

Масштабируемый параллельный алгоритм для моделирования трехмерной динамики гравитирующих систем методом частиц*

Н.В. Снытников¹

Институт Вычислительной Математики и Математической Геофизики СО РАН¹

Предложен параллельный алгоритм для численного решения системы уравнений Власова-Пуассона методом частиц. Используется новый метод динамической балансировки процессоров, которые распределяются между подобластями в соответствии с реальным числом модельных частиц. Метод декомпозиции области позволяет комбинировать сеточный (эйлеров) метод для решения уравнения Пуассона с лагранжевым методом частиц для решения уравнения Власова. Он учитывает физические особенности задач моделирования нестационарных вращающихся дисков (двумерных и трехмерных).

Ключевые слова: гравитирующие системы, уравнение Пуассона, уравнение Власова, метод частиц, динамическая балансировка.

1. Введение

Суперкомпьютерное моделирование динамики гравитирующих систем (таких как галактики или окологзвездные диски) является одной из наиболее трудных и актуальных проблем в современной вычислительной астрофизике. Хотя однопроцессорные численные методы были созданы и успешно используются уже более 30 лет [1–5], разработка соответствующих им параллельных алгоритмов требует значительных модификаций, связанных с адаптацией к новым аппаратным и программным архитектурам суперкомпьютеров. Другие трудности возникают из-за разнообразия моделируемых астрофизических процессов [6–8], для которых существуют специализированные программные пакеты [9–11] не всегда применимые для решения смежных задач.

В данной статье предложен параллельный алгоритм для моделирования вращающихся бесстолкновительных гравитирующих систем методом частиц в ячейках, и приведены результаты численных экспериментов (детальное описание алгоритмов приведено в [12]). Алгоритм может быть применен как для модели тонкого диска (когда отсутствует движение вещества в вертикальном направлении), так и для полностью трехмерной модели [13, 14]. Новым в этом подходе является метод динамической балансировки загрузки процессоров при вычислении траектории модельных частиц, которые в случае вращающихся систем могут многократно переходить между подобластями (и соответствующим им процессорам).

Решаемая система уравнений состоит из уравнения Власова для функции распределения вещества $f = f(t, \mathbf{r}, \mathbf{u})$, зависящей от времени t , пространственной координаты \mathbf{r} и вектора скорости \mathbf{u} , с заданным начальным значением $f^0(\mathbf{r}, \mathbf{u})$ (в виде некоторого вращающегося диска):

$$\frac{\partial f}{\partial t} + \mathbf{u} \frac{\partial f}{\partial \mathbf{r}} - \nabla \Phi(t, \mathbf{r}) \frac{\partial f}{\partial \mathbf{u}} = 0, \quad f(0, \mathbf{r}, \mathbf{u}) = f^0(\mathbf{r}, \mathbf{u})$$

и уравнения Пуассона для гравитационного потенциала $\Phi = \Phi(t, \mathbf{r})$ с краевыми условиями для изолированных систем:

$$\Delta \Phi(t, \mathbf{r}) = 4\pi G \rho(t, \mathbf{r}), \quad \Phi(t, \mathbf{r})|_{|\mathbf{r}| \rightarrow \infty} = 0,$$

*Разработка теоретических основ алгоритма выполнена при поддержке Российского Научного Фонда, грант № 14-11-00485. Программная реализация алгоритма и тестовые эксперименты выполнены при поддержке РФФИ, гранты № 16-07-00916, № 14-07-00241, № 14-01-00392.

где G – гравитационная постоянная.

Замыкает систему следующее уравнение для плотности $\rho = \rho(t, \mathbf{r})$, которое связывает ее с функцией распределения f :

$$\rho(t, \mathbf{r}) = \int_{\mathbf{u}} f(t, \mathbf{r}, \mathbf{u}) d\mathbf{u}.$$

Для решения уравнения Власова используется метод частиц в ячейках [1, 2], а для решения уравнения Пуассона — метод свёртки. Соответствующий параллельный алгоритм должен подразделять исходную вычислительную область на такие подобласти, что каждая из них могла бы быть обработана одним вычислительным устройством с 1-4 ГБ памяти, и распределять модельные частицы между процессорами так, чтобы они могли иметь доступ к значениям требуемых сеточных функций в подобластях, свободно двигаться между подобластями, и общее количество частиц, передаваемых между процессорами, было минимальным.

В типичном случае максимальное количество частиц, которое можно поместить в одно вычислительное устройство (CPU, GPU, Intel Phi), составляет 8-16 миллионов, а максимальное количество узлов сетки составляет $256 \times 256 \times 256$ для трехмерных задач и 4096×4096 для двумерных. Это означает, что большие вычислительные области (такие как $1024 \times 1024 \times 1024$) должны подразделяться на меньшие подобласти. В то же самое время модельные частицы могут многократно перелетать из одной области в другую в течение одного численного эксперимента (например, если рассматривается моделирование дисков на временных масштабах порядка десятков оборотов, где каждая частица двигается по эллиптической или эпициклической орбите вокруг общего центра масс), что создает проблему пересылок данных на каждом временном шаге. Кроме того, частицы могут быть распределены неравномерно между подобластями. Например, во вращающемся диске могут возникать гравитационные неустойчивости, которые приводят к появлению разнообразных структур — сгустков вещества или спиральных волн, которые способны передвигаться по всей вычислительной подобласти. Плотность вещества (и количество частиц) в этих структурах может быть на порядки больше фоновых значений. Следовательно, количество частиц в каждой подобласти может так же отличаться на порядки и меняться со временем, особенно в тех случаях, когда сгусток, содержащий большое количество частиц, двигается вокруг центра масс.

Разработанный параллельный алгоритм использует следующие свойства задачи и численного метода:

- Трудоемкость решения уравнения Власова (т.е. интегрирование траекторий PIC-частиц) существенно выше (от 10 до 100 раз), чем трудоемкость вычисления гравитационного потенциала. Это связано с тем, что среднее число модельных частиц на одну ячейку сетки в типичных расчетах находится в диапазоне $10 \div 100$.
- Чтобы удовлетворять условию Куранта устойчивости численного метода, частица может перелететь за один шаг не далее, чем в соседнюю ячейку. Это означает, что число частиц, которые должны перемещаться между смежными подобластями, намного меньше, чем общее число частиц в смежных подобластях (порядка $10 \cdot n_y$ для 2D задач и $10 \cdot n_y n_z$ для 3D задач, где n_y, n_z — это число всех граничных узлов между подобластями).

Общая схема параллельного алгоритма описана в разделе 2, а детальное описание алгоритмов приведено в статье [12]. Раздел 3 посвящен описанию метода вычисления гравитационного потенциала. В разделе 4 приводятся результаты экспериментов.

2. Параллельный алгоритм

Исходная вычислительная область (2D или 3D) равномерно подразделяется на K под-областей одинакового размера S_1, S_2, \dots, S_K в направлении X . Группа процессоров G_k назначается каждой подобласти S_k (см. Рис. 1). Количество процессоров в группе G_k равно P_k , $P_k \geq 1$. Общее количество процессоров равно $P = \sum P_k$. Количество узлов (ячеек) в каждой вычислительной подобласти выбирается таким образом, что все сеточные функции (гравитационный потенциал, плотность, сила) должны полностью помещаться в оперативную память одного вычислительного устройства и оставить в памяти достаточно места для хранения массивов с частицами (т.е. их пространственных координат и скоростей). Обычно это число равно $0.5 \div 2$ миллионов узлов (соответствует сетке 128^3 или 1024^2). Количество процессоров в группе G_k выбирается так, чтобы обеспечить приблизительно одинаковое количество частиц на одном процессоре (см. Алгоритм 2 в [12]). Это означает, что если под-область S_{k_1} содержит больше частиц, чем подобласть S_{k_2} , то число процессоров P_{k_1} в группе G_{k_1} будет больше или равно количеству P_{k_2} в группе G_{k_2} . В каждой группе процессоров G_k и соответствующей подобласти S_k имеется главный процессор g_k^0 , который не может быть переименован любой другой подобласти во время вычислительного эксперимента (даже в том случае, когда подобласть S_k не содержит частиц).

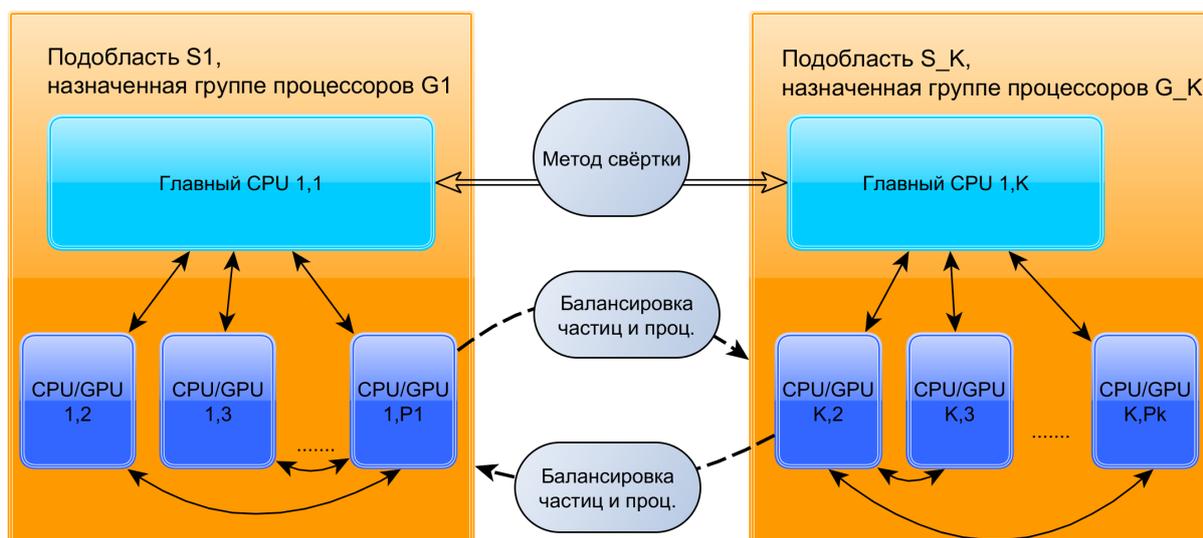


Рис. 1. Схема организации параллельных вычислений и балансировки нагрузки между процессорами. Главные процессоры в группе предназначены для вычисления гравитационного потенциала методом свёртки. Остальные процессоры в каждой группе занимаются вычислением новых координат частиц

В результате основной алгоритм выглядит следующим образом:

1. В нулевой момент времени ($t = 0$):
 - (a) для каждой подобласти S_k вычисляется общее количество частиц, соответствующее числу процессоров в группе G_k , и количество частиц, назначенное каждому из процессоров,
 - (b) на каждом процессоре выделяется память для массивов координат и скоростей частиц,
 - (c) выполняется генерация начальных координат и скоростей PIC частиц в области (на каждом процессоре) в соответствии с заданной пользователем функцией распределения.

2. Вычисляется сеточная функция плотности для PIC частиц. Выполняется суммирование плотности на главном процессоре группы: вызывается функция *MPI_Reduce* для процессоров из группы.
3. Вычисляется гравитационный потенциал с помощью параллельного метода свертки. В том числе выполняются коммуникации *MPI_Alltoall* между главными процессорами.
4. Для каждого процессора вычисляются новые координаты частиц для следующего шага. Межпроцессорные коммуникации на этом шаге отсутствуют.
5. Вычисляется количество частиц, которое должно находиться в каждой подобласти S_k на следующем шаге. Вычисляется количество процессоров, которое должно быть назначено каждой подобласти.
6. Перераспределяются частицы между процессорными группами. Выполняются межпроцессорные коммуникации *MPI_Gather* и *MPI_Scatter* для процессоров из группы.
 - (a) На каждом процессоре могут находиться три типа частиц: (а) частицы, которые остались внутри данной подобласти, (б) частицы, которые должны переместиться в левую смежную подобласть, и (в) частицы, которые должны переместиться в правую смежную подобласть. Общее количество перемещаемых между подобластями частиц невелико из-за необходимости выполнения условия Куранта: оно не может быть больше, чем число частиц, которые содержались в граничных ячейках подобласти. На практике это число существенно меньше и составляет менее 0.1% от общего количества частиц.
 - (b) Вычисляем новое число частиц для каждой из подобластей.
 - (c) Вычисляем число процессоров для каждой подобласти.
 - (d) Определяем те процессоры в каждой группе, которые должны быть переназначены на соседние подобласти (то есть переданы другим группам). Передаем с них те частицы, которые должны остаться в данной подобласти.
 - (e) Передаем частицы, которые перешли из данной подобласти в соседние подобласти.
7. Инкрементируется номер временного шага и выполняется переход на Шаг 2.

3. Метод свёртки для вычисления гравитационного потенциала

Для вычисления гравитационного потенциала был реализован метод свёртки, предложенный Хокни [1]. Его суть заключается в том, что вместо задачи Дирихле для уравнения Пуассона в бесконечной области:

$$\begin{aligned}\Delta\Phi(\mathbf{x}) &= \rho(\mathbf{x}), \\ \Phi(\mathbf{x})|_{\mathbf{x}\rightarrow\infty} &= 0,\end{aligned}\tag{1}$$

выполняется эффективное вычисление интеграла, представляющего фундаментальное решение уравнения Пуассона:

$$\Phi(\mathbf{x}_0) = - \int \frac{\rho(\mathbf{x})d\mathbf{x}}{|\mathbf{x}_0 - \mathbf{x}|}.\tag{2}$$

В дискретном случае в декартовой системе координат на равномерной сетке с числом узлов $N_x \times N_y \times N_z$ и сеточными шагами h_x, h_y, h_z решение будет записываться в следующем виде:

$$\Phi(x_0, y_0, z_0) = - \sum_{i=1}^{N_x-1} \sum_{j=1}^{N_y-1} \sum_{k=1}^{N_z-1} \frac{q_{i,j,k}}{\sqrt{(x_i - x_0)^2 + (y_j - y_0)^2 + (z_k - z_0)^2}},\tag{3}$$

где $q_{i,j,k} = \rho_{i,j,k} \cdot (h_x h_y h_z)$ — значения зарядов (масс), находящихся в узлах сетки.

Воспользовавшись дискретным аналогом теоремы о свёртке и алгоритмом быстрого преобразования Фурье, можно вычислить сумму (3) за $O(N_x N_y N_z (\log_2 N_x + \log_2 N_y + \log_2 N_z))$ операций, что намного быстрее прямого способа вычисления фундаментального решения для уравнения Пуассона и соответствует трудоёмкости решения задачи Дирихле для уравнения Пуассона методом разделения переменных [15].

Для того, чтобы применить дискретное преобразование Фурье, необходимо устранить неопределенность в точке $\mathbf{x} = \mathbf{x}_0$ и определить функции K и ρ так, чтобы они стали периодическими. Это решается с помощью определения сеточной функции K следующим образом:

$$K(x, y, z) = \begin{cases} \frac{1}{0.5 \min(h_x, h_y, h_z)}, & \sqrt{x^2 + y^2 + z^2} = 0, \\ \frac{1}{\sqrt{x^2 + y^2 + z^2}}, & \sqrt{x^2 + y^2 + z^2} > 0. \end{cases}$$

Вторая проблема решается с помощью введения фиктивных дополнительных подобластей, дублирующих область решения по каждому направлению в два раза, и доопределения в них функции K так, чтобы она стала периодической во всей новой области, а функция ρ — равной нулю [16].

Параллельная реализация данного алгоритма заключается в разбиении на подобласти и применении метода транспозиции данных (стандартного подхода для многомерного преобразования Фурье [18]):

1. Вычислительная область в 2D или 3D подразделяется на подобласти в направлении X.
2. Применяется прямое быстрое преобразование Фурье (БПФ) к сеточным функциям плотности ядра потенциала в направлении Y и в направлении Z.
3. Выполняется транспозиция «слоев» из направления Y в направление X.
4. Применяется прямое БПФ в направлении X. Перемножается образ сеточной функции ядра на образ функции плотности. Применяется обратное БПФ в направлении X к полученному результату.
5. Выполняется обратная транспозиция «слоев» из направления X в направление Y.
6. Применяется обратное БПФ в направлении Z и обратное БПФ в направлении Y.

Для выполнения быстрого преобразования Фурье использовалась библиотека FFTW [17].

4. Тестовые результаты

Тестовые эксперименты проводились на суперкомпьютерах Сибирского суперкомпьютерного центра, Межведомственного суперкомпьютерного центра и суперкомпьютера «Ломоносов» в МГУ. В Таблице 1 представлены результаты тестовых экспериментов для параллельного метода свертки в случае плоского двумерного распределения для сеток 16384×16384 и 32768×32768 . Данный алгоритм обладает хорошей масштабируемостью на большое число процессоров и позволяет вычислять гравитационный потенциал на сетках с числом узлов более 1 млрд. за время менее 1 секунды.

В Таблице 2 представлены результаты «нагрузочного» эксперимента для расчета одного временного шага. В качестве начального распределения $f^0(\mathbf{r}, \mathbf{u})$ задавался диск Маклорена с твердотельным вращением и разными дисперсиями скоростей частиц. В качестве технических параметров для полностью трехмерной модели задавалась сетка $1024 \times 1024 \times 1024$,

Таблица 1. Эксперименты по оценке производительности алгоритма для сетки 16384×16384 и 32768×32768 при разном количестве процессоров суперкомпьютера «Ломоносов» МГУ.

Число процессоров N	Время (с) 16384×16384	Время (с) 32768×32768
64	1.75	—
128	1.1	—
256	0.5	2.4
512	0.35	1.3
1024	0.25	0.65

Таблица 2. Время расчета одного временного шага для параллельного метода решения системы Власова-Пуассона. Относительное разбиение времени по основным программным процедурам и межпроцессорным коммуникациям сгруппировано следующим образом: *Particles*: вычисление координат частиц (межпроцессорные коммуникации отсутствуют), *MPI_Reduce*: суммирование плотности для подобласти, *MPI_Alltoall*: вычисление гравитационного потенциала, *FFT*: быстрое преобразование Фурье (межпроцессорные коммуникации отсутствуют), *MPI_Bcast*: пересылка потенциала от главного процессора группы на дочерние, *MPI_Gatherv*, *MPI_Scatterv*: балансировка процессоров и перераспределение частиц.

Общее время	7 секунд
Particles	60%
MPI_Reduce	3%
MPI_Alltoall	7.5%
FFT	7.5%
MPI_Bcast	7%
MPI_Gatherv, MPI_Scatterv	15%

10 миллиардов частиц и 1024 процессора. Общее количество подобластей и групп процессоров составляло 128.

Таблица показывает, что основная часть времени тратится на последовательные вычисления, в то время как на межпроцессорные коммуникации уходит не более 40%.

5. Заключение

В статье представлен параллельный алгоритм для решения системы уравнений Власова-Пуассона методом частиц. Показано, что данный алгоритм обладает хорошей производительностью и может применяться для суперкомпьютерного моделирования нестационарных задач астрофизики (динамики вращающихся галактик и газопылевых протопланетных дисков), для которых необходимо проведение серийных расчетов с десятками тысяч временных шагов.

Литература

1. Hockney, R.W., Eastwood, J.W. Computer Simulation Using Particles. McGraw-Hill. New York. 1981. 540 p.
2. Березин Ю.А., Вшивков В.А. Метод частиц в динамике разреженной плазмы. Наука. Новосибирск. 1980. 96 с.
3. Barnes J.E., Hut P. A Hierarchical $O(N \log N)$ Force-Calculation Algorithm // Nature. 1986. Vol. 324, P. 446-449.
4. Gingold R.A. and Monaghan J.J. Smoothed particle hydrodynamics - Theory and application to non-spherical stars // Monthly Notices of the Royal Astronomical Society. 1977. Vol. 181. P. 375-389.
5. Colella P., Woodward P.R. The Piecewise Parabolic Method (PPM) for gas-dynamical simulations // Journal of Computational Physics. 1984. Vol. 54, No. 1. P. 174-201.
6. Dubey A., Antypas K., Ganapathyc M.K., et al. Extensible component-based architecture for FLASH, a massively parallel, multiphysics simulation code // Parallel Computing. 2009. Volume 35. P. 512-522.
7. Springel V., Yoshida N., White S.D.M. GADGET: a code for collisionless and gasdynamical cosmological simulation // New Astronomy. 2001. Vol. 6. P. 79-117.
8. Pearce F.R., Couchman H.M.P. Hydra: a parallel adaptive grid code // New Astronomy. 1997. Vol. 2, No. 5. P. 411-427.
9. Springel V. et al. Simulations of the formation, evolution and clustering of galaxies and quasars // Nature. 2005. Vol. 435. P. 629.
10. Klypin A.A. et al. Dark Matter Halos in the Standard Cosmological Model: Results from the Bolshoi Simulation // Astrophysical Journal. 2011. Vol. 740. P. 102.
11. Feng Y., Di Matteo T., Croft R.A.C., Bird S., Battaglia N., Wilkins S. BlueTides: First galaxies and reionization // Monthly Notice of Royal Astronomical Society. 2015 (submitted)
12. Снытников Н.В., Вшивков В.А. Метод декомпозиции области для суперкомпьютерного моделирования гравитирующих систем // СУПЕРКОМПЬЮТЕРНЫЕ ДНИ В РОССИИ. Труды международной конференции. Под редакцией чл.-корр. РАН Вл.В. Воеводина. 2015. С. 572-580.

13. Снытников В.Н., Вшивков В.А., Кукшева Э.А., Неупокоев Е.В., Никитин С.А., Снытников А.В. Трехмерное численное моделирование нестационарной гравитирующей системы многих тел с газом // Письма в астрономический журнал. 2004. Т. 30, №.2. 146-160.
 14. Вшивков В.А., Снытников В.Н., Снытников Н.В. Моделирование трехмерной динамики вещества в гравитационном поле на многопроцессорных ЭВМ // Вычислительные технологии. 2006, Т.11. N.2. С.15-27.
 15. Самарский А.А., Андреев В.Б. Разностные методы для эллиптических уравнений. М.: Наука, 1976. 352 с.
 16. Eastwood J.W., Brownrigg D.R.K. Remarks on the Solution of Poisson's Equation for Isolated Systems // Journal of Computational Physics. 1979. Vol. 32, P.24-38.
 17. M. Frigo, and S.G. Johnson. FFTW software. URL: <http://www.fftw.org> (дата обращения: 01.02.2016)
 18. Ayala O., Wang L.P. Parallel implementation and scalability analysis of 3D Fast Fourier Transform using 2D domain decomposition // Parallel Computing. 2013. Vol.39. P. 58-77
-

Scalable parallel algorithm for simulation of 3D dynamics of gravitating systems using particle-in-cell method

N.V. Snytnikov¹

Institute of Computational Mathematics and Mathematical Geophysics¹

We developed parallel algorithm for solving Vlasov-Poisson systems of equations using particle-in-cell method. It uses new technique of dynamic load balancing for processors, which are distributed between subdomains in correspondance with the number of modeling particles located in the subdomain. Domain decomposition method combines grid (eulerian) method for solving Poisson equation with lagrangian paricle method for solving Vlasov equation. It takes into account physical features of the modeling nonstationary rotating disks (both 2D and 3D).

Keywords: gravitating systems, Poisson equation, Vlasov equation, particle-in-cell method, dynamic load balancing.

References

1. Hockney, R.W., Eastwood, J.W. Computer Simulation Using Particles. McGraw-Hill. New York. 1981. 540 p.
2. Berezin Ju.A., Vshivkov V.A. Metod chastic v dinamike razrezhennoj plazmy. [Berezin Yu.A. Vshivkov V.A. Particle-in-cell method in the dynamics of low-density plasma] Nauka. Novosibirsk, 1980. 96 p.
3. Barnes J.E., Hut P. A Hierarchical O(NlogN) Force-Calculation Algorithm // Nature. 1986. Vol. 324, P. 446-449.
4. Gingold R.A. and Monaghan J.J. Smoothed particle hydrodynamics - Theory and application to non-spherical stars // Monthly Notices of the Royal Astronomical Society. 1977. Vol. 181. P. 375-389.
5. Colella P., Woodward P.R. The Piecewise Parabolic Method (PPM) for gas-dynamical simulations // Journal of Computational Physics. 1984. Vol. 54, No. 1. P. 174-201.
6. Dubeya A., Antypasb K., Ganapathyc M.K., et al. Extensible component-based architecture for FLASH, a massively parallel, multiphysics simulation code // Parallel Computing. 2009. Volume 35. P. 512-522.
7. Springel V., Yoshida N., White S.D.M. GADGET: a code for collisionless and gasdynamical cosmological simulation // New Astronomy. 2001. Vol. 6. P. 79-117.
8. Pearce F.R., Couchman H.M.P. Hydra: a parallel adaptive grid code // New Astronomy. 1997. Vol. 2, No. 5. P. 411-427.
9. Springel V. et al. Simulations of the formation, evolution and clustering of galaxies and quasars // Nature. 2005. Vol. 435. P. 629.
10. Klypin A.A. et al. Dark Matter Halos in the Standard Cosmological Model: Results from the Bolshoi Simulation // Astrophysical Journal. 2011. Vol. 740. P. 102.
11. Feng Y., Di Matteo T., Croft R.A.C., Bird S., Battaglia N., Wilkins S. BlueTides: First galaxies and reionization // Monthly Notice of Royal Astronomical Society. 2015 (submitted)

12. Snytnikov N.V., Vshivkov V.A. Metod dekompozitsii oblasti dlya superkomp'yuternogo modelirovaniya gravitiruyushchikh sistem. [Domain decomposition method for supercomputer simulation of gravitating systems] // Russian Supercomputing Days. Proceedings of International Conference. 2015. P. 572-580.
13. Snytnikov V.N. et al. Three-Dimensional Numerical Simulation of a Nonstationary Gravitating N-Body System with Gas // *Astronomy Letters*. 2004. Vol. 30. P. 124-137.
14. Vshivkov V.A., Snytnikov V.N., Snytnikov N.V. Modelirovanie trekhmernoy dinamiki veshchestva v gravitatsionnom pole na mnogoprotsessornykh EVM [Simulation of three-dimensional dynamics of matter in gravitational field with the use of multiprocessor computer]. // *Vychislitelnye tekhnologii (Computational Technologies)*. 2006. Vol. 11, No. 2. P. 15-27.
15. Samarskii A.A., Andreev V.B. Raznostnye metody dlya ellipticheskikh uravneniy [Difference methods for elliptic equations]. Moscow: Nauka, 1976. 352 p.
16. Eastwood J.W., Brownrigg D.R.K. Remarks on the Solution of Poisson's Equation for Isolated Systems // *Journal of Computational Physics*. 1979. Vol. 32, P.24-38.
17. Frigo M., and Johnson S.G. FFTW software. URL: <http://www.fftw.org> (access date: 01.02.2016)
18. Ayala O., Wang L.P. Parallel implementation and scalability analysis of 3D Fast Fourier Transform using 2D domain decomposition // *Parallel Computing*. 2013. Vol.39. P. 58-77.