

УДК 519.632.4

## ПАРАЛЛЕЛЬНЫЙ АЛГОРИТМ РЕШЕНИЯ ЗАДАЧ ГРАВИТАЦИОННОЙ ФИЗИКИ, ОСНОВАННЫЙ НА ДЕКОМПОЗИЦИИ ОБЛАСТИ

Э. А. Кукшева<sup>1</sup>, В. Н. Снытников<sup>2</sup>

Разработан параллельный алгоритм для решения нестационарных трехмерных задач бесстолкновительной динамики свыше  $10^9$  тел в самосогласованном гравитационном поле в декартовых координатах. При распараллеливании алгоритма применен метод декомпозиции вычислительной области, что привело к созданию параллельной программы для суперкомпьютеров с распределенной памятью. Полученную программу можно использовать для расчетов на сетках, содержащих более  $1000^3$  узлов. Представлено решение одной из таких задач динамики развития неустойчивости, которое получено на суперЭВМ с помощью созданной программы. Работа выполнена по проекту программ Президиума РАН “Происхождение и эволюция звезд и галактик” (координатор академик А. А. Боярчук) и интеграционному проекту СО РАН № 26 (координатор академик Б. Г. Михайленко).

**Ключевые слова:** математическое моделирование, суперЭВМ, параллельное программирование, астрофизика.

**1. Введение.** Численное моделирование стало одним из необходимых и широко используемых инструментов астрофизики. Важная область его применения — решение задач бесстолкновительной гравитационной физики. Сюда можно отнести, в частности, исследования коллективной динамики тел в околозвездных дисках и нестационарных процессов в галактиках, изучение галактических скоплений и крупномасштабной структуры Вселенной. Важную роль в моделировании бесстолкновительной динамики играют методы “частицы–сетка”, получившие в литературе название РМ (Particle–Mesh methods) [1]. Метод РМ состоит в приписывании всем частицам определенного ядра, с помощью которого на введенной сетке рассчитывается вносимая каждой частицей плотность и другие усредняемые функции. Далее на сетке, к примеру, по функции плотности от координат находится распределение гравитационного потенциала. Используя последнюю функцию, находится сила, действующая на каждую частицу со стороны всех остальных тел. Практически РМ-метод предполагает математическое моделирование бесстолкновительных систем на основе физически первых принципов при выполнении определенных условий для числа частиц, сетки и представления физических параметров в вычислительных экспериментах [1]. Подобные вычислительные эксперименты с использованием программ по РМ-методу позволяют находить детальные немаксвелловские функции распределения частиц в 6-мерном фазовом пространстве, отслеживать тонкие эффекты с участием резонансных частиц и определять с контролируемой точностью физические характеристики моделируемых бесстолкновительных систем [2, 3].

Одно из важнейших приложений РМ-метода связано с изучением динамики развития неустойчивости в различных гравитационных системах. В этих случаях необходимы размеры вычислительной сетки порядка  $1000^3$  узлов в трехмерных по пространственным переменным расчетах, а для обеспечения уровня флуктуаций плотности в 1–10% в ячейке должно быть до  $10^3$ – $10^4$  частиц. Тогда общее количество частиц для указанной сетки оценивается в  $10^{12}$ . В принципе, уже современные петафлопные и проектируемые экзафлопные суперкомпьютеры на первый взгляд могут располагать техническими ресурсами для решения задач с указанными параметрами. В нашей работе усилия направляются в первую очередь на создание работоспособного кода для кластерной суперЭВМ. Достижение высокой эффективности распараллеливания как показателя перехода от одного числа процессоров к другому может рассматриваться для сеток порядка  $1000^3$  как следующий этап в развитии параллельных алгоритмов и программ.

В работах [4, 5] были описаны параллельные программы для решения нестационарных трехмерных задач в цилиндрических и декартовых координатах на кластерных системах. Для указанных программ

<sup>1</sup> Институт катализа им. Г. К. Борескова, просп. акад. Лаврентьева, 5, 630090, г. Новосибирск; науч. сотр., e-mail: kuksheva@catalysis.ru;

<sup>2</sup> Институт катализа им. Г. К. Борескова, просп. акад. Лаврентьева, 5, 630090, г. Новосибирск; Новосибирский государственный университет, ул. Пирогова, 2, 630090, г. Новосибирск; ст. науч. сотр., доцент, e-mail: snyt@catalysis.ru

имелось естественное ограничение на рабочие сетки — не более  $256^3$  узлов как следствие типичного в то время размера памяти 1–2 Гб на одном процессоре. Это ограничение возникает из-за выбора самого простого метода распараллеливания задачи, когда копии трехмерных массивов с сеточными значениями потенциала и плотности хранятся во всех процессорах. Такой метод сразу упрощает распараллеливание метода частиц и уменьшает число пересылок. Однако сетки размером более  $256^3$ – $512^3$  узлов не помещаются в памяти процессора на параллельных вычислительных системах с распределенной памятью. Так как большинство суперЭВМ, на которых мы проводим наши вычислительные эксперименты, относятся к кластерным системам, то этот недостаток является критическим. Поэтому главная цель данной работы состояла в том, чтобы предложить и реализовать другой метод распараллеливания, позволяющий работать с вычислительными сетками размером более  $256^3$  узлов, а именно — метод декомпозиции области для действующих суперкомпьютеров с распределенной памятью.

**2. Постановка задачи.** Система уравнений гравитационной физики без учета столкновений состоит из кинетического уравнения Власова–Лиувилля и уравнения Пуассона для потенциала гравитационного поля. Уравнение Власова–Лиувилля в приближении усредненного самосогласованного поля записывается в виде

$$\frac{\partial f}{\partial t} + \mathbf{v} \frac{\partial f}{\partial \mathbf{x}} + \mathbf{a} \frac{\partial f}{\partial \mathbf{v}} = 0, \tag{1}$$

где  $f(t, \mathbf{x}, \mathbf{v})$  — зависящая от времени одночастичная функция распределения по координатам и скоростям,  $\mathbf{a} = \frac{\mathbf{F}}{m} = -\nabla\Phi$  — ускорение частиц. Гравитационный потенциал  $\Phi$ , в котором происходит движение, разделяется на две части:  $\Phi = \Phi_1 + \Phi_2$ , где  $\Phi_1$  в зависимости от моделируемых условий может представлять потенциал внешних сил. Вторая часть  $\Phi_2$  потенциала определяется совокупным распределением движущихся частиц и удовлетворяет уравнению Пуассона ( $G$  — гравитационная постоянная)  $\Delta\Phi_2 = 4\pi G\rho$ , которое в декартовой системе координат запишется в виде

$$\frac{\partial^2 \Phi_2}{\partial x^2} + \frac{\partial^2 \Phi_2}{\partial y^2} + \frac{\partial^2 \Phi_2}{\partial z^2} = 4\pi G\rho. \tag{2}$$

Для изолированной системы граничное условие состоит в нулевом значении потенциала на бесконечности:  $\Phi_{r \rightarrow \infty} = 0$ .

По заданной для начального момента времени функции плотности от координат  $\rho(\mathbf{x})$  частицы располагаются в расчетной области. Начальная функция может быть достаточно произвольной. К примеру, часто задается плоский диск с осесимметричным распределением поверхностной плотности или шар. Начальные скорости частиц определяют движение изолированной системы вокруг своего центра масс. В качестве основных величин обезразмеривания нами выбраны характерный линейный размер гравитирующей системы  $R_*$ , ее масса  $M_*$  и гравитационная постоянная  $G$ . Тогда  $V_* = \sqrt{\frac{GM_*}{R_*}}$ ,  $t_* = \frac{R_*}{V_*}$ ,  $\Phi_* = V_*^2$ , где  $V_*$ ,  $t_*$  и  $\Phi_*$  — единицы для скорости частиц, времени и потенциала. Все дальнейшие выкладки приведены в безразмерном виде.

**3. Численные методы.**

**3.1. Уравнение Власова.** Кинетическое уравнение Власова решается методом частиц-в-ячейках [1], который использовался нами ранее в работах [5, 6]. В пространственной области в виде параллелепипеда вводится сетка, которая делит область на ячейки. Модельные частицы, в общем случае с собственной приписанной им массой и другими характеристиками, имеют индивидуальные координаты и могут перемещаться из ячейки в ячейку в соответствии со своими скоростями, определяя функцию распределения  $f(t, \mathbf{x}, \mathbf{v})$  по скоростям и координатам. Функция плотности  $\rho(t, \mathbf{x}) = \int f(t, \mathbf{x}, \mathbf{v}) d\mathbf{v}$  восстанавливается с помощью интерполяции массы каждой частицы в узлы введенной сетки по известным координатам частицы с ядром PIC (Particle-In-Cell) [1, 5, 6]. Уравнения движения частицы  $i$  имеют вид  $\frac{d\mathbf{v}_i}{dt} = \frac{\mathbf{F}_i}{m_i} = \mathbf{a}_i$ ,  $\frac{d\mathbf{x}_i}{dt} = \mathbf{v}_i$ . Численное решение этих уравнений осуществляется по схеме

$$\begin{aligned} v_x^{n+1} &= v_x^n + \tau a_x^n, & v_y^{n+1} &= v_y^n + \tau a_y^n, & v_z^{n+1} &= v_z^n + \tau a_z^n, \\ x^{n+1} &= x^n + \tau v_x^{n+1}, & y^{n+1} &= y^n + \tau v_y^{n+1}, & z^{n+1} &= z^n + \tau v_z^{n+1}, \end{aligned} \tag{3}$$

где  $\tau$  — шаг по времени и  $n$  — номер этого шага.

Для нахождения ускорения каждой частицы используется интерполяция значения сеточной вектор-функции  $\mathbf{F}$  в местоположение частицы с координатами ее центра  $x^n, y^n, z^n$ , а значения компонент силы  $\mathbf{F}^n$  вычисляются из значений сеточной функции потенциала  $\Phi^n$  со вторым порядком аппроксимации [5].

**3.2. Уравнение Пуассона.** Уравнение Пуассона в прямоугольной сеточной области с регулярной сеткой узлов  $\{x_i = h_x i, y_k = h_y k, z_l = h_z l, i = 0, \dots, I, k = 0, \dots, K, l = 0, \dots, L\}$  аппроксимируется схемой второго порядка

$$\frac{\phi_{i+1,k,l} - 2\phi_{i,k,l} + \phi_{i-1,k,l}}{h_x^2} + \frac{\phi_{i,k+1,l} - 2\phi_{i,k,l} + \phi_{i,k-1,l}}{h_y^2} + \frac{\phi_{i,k,l+1} - 2\phi_{i,k,l} + \phi_{i,k,l-1}}{h_z^2} = 4\pi\rho_{i,k,l}.$$

Полученная система уравнений решается применением трехкратного преобразования Фурье, которое реализуется на основе процедуры быстрого преобразования Фурье. Для этого необходимо независимо найти значение потенциала на границе области.

**3.3. Граничные условия для уравнения Пуассона.** В соответствии с граничным условием для изолированной системы, состоящем в убывании потенциала изолированной системы тел как  $r^{-1}$  до нуля на бесконечности, необходимо вычислять значение потенциала тел на границе. Это можно сделать с помощью фундаментального решения уравнения Пуассона. Его дискретный аналог записывается в виде

$$\phi_{i',k',l'} = \sum_{i,k,l} \frac{m_{i,k,l}}{R_{i',k',l'}^{i,k,l}}, \tag{4}$$

где  $R_{i',k',l'}^{i,k,l}$  — расстояние от узла с точечной массой  $m_{i,k,l}$  до узла, в котором вычисляется потенциал  $\phi_{i',k',l'}$ . Значение сеточной функции потенциала  $\phi_{i',k',l'}$  находится через суммирование вкладов точечных масс  $m_{i,k,l}$ , рассчитанных из значений сеточной функции плотности. В нашем трехмерном случае следует вычислять потенциал на шести граничных гранях пространственной области, поэтому формула (4) дополняется следующими выражениями:

$$\begin{aligned} i = 0, \dots, I, \quad k = 0, \dots, K, \quad l = 0, \dots, L; \quad i' = 0, \dots, I, \quad k' = 0, \dots, K, \quad l' = 0, l' = L; \\ i' = 0, \dots, I, \quad k' = 0, k' = K, \quad l' = 0, \dots, L; \quad i' = 0, i' = I, \quad k' = 0, \dots, K, \quad l' = 0, \dots, L; \\ m_{i,k,l} = V\rho_{i,k,l}, \quad V = h_x h_y h_z, \end{aligned}$$

где  $\rho_{i,k,l}$  — плотность частиц в узле  $(i, k, l)$  и  $V$  — объем ячейки. Таким образом, чтобы вычислить потенциал на границе (4), необходимо определить сеточные значения плотности и найти сеточные значения потенциала граничных ячеек. Как показано в [6], этот способ вычисления потенциала на границе позволяет проводить расчеты при нахождении тел вблизи и на границах расчетной области.

**3.4. Начальные условия.** По заданной для начального момента времени функции плотности от пространственных координат определяются сеточные значения массы в ячейках. Затем в ячейки равномерным образом помещаются частицы, число которых зависит от сеточного значения массы в ячейках. Далее определяются значения скоростей в соответствии с начальной функцией  $f(0, \mathbf{x}, \mathbf{v})$ .

Например, пусть задана модель плоского диска с осесимметричным распределением поверхностной плотности в виде

$$\sigma(r) = \begin{cases} \sigma_0 \sqrt{1 - \left(\frac{r}{R_0}\right)^2}, & r \leq R_0, \\ 0, & r > R_0, \end{cases} \tag{5}$$

где  $R_0$  — начальный радиус диска. Значения цилиндрических координат частиц по  $\varphi$  и  $z$  находятся по правилу  $\varphi \in \text{rnd}[0; 2\pi], z = z_0$ , где  $\text{rnd}$  — функция, генерирующая случайные числа с равномерным распределением на интервале  $[0; 2\pi]$  и  $z_0$  — центральная плоскость области по  $z$ . Средние скорости “ $v_r$ ”, “ $v_z$ ” в ячейке равны нулю, скорость  $v_\varphi$  частиц определяется из соотношения

$\frac{v_\varphi^2}{r} = \frac{\partial \Phi}{\partial r}$ , а далее изменяется в соответствии с  $f(0, \mathbf{x}, \mathbf{v})$ . В результате начальные скорости модельных частиц в диске подчиняются нормальному закону со значением дисперсии по всем трем направлениям пространства  $dv_r, dv_\varphi, dv_z$ .

**4. Параллельная реализация.** Для реализации приведенных численных методов необходимо хранить как минимум два трехмерных массива для сеточных значений потенциала  $[\Phi]$  и плотности  $[M]$

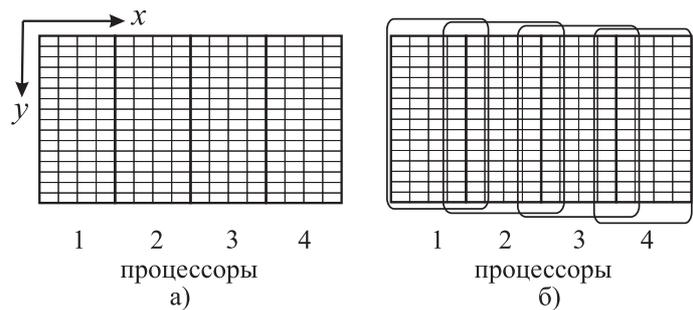


Рис. 1. Схема разбиения массивов по процессорам: разбиение массива плотности  $[M]$  (а), массива потенциала  $[\Phi]$  (б)

и шесть одномерных массивов для координат и скоростей частиц. Главное требование к данной параллельной реализации состояло в том, чтобы была возможность проведения расчетов с вычислительными сетками  $[\Phi]$  и  $[M]$  размером от  $1000^3$  узлов. Чтобы поместить в процессор целиком один такой массив, необходимо иметь 8 Гб оперативной памяти. Характерный объем оперативной памяти процессорного элемента на сегодня составляет 2–4 Гб на доступных нам кластерах. Следовательно, необходимо распределять массивы по процессорам.

Для этого расчетная область делится на слои вдоль оси абсцисс, число которых соответствует числу процессоров. Согласно этому делению, в каждый процессор попадает часть массива  $[M]$  и  $[\Phi]$ , как это показано на рис. 1. Для наглядности деление показано на плоскости. Из рисунка видно, что имеется перекрытие областей, находящихся в обоих соседних процессорах. Для массива  $[M]$  это перекрытие составляет один слой массива слева и справа, а для массива  $[\Phi]$  — по два слоя слева и справа. Перекрытия  $[\Phi]$  необходимы для правильного вычисления силы, три компоненты которой являются производными от потенциала  $\frac{F}{m} = -\nabla\Phi$ . Перекрытия плотности используются, чтобы правильно учитывать вклад в нее частиц, находящихся в соседнем процессоре при интерполяции с ядром РС.

Схема параллельного алгоритма показана на рис. 2. Начальное распределение частиц по пространству и задание их скоростей, а также вычисление начальной сеточной плотности выполняется параллельно, что не отмечено на рисунке. Частицы распределяются по процессорам в соответствии со своими получившимися начальными координатами. Программа реализована на языке C++ с помощью системы передачи сообщений MPI. При выполнении блоков программы “Вычисление граничных условий для уравнения Пуассона” и “Решение уравнения Пуассона методом Фурье” используются коллективные операции MPI сбора и рассылки во все процессоры. В блоках “Вычисление новых координат частиц” и “Вычисление плотности по новым координатам частиц” пересылки “точка–точка” происходят между соседними процессорами. Блок “Сохранение промежуточных результатов” работает через определенное число шагов по времени. Здесь посредством коллективной операции данные собираются в нулевом процессоре, который затем производит запись их на диск.

**4.1. Уравнение Власова.** Решение уравнения Власова на каждом шаге по времени заключается в том, что в цикле по всем частицам из данного процессора согласно формулам (3) вычисляется новое положение каждой частицы. Все частицы, “вылетевшие” за пределы процессора, помещаются во временные массивы: один — для частиц, летящих налево, другой — для частиц, летящих направо. Таким образом, в цикле по частицам формируется два массива с вылетающими частицами. После завершения цикла по частицам все процессоры начинают обмен с соседями справа и слева массивами с выбывшими частицами, при этом получая от них прибывшие частицы (рис. 3).

Поскольку не известно, сколько частиц прибудет и убудет в каждом данном процессоре на каждом

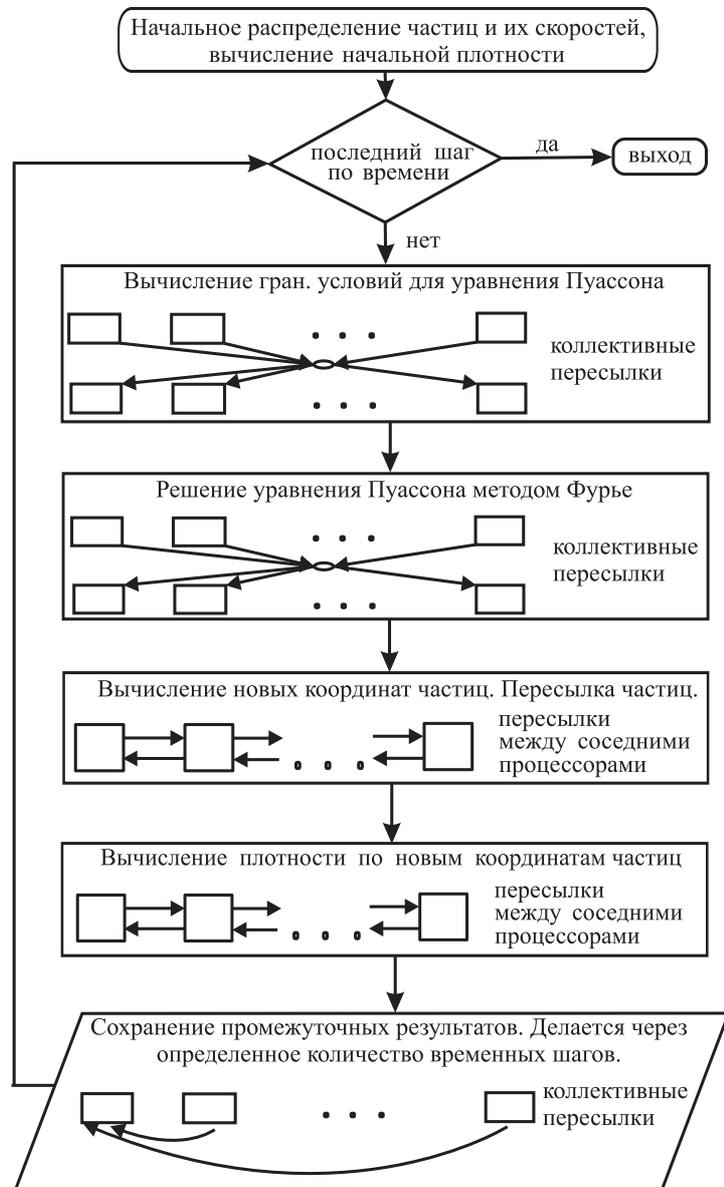


Рис. 2. Схема работы параллельной программы

шаге, возникает проблема организации данных для постоянно меняющихся частиц. Эта проблема была решена следующим образом. Частицы хранятся в шести массивах двойной точности: три массива для хранения координат и три — для хранения компонент скоростей частиц. При инициализации размер массивов частиц в каждом процессоре выбирается с запасом, реальное число частиц в этих массивах обычно меньше. Этот размер зависит от числа частиц и, если их больше 1000, берется равным  $\frac{3jm}{N}$ , где  $jm$  — общее число частиц и  $N$  — число процессоров. При “вылете” частицы из процессора она переносится во временный массив, а на ее место записывается последняя частица из массива частиц, при этом число частиц уменьшается. Это происходит в цикле по частицам (рис. 4). После окончания цикла по частицам после пересылки все вновь прибывшие частицы записываются в конец массива. На этом этапе число частиц увеличивается. Здесь нужно еще раз подчеркнуть, что размерность всего массива частиц не меняется, а меняется только число ячеек, фактически участвующих в выполнении алгоритма.

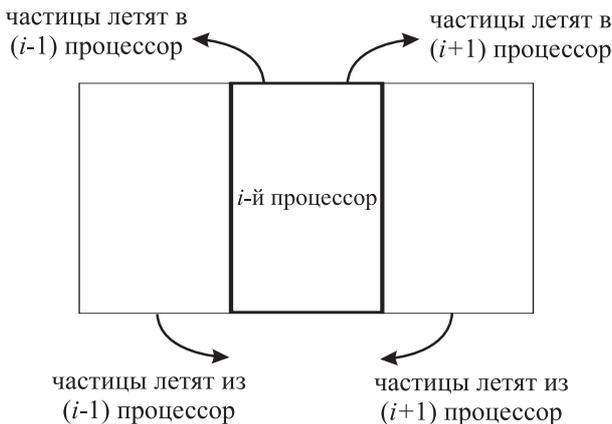


Рис. 3. Схема обмена частицами процессора  $i$  с его соседями

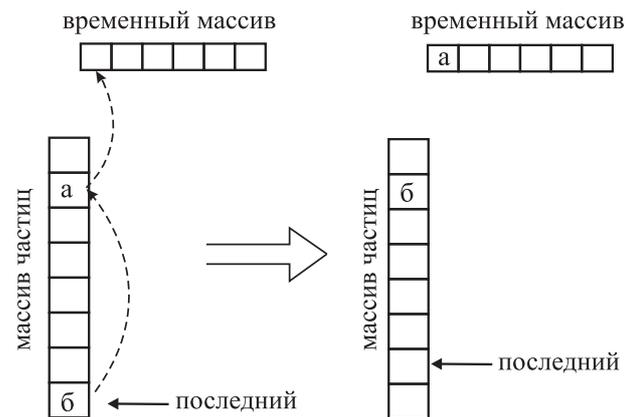


Рис. 4. При выбывании частицы “а” из процессора последняя частица в массиве “б” занимает ее место. Число частиц массива уменьшается

Так как за один шаг по времени частица не может пересечь более одной ячейки массива потенциала  $[\Phi]$ , межпроцессорные коммуникации, связанные с миграцией частиц в таких специально подобранных задачах, как диск (5), обычно не велики. Путем проведения расчетов было установлено, что общее число частиц во всех процессорах, участвующих в межпроцессорных коммуникациях, в среднем составляет примерно 0.001% от общего количества частиц. Например, в расчете на восьми процессорах с  $10^8$  частицами в обменах участвовало 104 частицы, что соответствует 0.0001%. Значение этого числа сильно зависит от самой задачи: если частицы активно двигаются в пространстве с большими скоростями, то пересылка больше, в противном случае — меньше. Кроме того, на число обменов влияет количество процессоров: чем больше процессоров, тем больше “швов” (рис. 1) и обменов, с падением эффективности распараллеливания.

Очевидно, что в процессе счета некоторые процессоры могут быть более загружены частицами, чем другие, а также теоретически возможно, что большинство частиц окажется в одном процессоре. В последнем случае массив частиц переполнится и работа программы завершится. Задание начальных данных должно предвидеть такие ситуации. Естественно, нельзя распределять частицы по пространству так, чтобы все они в своей динамике попали в один процессор. Наиболее часто встречается ситуация, когда некоторые процессоры поначалу вообще не имеют частиц. В качестве примера можно указать на задачи разлета частиц из небольшой области в окружающее пространство. Здесь возникает проблема балансировки загрузки процессоров, которая авторами в этой статье не затрагивается.

После определения новых координат частиц необходимо вычислять сеточные значения плотности. Каждый процессор вычисляет свой слой массива плотности  $[M]$  без обменов. Однако в конце этого этапа приходится обмениваться двумерными массивами вкладов в плотность от соседних процессоров для граничных слоев. Полученная плотность используется для решения уравнения Пуассона.

**4.2. Решение уравнения Пуассона.** Каждый процессор имеет в своей памяти только свой слой исходного массива плотности  $[M]$  для нахождения потенциала и только свой слой массива потенциала  $[\Phi]$ . Поэтому внутри процедуры решения уравнения Пуассона все процессоры многократно обмениваются частями массива  $[\Phi]$  со всеми процессорами. Эта часть алгоритма наиболее насыщена неизбежными меж-

процессорными коммуникациями. Для обращения к функциям быстрого преобразования Фурье в этой части алгоритма использовалась библиотека MKL через ее “обертку” для процедур свободно распространяемой библиотеки FFTW.

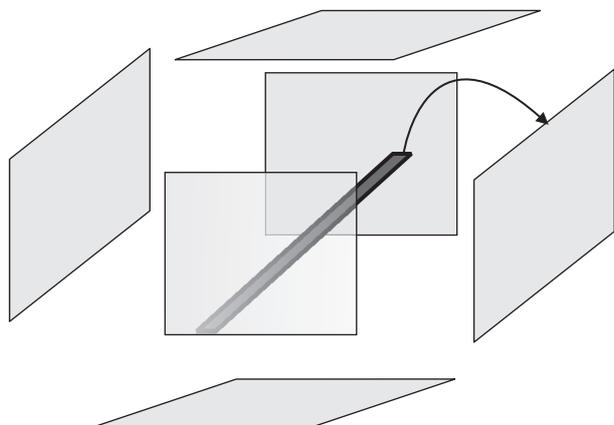


Рис. 5. Каждый процессор находит вклад своей области пространства в общий потенциал на границе. После этого происходит пересылка массивов граничных условий и суммирование вкладов

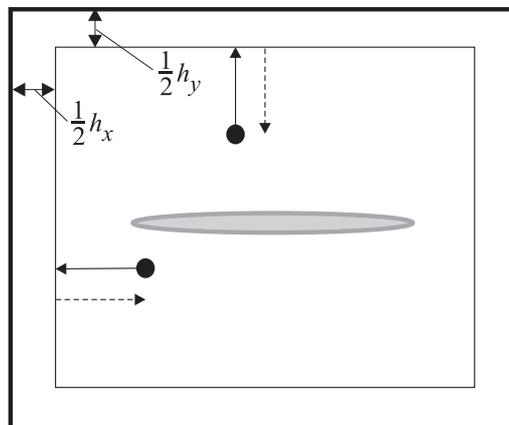


Рис. 6. На расстоянии половины ячейки до границы области частицы отражаются

Время (сек.) выполнения вычислительных процедур в зависимости от размерности задачи и числа процессоров  $N$

Размерность задачи	Время работы (сек.)							
	512 <sup>3</sup> узлов, 10 <sup>9</sup> частиц				1024 <sup>3</sup> узлов, 10 <sup>9</sup> частиц			
Число процессоров $N$	32	64	128	256	32	64	128	256
Граничные условия	10	6	2	1	290	171	85	45
Уравнение Пуассона	58	57	81	106	345	361	657	651
Динамика частиц	27	36	22	3	19	14	7	3
Вычисление плотности	4	2	1	1	4	4	3	4
Время выполнения одной итерации задачи	99	101	106	111	658	550	752	703

**4.3. Граничные условия.** Для решения уравнения Пуассона методом Фурье необходимо определить значение потенциала по формуле (4) на шести граничных плоскостях, распределенных по всем процессорам 3D-сетки. Для этого необходима также вся распределенная 3D-сетка. Нами было решено, что каждый процессор хранит граничные условия отдельно в шести двумерных массивах. Будем их называть “массивы граничных условий”. Каждый процессор находит вклад своей области пространства в общий потенциал на границе. После этого происходит пересылка массивов граничных условий и суммирование вкладов посредством коллективной редуцированной операции MPI (рис. 5). Для уменьшения числа операций вычисление потенциала происходит не в каждом узле массивов граничных условий, а с определенным интервалом на более грубой сетке. Значения остальных ячеек определяются линейной интерполяцией. Размер этой загрубленной сетки является входным параметром программы.

Сетки потенциала  $[\Phi]$  и плотности  $[M]$  в данной реализации полностью совпадают. Чтобы по формуле (4) не делить на ноль, частицы отражаются, не долетая половины ячейки до границы вычислительной области (рис. 6). Таким образом, частиц на границе нет и плотность там нулевая. Вклад граничных узлов в плотность нулевой, поэтому в алгоритме суммирования эти узлы можно не учитывать.

**4.4. Обсуждение эффективности параллельной программы.** Для параллельных программ принято оценивать эффективность. В таблице представлено время работы отдельных вычислительных

процедур алгоритма в зависимости от числа процессоров и размерности задачи. В последней строчке записана сумма времен работы всех четырех процедур, что и составляет один шаг алгоритма по времени, не считая диагностику. Указанные в таблице времена не являются постоянными и могут несколько меняться от запуска к запуску. Измерения проводились на кластере MVS-100K в МСЦ (Межведомственный суперкомпьютерный центр РАН). Остальные параметры в расчетах брались теми же, что и в разделе “Численные эксперименты” настоящей статьи (см. ниже).

Из таблицы следует, что все процедуры, кроме решения уравнения Пуассона, показали ускорение при увеличении числа процессоров. Однако так как решение уравнения Пуассона занимает львиную долю работы алгоритма на заданных параметрах задачи, рассматриваемая реализация в целом не имеет роста эффективности при увеличении числа процессоров. Это связано с большими пересылками внутри процедуры, когда все процессоры обмениваются со всеми большими частями 3D-массивов. Плохая распараллеливаемость — известное свойство уравнений эллиптического типа. По-видимому, здесь мы сталкиваемся с ограничением на использованные прямые методы для решения уравнения Пуассона. Это ограничение, возможно, удастся преодолеть на путях использования многосеточных итерационных подходов. Однако у последних есть известный недостаток — необходимость большого числа итераций для достижения требуемой точности. Тем самым, исходя из данных таблицы, можно сделать вывод, что решать поставленную физическую задачу нужно на минимальном числе процессоров, в которое помещается данная задача. При этом время счета минимально, и программа позволяет получать необходимые физические результаты, в ряде случаев, с использованием контрольной точки.

**5. Численные эксперименты.** Основной целью данной работы являлось решение задачи на сетках  $1000^3$  узлов с числом частиц до  $10^3$  в ячейке. Однако практическое проведение расчетов с такими параметрами требует преодоления дополнительных организационных и технических проблем. Поэтому нами выбраны начальные условия в виде диска, что позволило примерно в 4 раза уменьшить расчетную сетку по  $z$ -координате. За счет этого удалось снизить общее число частиц до  $10^9$ . На рис. 7 приведены результаты моделирования динамики формирования сгустков в плоском диске из тел в их самосогласованном гравитационном поле при развитии неустойчивости Джинса. Нами исследовалось поведение численного решения при сгущении вычислительной сетки и увеличении числа частиц. Начальная плотность тел в диске задавалась согласно формуле (5), начальный радиус диска был равен 3, его масса — 1, центрального тела и внешних потенциалов не было, начальные дисперсии скоростей  $dv_r = 0.02$ ,  $dv_\varphi = 0.01$ ,  $dv_z = 0.1$ . Расчетная область задавалась как  $10 \times 10 \times 10$ . Для выбранных параметров характерное значение длины Джинса составляло  $\lambda_J \sim 0.1$ . Интерес представляют расчеты с шагом сетки

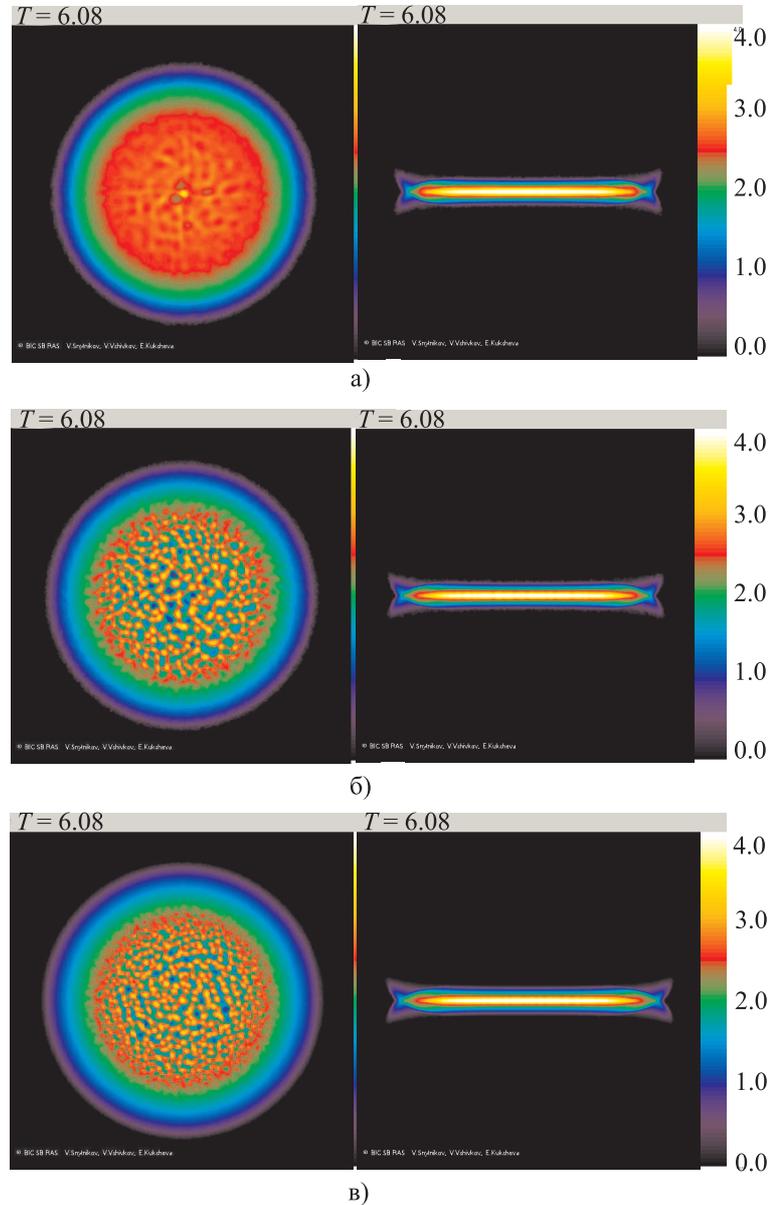


Рис. 7. Распределение логарифма поверхностной плотности вещества в экваториальной и меридиональной плоскостях диска в момент времени  $T = 6.08$  для трех расчетов

$h \ll \lambda_J$ .

На рис. 7 приведены распределения поверхностной плотности в экваториальной и меридиональной плоскостях диска в момент времени  $T = 6.08$  для трех расчетов. Рис. 7а — расчет выполнен на сетке с  $128^3$  узлов и  $10^8$  модельных частиц, рис. 7б — сетка  $256^3$ , число частиц тоже  $10^8$ , рис. 7в — сетка  $1024^2 \times 256$ , число частиц  $10^9$ . Расчеты 7а и 7б выполнялись с помощью программы, реализация которой описана в работе [5]. Расчет 7в выполнен в Сибирском суперкомпьютерном центре на кластере НКС-30Т с помощью нашей программы с декомпозицией области. В каждом узле кластера НКС-30Т имеется два четырехъядерных процессора Intel Xeon E5450 (3.00GHz) и 16 Гб памяти.

Из приведенных рисунков следует, что макроскопическая динамика тел для всех вариантов близка друг к другу. Частицы разлетаются в самосогласованном гравитационном поле, толщина диска увеличилась, плотность и температура (дисперсии) функции распределений по скоростям упали. В центральной области диска возникли условия для развития неустойчивости Джинса в бесстолкновительной системе. Однако на сетке  $128^3$  с  $h \sim \lambda_J$  (рис. 7а) и средним числом модельных частиц в ячейке порядка  $10^3$  обнаруживается только тенденция к формированию сгустков. Уменьшение  $h$  в 2 раза на сетке  $256^3$  с выполнением условия  $h < \lambda_J$  позволило выявить формирование сгустков на масштабе  $\lambda_J$ , хотя число частиц в ячейке уменьшилось в 2–4 раза. Дальнейшее уменьшение шага  $h$  в 4 раза по двум направлениям с заданием сетки  $1024^2 \times 256$  узлов и увеличение общего числа частиц в 10 раз для приближенного сохранения числа частиц в ячейке  $\sim 10^3$  позволило надежно рассчитать структуру сгустков, возникающую при развитии неустойчивости Джинса на ее нелинейных стадиях (рис. 7в).

**6. Заключение.** Таким образом, нами предложен алгоритм и разработана параллельная программа для решения трехмерных задач гравитационной физики без столкновений на сетках масштаба  $1000^3$  для существующих кластерных суперЭВМ. Распараллеливание основано на декомпозиции пространственной области. Как показали расчеты на доступных нам кластерах, программа позволяет проводить необходимые вычислительные эксперименты на суперкомпьютерах с распределенной памятью.

Авторы выражают благодарность академику В. Н. Пармону за поддержку и интерес к данному исследованию.

#### СПИСОК ЛИТЕРАТУРЫ

1. Хокни Р., Иствуд Дж. Численное моделирование методом частиц. М.: Мир, 1987.
2. Снытников В.Н., Вишнев В.А., Кукушева Э.А., Никитин С.А., Неупокоев Е.В., Снытников А.В. Трехмерное численное моделирование нестационарной гравитирующей системы многих тел с газом // Письма в астрономический журнал. 2003. **29**, № 12. 146–160.
3. Боронина М.А., Вишнев В.А., Левичев Е.Б., Никитин С.А., Снытников В.Н. Алгоритм для трехмерного моделирования ультрарелятивистских пучков // Вычислительные методы и программирование. 2007. **8**, № 2. 203–210.
4. Вишнев В.А., Снытников В.Н., Снытников Н.В. Моделирование трехмерной динамики вещества в гравитационном поле на многопроцессорных ЭВМ // Вычислительные технологии. 2006. **11**, № 2. 15–27.
5. Кукушева Э.А., Снытников В.Н. Параллельный алгоритм и программа Key7D для решения нестационарных трехмерных задач гравитационной физики // Вычислительные технологии. 2007. **12**, № 1. 35–44.
6. Кукушева Э.А., Снытников В.Н. Параллельная реализация фундаментального решения уравнения Пуассона // Вычислительные технологии. 2005. **10**, № 4. 63–71.

Поступила в редакцию  
12.04.2010