Оптимизация переупорядочивания модельных частиц при реализации метода частиц в ячейках на GPU

А. А. Романенко 1, **А. В. Снытников** 1, 2

¹ Новосибирский государственный университет
² Институт вычислительной математики и математической геофизики СО РАН Новосибирск, Россия

Аннотация

Представлено описание реализации метода частиц в ячейках на GPU. Основным недостатком метода, с точки зрения затрат по времени, является функция переупорядочивания частиц между ячейками. Изложена оригинальная методика оптимизации данного этапа расчета, позволяющая избавиться от атомарных операций. Приведены результаты тестирования производительности на ряде современных графических процессоров.

Ключевые слова

трехмерная модель, плазма, GPU, оптимизация

Благодарности

Работа выполнена в рамках государственного задания ИВМиМГ СО РАН (проект № 0315-2019-0009), реализация на СUDA выполнена при поддержке гранта РФФИ № 18-07-00364, оптимизация алгоритма выполнена при частичной поддержке гранта РФФИ № 19-07-00085

Для цитирования

Романенко А. А., Снытников А. В. Оптимизация переупорядочивания модельных частиц при реализации метода частиц в ячейках на GPU // Вестник НГУ. Серия: Информационные технологии. 2019. Т. 17, № 1. С. 82–89. DOI 10.25205/1818-7900-2019-17-1-82-89

Reordering Optimization for GPU Implementation of Particle-in-Cell Method

A. A. Romanenko¹, A. V. Snytnikov^{1,2}

¹ Novosibirsk State University
² Institute of Computational Mathematics and Mathematical Geophysics SB RAS
Novosibirsk, Russian Federation

Abstract

Particle-In-Cell (PIC) method is widely used for plasma simulation and the GPUs appear to be the most efficient way to run this method. In this work we propose a technique that enables to speedup one of the most time-consuming operations in the GPU implementation of the PIC method. The operation is particle reordering, or redistribution of particles between cells, which is performed after pushing. The reordering operation provides data locality which is the key performance issue of the PIC method. We propose to divide the reordering into two stages. First, gather the particles that are going to leave a particular cell into buffer arrays, the number of arrays being equal to number of neighbour cells (26 for 3D case). Second, each neighbour cell copies the particles from the necessary array to her own particle array. Since the second operation is done in 26 threads independently with no synchronization, waiting and involves no critical sections, semaphores, mutexes, atomic operations etc. It results in the more than 10 times reduce of the reordering time compared to the straightforward reordering algorithm. Futhermore, we eliminate the 26 buffer arrays in the following way: the particles are just labeled instead of moving to buffer array. It enables to keep all the advantages with no memory wasted for buffer arrays.

Keywords

3D model, plasma, GPU, optimization

Acknowledgements

This work was conducted within the framework of the budget project no. 0315-2019-0009 for ICMMG SB RAS. CUDA implementation was supported by RFBR grant no. 18-07-00364, and the optimization of the algorithm was partially supported by RFBR grant no. 19-07-00085

© А. А. Романенко, А. В. Снытников, 2019

For citation

Romanenko A. A., Snytnikov A. V. Reordering Optimization for GPU Implementation of Particle-in-Cell Method. *Vestnik NSU. Series: Information Technologies*, 2019, vol. 17, no. 1, p. 82–89. (in Russ.) DOI 10.25205/1818-7900-2019-17-1-82-89

Введение

Для численного моделирования плазмы часто используется метод частиц в ячейках (англ. Particle-in-Cell method, PIC) [1; 2]. Моделирование плазменной турбулентности может проводится либо в гидродинамическом приближении — путем добавки в уравнения магнитной гидродинамики (МГД) дополнительных слагаемых, обеспечивающих аппроксимацию турбулентности, либо в кинетическом приближении — путем решения непосредственно кинетического уравнения Власова (или Больцмана). Первый вариант является в данном случае неприемлемым в силу того, что характер возникающей турбулентности экспериментально и теоретически слабо изучен, а также в силу того, что плазма (в данном случае — субтермоядерная плазма установки ГОЛ-3, ИЯФ СО РАН [3] и плазма перспективных установок управляемого термоядерного синтеза) не является даже приближенно равновесной, так что уравнения МГД не могут быть использованы.

Кинетическое уравнение может быть решено либо методом частиц в ячейках, либо прямым конечно-разностным методом. В обоих случаях существует пространственная сетка для решения уравнений Максвелла. При использовании метода частиц в каждую ячейку сетки добавляются модельные частицы, уравнения движения которых представляют собой уравнения характеристик для кинетического уравнения Власова, при использовании прямого конечно-разностного метода вводится дополнительная сетка в пространстве скоростей, так что возникает сетка в 6-мерном пространстве. Таким образом, вариант с использованием метода частиц является более затратным по количеству операций, но значительно более экономичным по памяти по сравнению с прямым конечно-разностным методом.

Итак, только метод частиц в ячейках обеспечивает возможность решения задачи. Все варианты построения более хорошего (быстрого) метода на данный момент связаны с внесением тех или иных некорректных упрощений в физическую постановку задачи. По сути дела, расчеты с использованием метода частиц в ячейках проводятся именно с целью проверки того, какая из упрощенных физических моделей (МГД, модель на основе первых моментов уравнения Больцмана...) будет в данном случае корректной.

Вычислительные эксперименты показывают, что при реализации метода частиц в ячейках наиболее важной характеристикой оборудования является доступ к оперативной памяти. С этой точки зрения целесообразно использовать для реализации метода частиц в ячейках гибридные вычислительные системы на базе графических процессоров (GPU) [4], которые обладают высокой пропускной способностью по доступу к памяти за счет своей архитектуры и таких аппаратных средств, как текстурный кэш и разделяемая память. Кроме того, большое количество ядер на GPU, так же как и на ускорителях Intel Xeon Phi, может быть использовано для создания более быстрой реализации метода частиц, чем на «обычных» многоядерных процессорах, таких как Intel Xeon, Intel Nehalem, IBM Power и др.

Кроме того, необходимость перехода на GPU диктуется тем, что среди наиболее мощных компьютеров мира имеется тенденция к увеличению доли гибридных суперЭВМ. В 2018 г. это выглядит следующим образом: суперЭВМ на основе Intel Xeon Phi и Nvidia Volta (или Nvidia Pascal) в Тор500 и Тор10: № 1, 2, 5, 6, 7, 9 1 . Такая тенденция поддерживается тем, что энергоэффективность гибридных систем выше, чем систем, имеющих традиционную архитектуру.

¹ Top500. Список 500 наиболее мощных компьютеров мира (ноябрь 2018). URK: https://www.top500.org/lists/2018/11/.

В последнее время появилось много программных пакетов для моделирования динамики плазмы на основе метода частиц в ячейках, использующих графические ускорители (GPU), например, PIConGPU [5] и ALaDyn [6] и др. [7; 8].

Вместе с тем многие имеющиеся коды для расчета динамики плазмы на GPU ориентированы на конкретный вычислительный либо алгоритмический вопрос и с точки зрения физических расчетов носят предварительный характер. В итоге имеется или корректно работающий численный код, которому не хватает вычислительной мощности, или очень быстро работающий код для гибридной суперЭВМ, физическая корректность результатов которого вызывает сомнения, потому что в его разработке участвовали только специалисты по суперЭВМ. Описываемый в данной работе код разрабатывался при участии физиков [9; 10], что освобождает его от упомянутых выше недостатков.

Постановка задачи

Математическая модель высокотемпературной бесстолкновительной плазмы представляется кинетическим уравнением Власова и системой уравнений Максвелла [1], которые в безразмерной форме имеют следующий вид:

$$\begin{split} \frac{\partial f_{i,e}}{\partial t} + \mathbf{v} \frac{\partial f_{i,e}}{\partial \mathbf{r}} + \mathbf{F}_{i,e} \frac{\partial f_{i,e}}{\partial \mathbf{p}} &= 0, \quad \mathbf{F}_{i,e} = q_{i,e} \bigg(\mathbf{E} + \frac{1}{c} \big[\mathbf{v}, \mathbf{B} \big] \bigg), \\ \mathbf{rot} \ \mathbf{B} &= \frac{4\pi}{c} \, \mathbf{j} + \frac{1}{c} \frac{\partial \mathbf{E}}{\partial t}, \\ \mathbf{rot} \ \mathbf{E} &= -\frac{1}{c} \frac{\partial \mathbf{B}}{\partial t}, \\ \mathbf{div} \ \mathbf{E} &= 4\pi \rho, \\ \mathbf{div} \ \mathbf{B} &= 0. \end{split}$$

Здесь индексами i и e помечены величины, относящиеся к ионам и электронам соответственно; $q_e = -1$, $q_i = m_e/m_i$; $f_{i,e}(t,\mathbf{r},\mathbf{p})$ — функция распределения частиц; $m_{i,e}$, $\mathbf{p}_{i,e}$, $\mathbf{r}_{i,e}$ — масса, импульс, положение иона или электрона; \mathbf{E} , \mathbf{B} — напряженности электрического и магнитного полей. Для перехода к безразмерному виду в качестве единиц используются следующие базовые величины:

- скорость света $c = 3 \times 10^{10}$ см/с;
- плотность плазмы $n_0 = 10^{14} \,\mathrm{cm}^{-3}$;
- плазменная электронная частота $\omega_e = 5.6 \times 10^4 \, n^{1/2} \, \mathrm{c}^{-1}$.

В начальный момент времени в трехмерной области решения, имеющей форму прямо-угольного параллелепипеда

$$0 < x < L_x$$
, $0 < y < L_y$, $0 < z < L_z$,

находятся плазма, состоящая из электронов и ионов водорода, и пучок электронов. Заданы плотности электронов пучка n_b и электронов плазмы $n_e=1-n_b$. Плотность ионов плазмы равна сумме плотностей электронов пучка и электронов плазмы. Температура электронов плазмы T_e и пучка T_b ; температура ионов считается нулевой $T_i=0$. Начальное распределение частиц по скоростям максвелловское с плотностью распределения

$$f(v) = \frac{1}{\Delta v \sqrt{\pi}} \exp\left(-\frac{(v - v_0)^2}{2(\Delta v)^2}\right),$$

где Δv – разброс частиц по скоростям $(T_b = \Delta v^2)$, v_0 – средняя скорость пучка. Средняя скорость ионов и электронов фона нулевая. Все частицы распределены по области равномер-

но, начальная средняя скорость пучка направлена по X и равна $v_0 = 0, 2$. Граничные условия периодические.

В расчетах представляет интерес развитие отдельно взятой неустойчивой моды, поэтому длина области в направлении X выбрана равной одной длине исследуемой плазменной волны, $L_v = L_z = 4h \ (h-$ шаг сетки). Эти эксперименты описаны в статьях [9; 10].

Метод частиц в ячейках

Основываясь на книге [2], приведем описание идеи метода. Пусть задача записана в абстрактной операторной форме:

$$\frac{\partial q}{\partial t} + Aq = 0. {1}$$

Здесь q(t,u) — вектор-функция со значениями в R^n , u — вектор независимых переменных с областью изменения в R^m , $0 < t < \infty$.

В реальных задачах роль уравнения (1) выполняет кинетическое уравнение (уравнение Власова или Больцмана) или уравнения гидродинамики. Решение задачи (1) представляется в виде следующей интерполяционной формулы:

$$q = \sum_{j=1}^{N} Q_{j} R(u, u_{j}(t)).$$

Этот переход называют разбиением среды на модельные частицы. Функция $R(u,u_j(t))$ называется ядром (или форм-фактором) модельной частицы. Эта функция описывает распределение некоторого признака (массы, заряда, скорости...) в рамках одной частицы.

Далее, если представить функцию q в виде

$$q = \sum_{j=1}^{N} Q_{j} R(r, r_{j}(t)) \delta(p - p_{j}(t)),$$

где r — радиус-вектор частицы, p — импульс частицы, то можно показать, что решение уравнения (1) тождественно решению следующей динамической системы:

$$\frac{\partial r_j}{\partial t} = p_j, \dots \quad \frac{\partial p_j}{\partial t} = F(r_j, p_j), \quad j = 1..N.$$
 (2)

Здесь F – вектор обобщенного поля.

Исключительно важно, что переход к модельным частицам не означает замены реальной физической системы, где число частиц — молекул, атомов, ионов и пр. — порядка числа Авогадро ($N_{\rm A}=6.02\times10^{23}$) на некую упрощенную систему из значительно меньшего количества таких же частиц, но «более крупного размера» (даже для самых больших расчетов на супер-ЭВМ N на данный момент не превышает 10^{12}). Система уравнений (2) является не более чем математическим формализмом для решения уравнения (1), т. е. на данном этапе никакого нарушения математической строгости не происходит, система уравнений (2) точно так же описывает физический процесс, как и исходное уравнение (1).

Общая схема расчетов по методу частиц в ячейках

Вначале выполняется задание начальной конфигурации распределения вещества в расчетной области и полей. Производится распределение частиц по ячейкам так, чтобы плотности (или токи), вычисленные по частицам, соответствовали заданной конфигурации.

Далее на каждом временном шаге процесса моделирования выполняется:

- 1) вычисление токов по значениям координат и скоростей модельных частиц;
- 2) расчет электрического и магнитного полей по токам;
- 3) сдвиг частиц: вычисление новых значений координат и скоростей модельных частиц с учетом новых значений поля.

При разбиении пространства на ячейки на 3-м шаге возникает необходимость перераспределения частиц между ячейками.

Оптимизация расчета с помощью переупорядочивания частиц

Для того чтобы подчеркнуть значение полученной на GPU производительности, вначале приведем данные о производительности на CPU: для процессора Intel Xeon E5540 на 4 его ядрах время вычисления сдвига частиц было в 41,6 раза больше, чем для GPU Nvidia Tesla K40, т. е. 11,7 с (кинетический вариант развития неустойчивости, 1000 модельных частиц в ячейке).

Причина этого заключается в том, что частицы обычно, в большинстве программ на основе метода частиц, хранятся в памяти без учета их расположения в пространстве моделирования, т. е. следующая по номеру в массиве координат частица может быть расположена в пространстве моделирования очень далеко от предыдущей. Это значит, что значения полей при расчете сдвига частиц будут взяты из совершенно другой части трехмерного массива, в виде которого хранятся поля. Это, в свою очередь, означает, что использование кэш-памяти будет очень неэффективным.

Поскольку еще в работе [11] было показано, что переход от хранения координат и скоростей частиц в одном массиве без учета расположения в пространстве (1 вариант) к отдельным массивам по каждой ячейке (2 вариант) приводит к ускорению движения частиц в 1,5–2 раза, что подтверждено и в данной работе (см. таблицу), то частицы хранятся распределенными по ячейкам (напомним, что количество частиц в ячейке около 1 000). Но при этом возникает проблема переупорядочивания модельных частиц, т. е. передачи частиц, перелетающих в другие ячейки, на хранение в массивы, связанные с этими другими ячейками (см. рисунок, 1, 2). Напомним, что по соображениям устойчивости метода частиц модельные частицы могут перелетать только в соседнюю ячейку.

Конфликт чтения-записи (см. рисунок, 2), возникает в силу того, что каждое перемещение частицы выполняется отдельным потоком CUDA, при этом необходимо считать частицу из массива частиц соседней ячейки и удалить ее оттуда, т. е. выполняется и чтение, и запись одновременно несколькими потоками, а для мощных GPU (Volta, Pascal и др.) несколькими тысячами потоков. Для разрешения конфликта в данном случае используются атомарные операции CUDA, что существенно замедляет выполнение алгоритма в целом. Поэтому есть необходимость выполнять операцию переупорядочивания по-другому. В данной статье предложен альтернативный более быстрый вариант.

Время выполнения различных этапов вычислительного алгоритма на нескольких GPU (оптимизированный вариант без атомарных операций) по сравнению с неоптимизированным (полужирный шрифт) вариантом на Kepler K40.

Время указано в микросекундах

Execution time for different stages of the computational algorithm for various GPUs (optimized version without atomic operations) compared to non-optimized algorithm on Kepler K40 (bold font). Time is given in microseconds

Этап алгоритма	GPU			
	Kepler K40		Kepler K80	Pascal P100
Сдвиг частиц	433.8	433.8 *	348.06 *	338.1 *
Расчет поля	31.7	31.7 *	25.4 *	19.3 *
Переупорядочивание	27167	1131 *	446.43 *	98.9 *

^{*} Атомарные операции не используются.

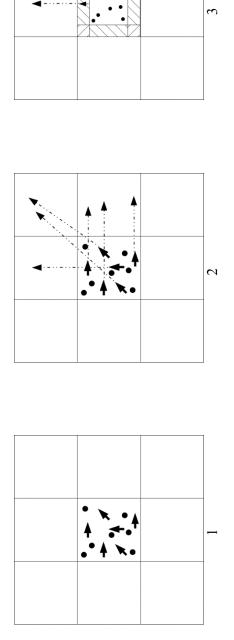


Схема распределения частиц в ячейках

(частицы, которые не покидают данную ячейку, показаны черными точками, а частицы, которые должны перелеть в одну из соседних ячеек, – стрелками, указывающими в соответствующем направлении;

для простоты показан двумерный случай)

- $I- \Pi$ ример частиц в ячейке после вычисления новых координат и импульсов, соседние ячейки схематически показаны пустыми.
- 2 Конфликт чтения-записи для операции перемещения частиц в соседние ячейки. Каждая пунктирная линия показывает перемещение отдельной частицы из исходной ячейки в соседнюю; поскольку каждая такая операция производится отдельным потоком, то пересечение пунктирных линий символически показывает конфликт чтения-записи.
 - 3 Разрешение конфликта чтения-записи с помощью буферов (показаны штриховкой). Каждая пунктирная линия показывает перемещение отдельной частицы из исходной ячейки в соседнюю. Видно, что в данном случае они не пересекаются

An example of particles-in-cell

the particle remaining in the present cells are shown with circles, and the particles flying to some other cells are shown with arrows pointing to corresponding direction. 2D case is shown for simplicity)

- I An example of particles-in-cell after computing new coordinates and impulses, the neighbour cells are shown to be empty
- 2 Read-and-write conflict for moving particles to neighbour cells. Each dash-dotted line shows moving of a single particle from the original cell to a neighbour one. Since each moving is performed by a single thread, the crossing of lines shows a read-and write conflict.
- 3 Solving read-and-write conflict using buffers (shown by hatching). Each dash-dotted line shows moving of a single particle from the original cell to a neighbour one. It is seen that in this case the line do not cross

Оптимизация расчета с помощью переупорядочивания частиц

Следующая схема предложена для сокращения времени переупорядочивания модельных частиц:

- 1) функция, исполняемая на GPU и производящая переупорядочивание, разделяется на две части;
- 2) в каждой ячейке формируется список модельных частиц, которые должны быть перемещены в каждую из соседних ячеек;
- 3) заполняется матрица размером 3³, в которой записано, сколько модельных частиц должно быть перемещено в каждую из соседних ячеек (всего таких ячеек 26);
- 4) перемещаемые частицы копируются в буфера, соответствующие соседним ячейкам (см. рисунок, 3).

При этом ни разу не возникает конкуренция за доступ к памяти, именно поэтому удается избавиться от использования атомарных операций.

Список литературы / References

- 1. **Бэдсел Ч., Лэнгдон А.** Физика плазмы и численное моделирование. М.: Энергоатомиздат, 1989.
 - Birdsall C. K., Langdon A. Plasma Physics via Computer Simulation. CRC Press, 2004.
- 2. **Григорьев Ю. Н., Вшивков В. А., Федорук М. П.** Численное моделирование методами частиц в ячейках. Новосибирск: Изд-во СО РАН, 2004. **Grigoryev Yu. N., Vshivkov V. A., Fedoruk M. P.** Numerical Particle-in-Cell Methods. Theory and applications. Utrech, Boston, 2002.
- 3. **Astrelin V. T., Burdakov A. V., Postupaev V. V.** Generation of ion-acoustic waves and suppression of heat transport during plasma heating by an electron beam. *Plasma Physics Reports*, 1998, vol. 24 (5), p. 414–425.
- 4. **Боресков А. В., Харламов А. А.** Основы работы с технологией СUDA. ДМК Пресс, 2010.
 - Boreskov A. V., Harlamov A. A. Basics of the CUDA technology. DMK Press, 2010.
- 5. Burau H., Widera R., Hoenig W., Juckeland G., Debus A., Kluge T., Schramm U., Cowan T., Sauerbrey R., Bussmann M. PIConGPU: A Fully Relativistic Particle-in-Cell Code for a GPU Cluster. *IEEE Transactions on Plasma Science*, 2010, vol. 38, p. 2831–2839. DOI 10.1109/TPS.2010.2064310
- 6. **Rossi F., Londrillo P., Sgattoni A., Sinigardi S., Turchetti G.** Towards robust algorithms for current deposition and dynamic load-balancing in a GPU particle in cell code. In: AIP Conference Proceedings, 2012, vol. 1507, iss. 1, p. 184–192. DOI 10.1063/1.4773692
- 7. **Kong X., Huang M., Ren Ch., Decyk V.** Particle-in-cell simulations with charge-conserving current deposition on graphic processing units. *Journal of Computational Physics*, 2011, vol. 230, iss. 4, p. 1676–1685. DOI 10.1016/j.jcp.2010.11.032
- 8. **Rieke M., Trost T., Grauer R.** Coupled Vlasov and two-fluid codes on GPUs. *Journal of Computational Physics*, 2015, vol. 283, p. 436–452. DOI 10.1016/j.jcp.2014.12.016
- 9. **Месяц Е. А., Снытников А. В., Лотов К. В.** О выборе числа частиц в методе частиц-вячейках для моделирования задач физики плазмы // Вычислительные технологии, 2013. Т. 18, № 6. С. 83–96.
 - Mesyats E. A., Snytnikov A. V., Lotov K. V. On the selection of particle number in Particle-in-Cell method for plasma simulation. *Vychistilte'nye Tehnologii*, 2013, vol. 18, no. 6, p. 83–96. (in Russ.)

- 10. Lotov K. V., Timofeev I. V., Mesyats E. A., Snytnikov A. V., Vshivkov V. A. Note on quantitatively correct simulations of the kinetic beam-plasma instability. *Physics of Plasmas*, 2015, vol. 22, no. 2, p. 024502. DOI 10.1063/1.4907223.
- 11. **Tskhakaya D., Schneider R.** Optimization of PIC codes by improved memory management. *Journal of Computational Physics*, 2007, vol. 225, iss. 1, p. 829–839. DOI 10.1016/j.jcp. 2007.01.002.

Материал поступил в редколлегию Received 11.12.2018

Сведения об авторах / Information about the Authors

- **Романенко Алексей Анатольевич**, кандидат технических наук, доцент, заместитель декана факультета информационных технологий Новосибирского государственного университета (ул. Пирогова, 1, Новосибирск, 630090, Россия)
- **Alexey A. Romanenko**, Candidate of Science (Technology), Associate Professor, Deputy Dean of Faculty of Information Sciences, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation)

arom@ccfit.nsu.ru

- Снытников Алексей Владимирович, кандидат физико-математических наук, старший преподаватель факультета информационных технологий Новосибирского государственного университета (ул. Пирогова, 1, Новосибирск, 630090, Россия), старший научный сотрудник Института вычислительной математики и математической геофизики СО РАН (пр. Академика Лаврентьева, 6, Новосибирск, 630090, Россия)
- **Alexey V. Snytnikov**, Candidate of Science (Physics and Mathematics), Senior Teacher of Faculty of Information Sciences, Novosibirsk State University (1 Pirogov Str., Novosibirsk, 630090, Russian Federation), Senior Researcher at Institute of Computational Mathematics and Mathematical Geophysics SB RAS (6 Academician Lavrentiev Ave., Novosibirsk, 630090, Russian Federation)

snytav@ssd.sscc.ru