ЖУРНАЛ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И МАТЕМАТИЧЕСКОЙ ФИЗИКИ, 2023, том 63, № 12, с. 1993–2015

МАТЕМАТИЧЕСКАЯ ФИЗИКА

УДК 519.635

Посвящается профессору Евгению Михайловичу Шахову в связи с его 90-летием

АЛГОРИТМ ПАРАЛЛЕЛИЗАЦИИ ПО ДАННЫМ ДЛЯ МЕТОДА ПРЯМОГО СТАТИСТИЧЕСКОГО МОДЕЛИРОВАНИЯ ТЕЧЕНИЙ РАЗРЕЖЕННОГО ГАЗА НА ОСНОВЕ ТЕХНОЛОГИИ OPENMP[1)]

© 2023 г. Н. Ю. Быков[1,2,*], С. А. Фёдоров[1,2]

[1] *119333 Москва, ул. Вавилова, 44, к. 2, ФИЦ ИУ РАН, Россия*

[2] *195251 С.-Петербург, ул. Политехническая, 29, С.-Петербургский политехнический университет Петра Великого, Россия*

**e-mail: nbykov2006@yandex.ru*

Поступила в редакцию 26.06.2023 г.
Переработанный вариант 05.07.2023 г.
Принята к публикации 22.08.2023 г.

Рассмотрен алгоритм параллелизации по данным для метода прямого статистического моделирования течений разреженного газа. Выполнен анализ масштабирования производительности основных процедур алгоритма. Показано удовлетворительное масштабирование производительности параллельной процедуры индексации частиц, предложен алгоритм ускорения работы данной процедуры. На примерах решения задач о свободном потоке и обтекании конуса для 28-ядерного узла с общей памятью получено приемлемое ускорение работы всего алгоритма. Проведено сравнение эффективности алгоритма параллелизации по данным и алгоритма декомпозиции расчетной области для свободного течения. С использованием разработанного параллельного кода выполнено исследование обтекания конуса сверхзвуковым разреженным потоком. Библ. 39. Фиг. 18. Табл. 2.

Ключевые слова: метод прямого статистического моделирования Монте-Карло, параллельные алгоритмы, параллелизация по данным, OpenMP, разреженный газ, обтекание конуса.

DOI: 10.31857/S0044466923120086, **EDN:** JAJPFH

ВВЕДЕНИЕ

Решение современных задач внешней аэродинамики для условий разреженной атмосферы, разработка двигателей малой тяги космических летательных аппаратов, устройств микроэлектромеханических и наноэлектромеханических систем, а также многочисленные приложения в области вакуумных технологий требуют совершенствования методов расчета течений разреженного газа, разработки соответствующих программ и алгоритмов. Математическая модель расчета разреженных течений базируется на решении интегродифференциального уравнения Больцмана [1], [2]. Численное решение обобщенного уравнения Больцмана в общем случае произвольной геометрии задачи с физико-химическими процессами, протекающими в объеме газовой фазы, затруднительно. Данное обстоятельство связано со сложностью “точного” расчета интеграла столкновений [3], [4]. Широкое развитие получили методы решения уравнения Больцмана, основанные на возможности использования упрощенной модели интеграла столкновений [5–10]. Однако модификации таких методов для численного моделирования течений химически реагирующей газовой смеси еще находятся в стадии развития. Наиболее распространенным в настоящее время методом расчета течений разреженного газа является метод прямого статистического моделирования (ПСМ) Монте-Карло [11–13].

Метод ПСМ относится к стохастическим методам решения уравнения Больцмана. Метод предполагает моделирование движения и столкновений модельных частиц, каждая из которых

[1)]Работа выполнена при финансовой поддержке РНФ (код проекта № 22-11-00078). Вычислительные ресурсы предоставлены Суперкомпьютерным центром “Политехнический”.

представляет F_N реальных атомов или молекул ($F_N \gg 1$). Макроскопические параметры течения получаются путем усреднения параметров частиц.

Метод был предложен австралийским ученым Грэмом Бердом [14]. Метод ПСМ реализован в ряде коммерческих пакетов (например, Monaco [15], SMILE [16]), свободно распространяемых кодов [17] и исследовательских программ [18]. К достоинствам метода относится возможность моделирования трехмерных течений со сложной геометрией с учетом физико-химических процессов в объеме движущейся среды и на поверхности обтекаемых тел. К недостаткам относится вычислительная ресурсоемкость метода, прямо пропорциональная числу модельных частиц в поле течения и возрастающая с уменьшением характерного числа Кнудсена. Для преодоления данного недостатка возможна как разработка экономичных схем расчета, требующих относительно малого числа частиц в ячейке [19], [20], так и адаптация алгоритма ПСМ для расчета на компьютерных кластерах различной архитектуры. В настоящей работе рассматривается второй подход для ускорения работы алгоритма ПСМ.

Разработка высокопроизводительных алгоритмов ПСМ началась с середины 1990-х гг. и была обусловлена интенсивным развитием многопроцессорных компьютеров. Наиболее популярным методом параллелизации являлся метод декомпозиции расчетной области [21–25], однако были предложены и альтернативные подходы, связанные с декомпозицией по данным [26] и декомпозицией по статистически независимым ветвям нестационарной задачи [27–29].

В настоящее время доступные суперкомпьютерные ресурсы имеют гетерогенную архитектуру – в общем случае они включают несколько узлов и несколько многоядерных процессоров в каждом узле вместе с графическими процессорами. Гетерогенная архитектура суперкомпьютеров открывает возможности для разработки многоуровневых высокопроизводительных алгоритмов, комбинирующих различные методы параллелизации. Такая разработка требует понимания возможностей каждого из упомянутых выше методов. На взгляд авторов наименее изучены возможности подхода, ориентированного на параллелизацию по данным. Работы, содержащие детальное описание техники распараллеливания по данным, а также тестовые эталонные задачи для оценки эффективности соответствующих параллельных алгоритмов ПСМ, практически отсутствуют. Настоящая работа посвящена разработке одноуровневого алгоритма параллелизации по данным и анализу его эффективности. Развитие низкоуровневых алгоритмов важно для ускорения расчетов на вычислительных системах с общей памятью, как в случае самостоятельного использования, так и в качестве одного из уровней многоуровневого алгоритма. Также такие алгоритмы будут несомненно полезны для современных персональных компьютеров и серверов с общей памятью.

Настоящая статья организована следующим образом. В разд. 1 представлено описание классического последовательного алгоритма ПСМ. В разд. 2 приведено описание алгоритма параллелизации по данным и предложены его модификации. Анализ ускорения работы соответствующего программного кода и эффективности параллельного алгоритма представлен в разд. 3. Рассмотрены задачи о моделировании свободного потока и обтекания конуса разреженным газом. Проведено сопоставление с результатами применения алгоритма декомпозиции по расчетной области. В следующем разделе обсуждаются физические аспекты обтекания конуса разреженным сверхзвуковым потоком. В заключение сформулированы общие выводы по проведенному исследованию.

1. МЕТОД ПСМ И ВЫЧИСЛИТЕЛЬНЫЙ АЛГОРИТМ

В своей последней монографии [13] Грэм Берд предлагает следующую характеристику метода: “Метод ПСМ обеспечивает вероятностное физическое моделирование газового потока путем одновременного отслеживания движения репрезентативных модельных молекул в физическом пространстве”. Берд разрабатывал основы метода исходя из физических представлений о динамике разреженного газа и не опирался на формализованную математическую модель [11], [14]. Однако последующие исследования [30], [12] показали, что ПСМ может рассматриваться как стохастический метод решения уравнения Больцмана:

$$\frac{\partial (nf)}{\partial t} + \mathbf{c} \cdot \frac{\partial}{\partial \mathbf{r}}(nf) = \int_{-\infty}^{+\infty} \int_{0}^{4\pi} n^2 \left(f^* f_1^* - f f_1 \right) c_r \sigma d\Omega dc_1. \tag{1}$$

Здесь t – время, $\mathbf{r}$ – радиус-вектор в физическом пространстве, n – концентрация газа, f – функция распределения частиц по скоростям, для которой выполняется следующее условие нормировки:

$$\int_{-\infty}^{+\infty} f d\mathbf{c} = 1,$$

где $\mathbf{c}$ – вектор скорости, c_r – относительная скорость сталкивающихся частиц, $\sigma d\Omega$ – дифференциальное сечение столкновения, нижний индекс 1 соответствует полевым молекулам, верхний индекс * соответствует функции распределения послестолкновительных скоростей частиц. Уравнение (1) справедливо для однокомпонентного одноатомного газа и записано в отсутствие действия внешних сил. Для случая химически реагирующей газовой смеси рассматривается обобщенное уравнение Больцмана [2].

Классический вычислительный алгоритм ПСМ для расчета стационарного течения приведен на фиг. 1. Алгоритм предполагает этап предобработки, связанный с построением сетки, вычислением необходимых констант и обнулением/заполнением используемых массивов. Далее следует основной цикл, проводящий моделирование по времени. Шаги цикла можно разделить на два последовательных блока. Первая часть шагов соответствует времени установления течения t_S. После установления течения ($t > t_S$) на каждом временном шаге второго блока цикла после выполнения основных процедур осуществляется выборка параметров модельных частиц. Отличие второго блока от первого заключается только в реализации дополнительной процедуры выборки параметров (ПВП) в каждой ячейке сетки.

Внутри цикла по времени реализуются основные процедуры метода. Предполагается расщепление процессов передвижения частиц в пространстве и их столкновений на временном шаге.

Процедура передвижения частиц (ППЧ) соответствует моделированию конвективного переноса частиц согласно (1). ППЧ предполагает перебор в цикле всех частиц и перемещение каждой частицы i в пространстве за шаг Δt:

$$\mathbf{r}_i = \mathbf{r}_{0,i} + \mathbf{c}_i \Delta t;$$

здесь $\mathbf{r}_i$, $\mathbf{r}_{0,i}$ – радиус-вектор конечного и начального положений частицы соответственно, $\mathbf{c}_i$ – скорость частицы. Моделирование осесимметричного течения в методе ПСМ имеет некоторые особенности. Вычисление компонент вектора $\mathbf{r}_i$ (x_i, y_i) (y – радиальная координата) происходит согласно выражениям [11–13]:

$$x_i = x_{0,i} + u_{0,i}\Delta t,$$

$$y_i = \left\{\left(y_{0,i} + v_{0,i}\Delta t\right)^2 + \left(w_{0,i}\Delta t\right)^2\right\}^{1/2},$$

где $(x_{0,i}, y_{0,i})$ – начальные координаты частицы, $(u_{0,i}, v_{0,i}, w_{0,i})$ – начальные компоненты скорости частицы. Компоненты скорости частиц в осесимметричном течении:

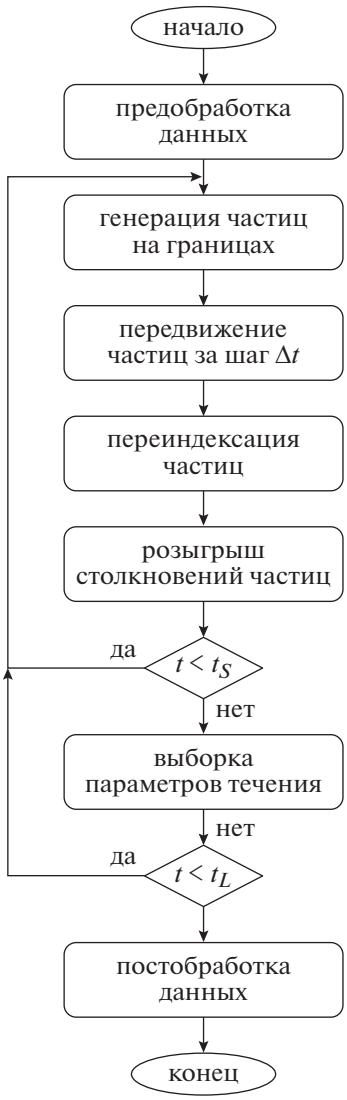
$$u_i = u_{0,i},$$

$$v_i = \left\{v_{0,i}(y_{0,i} + v_{0,i}\Delta t) + w_{0,i}^2\Delta t\right\}^{1/2}/y,$$

$$w_i = \left\{w_{0,i}(y_{0,i} + v_{0,i}\Delta t) - v_{0,i}w_{0,i}\Delta t\right\}/y.$$

Скорость частицы в ППЧ может изменяться при ее столкновении с поверхностью, поэтому ППЧ также включает установление факта пересечения частицы с поверхностью тела или границей области при передвижении на данном временном шаге. В случае факта пересечения моделируется отражение частиц от поверхности тела/границы или происходит удаление частиц из расчета.

Процедура розыгрыша столкновений частиц (ПСЧ) отвечает за моделирование процесса столкновений (правая часть (1)). ПСЧ предусматривает перебор в цикле всех ячеек сетки, в которых моделируются столкновения. В каждой ячейке выбираются пары частиц для розыгрыша парных столкновений и, в случае реализации столкновений, определяются послестолкновительные скорости частиц. Число сталкивающихся пар согласно кинетической теории зависит от плотности частиц в ячейке и от средней величины произведения относительной скорости частиц на сечение столкновения. Специфика реализации ПСЧ зависит от выбора схемы столкновений.



Фиг. 1. Классический алгоритм ПСМ для расчета стационарного течения.

В настоящей работе число столкновений в ячейке j определялось согласно модели без временного счетчика (NTC, no-time counter) [12], [13]:

$$N_c = 1/2\left(N_{p,j} - 1\right) N_{p,j} F_N \left(\sigma c_r\right)_{\max} \Delta t / V_j. \tag{2}$$

Здесь $N_{p,j}$ – число модельных частиц в ячейке, σ – сечение столкновения, индекс max означает максимально возможное значение в ячейке, V_j – объем ячейки. Выражение (2) записано для однокомпонентного газа.

Переход от ППЧ к ПСЧ требует дополнительной процедуры индексации модельных частиц (ПИЧ), так как в процессе передвижения частицы могут покинуть исходную ячейку или саму расчетную область. Основная задача ПИЧ состоит в установлении соответствия между номерами частиц и ячейками, в которых они находятся, что необходимо для выбора сталкивающихся пар частиц в ПСЧ. ПИЧ играет важную роль в масштабировании производительности и может быть реализована несколькими способами, описанными ниже.

Классическим вариантом реализации (ПИЧК) является объявление двумерного массива элементов L [2][Np], где Np – число частиц в расчетной области [11], [12]. Для простоты восприятия здесь и далее полагается, что индекс массива начинается с единицы. Используемая организация

хранения данных в массивах соответствует языку программирования C – первый индекс отвечает за номер строки, второй – за номер столбца, а хранение в памяти проводится по сплошным строкам. В строке L [1] длиной Np хранится номер ячейки для каждой частицы. Вторая строка L [2] содержит номера частиц, сгруппированные в порядке их расположения по ячейкам ("склад" номеров частиц). Например, если первая ячейка содержит две частицы с номерами 32 и 17, а вторая – три частицы с номерами 5, 64 и 89, то первые пять элементов будут равны (L [2]{32, 17, 5, 64, 89}). Дополнительно объявляется вспомогательный массив I[2][Nc], где Nc – число ячеек. Строка I[1] содержит число частиц в каждой ячейке. Строка I[2] для каждой ячейки содержит указатель на порядковое начало (смещение начального индекса) "склада" частиц, принадлежащих ячейке в строке L [2]. Для описанного выше примера (I[1]{2, 3}, I[2]{1, 3}).

ПИЧК состоит из нескольких циклов. Цикл по частицам определяет номер ячейки для каждой частицы, заполняет строку L [1]. В цикле происходит инкрементирование счетчика I[1][X], где X – номер ячейки. Второй цикл по числу частиц позволяет организовать последовательное хранение номеров частиц в ячейках в строке L [2] и также предусматривает инкрементирования счетчика I[1][X]. В ПИЧК рассчитывается и указатель I[2][X] на начало "склада" номеров частиц для ячейки X в строке L [2]. В результате работы ПИЧ для каждой ячейки известно число частиц I[1][X] в ней, порядковое начало "склада" номеров частиц, соответствующее ячейке X в строке L [2], и заполнена сама строка L [2]. Таким образом, имеется вся информация для выбора пар частиц для последующего розыгрыша столкновений в процедуре ПСЧ.

Недостатком указанной классической процедуры ПИЧК является наличие двух циклов по частицам, в каждом из которых производится инкрементирование счетчика числа частиц в каждой ячейке. При распараллеливании это инкрементирование необходимо защищать средствами синхронизации, что сказывается на масштабировании производительности алгоритма.

В настоящей работе предлагается рассмотреть также альтернативный алгоритм процедуры ПИЧ. Модификация процедуры (ПИЧМ) предусматривает резервирование одинакового размера "склада" Nw для номеров частиц в строке L [2] для каждой ячейки. Такой шаг позволяет избавиться от расчета строки L [1] с инкрементированием счетчика I[1][X], а значит, в перспективе – от необходимости синхронизации при распараллеливании. При этом теряется "сплошность" хранения всех номеров частиц в L [2]. Размер "склада" Nw должен быть равен максимально возможному числу частиц, которое может оказаться в любой ячейке в любой моделируемый момент времени. Адрес начала "склада" в строке L [2] для ячейки с номером X определяется как (X – 1) * Nw + 1. Такой подход позволяет ограничиться одним циклом по частицам для заполнения строки L [2] и для определения числа частиц в каждой ячейке I[1][X], но не исключает одну из двух процедур инкрементации элементов строки I[1]. С точки зрения использования оперативной памяти такой подход с одной стороны требует большей размерности "с запасом" массива L. Максимальный размер массива должен быть увеличен в зависимости от типа течения и возможной амплитуды изменения числа частиц в ячейках, так как нужно предусмотреть именно крайний случай – максимальное число частиц в одной из ячеек. С другой стороны, отпадает необходимость в строках L [1] и I[2]. Соответственно массивы L и I могут быть объявлены одномерными: LN[Np], IN[Nc].

Следует отметить возможность еще одной модификации классического варианта ПИЧ, связанной с передачей функции определения соответствия ячейки каждой частице (первый цикл заполнения строки L [1]), в блок передвижения частиц. Такой подход реализован в [25].

Первой процедурой основного цикла является процедура генерации частиц (ПГЧ) на границах области она следует перед ППЧ. Поток числа частиц на единичный элемент поверхности в единицу времени определяется как [11–13]:

$$F^{+} = n\left(kT/(2\pi m)\right)^{1/2}\left\{\exp\left(-s^{2}\cos^{2}\theta\right) + \pi^{1/2}s\cos\theta\left(1+\operatorname{erf}(s\cos\theta)\right)\right\}, \tag{3}$$

где m – масса частиц, T – температура газа, s – скоростное отношение ($s = c_0/(2RT)^{1/2}$, R – газовая постоянная), θ – угол между вектором среднемассовой скорости газа $\mathbf{c}_0$ и вектором нормали к граничной поверхности, k – постоянная Больцмана.

Последним блоком алгоритма является блок постобработки, который выполняется после достижения заданного времени $t_L > t_S$. В данном блоке рассчитывается концентрация, компоненты среднемассовой скорости и температуры газа в ячейке j:

$$n_j = \sum_{k=1}^{N_A} N_{p,j,k} F_N / (N_A V_j),$$
$$u_{0,j} = \sum_{k=1}^{N_A} \sum_{i=1}^{N_{p,j,k}} u_{i,j,k} \Big/ \left(\sum_{k=1}^{N_A} N_{p,j,k} \right), \tag{4}$$

$$T_{x,j} = \frac{m}{k} \left(\sum_{k=1}^{N_A} \sum_{i=1}^{N_{p,j,k}} u_{i,j,k}^2 \Big/ \left(\sum_{k=1}^{N_A} N_{p,j,k} \right) - u_{0,j}^2 \right), \tag{5}$$

где N_A – число временных шагов, используемых для выборки параметров, соответственно индекс k относится к временному шагу, на котором происходит выборка, m – масса частицы. Суммы

$$\sum_{k=1}^{N_A} N_{p,j,k}, \quad \sum_{i=1}^{N_{p,j,k}} u_{i,j,k}, \quad \sum_{i=1}^{N_{p,j,k}} u_{i,j,k}^2,$$

вычисляются в процедуре ПВП. Компоненты среднемассовой скорости $v_{0,j}$ и $w_{0,j}$ рассчитываются аналогично (4), а компоненты температуры $T_{y,j}$ и $T_{z,j}$ аналогично (5). Полная температура газа определяется как $T_j = (T_{x,j} + T_{y,j} + T_{z,j})/3$.

2. АЛГОРИТМ РАСПАРАЛЛЕЛИВАНИЯ ПО ДАННЫМ ДЛЯ УЗЛА КЛАСТЕРА С ОБЩЕЙ ПАМЯТЬЮ

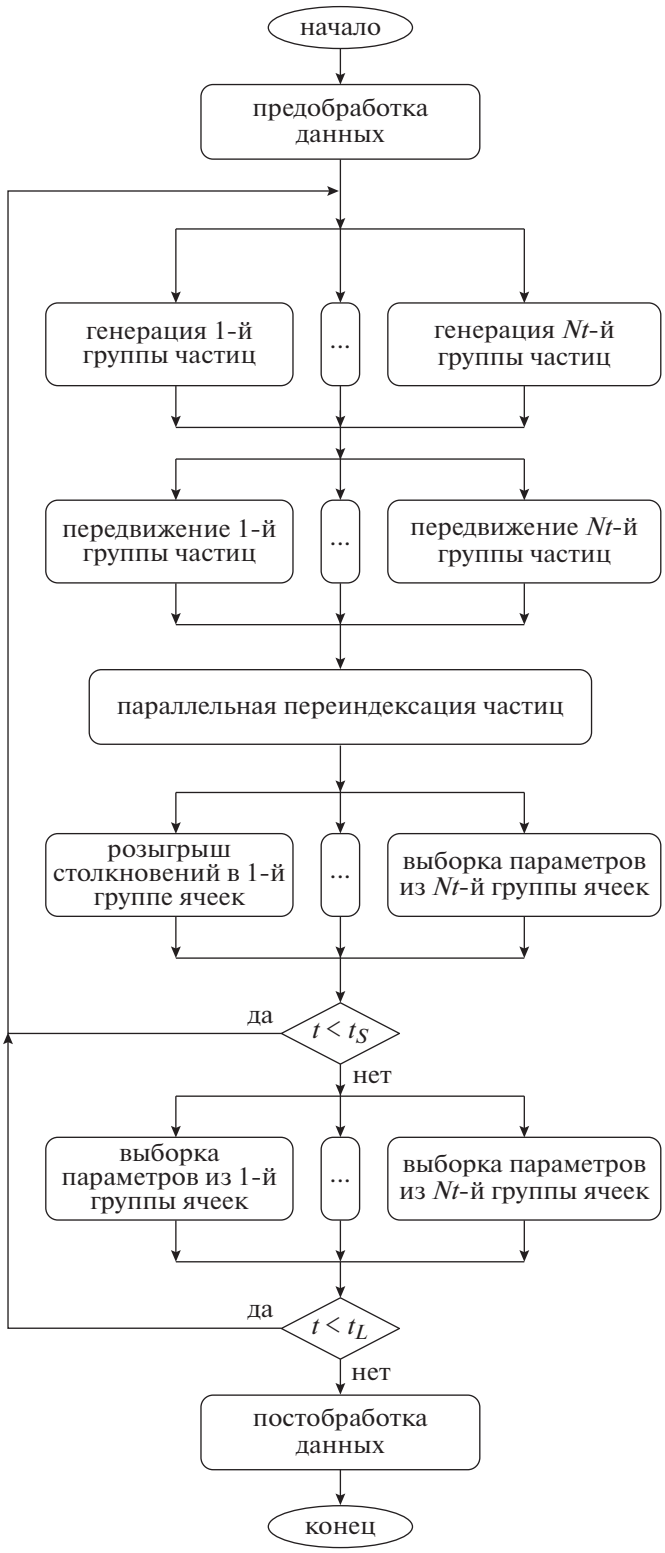
Согласно методу ПСМ распараллеливание по данным строится на следующих принципах: (i) независимости моделирования передвижения каждой частицы от передвижения других частиц, (ii) независимости моделирования процесса столкновений в каждой ячейке от столкновений частиц в других ячейках.

Версия распараллеливания алгоритма по данным представлена на фиг. 2. Предполагается параллельная работа Nt потоков (threads), каждого на отдельном ядре. Для распараллеливания используется модель параллельного программирования соединение-разветвление (fork–join) посредством технологии OpenMP.

В процедуре ППЧ реализуется параллельный расчет передвижения частиц. Каждый поток передвигает и при необходимости сталкивает с границами тел/области свою группу частиц. Основной цикл по частицам распараллелен средствами неявной многопоточности (директива parallel for) без необходимости в синхронизации по данным.

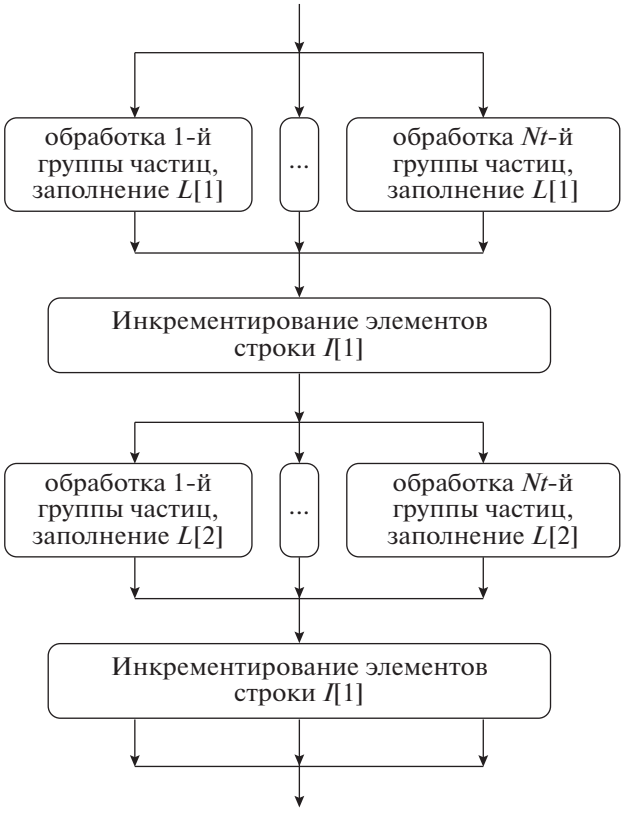
В процедуре ПСЧ реализуется параллельный расчет столкновений в ячейках. Каждый поток обрабатывает свою группу ячеек. Распараллеливание процессов столкновения частиц между потоками осуществляется средствами OpenMP без использования средств синхронизации.

Распараллеливание алгоритма требует учета специфики процедуры индексации. В случае классической ПИЧ первый цикл по частицам в этой процедуре распараллеливается стандартными средствами OpenMP по модели соединение–разветвление. Внутри цикла для каждой частицы определяется ячейка X, в которой находится частица и инкрементируется счетчик частиц в этой ячейке I[1][X]. Из-за того, что совершенно разные частицы могут оказаться в ячейке X, данное инкрементирование счетчика должно проводиться, используя средства синхронизации. Это одно из немногих “узких” мест алгоритма. Непосредственное использование для этой синхронизации директивы critical section, предлагаемой в OpenMP, приводит к неудовлетворительным результатам по масштабированию, поэтому при выполнении данной работы для улучшения масштабирования производительности применялось два подхода по организации согласованного инкрементирования указанного счетчика частиц. Первый подход состоял в использовании атомарного обновления (atomic update), а второй – в применении свертки (reduction) целиком для строки I[1] двумерного массива I, описанного выше. Первый способ задействует атомарное инкрементирование счетчка I[1][X]. Второй способ стал доступен относительно недавно после реализации в используемом компиляторе gcc стандарта OpenMP 4.5. Он подразумевает использование отдельного экземпляра строки I[1] для каждого потока без всякой синхронизации, а затем



Фиг. 2. Версия распараллеливания алгоритма по данным.

свертку всех строк со счетчиками в одну. Во втором цикле блока ПИЧК, в котором происходит заполнение строки L [1], также применялись стандартные средства OpenMP. При этом использовался атомарный захват значения счетчика I[1][X] с предварительным его инкрементированием (atomic capture). Укрупненная блок-схема этой процедуры представлена на фиг. 3.



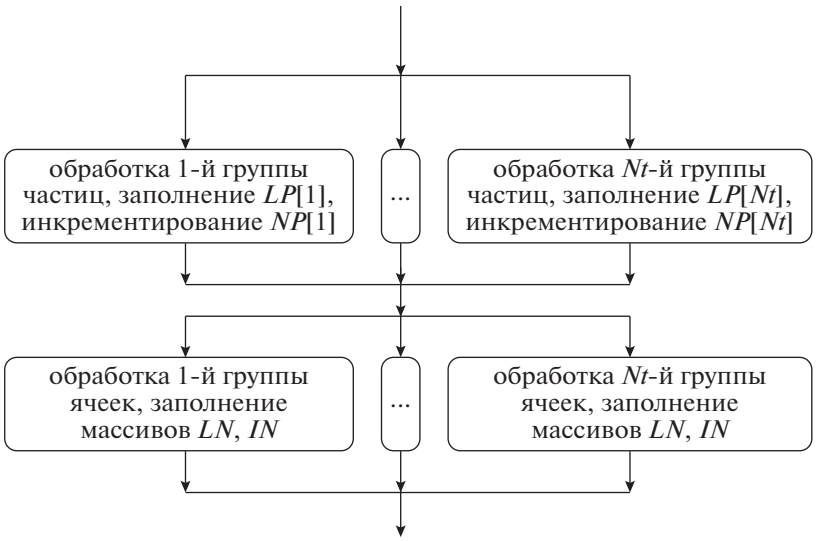
Фиг. 3. Укрупненная блок-схема процедуры ПИЧК.

В случае модифицированного варианта ПИЧМ с дополнительными массивами (далее ПИЧМД) требование по согласованному инкрементированию счетчика числа частиц в ячейке было сознательно исключено. Для этого заводятся дополнительные двумерные массивы, аналогичные массивам LN и IN, но имеющие дополнительную размерность, равную максимально возможному числу используемых вычислительных потоков: LP[Nt][Np], IP[Nt][Np], где Nt – количество используемых потоков. Каждый поток T работает со своей строкой LP[T] и IP[T]. В результате распараллеливание реализуется аналогично ПИЧК, но без всяких средств синхронизации. Как необходимость, после завершения цикла по частицам организуется дополнительный цикл по ячейкам, внутри которого происходит формирование уже единых массивов LN и IN за счет суммирования соответствующих элементов массива IP и слияния элементов LP. Данный цикл также распараллеливается с использованием модели соединение–разветвление, но тоже без всяких средств синхронизации. Указанный подход является относительно ресурсозатратным в плане использования оперативной памяти за счет хранения массивов LP[Nt][Np], IP[Nt][Np]. Увеличение размера массива LN при использовании данного подхода оказывается пропорциональным числу используемых потоков. Однако такой подход позволяет программно исключить наличие критической секции, атомарного обновления, атомарного захвата и свертки массива счетчиков – т.е. исключить необходимость в синхронизации. Укрупненная блок-схема этой процедуры представлена на фиг. 4.

Следует отметить возможность реализации упомянутой в предыдущем разделе дополнительной модификации классического варианта ПИЧ, связанной с определением соответствия ячейки каждой частице непосредственно в блоке передвижения частиц ППЧ. Однако такой подход приведет лишь к перераспределению вычислительной работы между процедурами ППЧ и ПИЧ, увеличив размер параллельного региона в ППЧ и уменьшив его размер в ПИЧ.

К вспомогательным процедурам алгоритма, которые вносят существенно меньший вклад в вычислительное время, относятся процедуры ПГЧ и ПВП.

В процедуре ПГЧ реализуется цикл по числу вбрасываемых в область частиц. Данное число, как правило, существенно меньше числа модельных частиц в области, поэтому распараллелива-



Фиг. 4. Укрупненная блок-схема процедуры ПИЧМД.

ние данной процедуры не оказывает большого влияния на масштабирование производительности алгоритма – ее вклад в общее время работы алгоритма менее 1%. Каждый поток генерирует свою часть вбрасываемых частиц. Цикл распараллелен директивой parallel for по полному числу вбрасываемых частиц на данном временном шаге. При этом инкрементирование общего числа частиц в области выполняется синхронизированно между потоками, используя атомарные операции OpenMP (omp atomic). Возможно также использование других подходов: свертки (reduction) счетчика или полного исключения синхронизации при явном разбиении цикла при параллелизации с выбором номеров вбрасываемых данным потоком частиц с шагом, равным общему числу потоков.

Последним блоком цикла по времени в ПСМ идет процедура выборки параметров ПВП. Для ее параллелизации перед циклом по ячейкам используется та же модель соединения–разветвления без синхронизации между потоками.

В результате описанной техники распараллеливания синхронизация по данным между потоками в основных процедурах задействуется только при переиндексации частиц в одной из трех версий алгоритма – ПИЧ+atomic. Алгоритм со сверткой ПИЧ+reduction и алгоритм с модифицированным блоком переиндексации ПИЧМД не требуют использования средств синхронизации при работе циклов.

Говоря об особенностях параллелизации по данным, стоит также отметить, что в начале работы кода средствами OpenMP создается пул потоков, работающий на протяжении всего моделирования. Каждый поток имеет свой генератор случайных чисел, обрабатывает в каждом блоке метода ПСМ свою часть частиц и свою часть ячеек без дополнительных проверок, связанных с номерами потоков и т.п. Авторами также был проверен подход, когда при обработке частиц каждый поток занимается очередной частицей или очередной ячейкой из соответствующего массива, например, каждой Nt-й частицей (ячейкой), где Nt – число потоков. Для этого использовалась политика планирования циклов schedule (static, 1). Такой подход не дал отличительных результатов, что можно объяснить удержанием в ядре данных для соседней частицы, которые нужны другому потоку, работающему на другом ядре.

Следует отметить необходимость в использовании потоко-безопасного генератора псевдослучайных чисел при реализации параллельного алгоритма. Применяемый генератор использует массив mk_m из 250 случайных чисел. Генерация очередного псевдослучайного числа x проводится как $x = (mk_m[mk_k])^{mk_m[mk_l]}$; mk_m[mk_k] = x, где mk_k – номер возводимого в степень числа в массиве mk_m, а mk_l – номер степени, в которую возводится это число. Процесс генерации псевдослучайных чисел определяется тройкой чисел, одно из которых – элемент массива mk_m[250], а два других – номера mk_k, mk_l. Для удовлетворения требования ПСМ по организации потоко-независимого моделирования каждая из этих троек является локальной для своего

потока. При этом начальное состояние для каждого потока разное и задается с использованием стандартного генератора псевдослучайных чисел библиотеки языка C.

3. РЕЗУЛЬТАТЫ ПРИМЕНЕНИЯ ПАРАЛЛЕЛИЗАЦИИ ПО ДАННЫМ И СРЕДСТВ OPENMP

3.1. Свободный поток

В качестве одной из тестовых задач рассматривается моделирование осесимметричного однородного потока аргона. Плотность атомов в потоке $n_0 = 3 \times 10^{21}$ м$^{-1}$, температура $T_0 = 288.15$ K, скорость $c_0 = 948.6$ м/с, число Маха $M_0 = 3$. Расчетная область представляет собой цилиндр с радиусом $R_c = 1$ см и длиной боковой поверхности $H_c = 25$ см. Вектор среднемассовой скорости $\mathbf{c}_0$ направлен вдоль оси симметрии, совпадающей с осью цилиндра.

Атомы аргона вбрасываются через левую границу области (торец цилиндра) $x = 0$, поток атомов определяется согласно (3). Функция распределения по скоростям вбрасываемых частиц соответствует максвелловской:

$$f_0 = \left(\frac{m}{2\pi k T_0}\right)^{3/2} \exp\left(-\frac{m\left\{(u - c_0)^2 + v^2 + w^2\right\}}{2kT_0}\right).$$

От верхней границы области (боковой поверхности цилиндра) $y = R_c$ частицы отражаются зеркально:

$$f_{rf}(\mathbf{c}_{rf}) = f_i\left(\mathbf{c}_{rf} - 2c_{rf,n}\mathbf{n}\right),$$

где $\mathbf{c}_{rf}$ – вектор скорости отраженной молекулы, $\mathbf{n}$ – единичный вектор нормали к границе в точке соударения, f_{rf} – функция распределения отраженных молекул, f_i – функция распределения падающих частиц. Частицы, достигающие правой границы (правого торца цилиндра) $x = H_c$, исключаются из расчета. В начальный момент времени частицы в области отсутствуют.

Реализуется 10 000 шагов по времени, выборка параметров течения происходит на последней тысяче шагов. Число модельных частиц в области на последнем временном шаге равно 400 000. Число ячеек в области 10 000, сетка зонная прямоугольная, используется одна зона с одинаковым размером ячеек. Рассмотренное течение эквивалентно течению по трубе с зеркальным законом отражения от ее стенок. Число Кнудсена, рассчитанное по радиусу “трубы”, Kn ≈ 0.04.

В расчетах используются схема столкновений NTC и модель сфер переменной твердости (VHS, variable hard sphere) для определения скоростей частиц после столкновения с параметрами: показатель степени в законе степенной зависимости вязкости от температуры $\omega = 0.81$, диаметр столкновений $d_{\text{ref}} = 4.17 \times 10^{-10}$ м при температуре $T_{\text{ref}} = 273$ K. Техника пространственных весовых множителей не применяется.

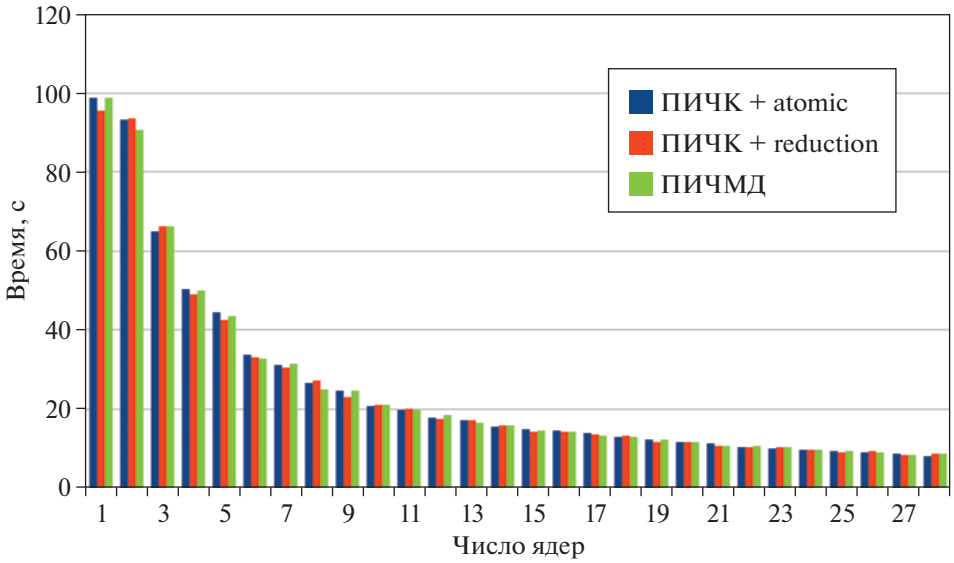
Результаты для времени расчета основного цикла последовательной версией кода ПСМ и двух вариантов реализации ПИЧ, описанных выше, представлены в табл. 1. По результатам теста можно сделать следующие выводы.

1. Реализация процедуры ПИЧ является соизмеримой по времени с процедурами ППЧ и ПСЧ. Данный вывод находится в согласии с выводами работы [31].

2. Переход от классической процедуры ПИЧК к модифицированной процедуре ПИЧМ позволяет сократить время счета на 10%.

Таблица 1. Время работы последовательного алгоритма метода ПСМ для свободного течения

Процедура	Алгоритм с ПИЧК	Алгоритм с ПИЧМ
ППЧ	98.9 с (37.2%)	98.9 с (41.3%)
ПСЧ	58.2 с (21.9%)	58.0 с (24.2%)
ПИЧ	101.7 с (38.2%)	78.5 с (32.8%)
Сумма ППЧ + ПСЧ + ПИЧ	258.8 с (97.3%)	235.5 с (98.4%)
Все время	266.0 с (100%)	239.3 с (100%)



Фиг. 5. Время работы процедур ППЧ для свободного течения в зависимости от числа ядер.

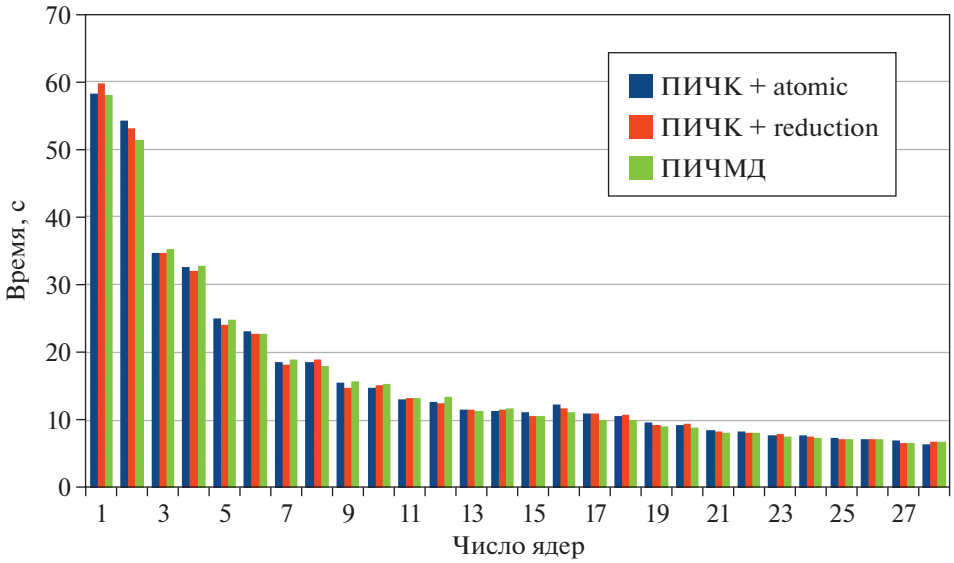
3. Процедуры ПГЧ и ПВП занимают существенно меньшее время обработки и их параллелизация носит второстепенный характер. Так, процедуры ППЧ, ПСЧ, ПИЧ занимают при моделировании более 97%.

4. Процедура ППЧ при функциональной простоте в используемой версии кода включает проверку всех частиц сеточной зоны на факт пересечения с границами и, как следствие, ее время выполнения при расчете свободного потока составляет более 30% от общего времени моделирования.

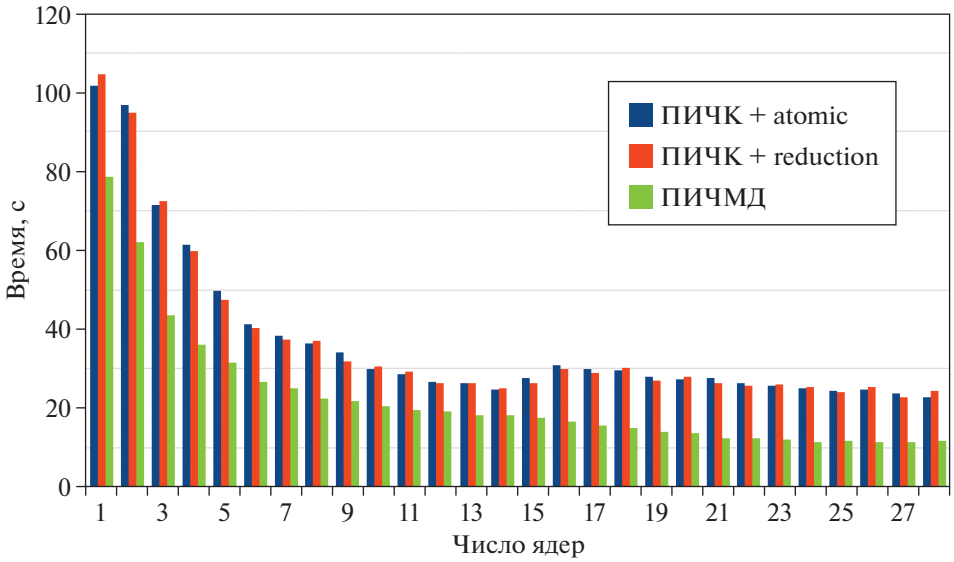
Для той же тестовой задачи проведено моделирование параллельной версией алгоритма. Для расчетов использовался один узел РСК Торнадо с общей памятью (СКЦ “Политехнический”). Узел состоит из двух процессоров Intel Xeon E5 2697 v3, каждый из которых содержит 2 контроллера памяти и 2 процессорных блока по 7 ядер в каждом. Каждое ядро выполнено по архитектуре Haswell, поддерживает гиперпоточность и расширенные инструкции AVX-2 (векторный модуль с плавающей запятой и векторное арифметико-логическое устройство). 14 ядер каждого блока имеют связанную кэш-память (СОЗУ) последнего уровня объемом 35 МБ (L3). Таким образом, в каждом узле содержится 28 ядер и 56 нитей (strands), или аппаратных потоков, для планирования на них потоков приложений. Каждый узел обеспечен 64 ГБ ОЗУ. Число используемых потоков не превышало число ядер на узле (28). Рассмотрены три версии кода: код с ПИЧК с атомарным захватом и обновлением счетчиков (далее ПИЧК + atomic), код с ПИЧК со сверткой массива счетчиков (далее ПИЧК + reduction) и предложенная модификация кода с ПИЧМ и дополнительными массивами.

Масштабирование производительности работы процедур и всего алгоритма по расчету однородного потока приведено на фиг. 5–8. Видно, что масштабирование производительности работы процедуры ППЧ является в целом приемлемым. Ускорение $S = Tp/Tp1$ (где Tp – время исполнения процедуры в параллельном коде, $Tp1$-время исполнения процедуры в последовательном коде) приблизительно равно 11 на 28 потоках, и 6 на 14 потоках. Потенциально увеличить ускорение можно за счет минимизации числа частиц, проверяемых на факт пересечения с границами области/обтекаемого тела. Ускорение процедуры ПСЧ демонстрирует чуть более слабые результаты. Такое ускорение процедур ППЧ и ПСЧ связано с удержанием совместно используемых данных в других вычислительных ядрах (например, в их кэш-памяти первого–третьего уровней). Стоит отметить, что в случае использования ПСЧ для осесимметричной задачи без техники пространственных весовых множителей на масштабирование производительности влияет неравномерность числа частиц в столкновительных ячейках, о чем указано далее.

Основной проблемой реализации алгоритма параллелизации по данным является неудовлетворительное масштабирование производительности алгоритма ПИЧК. Использование версии ПИЧК+atomic дает ускорение работы процедуры ПИЧ 4.5 на 28 ядрах и 4.1 на 14 ядрах. Исполь-



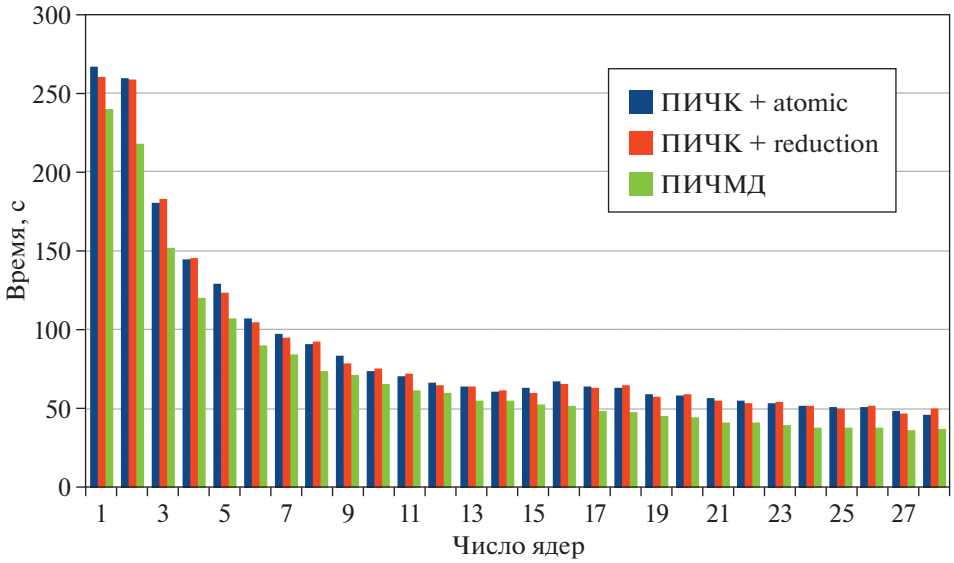
Фиг. 6. Время работы процедур ПСЧ для свободного течения в зависимости от числа ядер.



Фиг. 7. Время работы процедур ПИЧ для свободного течения в зависимости от числа ядер.

зование версии ПИЧК+reduction дает ускорение процедуры ПИЧ 4.6 на 28 ядрах и 4.2 на 14 ядрах. Эффективнее выглядит масштабирование производительности работы процедуры ПИЧМД. Прежде всего наблюдается снижение времени счета последовательного алгоритма. Кроме того, ускорение работы процедуры ПИЧ увеличивается до 4.3 на 14 ядрах и до 6.7 на 28 ядрах.

Ускорение процедур ППЧ, ПИЧ и ПСЧ предопределяет полное время работы алгоритмов. Использование версии ПИЧК+atomic дает ускорение работы всего алгоритма: 4.4 на 14 ядрах и 5.7 на 28 ядрах. Использование версии ПИЧК + reduction дает ускорение работы всего алгоритма: 4.2 на 14 ядрах и 5.2 на 28 ядрах. Наиболее предпочтительно выглядит масштабирование при использовании процедуры ПИЧМД. Как уже отмечалось, наблюдается снижение времени счета последовательного алгоритма. Ускорение работы всего алгоритма при этом увеличивается до 6.2 на 28 ядрах.



Фиг. 8. Время работы алгоритмов для свободного течения в зависимости от числа ядер.

Относительно слабое ускорение связано с недостатками самой модели параллельного программирования соединение–разветвление, когда распараллеливаются только участки кода, а не весь код (например, не сразу все указанные процедуры). Среди причин можно отметить следующие дополнительные обстоятельства:

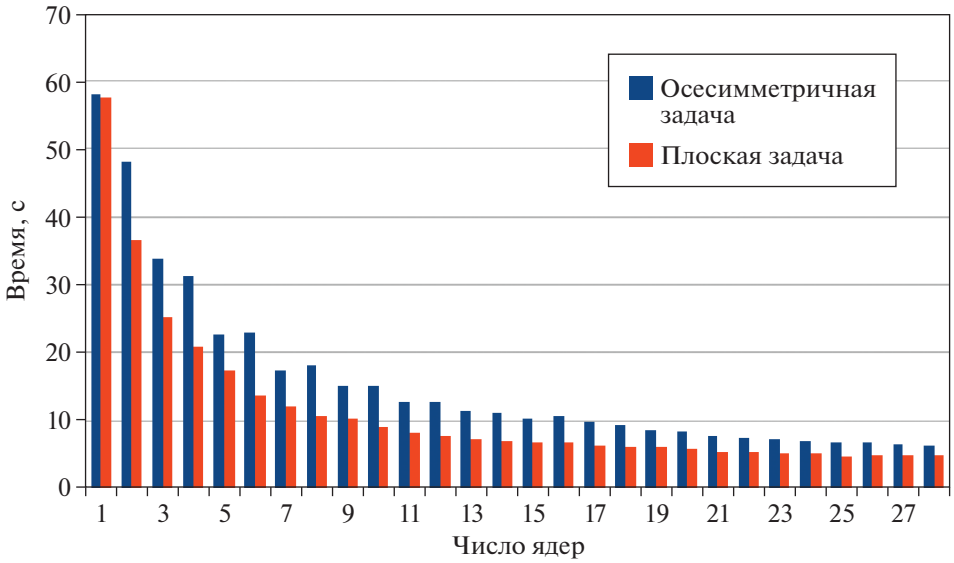
(i) параллельные регионы содержат средства синхронизации;

(ii) упомянутое ранее удержание совместно используемых данных в других вычислительных ядрах.

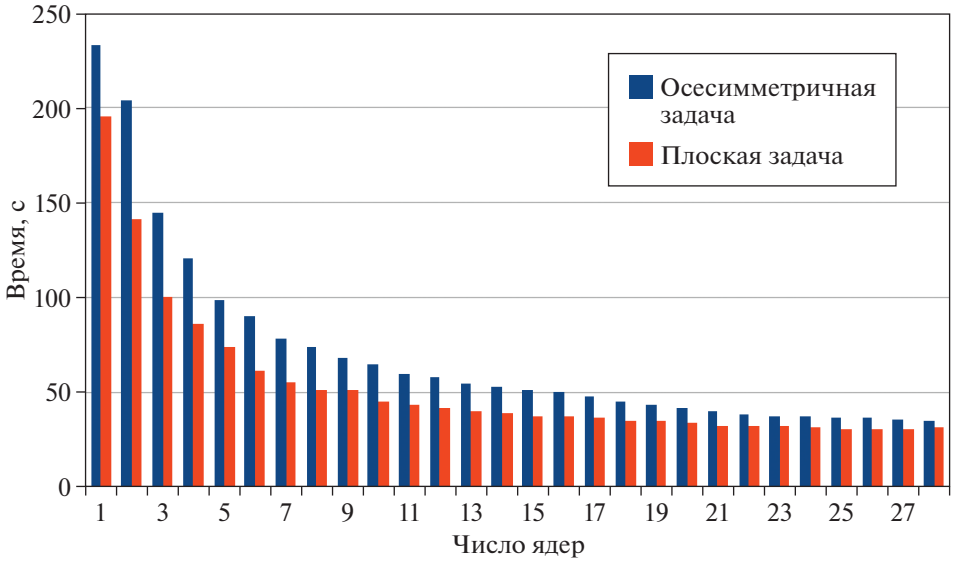
Ускорение алгоритма и, в частности, ПСЧ, несколько повышается с уменьшением числа Кнудсена из-за роста числа столкновений в ячейках. Также наблюдается незначительное повышение ускорения работы ПСЧ для алгоритма ПИЧМД (до 8.3 на 14 ядрах и до 11.5 на 28 ядрах) при переходе от осесимметричной к плоской задаче при аналогичном количестве частиц в расчетной области (фиг. 9). Данное обстоятельство связано с более равномерным распределением частиц в ячейках при моделировании плоского свободного потока. В осесимметричном случае “равномерность” распределения частиц по ячейкам может быть обеспечена за счет использования весовых пространственных множителей. Однако общее ускорение (см. фиг. 10), с учетом фактора снижения времени работы последовательного кода при расчете плоской задачи, оказывается близким к случаю расчета осесимметричной задачи (4.9 на 14 ядрах и 6.1 на 28 ядрах).

Методически важным моментом является сравнение эффективности алгоритма параллелизации по данным с алгоритмом декомпозиции расчетной области (АДО). В АДО каждое ядро “обрабатывает” свою подобласть, т.е. выполняет все основные процедуры ПСМ (ПГЧ, ППЧ, ПИЧ, ПСЧ, ПВП). Таким образом, “однотипно” распараллеливается весь код. К недостаткам АДО следует отнести необходимость на каждом временном шаге формирования массивов частиц, покидающих подобласть, и организации обменов такими массивами между ядрами.

Время работы алгоритма с параллелизацией по данным и процедурой ПИЧМД средствами OpenMP и алгоритма декомпозиции расчетной области средствами MPI в зависимости от числа используемых ядер показано на фиг. 11. В последнем случае решается тождественная задача о расчете свободного потока. Видно, что параллелизация по данным уступает в результативности АДО. Так, при задействовании 28 вычислительных ядер алгоритм с параллелизацией по данным и процедурой ПИЧМД средствами OpenMP имеет ускорение 6.36, а алгоритм декомпозиции расчетной области средствами MPI имеет ускорение 23.8. Однако здесь необходимо упомянуть главное преимущество обсуждаемого алгоритма, которое заключается в простой технике распараллеливания и минимальном изменении последовательного кода. Фактически в каждой процедуре перед соответствующим главным циклом добавляется одна строчка дополнительного кода. Вторым преимуществом является отсутствие процедур отбора частиц, покидающих подобласть и идентификации подобластей, “принимающих” частицы. В случае геометрически сложной



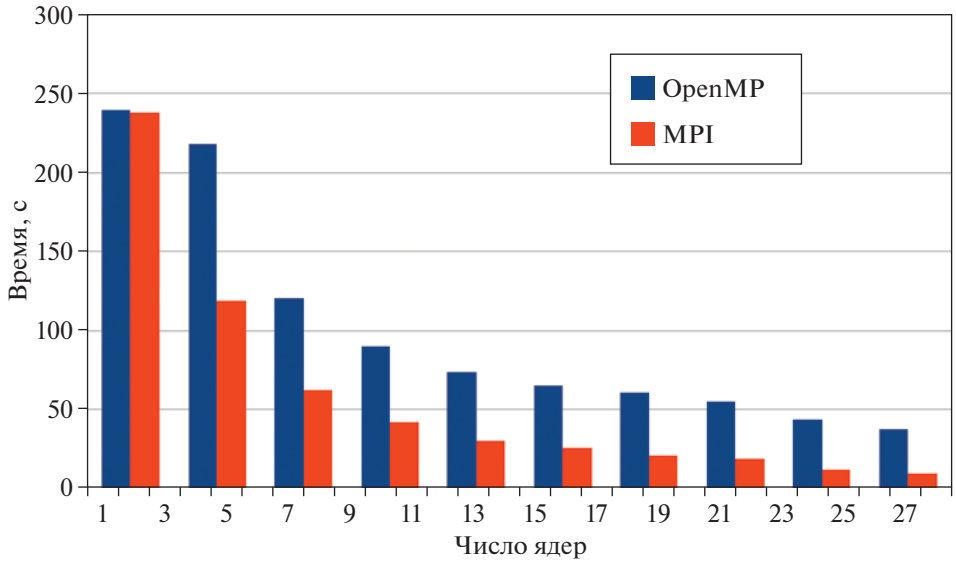
Фиг. 9. Время работы процедур ПСЧ для свободного осесимметричного и плоского течения в зависимости от числа ядер.



Фиг. 10. Время работы алгоритма ПИЧМД для свободного осесимметричного и плоского течения в зависимости от числа ядер.

расчетной области процедура передачи частиц между ядрами не является тривиальной. Наконец, в-третьих, в настоящем разделе рассматривается однородный поток, для которого балансировка загрузки работы ядер осуществляется простым, пропорциональным количеству ядер, делением расчетной области с учетом специфики осесимметричного течения. Также использование алгоритма декомпозиции по данным не требует резервирования размеров массивов, что позволяет проводить моделирование с бóльшим числом частиц. Применение АДО к реальной задаче требует разработки дополнительных алгоритмов балансировки, без которых метод декомпозиции по расчетной области показывает лишь удовлетворительные результаты.

Следует отметить, что общее время работы последовательного кода при решении рассматриваемой задачи занимает менее 15 мин, число частиц в области увеличивается от 0 до 400000. Сле-



Фиг. 11. Время работы алгоритма ПИЧМД с параллелизацией по данным (OpenMP) и алгоритма с параллелизацией по области (MPI) в зависимости от числа используемых ядер.

дует ожидать, лучших результатов по ускорению работы программы с параллелизацией по данным на реальной задаче, время последовательного выполнения которой существенно больше.

3.2. Обтекание конуса

В качестве второй тестовой задачи в данной работе рассматривалось осесимметричное обтекание острого конуса потоком аргона. Радиус основания конуса 0.045 м, высота 0.15 м, что соответствует углу полураствора 16.7°. Плотность частиц в набегающем потоке $n_0 = 6 \times 10^{20}$ м$^{-3}$, температура $T_0 = 288.15$ K, скорость потока $c_0 = 948$ м/с, число Маха $\mathrm{M}_0 = 3$. Размер расчетной области $H_c = 78$ см, $R_c = 22.5$ см.

Граничные условия аналогичны рассмотренным в предыдущем разделе. Отражение от поверхности конуса полагается диффузным с полной аккомодацией энергии, функция распределения для отраженных частиц:

$$f_{rf} = \left(\frac{m}{2\pi k T_w}\right)^{3/2} \exp\left(-\frac{m c_{rf}^2}{2k T_w}\right).$$

Температура поверхности конуса T_w равна температуре набегающего потока $T_w = T_0$.

Реализуется 12000 шагов по времени, выборка параметров течения происходит на последней тысяче шагов. Число модельных частиц в области на последнем шаге равно 20 млн. Число ячеек в области 31000, число сеточных зон 6. Число Кнудсена, рассчитанное по радиусу основания конуса $\mathrm{Kn} \approx 0.05$. Число Рейнольдса может быть оценено по формуле (см. [12]):

$$\mathrm{Re}_0 = \frac{2(5-2\omega)(7-2\omega)}{15\pi^{1/2}} \sqrt{\frac{\gamma}{2}} \frac{\mathrm{M}_0}{\mathrm{Kn}_0}. \tag{6}$$

Согласно (6), $\mathrm{Re}_0 = 75$.

Результаты для времени расчета основного цикла для последовательного кода ПСМ с двумя вариантами реализации ПИЧ, описанными выше, представлены в табл. 2. По результатам теста можно сделать следующие выводы.

1. Реализация процедуры ПИЧ является наибольшей по времени, но соизмеримой с процедурами ППЧ и ПСЧ.

2. Переход от классической процедуры ПИЧК к модифицированной процедуре ПИЧМ позволяет сократить время счета на 22.7%.

Таблица 2. Время работы последовательного алгоритма метода ПСМ для обтекания конуса

Процедура	Алгоритм с ПИЧК	Алгоритм с ПИЧМ
ППЧ	8125 с (27.4%)	5741 с (24.5%)
ПСЧ	5971 с (20.2%)	6539 с (28.0%)
ПИЧ	15524 с (52.4%)	11112 с (47.5%)
Сумма ППЧ + ПСЧ + ПИЧ	29620 с (92.9%)	23392 с (94.8%)
Все время	31890 с (100%)	24665 с (100%)

3. Процедуры ПГЧ и ПВП занимают существенно меньшее время обработки и их параллелизация носит второстепенный характер. Так, процедуры ППЧ, ПСЧ, ПИЧ занимают при моделировании более 92%.

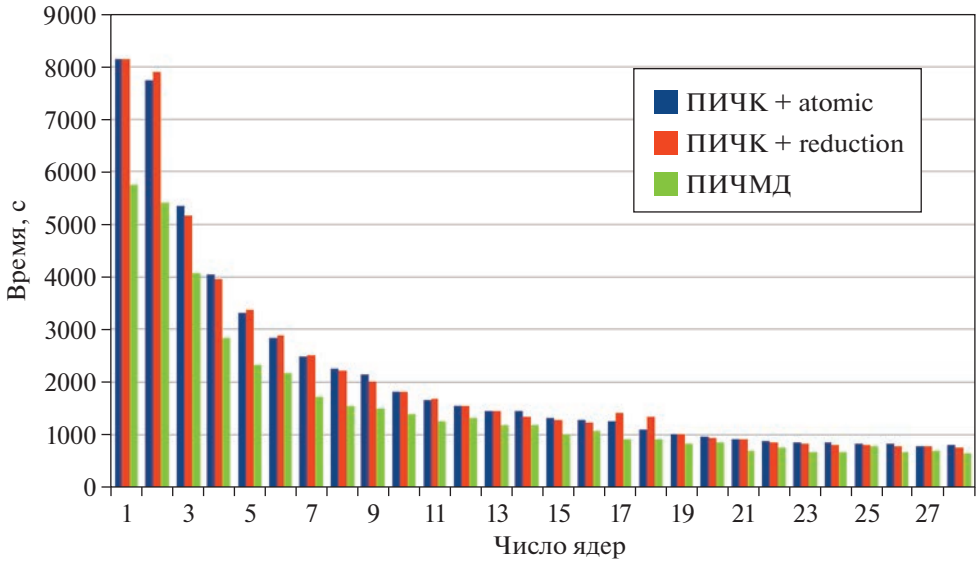
4. В процедуре ППЧ на пересечение с поверхностью конуса, а также с границами области проверяются только частицы, принадлежащие к соответствующим зонам. Как следствие, доля времени ППЧ в общем времени счета сокращается по сравнению с тестом для свободного потока.

Для той же тестовой задачи проведено моделирование параллельной версией алгоритма. Для расчетов использовался тот же узел РСК Торнадо с общей памятью (СКЦ “Политехнический”). Как и ранее, рассмотрены три версии кода по моделированию: код с ПИЧК с атомарным захватом и обновлением счетчиков (ПИЧК + atomic), код с ПИЧК со сверткой массива счетчиков (ПИЧК + reduction) и предложенная модификация кода с ПИЧМ и дополнительными массивами (ПИЧМД).

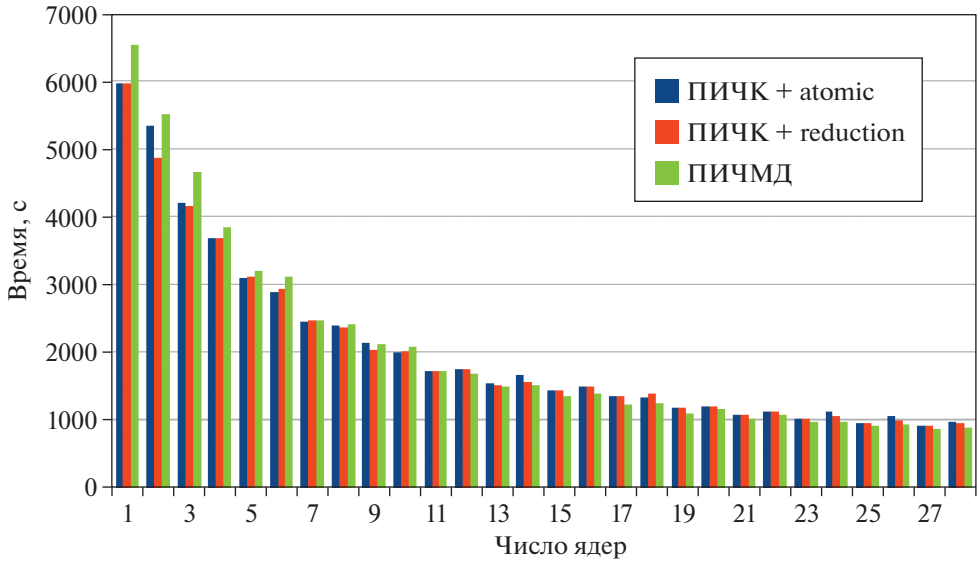
Масштабирование производительности работы процедур и всего алгоритма по расчету обтекания конуса приведено на фиг. 12–15. Видно, что масштабирование процедур ППЧ и ПСЧ является в целом аналогичным задаче о свободном потоке. Ускорение работы процедур составляет 7.4–10.2 на 28 потоках и 4.3–6.9 на 14 потоках.

Использование версии ПИЧК+atomic дает ускорение работы процедуры ПИЧ 6 на 28 ядрах и 4.7 на 14 ядрах. Использование версии ПИЧК + reduction дает ускорение процедуры ПИЧ 7.2 на 28 ядрах и 6.0 на 14 ядрах. Как и ранее, эффективнее выглядит масштабирование производительности работы процедуры ПИЧМД. Ускорение работы процедуры ПИЧ увеличивается до 4.9 на 14 ядрах и до 7.7 на 28 ядрах.

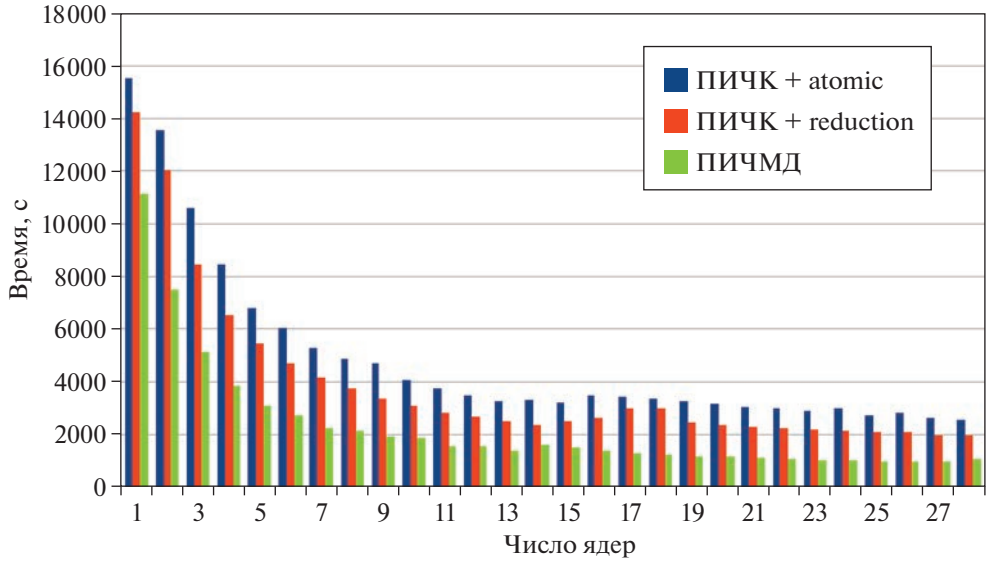
Использование версии ПИЧК + atomic дает ускорение работы всего алгоритма: 4.4 на 14 ядрах и 5.8 на 28 ядрах. Использование версии ПИЧК+reduction дает ускорение работы всего алгоритма: 5.0 на 14 ядрах и 6.9 на 28 ядрах. Наиболее эффективным является использование алгоритма



Фиг. 12. Время работы процедур ППЧ для обтекания конуса в зависимости от числа ядер.



Фиг. 13. Время работы процедур ПСЧ для обтекания конуса в зависимости от числа ядер.



Фиг. 14. Время работы процедур ПИЧ для обтекания конуса в зависимости от числа ядер.

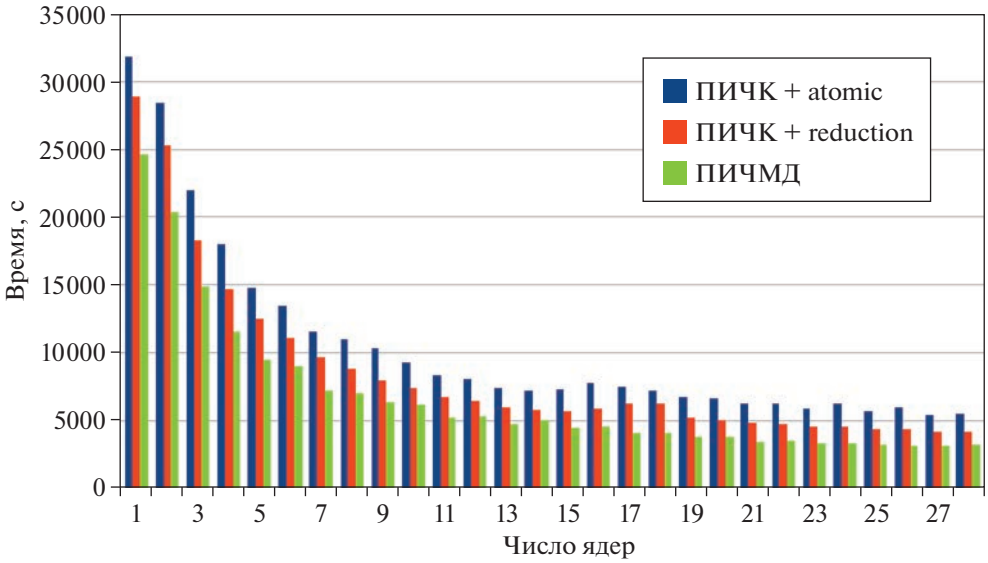
с процедурой ПИЧМД. Как уже отмечалось, наблюдается снижение времени счета последовательного алгоритма. Ускорение работы всего алгоритма увеличивается до 7.7 на 28 ядрах.

Как и ожидалось, ускорение работы кода на реальной задаче имеет чуть лучшие результаты, время последовательного выполнения которой существенно больше.

При исследовании производительности ряда процедур рассматриваемых алгоритмов можно заметить их деградацию при переходе на второй процессор и задействовании более 14 ядер. Данное обстоятельство связано с указанными ранее особенностями архитектуры узла и версии ядра Linux 3.10.0, используемой в том числе для наложения необходимых заплат, а также для поддержки файловой системы Lustre.

4. ФИЗИЧЕСКИЕ АСПЕКТЫ ОБТЕКАНИЯ КОНУСА

Обтекание тел простой формы сверхзвуковым потоком разреженного газа являлось предметом многочисленных исследований в 60–80-е гг. прошлого века. Обтекание конуса рассматри-



Фиг. 15. Время работы алгоритмов для обтекания конуса в зависимости от числа ядер.

валось в работах [32–35]. Основное внимание отводилось анализу аэродинамических характеристик конуса. Картина течения, в особенности, в донной части подробно не рассматривалась. Данное обстоятельство можно связать с резким падением плотности газа за основанием конуса и с возрастанием вычислительной сложности статистического моделирования течения в этой области. Использование высокопроизводительных алгоритмов уже позволяет провести расчет с достаточным количеством модельных частиц для изучения донного течения.

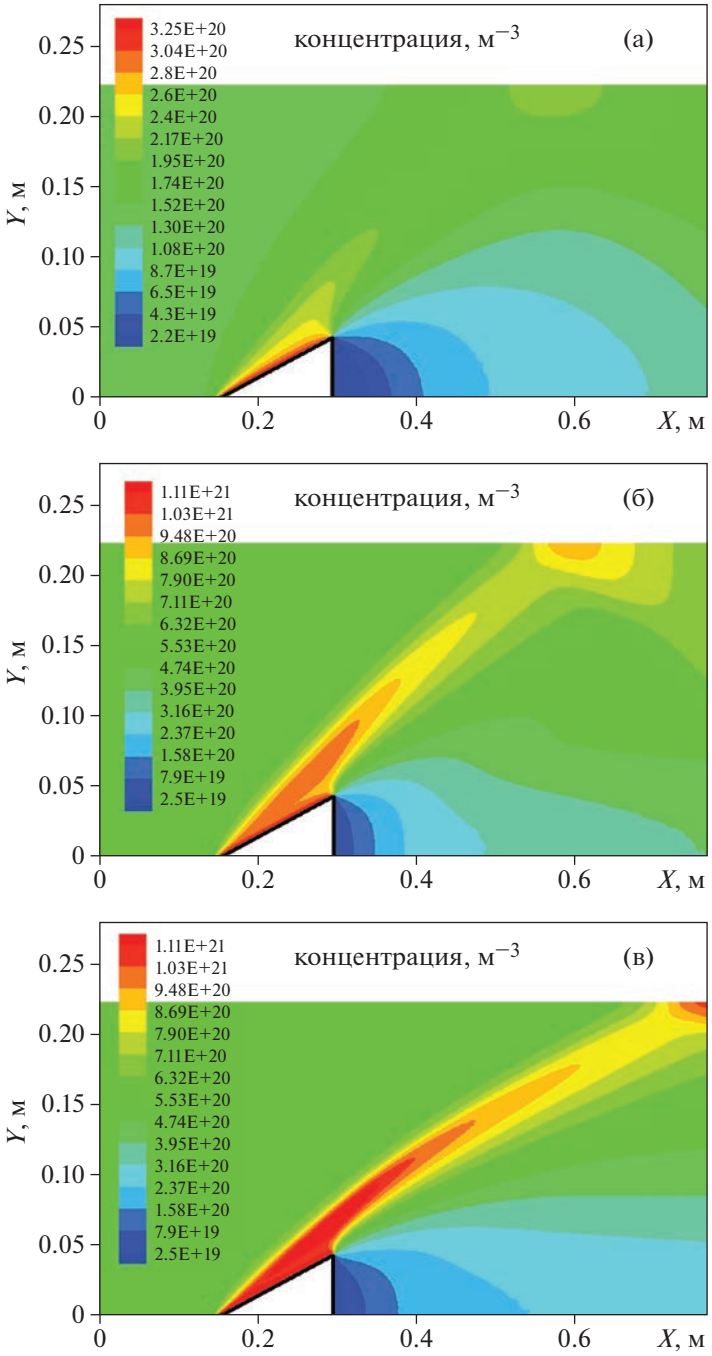
Ниже приведены результаты расчетов конуса (угол полураствора 16.7°) в постановке, приведенной в предыдущей секции, но с большим количеством ячеек (41 000) и модельных частиц (200 млн) в области. На фиг. 16, 17 приведены поля плотности и числа Маха для трех вариантов обтекания конуса, соответственно фиг. 16 а–в, 17 а–в: $\mathrm{Kn} \approx 0.2$, $M_0 = 3$ (вариант а); $\mathrm{Kn} \approx 0.05$, $M_0 = 3$ (вариант б); $\mathrm{Kn} \approx 0.05$, $M_0 = 5$ (вариант в).

При числе Кнудсена $\mathrm{Kn} = 0.2$ ударная волна сильно размыта и имеет наибольшую толщину. Интенсивность ударной волны быстро снижается с удалением от вершины конуса. На верхней границе ставились условия симметрии (зеркальное отражение атомов), координата отражения ударной волны от верхней поверхности $X_r \approx 0.55$ м. Течение между основной ударной волной и отраженной от верхней границы ударной волной ускоряется, но не достигает числа Маха на бесконечности. В донной части плотность газа падает примерно более чем в 10 раз по отношению к плотности набегающего потока. Течение в донной области является существенно дозвуковым ($M < 0.2$).

С уменьшением числа Кнудсена толщина ударной волны уменьшается, а интенсивность с удалением от вершины конуса снижается не так быстро, как для $\mathrm{Kn} = 0.2$. Несмотря на уменьшение толщины УВ, ее структура остается размытой. Координата отражения УВ от верхней границы $X_r = 0.6$. За ударной волной поток ускоряется практически до числа Маха на бесконечности. Область дозвукового течения в донной области несколько возрастает.

Увеличение числа Маха с 3 до 5 в варианте расчета при сохранении числа Кнудсена приводит к смещению координаты отражения УВ от верхней поверхности ($X_r \approx 0.75$). В донной области наблюдается существенно меньшая плотность по отношению к варианту б.

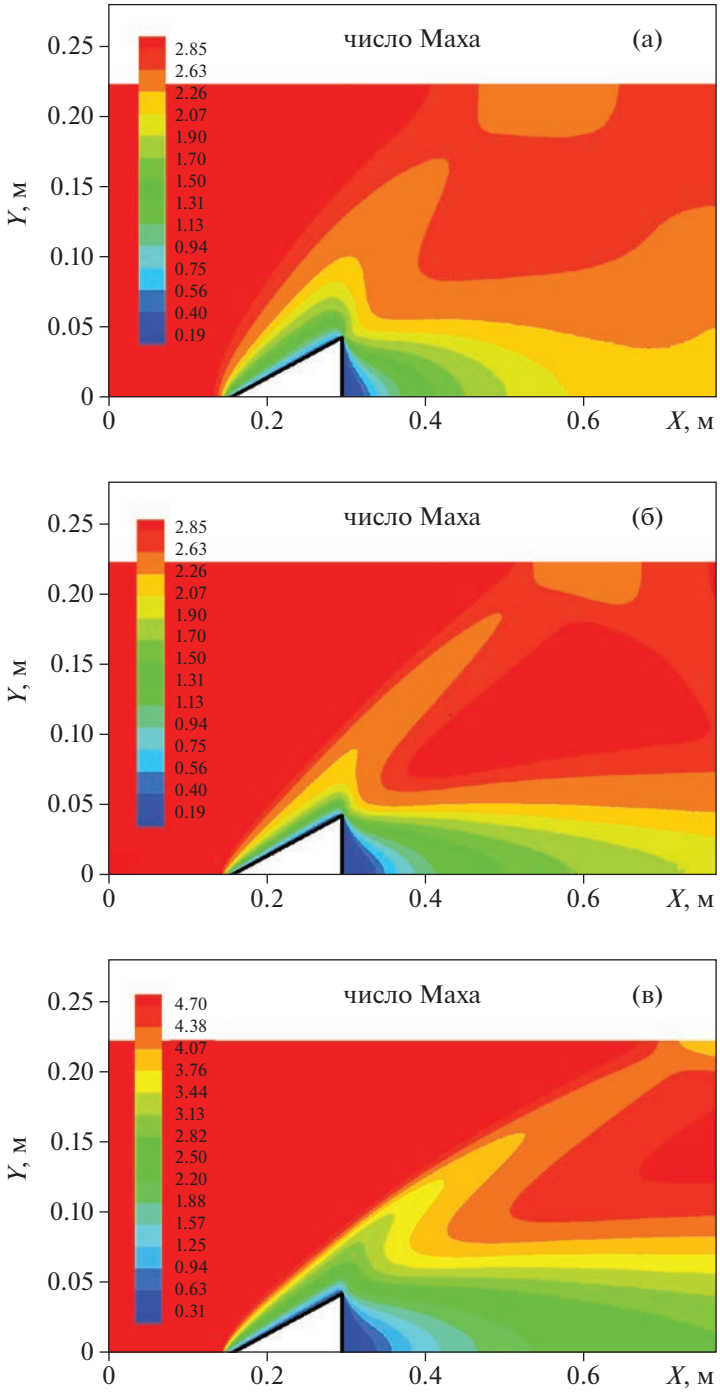
Для вариантов б и в ($\mathrm{Kn} \approx 0.05$) результаты расчетов можно приближенно сопоставить с данными об угле наклона ударной волны при обтекании конуса сверхзвуковым невязким потоком [36], [37]. В последнем случае течение полагается коническим, что позволяет упростить систему уравнений для определения угла наклона косого скачка. Согласно [37] угол наклона составляет 30.5° для угла полураствора конуса 20° при числе Маха набегающего потока 2.84 и 20.6° для угла полураствора 15° при $M_0 = 4.6$. Угол наклона скачка уменьшается с ростом числа Маха и уменьшением угла полураствора конуса. Для полученных результатов угол наклона скачка можно оценить лишь приближенно по углу наклона изолиний максимальной плотности. Рассматриваемое



Фиг. 16. Поле плотности для вариантов (а), (б) и (в).

течение разреженное и не является коническим. Угол наклона скачка находится в отмеченном выше диапазоне и составляет 28.2° для $M_0 = 3$ и 27° для $M_0 = 5$. Качественно результаты расчета полей плотности находятся в соответствии с результатами работы [35], в которой рассматривается обтекание конуса с углом полураствора 10° потоком азота.

Интересным аспектом обтекания конуса является формирование циркуляционного течения в донной области (фиг. 18). Для варианта расчета б ($M_0 = 3$, $Kn \approx 0.05$) в донной области формируется зона развитого циркуляционного течения. Для вариантов а ($M_0 = 3$, $Kn \approx 0.02$) и в ($M_0 = 5$, $Kn \approx 0.05$) падение плотности аргона за основанием конуса оказывается существенно больше. Течение в донной области становится близким к свободномолекулярному и область циркуляционного течения деградирует.

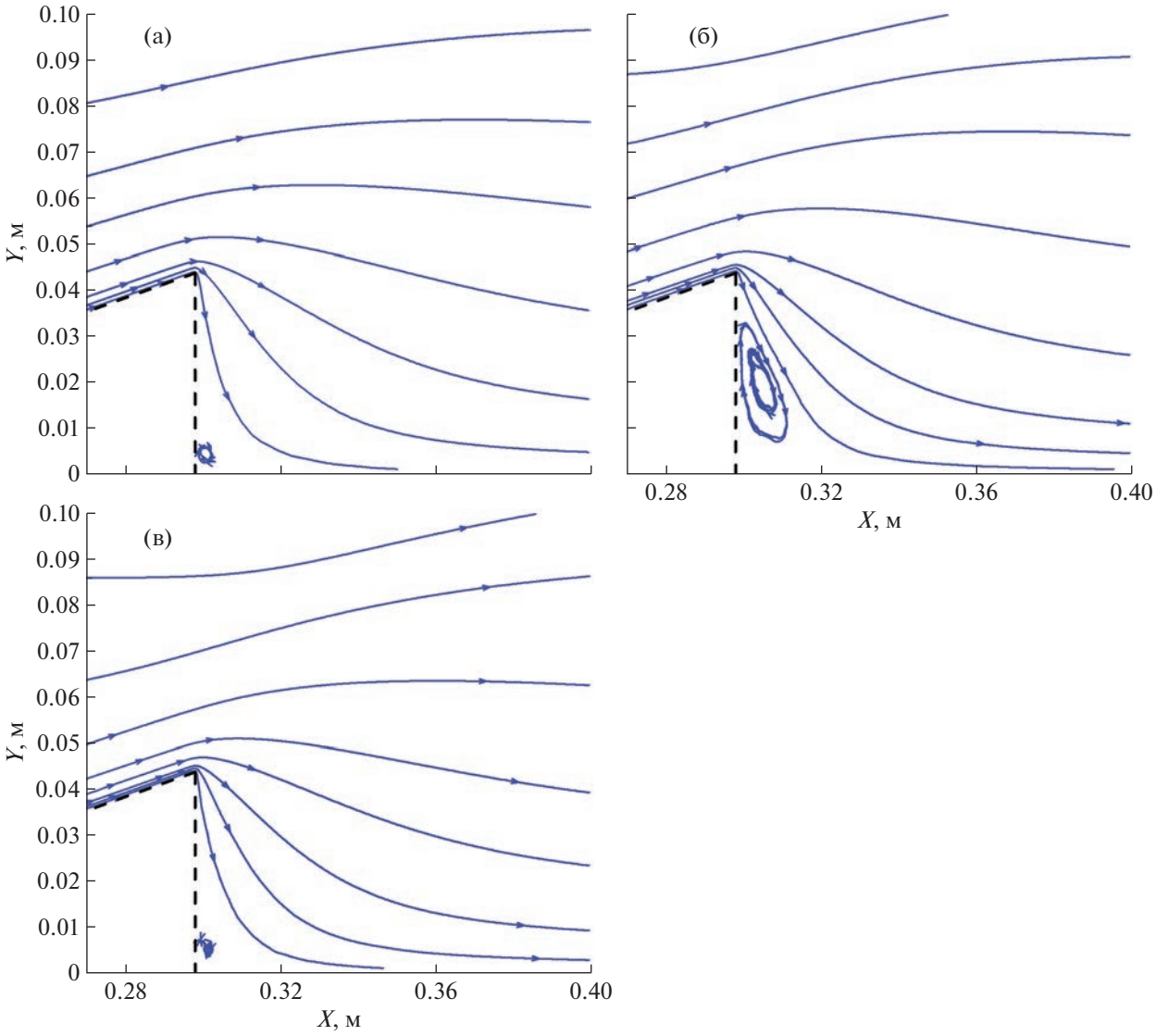


Фиг. 17. Поле числа Маха для вариантов (а), (б) и (в).

ВЫВОДЫ

Алгоритмы параллелизации необходимы для ускорения расчетов разреженных течений методом ПСМ на кластерах различной архитектуры. Одним из возможных подходов к параллелизации является параллелизация по данным. К достоинствам алгоритма параллелизации по данным относится простота реализации в имеющемся последовательном коде ПСМ средствами OpenMP, а также меньшие требования по резервированию памяти для массивов данных.

В рамках проведенного исследования разработанный алгоритм параллелизации по данным, реализованный средствами OpenMP, протестирован на двух задачах: о расчете свободного одно-



Фиг. 18. Линии тока для вариантов (а), (б) и (в).

родного потока и расчете осесимметричного обтекания конуса. Ускорение алгоритма оказалось удовлетворительным, около 6.5 на 28 потоках/ядрах. Наиболее “узким” местом работы алгоритма является плохая масштабируемость процедуры индексации частиц. Предложен алгоритм, позволяющий увеличить эффективность параллелизации данной процедуры и повысить значение ускорения работы всей программы до 8 на 28 потоках. Среди причин относительно слабого ускорения можно указать также недостаток используемой модели параллельного программирования соединение–разветвление, когда распараллеливаются только участки, а не весь код, и возможное удержание совместно используемых данных в других вычислительных ядрах.

Проведено сравнение эффективности работы алгоритма параллелизации по данным и алгоритма декомпозиции расчетной области. Последний в общем случае может быть реализован как средствами технологии MPI, так и OpenMP. Расчет свободного потока продемонстрировал существенно лучшую масштабируемость алгоритма декомпозиции расчетной области. Однако программная реализация алгоритма требует передачи данных о покидающих подобласти частицах между потоками/ядрами на каждом временном шаге, правильной “адресации” этих частиц в новые области, а также использования дополнительных алгоритмов балансировки загрузки. Таким образом, в случае применения алгоритма декомпозиции по области необходимо существенное изменение программного кода с его последующей отладкой.

С использованием параллельного кода выполнен расчет обтекания конуса с углом полураствора 16.7° разреженным сверхзвуковым потоком аргона. Распределение параметров в поле течения находится в согласии с результатами других теоретических исследований. Использование высокоэффективного алгоритма позволило изучить особенности течения в донной области, характеризуемой малой плотностью частиц. Показана деградация области циркуляционного течения с ростом числа Кнудсена и числа Маха.

Представленная работа является одним из шагов в развитии алгоритма декомпозиции по данным. Перспективным направлением является ускорение работы алгоритма за счет: реорганизации массива информации о частицах и перехода к использованию отдельных массивов для каждой ячейки, аналогично технологии, реализованной в PIC-коде (Particle-In-Cell) расчета течений плазмы [38], [39]; полному исключению процедуры индексации частиц за счет включения ряда ее операций в другие процедуры метода ПСМ. Также интерес представляет разработка двухуровневых алгоритмов, основанных на декомпозиции расчетной области средствами MPI совместно с алгоритмом с параллелизацией по данным средствами OpenMP, а также реализации алгоритма декомпозиции расчетной области средствами MPI при межузловом взаимодействии совместно с алгоритмом декомпозиции расчетной области средствами OpenMP при работе с общей памятью на каждом узле.

СПИСОК ЛИТЕРАТУРЫ

1. *Boltzmann L.* Weitere studien über das wärmegleichgewicht unter gasmolecülen // Wien. Acad. Sitzungsber. 1872. № 66. P. 275–370.
2. *Нагнибеда Е.А., Кустова Е.В.* Кинетическая теория процессов переноса и релаксации в потоках неравновесных реагирующих газов. СПб.: Изд-во Санкт-Петербургского гос. ун-та, 2003. 272 с.
3. *Додулад О.И., Клосс Ю.Ю., Потапов А.П., Черемисин Ф.Г., Шувалов П.В.* Моделирование течений разреженного газа на основе решения кинетического уравнения Больцмана консервативным проекционным методом // Ж. вычисл. матем. и матем. физ. 2016. Т. 56. № 6. С. 89–105.
4. *Черемисин Ф.Г.* Решение уравнения Больцмана в режиме сплошной среды // Ж. вычисл. матем. и матем. физ. 2023. Т. 63. № 2. С. 336–348.
5. *Bhatnagar P.L., Gross E.P., Krook M.* Model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems // Phys. Rev. 1954. Rev. 94. P. 511–524.
6. *Shakhov E.M.* Generalization of the Krook kinetic relaxation equation // Fluid Dynamics. 1968. V. 3. № 5. P. 95–96.
7. *Шахов Е.М.* Метод исследования движений разреженного газа. М.: Наука, 1974. 205 с.
8. *Titarev V.A.* Application of the Nesvetay Code for Solving Three-Dimensional High-Altitude Aerodynamics Problems // Comput. Math. and Math. Phys. 2020. V. 60. № 4. P. 737–748.
9. *Titarev V.A., Morozov A.A.* Arbitrary Lagrangian-Eulerian discrete velocity method with application to laser-induced plume expansion // Appl. Math. and Comput. VSI:Num Hyp. 2022. V. 429. P. 127241.
10. *Pfeiffer M., Mirza A., Nizenkov P.* Multi-species modeling in the particle-based ellipsoidal statistical Bhatnagar-Gross-Krook method for monatomic gas species // Phys. of Fluids. 2021. V. 33. P. 036106.
11. *Берд Г.* Молекулярная газовая динамика. М.: Мир, 1981.
12. *Bird G.A.* Molecular gas dynamics and the direct simulation of gas flows. Oxford: Clarenton Press, 1994.
13. *Bird G.A.* The DSMC Method. CreateSpace Independent Publishing Platform, 2013.
14. *Bird G.A.* Approach to translational equilibrium in a rigid sphere gas // Phys. Fluids. 1963. V. 6. P. 1518–1519.
15. *Dietrich S., Boyd I.D.* Scalar and parallel optimized implementation of the direct simulation Monte Carlo method // J. Comput. Phys. 1996. V. 126. № 2. P. 328–342.
16. *Ivanov M., Kashkovsky A., Gimelshein S., Markelov G., Alexeenko A., Bondar Y.A., Zhukova G., Nikiforov S., Vaschenkov P.* SMILE system for 2D/3D DSMC computations // Proc. of 25th Internat. Symposium on Rarefied Gas Dynamics. St. Petersburg, Russia. 2006. P. 21–28.
17. *Bykov N.Y., Gorbachev Yu.E.* Cluster formation in copper vapor jet expanding into vacuum: the direct simulation Monte Carlo. Vacuum. 2019. V. 163. P. 119–127.
18. *Plimpton S.J., Moore S.G., Borner A., Stagg A.K., Koehler T.P., Torczynski J.R., Gallis M.A.* Direct simulation Monte Carlo on petaflop supercomputers and beyond // Phys. Fluids. 2019. V. 31. P. 086101.
19. *Scanlon T., White C., Borg M.K., Cassineli P.R., Farbar E., Boyd I.D., Reese J., Brown R.* Open source Direct Simulation Monte Carlo (DSMC) chemistry modelling for hypersonic flows // AIAA Journal. 2015. V. 53. № 6. P. 1670–1680.
20. *Иванов М.С., Рогазинский С.В.* Метод прямого статистического моделирования в динамике разреженного газа // М.: ВЦ СО АН СССР. 1988. С. 177.

21. *Stefanov S., Roohi E., Shoja-Sani A.* A novel transient-adaptive subcell algorithm with a hybrid application of different collision techniques in direct simulation Monte Carlo (DSMC) // Physics of Fluids. 2022. V. 34. P. 092003.
22. *Wilmoth R.G.* Application of a parallel direct simulation Monte Carlo method to hypersonic rarefied flows // AIAA Journal. 1992. V. 30. № 10. P. 2447–2452.
23. *Ivanov M., Markelov G., Taylor S., Watts J.* Parallel DSMC strategies for 3D computations // Proc. parallel CFD'96. North Holland, Amsterdam, 1997. P. 485–492.
24. *Kyun Ho Lee, Seok Weon Choi* Interaction effect analysis of thruster plume on LEO satellite surface using parallel DSMC method // Comput. & Fluids. 2013. V. 80. P. 333–341.
25. *Revathi Jambunathan, Deborah A. Levin.* Advanced Parallelization Strategies Using Hybrid MPI-CUDA Octree DSMC Method for Modeling Flow Through Porous Media // Comput. and Fluids. 2017. V. 149. P. 70–87.
26. *Гришин И.А., Захаров В.В., Лукьянов Г.А.* Параллелизация по данным прямого моделирования Монте-Карло в молекулярной газовой динамике // Препринт № 3–98. Институт высокопроизводительных вычислений и баз данных. Санкт-Петербург. 1998.
27. *Быков Н.Ю., Лукьянов Г.А.* Параллельное прямое моделирование Монте-Карло нестационарных течений разреженного газа на суперкомпьютерах массивно-параллельной архитектуры // Препринт № 5–97. Институт высокопроизводительных вычислений и баз данных. Санкт-Петербург. 1998.
28. *Быков Н.Ю., Горбачев Ю.Е., Лукьянов Г.А.* Параллельное прямое моделирование методом Монте-Карло истечение газа в вакуум от импульсного источника // Теплофиз. и аэромеханика. 1998. Т. 5. № 3. С. 439–445.
29. *Shamseddine M., Lakkis I.* A Novel Spatio-Temporally Adaptive Parallel Three-Dimensional DSMC Solver for Unsteady Rare Þed Micro/Nano Gas Flows // Comput. and Fluids. 2019. Vol. 186. P. 1–14.
30. *Wolfgang Wagner.* A Convergence Proof for Bird's Direct Simulation Monte-Carlo Method for the Boltzmann Equation // J. of Statistical Physics. 1992. V. 66. № 3/4. P. 1011–1044.
31. *Gao D., Schwartzentruber T.E.* Optimizations and Open MP implementation for the direct simulation Monte Carlo method // Comput. & Fluids. 2011. V. 42. P. 73–81.
32. *Vogenitz F.W., Takata G.Y.* Rarefied hypersonic flow about cones and flat plates by Monte Carlo simulation // AIAA Journal. 1971. V. 9. Iss. 1. P. 94–100.
33. *Ерофеев А.И.* Расчет обтекания конуса под углом атаки гиперзвуковым потоком разреженного газа // Уч. Зап. ЦАГИ. 1979. Т. 10. № 6. С. 122–127.
34. *Ларина И.Н., Рыков В.А.* Пространственное обтекание конических тел потоком разреженного газа // Ж. вычисл. матем. и матем. физ. 1989. Т. 29. № 1. С. 110–117.
35. *Taylor J.C., Moss J.N., Hassan H.A.* Study of hypersonic flow past sharp cones // AIAA. 1989. P. 89–1713.
36. *Краснов Н.Ф.* Аэродинамика тел вращения. М.: Машиностр., 1964. 572 с.
37. *Kopal Z.* Tables of supersonic flow around Cones. Technical report N. 1. MIT., 1947.
38. *Tskhakaya D., Schneider R.* Optimization of PIC codes by improved memory management // J. of Comput. Phys. 2007. V. 225. P. 829–839.
39. *Романенко А.А., Снытников А.В.* Оптимизация переупорядочивания модельных частиц при реализации метода частиц в ячейках на GPU // Вестн. Новосибирского гос. ун-та. Серия: Информационные технологии. 2019. Т. 17. № 1. С. 82–89.