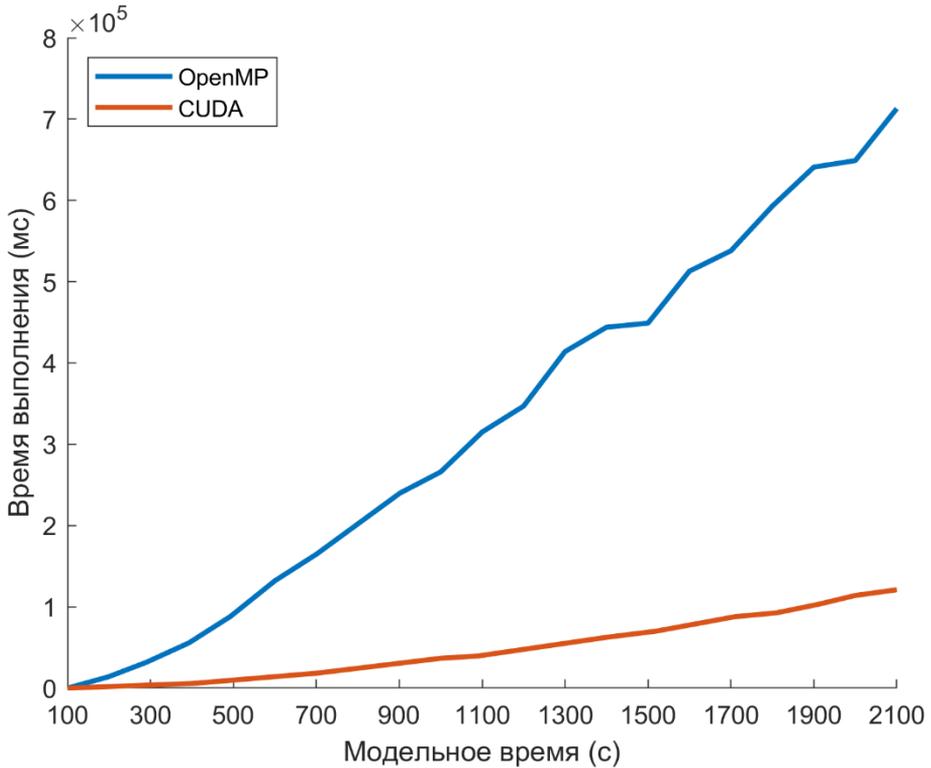


Рис. 6. Сравнение локальной производительности OpenMP и CUDA
Fig. 6. Local performance comparison between OpenMP and CUDA

В то же время данная технология не требует использования дополнительной памяти, поэтому версия модели с OpenMP может быть оптимальна для использования в системах с одним узлом процессоров с общей памятью. Версия модели с использованием технологии MPI позволила получить ускорение примерно в 8 раз относительно последовательной версии модели, что быстрее и версии с OpenMP. Версия с MPI имеет потенциал дальнейшего увеличения производительности модели при увеличении числа процессоров и может использовать потенциал суперкомпьютерных систем, однако требует выделения дополнительных объёмов памяти. Данная версия может быть оптимальна для решения задач с высокой вычислительной стоимостью, однако имеет более высокие требования к используемым вычислительным системам. Версия модели с распараллеливанием на графических процессорах показала наибольший прирост эффективности – до 16 раз относительно последовательной версии программы. Использование технологии CUDA в разработанной модели имеет наибольший потенциал среди рассмотренных технологий, однако минусом данного подхода является то, что графические процессоры, поддерживающие данную технологию, встречаются значительно реже обычных процессоров, то есть версия модели с данной технологией будет иметь строгие требования к техническим характеристикам вычислительной системы.

Список литературы / References

- [1]. Thomson D. J., Wilson J. D. History of Lagrangian Stochastic Models for Turbulent Dispersion // Geophysical Monograph Series / Lin J, Brunner D, Gerbig C, Stohl A, Luhar A, Webley P. Washington, D. C. (eds.): American Geophysical Union, 2013. pp. 19–36. doi: 10.1029/2012GM001238.

- [2]. Maronga B., Gryschka M., Heinze R., Hoffmann F., Kanani-Sühring F., Keck M., Ketelsen K., Letzel M. O., Sühring M., Raasch S. The Parallelized Large-Eddy Simulation Model (PALM) version 4.0 for atmospheric and oceanic flows: model formulation, re-cent developments, and future perspectives // *Geosci. Model Dev.* 2015, vol. 8(8), pp. 2515–2551. doi: 10.5194/gmd-8-2515-2015.
- [3]. Huttner S. Further development and application of the 3D microclimate simulation ENVI-met // 2012.
- [4]. Sofiev M., Vira J., Kouznetsov R., Prank M., Soares J., Genikhovich E. Construction of the SILAM Eulerian atmospheric dispersion model based on the advection algorithm of Michael Galperin // *Geosci. Model Dev.* 2015, vol. 8(11), pp. 3497–3522. doi: 10.5194/gmd-8-3497-2015.
- [5]. Glazunov A., Rannik Ü., Stepanenko V., Lykosov V., Auvinen M., Vesala T., Mammarella I. Large-eddy simulation and stochastic modeling of Lagrangian particles for footprint determination in the stable boundary layer // *Geosci. Model Dev.* 2016, vol. 9(9), pp. 2925–2949. doi: 10.5194/gmd-9-2925-2016.
- [6]. Auvinen M., Järvi L., Hellsten A., Rannik Ü., Vesala T. Numerical framework for the computation of urban flux footprints employing large-eddy simulation and Lagrangian stochastic modeling // *Geosci. Model Dev.* 2017. T. 10. № 11. C. 4187–4205. <https://doi.org/10.5194/gmd-10-4187-2017>.
- [7]. Simon H., Heusinger J., Sinsel T., Weber S., Bruse M. Implementation of a Lagrangian Stochastic Particle Trajectory Model (LaStTraM) to Simulate Concentration and Flux Footprints Using the Microclimate Model ENVI-Met // *Atmosphere*. 2021, vol. 12(8), pp. 977. doi: 10.3390/atmos12080977.
- [8]. Stein A. F., Draxler R. R., Rolph G. D., Stunder B. J. B., Cohen M. D., Ngan F. NOAA's HYSPLIT Atmospheric Transport and Dispersion Modeling System // *Bulletin of the American Meteorological Society*. 2015, vol. 96(12), pp. 2059–2077. doi: 10.1175/BAMS-D-14-00110.1.
- [9]. Ansys Fluent. Theory Guide 12.0. [Электронный ресурс]. URL: https://www.afs.enea.it/project/neptunius/docs/fluent/html/th/main_pre.htm (access: 01.05.2023).
- [10]. Morsi S. A., Alexander A. J. An investigation of particle trajectories in two-phase flow systems // *J. Fluid Mech.* 1972, vol. 55(02), 193 p. doi: 10.1017/S0022112072001806.
- [11]. Fletcher N. H. The physics of rainclouds. Cambridge University Press // *Quarterly Journal of the Royal Meteorological Society*. 1962, № 378 (88). C. 559–559. doi: 10.1002/qj.49708837821.
- [12]. Budd W. F. The Drifting of Nonuniform Snow Particles. Washington, D. C.: American Geophysical Union, 1966, pp. 59–70. doi: 10.1029/AR009p0059.
- [13]. Wamser C., Lykosov V. N. On the friction velocity during blowing snow // *Beitr Phys Atmosph.* 1995, vol. 68(1), pp. 85–94. <https://epic.awi.de/id/eprint/3270/>.
- [14]. Pope S. B. *Turbulent Flows*: Cambridge University Press, 2000, Issue 1. doi: 10.1017/CBO9780511840531.
- [15]. Durbin P. A. Stochastic differential equations and turbulent dispersion // NASA, 1983. <https://ntrs.nasa.gov/citations/19830014275>.
- [16]. Boughton B. A., Delaurentis J. M., Dunn W. E. A stochastic model of particle dispersion in the atmosphere // *Boundary-Layer Meteorol.* 1987, T. 40. № 1–2. C. 147–163. <https://doi.org/10.1007/BF00140073>.
- [17]. Reynolds A. M., Cohen J. E. Stochastic simulation of heavy-particle trajectories in turbulent flows // *Physics of Fluids*. 2002, vol. 14(1), pp. 342–351. doi: 10.1063/1.1426392.
- [18]. Blocken B. LES over RANS in building simulation for outdoor and indoor applications: A foregone conclusion? // *Build. Simul.* 2018, vol. 11(5), pp. 821–870. doi: 10.1007/s12273-018-0459-3.
- [19]. ENVI-met – Decoding urban nature. URL: <https://www.envi-met.com/> (access: 01.05.2023).
- [20]. Kadantsev E., Mortikov E., Zilitinkevich S. The resistance law for stably stratified atmospheric planetary boundary layers // *Q J R Meteorol Soc.* 2021, vol. 147(737). pp. 2233–2243. doi: 10.1002/qj.4019.
- [21]. Tkachenko E. V., Debolskiy A. V., Mortikov E. V. Intercomparison of Subgrid Scale Models in Large-Eddy Simulation of Sunset Atmospheric Boundary Layer Turbulence: Computational Aspects // *Lobachevskii J Math.* 2021, vol. 42(7), pp. 1580–1595. doi: 10.1134/S1995080221070234.
- [22]. Varentsov A. I., Stepanenko V. M., Mortikov E. V., Konstantinov P.I. Numerical simulation of particle transport in the urban boundary layer with implications for SARS-CoV-2 virion distribution // *IOP Conf. Ser.: Earth Environ. Sci.* 2020, vol. 611(1), pp. 012017. doi: 10.1088/1755-1315/611/1/012017.
- [23]. Varentsov A. I., Stepanenko V. M., Konstantinov P. I. High-resolution simulation of particle transport in the urban atmospheric boundary layer // *IOP Conf. Ser.: Earth Environ. Sci.* 2019, vol. 386(1), 012045. doi: 10.1088/1755-1315/386/1/012045.
- [24]. Glazunov A. V., Debolskiy A. V., Mortikov E. V. Turbulent Length Scale for Multilayer RANS Model of Urban Canopy and Its Evaluation Based on Large-Eddy Simulations // *Supercomputing Frontiers and Innovations*. 2021, vol. 8(4), pp. 100–116. doi: 10.14529/jsfi210409.

- [25]. Glazunov A., Mortikov E., Debolskiy A. Studies of Stable Stratification Effect on Dynamic and Thermal Roughness Lengths of Urban-Type Canopy Using Large-Eddy Simulation // *Journal of the Atmospheric Sciences*. 2023, vol. 80(1), pp. 31–48. doi: 10.1175/JAS-D-22-0044.1.

Информация об авторах / Information about authors

Александр Иванович ВАРЕНЦОВ – аспирант кафедры метеорологии и климатологии географического факультета МГУ имени М.В. Ломоносова. Сфера научных интересов: численное моделирование переноса аэрозолей в атмосфере, моделирование городского микроклимата и качества воздуха, изменение климата и опасные явления погоды, сезонные прогнозы.

Alexander Ivanovich VARENTSOV - postgraduate student of the Department of Meteorology and Climatology, Faculty of Geography, Lomonosov Moscow State University. Research interests: numerical modeling of aerosol transport in the atmosphere, simulation of urban microclimate and air quality, climate change and weather hazards, seasonal forecasts.

Очир Анатолевич ИМЕЕВ – студент магистратуры кафедры вычислительных технологий и моделирования факультета вычислительной математики и кибернетики МГУ имени М.В. Ломоносова. Сфера научных интересов: оптимизация вычислительных процессов, параллельные вычисления, математическое моделирование атмосферных процессов.

Ochir Anatolievich IMEEV – Master student of the Department of Computing Technologies and Modeling, Faculty of Computational Mathematics and Cybernetics, Lomonosov Moscow State University. Research interests: computational processes optimization, parallel computing, mathematical modeling of atmospheric processes.

Андрей Васильевич ГЛАЗУНОВ – доктор физико-математических наук, ведущий специалист лаборатории математического моделирования геофизических пограничных слоёв Научно-исследовательский вычислительного центра МГУ имени М.В. Ломоносова. Сфера научных интересов: математическое моделирование, численные методы, физика геофизических пограничных слоев, турбулентность, гидродинамика, моделирование климата, параллельные вычисления.

Andrey Vasilievich GLAZUNOV – Cand. Sci. (Phys.-Math). Leading Specialist of the Laboratory of Mathematical Modeling of Geophysical Boundary Layers, Research Computing Center, Lomonosov Moscow State University. Research interests: mathematical modeling, numerical methods, physics of geophysical boundary layers, turbulence, hydrodynamics, climate modeling, parallel computing.

Евгений Валерьевич МОРТИКОВ – кандидат физико-математических наук, заведующий лабораторией математического моделирования геофизических пограничных слоёв Научно-исследовательский вычислительного центра МГУ имени М.В. Ломоносова. Сфера научных интересов: математическое моделирование, численные методы, физика геофизических пограничных слоев, турбулентность, гидродинамика, моделирование климата, параллельные вычисления.

Evgeniy Valerievich MORTIKOV – Cand. Sci. (Phys.-Math.). Head of the Laboratory of Mathematical Modeling of Geophysical Boundary Layers, Research Computing Center, Lomonosov Moscow State University. Research interests: mathematical modeling, numerical methods, physics of geophysical boundary layers, turbulence, hydrodynamics, climate modeling, parallel computing.

Виктор Михайлович СТЕПАНЕНКО – доктор физико-математических наук, заместитель директора Научно-исследовательского вычислительного центра МГУ имени М.В. Ломоносова с 2019 года. Сфера научных интересов: математическое моделирование деятельного слоя и экосистем суши, математическое моделирование водоёмов и водотоков

суши, физика геофизических пограничных слоёв и турбулентности, геофизическая гидродинамика, моделирование климата, параллельные вычисления.

Viktor Mihajlovich STEPANENKO – Dr. Sci. (Phys.-Math.). Deputy Director of Research Computing Center, Lomonosov Moscow State University. Research interests: mathematical modeling of the active layer and terrestrial ecosystems, mathematical modeling of terrestrial water bodies and watercourses, physics of geophysical boundary layers and turbulence, geophysical hydrodynamics, climate modeling, parallel computing.