

УДК 519.6

## ИСПОЛЬЗОВАНИЕ ВЫСОКОПРОИЗВОДИТЕЛЬНЫХ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ ДЛЯ МОДЕЛИРОВАНИЯ ФАРЛЕЙ-БУНЕМАНОВСКОЙ НЕУСТОЙЧИВОСТИ

Д. В. Ковалёв<sup>1</sup>, А. П. Смирнов<sup>1</sup>, А. Б. Шмелев<sup>1</sup>

Фарлей-бунемановская неустойчивость плазмы наблюдается в E-области ионосферы Земли на высотах порядка 100 км. Для описания неустойчивости используется математическая модель, включающая в себя кинетическое уравнение для ионов, зависящее от пяти независимых переменных (двух пространственных координат, двух скоростных координат и времени); гидродинамические уравнения для плотности и температуры электронов, зависящие от трех независимых переменных (двух пространственных координат и времени) и двумерное уравнение Пуассона для потенциала турбулентного электрического поля. Задача решается численно на многомерной сетке, содержащей в среднем порядка  $10^9$  точек. Создан комплекс программ для решения такой задачи на суперкомпьютерных системах. Расчеты проводились на вычислительных комплексах СКИФ МГУ “Чебышёв” и IBM Blue Gene/P факультета ВМК МГУ. Приведены результаты анализа эффективности работы и показана хорошая масштабируемость созданного комплекса программ. На вариантах расчетов с большим объемом используемых данных (порядка 100 Гб) и числом процессоров порядка 2000 получено ускорение близкое к линейному. Работа выполнена в рамках программы СКИФ-ГРИД и при поддержке государственных контрактов № П-958 от 20 августа 2009 г. и № 02.740.11.0196 по Федеральной целевой программе “Научные и научно-педагогические кадры инновационной России”, а также РФФИ (код проекта 08-01-00721).

**Ключевые слова:** ионосфера, моделирование плазмы, неустойчивость в плазме, кинетическое уравнение, высокопроизводительные вычисления.

**1. Введение.** Фарлей-бунемановская неустойчивость — это низкочастотная неустойчивость в плазме, движимая достаточно сильным квазистационарным электрическим полем  $E_0$ , перпендикулярным магнитному полю  $B_0$  [1–4]. Эта неустойчивость возникает в слабоионизированной E-области ионосферы Земли, в которой электроны замагничены, в то время как ионы из-за частых столкновений с нейтральными частицами практически не подвержены влиянию поля  $B_0$ . Ниже E-области (примерно 95 км) неустойчивость не развивается из-за столкновительного затухания, а на высотах более 120 км ионы становятся замагниченными. Средние амплитуды колебаний плотности плазмы обычно не превосходят нескольких процентов. Типичные длины волн колебаний, которые были зафиксированы радаром, лежат в метровом диапазоне. В ходе развития неустойчивости наблюдается рост возмущенного электрического поля. Под насыщением неустойчивости понимается насыщение среднеквадратического значения возмущенного электрического поля, а также среднего уровня колебаний плотности плазмы.

Помимо земной ионосферы фарлей-бунемановская неустойчивость может развиваться в атмосферах других планет, на звездах, а также в других разновидностях плазмы, где электроны замагничены, а ионы не замагничены.

**2. Математическая модель неустойчивости.** Условия замагниченности электронов и незамагниченности ионов в E-области ионосферы могут быть записаны в виде [3, 4]

$$\frac{\nu_{en}}{\Omega_e} \ll 1, \quad \frac{\nu_{in}}{\Omega_i} > 1, \quad (1)$$

где  $\nu_{en}$ ,  $\nu_{in}$  — средние частоты столкновений электронов и ионов с нейтральными частицами и  $\Omega_{e,i} = \frac{eB_0}{m_{e,i}}$  — гирочастоты электронов и ионов, массы которых равны  $m_{e,i}$ . В E-области ионосферы соотношение частот

<sup>1</sup> Московский государственный университет имени М. В. Ломоносова, факультет вычислительной математики и кибернетики, ГСП-1, Ленинские горы, 119991, Москва; Д. В. Ковалёв, мл. науч. сотр., e-mail: dmitry.kovalev@mail.ru; А. П. Смирнов, доцент, e-mail: sap@cs.msu.su; А. Б. Шмелев, ст. науч. сотр., e-mail: alex\_shmelev@inbox.ru

столкновений ионов и электронов с нейтральными частицами почти постоянно:  $\nu_{en} \simeq 10 \nu_{in}$ . Масса ионов приблизительно равна тридцати протонным массам.

Для описания движения электронов используется гидродинамическая модель. В E-области ионосферы выполнено условие  $\omega \ll \nu_{en}$ , где  $\omega$  — преобладающая волновая частота в турбулентной плазме, поэтому можно пренебречь инерцией электронов. Низкочастотные процессы в плазме имеют электростатическую природу, так как приводят к незначительным изменениям магнитного поля, поэтому возмущенное электрическое поле можно описывать с помощью электростатического потенциала  $\Phi$ . При таких предположениях на основании гидродинамической теории можно получить уравнение  $\frac{\partial n_e}{\partial t} + \nabla \cdot \Gamma_e = 0$ , где электронные потоки заданы следующим образом:

$$\Gamma_e \equiv n_e \mathbf{V}_e = - \frac{\hat{\nu}_{en}}{m_e \Omega_e^2} \left[ \nabla (T_e n_e) - e n_e (\nabla \Phi - \mathbf{E}_0) \right] + n_e \mathbf{V}_0 + \frac{e n_e}{m_i \Omega_i} \mathbf{e}_z \times \nabla \Phi + \frac{1}{e B_0^2} \nabla (n_e T_e) \times \mathbf{B}_0.$$

Здесь  $\mathbf{V}_e$  — среднемассовая скорость электронов,  $\hat{\nu}_{en} = \nu_{en} \sqrt{\frac{T_e}{T_i}}$  — зависимость частоты столкновений от температуры,  $\Phi$  — возмущенный электрический потенциал,  $T_e$  — температура электронов,  $T_i$  — температура ионов (она же температура электронов в начале моделирования),  $\mathbf{E}_0$  — постоянное электрическое поле, перпендикулярное геомагнитному полю  $\mathbf{B}_0$ ,  $\mathbf{e}_z = \frac{\mathbf{B}_0}{B_0}$  — единичный вектор вдоль магнитного поля,  $\mathbf{V}_0 = c \mathbf{E}_0 \times \frac{\mathbf{e}_z}{B_0}$  — дрейфовая скорость,  $\nabla = e_x \frac{\partial}{\partial x} + e_y \frac{\partial}{\partial y}$  — оператор по координатам, перпендикулярным  $\mathbf{B}_0$ ,  $x$  и  $y$  — координаты вдоль  $\mathbf{V}_0$  и  $\mathbf{E}_0$  соответственно. Температура выражается в джоулях.

Развитие неустойчивости исследуется в прямоугольной области. При решении уравнения для электронной плотности, как и при решении остальных уравнений из модели фарлей-бунемановской неустойчивости, на границе области моделирования используются периодические граничные условия. Начальные условия для плотности электронов  $n_e$ , как и для плотности ионов  $n_i$ , которая будет описана далее, задаются в виде периодической функции таким образом, чтобы начальное значение электрического поля ( $-\nabla \Phi$ ) было на порядок меньше внешнего поля  $\mathbf{E}_0$ . Это условие выполняется, если разность между начальными значениями плотности электронов и ионов мала.

Уравнение изменения температуры электронов описывается следующим нелинейным диффузионно-конвекционным уравнением [5]:

$$n_e^{2/3} \frac{d}{dt} \left( \frac{T_e}{n_e^{2/3}} \right) = \frac{2}{3} \mu_e \hat{\nu}_{en} V_e^2 - \delta_e \hat{\nu}_{en} (T_e - T_n). \quad (2)$$

Здесь  $\frac{d}{dt} = \frac{\partial}{\partial t} + V_{ex} \frac{\partial}{\partial x} + V_{ey} \frac{\partial}{\partial y}$  — лагранжева производная по времени,  $\mu_e = \frac{m_e m_n}{m_e + m_n} \approx m_e$  — приведенная масса, параметр  $\delta_e$  отвечает за относительное количество энергии, потерянной при одном неупругом соударении (величина  $\delta_e$  на два порядка больше соответствующей величины для упругого соударения  $2m_e/m_n$ ),  $m_n$  — средняя масса нейтральной частицы (в E-области примерно равна ионной массе).

Выражение, стоящее в левой части уравнения (2), описывает бесстолкновительное адиабатическое изменение температуры электронов и тепловую адвекцию. Первый член в правой части описывает столкновительный нагрев, а второй — охлаждение при электрон-нейтральных соударениях.

Кинетическое уравнение для ионов с учетом (1) и затухания Ландау записывается с помощью столкновительного члена вида БГК (Bhatnagar–Gross–Krook) [6, 7]

$$\frac{\partial f}{\partial t} + (\mathbf{v} \cdot \nabla) f + \frac{e(\mathbf{E}_0 - \nabla \Phi)}{m_i} \cdot \frac{\partial f}{\partial \mathbf{v}} = -\nu_{in} (f - f_0),$$

где  $\mathbf{v}$  — скорость ионов и  $f(\mathbf{r}, \mathbf{v}, t)$  — функция распределения ионов.

Невозмущенная функция ионов является максвелловской:

$$f_0(\mathbf{r}, \mathbf{v}, t) = n_i \left( \frac{m_i}{2\pi v T_i} \right)^{3/2} \exp \left( - \frac{m_i v^2}{2 T_i} \right).$$

Здесь  $n_i(\mathbf{r}, t) = \int f(\mathbf{r}, \mathbf{v}, t) d^3 v$ .

Уравнение Пуассона для электрического потенциала имеет вид  $\varepsilon_0 \nabla^2 \Phi = e(n_e - n_i)$ .

**3. Методы решения.** Основная трудность при решении уравнений для электронной температуры и плотности заключается в том, что при моделировании неустойчивости необходимо, во-первых, проводить вычисления как можно в большей области, а во-вторых, как можно лучше аппроксимировать плотности и температуру, т.е. увеличивать количество точек на характеристическую единицу длины [8]. Из физических соображений для разрешения дебаевского радиуса на единицу характеристической длины, равной отношению  $\frac{v_{T_i}}{\nu_{in}}$  ( $v_{T_i}$  — тепловая скорость ионов), должно приходиться как минимум 5–10 точек, в то время как для моделирования процесса в целом в плоскости  $(x, y)$  желательно рассматривать область длиной по каждому направлению как минимум в 20 характеристических длин. Многие испробованные разностные схемы для конвекции (Лакс–Вендрофф, Лакс–Фридрихс, МакКормак, различные комбинации этих методов и некоторые другие схемы) не дают достаточной аппроксимации на сетках с большим шагом по пространству (когда на единицу длины приходится 5–10 точек) и оказываются неустойчивыми при больших значениях скоростей переноса.

Для проведения расчетов и соблюдения условий развития устойчивости приходится значительно уменьшать шаг по времени, что приводит к увеличению времени выполнения алгоритма. Для того чтобы устранить этот недостаток и уменьшить вычислительное время, в уравнения для электронной плотности и температуры подставляются потоковые части и явно выделяются конвекционные и диффузионные члены. Для решения полученного уравнения (для температуры или плотности электронов) применяется метод расщепления по физическим процессам, т.е. уравнение разделяется на диффузионную и конвекционную части, которые решаются последовательно с использованием решения предыдущего уравнения в качестве начального условия. Для решения обеих подзадач используются разностные схемы второго порядка точности по пространственной сетке. Суммарная аппроксимация имеет второй порядок точности по пространственной сетке и первый порядок точности по времени. Построение схемы второго порядка точности по времени крайне сложно в силу нелинейной зависимости электронной плотности от ионной плотности, электрического потенциала и электронной температуры.

Для численного нахождения решения электронного уравнения на одном шаге по времени используется следующий алгоритм: сначала находится решение диффузионной части с начальными условиями, взятыми с предыдущего шага по времени, а затем полученное решение подставляется в качестве начального условия при решении конвекционной части. Диффузионную часть нелинейного уравнения предлагается решать с помощью явной диффузионной схемы (явная схема в данном случае устойчива, так как условие Куранта налагает достаточно жесткие условия на значения шагов по времени и по пространству), а конвекционную часть — с помощью численного интегрирования по характеристикам на основе интерполяции по Лагранжу по пяти точкам вдоль каждого из направлений либо схемы Фромма.

Численный алгоритм решения кинетического уравнения основан на методе расщепления. Основное уравнение разделяется на два более простых для решения уравнения следующим образом [8]:

$$\frac{\partial f}{\partial t} + v_x \frac{\partial f}{\partial x} + v_y \frac{\partial f}{\partial y} = 0, \quad (3)$$

$$\frac{\partial f}{\partial t} - \frac{e}{m_i} \frac{\partial \Phi}{\partial x} \frac{\partial f}{\partial v_x} + \frac{e}{m_i} \left( E_0 - \frac{\partial \Phi}{\partial y} \right) \frac{\partial f}{\partial v_y} = -\nu_{in}(f - f_0). \quad (4)$$

Решение на одном шаге по времени находится так: сначала выполняется перенос в пространстве координат (3) с использованием значений функции распределения с предыдущего шага по времени, затем полученное решение подставляется в качестве начального значения для переноса в пространстве скоростей (4). Оба расщепленные уравнения представляют собой набор из двумерных уравнений переноса, причем с постоянными скоростями, которые позволяют использовать быстрые и эффективные методы для нахождения решения. В настоящей статье используется интегрирование по характеристикам.

Уравнение Пуассона решается с помощью спектрального разложения с использованием алгоритма быстрого дискретного преобразования Фурье.

Схематический алгоритм решения уравнений, описывающих неустойчивость, на одном шаге по времени выглядит следующим образом.

1. Считая известными плотности электронов и ионов, решить уравнение Пуассона.
2. Решить ионное уравнение.
3. Решить  $N$  раз уравнение для электронов совместно с уравнением Пуассона с шагом по времени, в  $N$  раз меньшим по сравнению с тем, что использовался при решении кинетического уравнения для ионов.

Необходимость более частого вычисления решения электронного уравнения связана с тем, что скорость электронов на порядок больше скорости ионов. Условие Куранта для электронов из-за большей

скорости более жесткое, чем для ионов, поэтому приходится производить  $N$  расчетов ( $N \approx 20\text{--}40$  в зависимости от параметров) с меньшим шагом по времени для того, чтобы сохранить устойчивость разностных схем. Уравнение Пуассона пересчитывается, так как при движении электронов меняется электрическое поле.

**4. Методы распараллеливания.** Для удобства дальнейшего изложения введем следующие обозначения:  $N_x, N_y$  — количество точек дискретной сетки, описывающей область моделирования, вдоль направлений  $Ox$  и  $Oy$  соответственно. Точки сетки имеют нумерацию от 0 до  $N_w - 1$ ,  $w = x, y$  вдоль каждого направления, каждая точка однозначно идентифицируется своими номерами  $(i, j)$  вдоль направлений  $Ox$  и  $Oy$ . Обозначим  $M$  — количество используемых вычислительных узлов кластера. Рассмотрим случай  $M < N_x$ , так как только наиболее современные вычислительные комплексы имеют количество узлов, равное 2048 и более, в то время как физический интерес представляют решения задач с количеством точек сетки большим 2048 вдоль каждого из направлений.

Предположим сбалансированность нагрузки на узлы вычислительного комплекса, т.е.  $N_x$  делится на  $M$  без остатка. Изложенные алгоритмы могут быть легко расширены и на несбалансированный случай. В этом случае часть узлов вычислительного комплекса будет выполнять меньшую долю расчетов, чем остальные, что снизит общую производительность системы и увеличит время вычислений. Будем считать, что пользователь заинтересован в повышении эффективности расчетов и, выбирая  $N_x, N_y$  и  $M$ , старается оптимальным образом распределить данные между вычислительными узлами.

Распараллеливание алгоритмов для всех уравнений базируется на общем принципе и выполняется с помощью использования технологии MPI. Массив данных разбивается вдоль выделенного направления на  $M$  блоков, содержащих равное количество точек сетки. В каждый блок с номером  $i$  ( $i \in [0, M - 1]$ ), который размещается на  $i$ -м узле вычислительного комплекса, входят точки сетки, для которых номер точки вдоль выбранного направления (допустим,  $Ox$ ) лежит внутри сегмента  $\left[ i \frac{N_x - 1}{M}, (i + 1) \frac{N_x - 1}{M} \right]$ . Так как внутри блока для фиксированных номеров точек вдоль направления  $Ox$  находятся все точки вдоль направления  $Oy$ , то на каждом вычислительном узле можно независимо проводить расчеты по одномерным разностным схемам вдоль направления  $Oy$ . Если узел состоит из нескольких ядер или процессоров, то для распределения вычислений внутри узла можно дополнительно использовать технологию OpenMP.

При расчетах вдоль направления  $Ox$  необходимо проводить обмен данными между вычислительными узлами. Учитывая, что для вычислений используются разностные схемы с предопределенными шаблонами, всегда можно точно определить, какие узлы будут обмениваться данными. С учетом периодических граничных условий вычислительные узлы образуют кольцо, т.е. у каждого узла есть соседи справа и слева и каждый узел однозначно идентифицируется по своим соседям. Узлом слева (справа) для некоторого узла называется узел, на котором хранятся точки сетки с номерами по оси, вдоль которой проводилось разбиение, меньшими (большими), чем на самом узле. Соседними называются узлы, содержащие смежные блоки данных. Условие периодичности задает соседей для граничных узлов, так что для крайнего правого узла в группе соседом справа является самый левый сосед в этой же группе.

**4.1. Уравнение для электронов.** В этом разделе будут рассмотрены алгоритмы решения уравнения для плотности электронов. Схема вычислений, используемая для решения уравнения для температуры электронов, полностью аналогична.

Метод решения двумерного уравнения для плотности электронов, изложенный ранее, заключается в вычислении:

- 1) конвективной части (интегрировании по характеристикам вдоль оси  $Ox$ , а затем вдоль оси  $Oy$ );
- 2) диффузионной части и линейной добавки.

Как конвекция, так и диффузия (явная схема) вдоль направления  $Oy$  могут быть рассчитаны каждым узлом независимо. Линейная добавка рассчитывается с помощью разностной схемы первого порядка с симметричными весами, поэтому также может быть вычислена независимо на каждом узле.

Для расчета конвекции вдоль оси  $Ox$  для точек с номером  $k$  вдоль этой же оси необходимо знать значения в точках  $k \pm 1, 2, 3, \dots$  в зависимости от шаблона. Для интерполяции по пяти точкам необходимо знать значения в точках  $k \pm 1, 2$ . Алгоритм действия достаточно прост. Если такие точки размещены на том же узле, где проводятся вычисления, то производится расчет конвективной составляющей. Если нет, то проводится обмен данными с соседями слева и справа (при необходимости с соседями этих соседей). Так как заранее известно, с какими узлами будет производиться обмен, то целесообразно провести его перед вычислениями, а дальше независимо выполнить расчеты на каждом узле.

Здесь предполагается, что количество точек сетки вдоль направления  $Ox$ , приходящееся на каж-

дый узел, мало. Иными словами, время, необходимое для расчетов конвекции вдоль  $Ox$  для точек, над которыми можно провести расчеты без обмена данными с соседними узлами, мало либо такие точки отсутствуют (например, на узел приходится одна точка вдоль оси  $Ox$ ). В противном случае можно использовать асинхронный обмен данными непосредственно во время расчетов конвекционной составляющей для “внутренних” для каждого узла точек.

Абсолютно аналогично проводится обмен данными для расчетов диффузионной составляющей.

Как результат, обобщенный алгоритм вычисления можно записать так:

- 1) обмен данными между соседними узлами,
- 2) расчет конвективной части вдоль  $Ox$  и  $Oy$ ,
- 3) обмен данными между соседними узлами,
- 4) расчет диффузионной части,
- 5) подсчет линейной добавки.

**4.2. Кинетическое уравнение.** Для распараллеливания алгоритма решения кинетического уравнения используется метод, описанный в предыдущем разделе.

Данные разбиваются на блоки вдоль оси  $Ox$ . Вычисления переноса вдоль оси  $Oy$ , а также перенос в пространстве скоростей  $Vx, Vy$ , а также вычисление столкновительного интеграла проводятся независимо каждым узлом. Для вычисления переноса вдоль оси  $Ox$  осуществляется обмен данными между соседними узлами по аналогичному принципу, как и при расчете конвективной части электронного уравнения.

**4.3. Уравнение Пуассона.** Для решения уравнения Пуассона использовалась библиотека FFTW, реализующая быстрое дискретное преобразование Фурье. Разные версии этой библиотеки поддерживают различные типы параллельных вычислений: MPI и OpenMP.

Версия с OpenMP предназначена для расчетов на одном (возможно, многопроцессорном, но с общей памятью) вычислительном узле, что не является оптимальным с точки зрения использования памяти и эффективности используемых вычислительных средств для данной задачи. Проблема с памятью заключается в том, что даже на средних двумерных задачах с размером сетки 4096 по двум направлениям на компьютере, использующем 8 байт для представления вещественного числа, один массив занимает 128 Мб памяти. Увеличение сетки в два раза по всем направлениям приводит к четырехкратному увеличению размера массива в памяти. При вычислениях используется как минимум два массива, которые фактически занимают всю имеющуюся оперативную память, что особенно критично для многопроцессорных вычислительных комплексов с распределенной памятью (и это без учета использования четырехмерного массива для ионов), так как на таких комплексах на один узел обычно выделяется 1–2 Гб памяти.

Версия с MPI позволяет задействовать количество процессоров, равное количеству точек сетки вдоль направления распараллеливания, поэтому для решения уравнения Пуассона использовалась MPI-версия библиотеки FFTW.

При подборе оптимального количества узлов кластера для решения уравнения Пуассона необходимо учитывать время, которое будет затрачено на преобразование данных о возмущенном электрическом поле в формат, используемый алгоритмами решения гидродинамических уравнений для электронов и кинетического уравнения для ионов. Такое преобразование необходимо, когда число узлов, используемое для решения уравнения Пуассона и электронного или ионного уравнения, различно.

**5. Анализ эффективности распараллеливания.** Для анализа эффективности распараллеливания программного комплекса были произведены тестовые расчеты на двух вычислительных комплексах IBM Blue Gene/P и СКИФ МГУ “Чебышёв”.

Для тестирования были выбраны три задачи на сетках с 1024, 2048 и 4096 точками по каждому направлению в пространстве координат и 30 точками по каждому направлению в пространстве скоростей. Электронное уравнение вычислялось 35 раз, а уравнение Пуассона вычислялось 36 раз на один ионный шаг по времени, количество ионных шагов по времени равно 10. Под решением электронного уравнения подразумевается решение системы уравнений для электронной плотности и температуры.

При расчетах на IBM Blue Gene/P использовался гибридный режим OpenMP–MPI, в котором в качестве одного узла выделяется четырехъядерный процессор, а на СКИФ МГУ использовался MPI-режим, в качестве узла выступало ядро процессора. Далее будет представлен анализ суммарного времени выполнения программы.

На рис. 1а приведен график ускорения относительно выполнения задачи на 128 узлах, полученный для расчетов на вычислительном комплексе IBM Blue Gene/P. Коэффициент загрузки (рис. 1б) определялся по формуле  $zgr = \frac{128T_{128}}{NT_N}$ , где  $T_N$  — время выполнения задачи на  $N$  узлах. Коэффициент загрузки показывает, насколько эффективнее выполняется задача на  $N$  узлах относительно расчета

на 128 узлах. С помощью этого коэффициента также можно определить, при каком количестве используемых вычислительных узлов затрачивается минимальное суммарное процессорное время для расчетов. Коэффициент загрузки может служить альтернативной мерой определения эффективности распараллеливания программы (обычно принято использовать в качестве такой меры ускорение) и позволяет отследить падение суммарной производительности при увеличении числа вычислительных узлов. В идеальном случае (без учета времени межпроцессорного обмена) коэффициент загрузки равен 1. Если он меньше 1, то задача эффективнее использует вычислительные мощности при использовании 128 узлов (это соответствует ситуации, когда при переходе от 128 узлов к  $N$  узлам суммарное процессорное время увеличивается); если же он больше 1, то эффективнее использовать  $N$  узлов (противоположная ситуация).

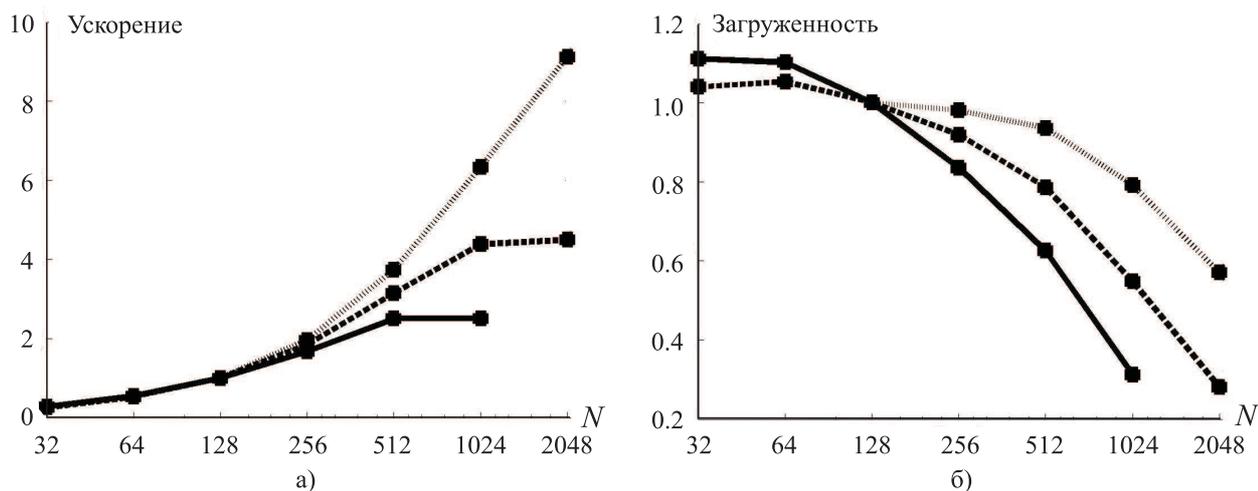


Рис. 1. Графики ускорения (а) и коэффициента загрузки (б), полученных при решении двумерной тестовой задачи, относительно расчета на 128 узлах. По горизонтальной оси отложено число используемых узлов, по вертикальной — полученные ускорение/коэффициент. Сплошная линия соответствует расчету с количеством точек сетки, равным 1024 вдоль каждого направления, пунктирная линия — расчету с 2048 точками сетки вдоль каждого направления, линия из точек — расчету с 4096 точками сетки вдоль каждого направления

Анализируя данные, отображенные на рис. 1, можно отметить, что программный код эффективно выполняется на достаточно большом количестве узлов. Для расчетов с большим объемом вычислений на один узел наблюдается ускорение, близкое к линейному. При уменьшении количества вычислений, приходящихся на один узел, в соответствии с законом Амдала ускорение падает. Аналогичные результаты получаются при анализе поведения коэффициента загрузки, который с увеличением количества вычислительных узлов уменьшается.

При рассмотрении времен выполнения алгоритмов решения уравнений по отдельности можно отметить, что для наиболее вычислительно сложного ионного уравнения ускорение практически линейное и коэффициент загрузки близок к 1 для любого количества узлов. Менее сложные уравнения для электронов и электрического потенциала наиболее эффективно рассчитываются, если количество узлов не превышает некоторый предел, зависящий от параметров сетки. Далее ускорение времени работы алгоритма падает из-за увеличения доли времени, необходимой для обмена данными между узлами, в суммарном времени выполнения программы. Решать уравнения для ионов и электронов на разном числе процессоров также не всегда оптимально из-за необходимости согласования данных между узлами для решения уравнения Пуассона.

Масштабируемость программного кода (т.е. способность увеличить производительность пропорционально дополнительным ресурсам) проверим с помощью сравнения времени выполнения выбранной тестовой программы и времени выполнения программы, для которой объем вычислений и количество вычислительных ресурсов было увеличено в  $N$  раз. Если отношение времен близко к единице, то программный код масштабируем. Если отношение много меньше единицы, то программный код эффективнее использует вычислительные мощности при меньшем объеме данных (при увеличении количества вычислений и вычислительных узлов производительность системы падает). Значение коэффициента, большее единицы, чаще всего означает, что программный код эффективнее использует вычислительные мощности

при больших объемах данных для выбранного набора тестов.

Для сравнения были выбраны следующие два варианта:

- 1) задача с 2048 точками на 256 узлах и задача с 4096 точками на 1024 узлах,
- 2) задача с 1024 точками на 64 узлах и задача с 2048 точками на 256 узлах.

Для обоих тестов наблюдается четырехкратное увеличение числа ресурсов и объема данных между выбранными расчетами. Для первого варианта время выполнения задач составило 27.8 и 35.5 секунд соответственно, для второго — 23.84 и 27.8 секунд соответственно. В обоих случаях отношения времен выполнения близко к 1, что свидетельствует о хорошей масштабируемости программного кода для исследуемого объема данных.

Тестирование задачи также проводилось на вычислительном комплексе СКИФ МГУ “Чебышёв”. Задача выполнялась в MPI-режиме, при котором вычислительный узел представлял собой одно ядро процессора. В силу высокой загруженности комплекса тестирование проводилось максимум на 512 узлах, при этом из-за ограничений на размер оперативной памяти, приходящейся на один узел, задача с 4096 точками была выполнена только на 512 узлах (на меньшем количестве узлов ей не хватало памяти), поэтому эта тестовая задача не использовалась для анализа полученного ускорения.

На рис. 2 представлены результаты расчетов на СКИФ МГУ “Чебышёв”. Тестовая задача с 1024 точками при переходе от 32 узлов к 512 ускоряется в 12 раз (увеличение числа узлов в 16 раз). Тестовая задача с 2048 точками не запускалась на 32 и 64 узлах в силу ограничений на оперативную память, при переходе от 128 узлов к 512 узлам было получено ускорение 5.39 (увеличение количества узлов в 4 раза). Суперлинейное ускорение может объясняться тем, что при увеличении числа процессоров уменьшается объем данных, используемых вычислительным узлом, что уменьшает количество промахов мимо кэша при считывании данных из четырехмерного массива, используемого для хранения данных о функции распределения ионов, и позволяет более рационально использовать кэш-память процессора.

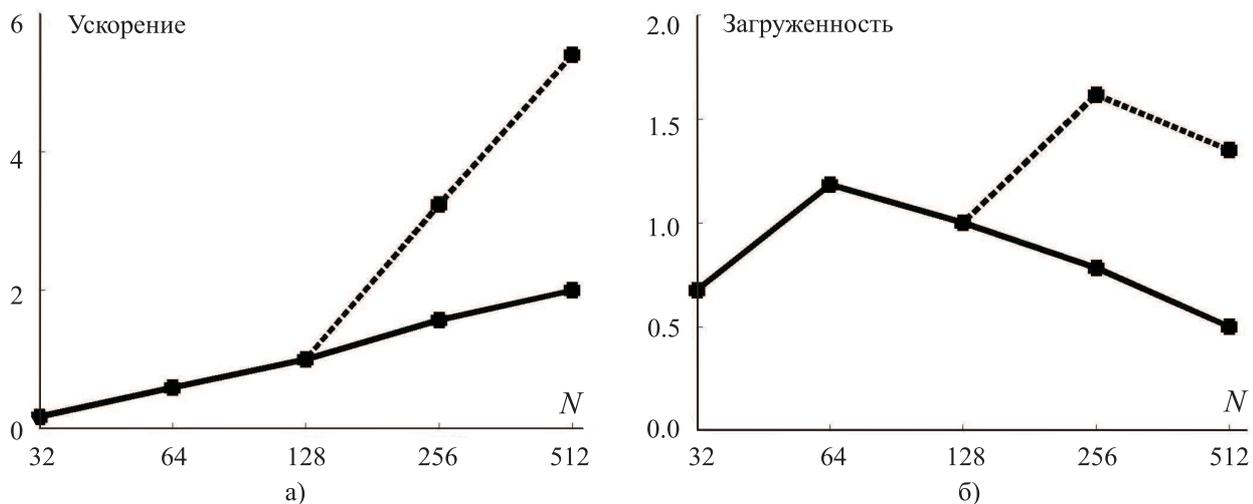


Рис. 2. Графики ускорения (а) и коэффициента загруженности (б), полученных при решении двумерной тестовой задачи, относительно расчета на 128 узлах. По горизонтальной оси отложено число используемых узлов, по вертикальной — полученные ускорение/коэффициент. Сплошная линия соответствует расчету с количеством точек сетки, равным 1024 вдоль каждого направления, пунктирная линия — расчету с 2048 точками сетки вдоль каждого направления

Коэффициент загруженности для тестовой задачи с 2048 точками для всех расчетов больше 1, что свидетельствует об эффективном выполнении задачи при увеличении числа процессоров. Для тестовой задачи с меньшим количеством точек наблюдается плавное снижение коэффициента загруженности из-за уменьшения доли времени выполнения задачи, затрачиваемого на вычисления, и увеличения доли времени, затрачиваемого на операции обмена данными.

#### СПИСОК ЛИТЕРАТУРЫ

1. Farley D.T. A plasma instability resulting in field-aligned irregularities in the ionosphere // J. Geophys. Res. 1963. 68. 6083–6097.

2. *Buneman O.* Excitation of field aligned sound waves by electron streams // *Phys. Rev. Lett.* 1963. **10**. 285–288.
3. *Kovalev D.V., Smirnov A.P., Dimant Y.S.* Modeling of the Farley–Buneman instability in the E-region ionosphere: a new hybrid approach // *Ann. Geophys.* 2008. **26**, N 9. 2853–2870.
4. *Dimant Y.S., Oppenheim M.M.* Ion thermal effects on E-region instabilities: linear theory // *J. Atmos. Terr. Phys.* 2004. **66**, N 17. 1639–1654.
5. *Ковалёв Д.В., Смирнов А.П., Димант Я.С.* Моделирование нелинейного развития фарлей-бунемановской неустойчивости с учетом электронных тепловых эффектов // *Физ. плазмы.* 2009. **35**, № 7. 657–664.
6. *Ковалёв Д.В., Смирнов А.П., Димант Я.С.* Исследование кинетических эффектов, возникающих при моделировании фарлей-бунемановской неустойчивости // *Физ. плазмы.* 2009. **35**, № 5. 465–471.
7. *Bhatnagar P.L., Gross E.P., Krook M.* A model for collision processes in gases. I. Small amplitude processes in charged and neutral one-component systems // *Phys. Rev.* 1954. **94**, N 3. 511–525.
8. *Ковалёв Д.В.* Моделирование фарлей-бунемановской неустойчивости с использованием четырехмерного кинетического уравнения // *Матем. моделирование.* 2008. **20**, № 12. 89–104.

Поступила в редакцию  
12.04.2010

---