

ИСПОЛЬЗОВАНИЕ ПАРАЛЛЕЛЬНЫХ ВЫЧИСЛЕНИЙ В РЕСУРСООЕМКИХ ЗАДАЧАХ МОДЕЛИРОВАНИЯ ПРОЦЕССОВ ДВИЖЕНИЯ И ВЗАИМОДЕЙСТВИЯ ЧАСТИЦ В ПЛАЗМЕ ПРИ СИНТЕЗЕ УГЛЕРОДНЫХ НАНОСТРУКТУР

Г.В. Абрамов¹

А.Н. Гаврилов²

А.Л. Ивашин²

И.С. Толстова²

agwl@yandex.ru

ganinvrn@yandex.ru

ivashin.alexei@gmail.com

irin2102ka@mail.ru

¹ Воронежский государственный университет, Воронеж, Российская Федерация

² Воронежский государственный университет инженерных технологий,
Воронеж, Российская Федерация

Аннотация

Использование современных технологий параллельных вычислений позволяет решать сложные ресурсоемкие задачи моделирования физических процессов на персональном компьютере. Рассмотрено численное решение математической модели движения и взаимодействия частиц многокомпонентной плазмы при электродуговом синтезе углеродных наноструктур. Наличие большого числа ($10^{16} \dots 10^{17}$) взаимодействующих частиц, участвующих в расчете на каждом временном шаге, требует значительных затрат машинных ресурсов и времени для вычислений. Традиционно модели такого типа решают численно с использованием суперкомпьютеров или облачных вычислений. Применение технологий параллельных вычислений *GPGPU* и технологии *Nvidia CUDA* позволяет выполнять вычисления общего назначения на базе графических процессоров видеокарты. Представленное алгоритмическое обеспечение на основе метода крупных частиц позволяет получить численное решение рассматриваемой модели в допустимое время на персональном компьютере. Предложены подходы к распараллеливанию вычислений с учетом синхронизации результатов. Перечислены трудности, которые возникают при программной реализации алгоритмов поиска столкновений частиц в плазме, и предложены методы их преодоления. Рассчитаны эффективности различных параллельных алгоритмов

Ключевые слова

Синтез углеродных наноструктур, метод крупных частиц, параллельное программирование, графический процессор, GPU, CPU, Nvidia CUDA

Поступила в редакцию 22.01.2018
© МГТУ им. Н.Э. Баумана, 2018

Введение. Современные технологии, доступные в настоящее время широкому кругу пользователей, дают возможность выполнять вычисления, которые ранее были доступны только при использовании специализированных аппаратных комплексов. Персональные компьютеры содержат аппаратные и программные средства, позволяющие обрабатывать большие объемы данных в параллельном

или конкурентном режиме. С использованием анализа особенностей решаемых задач можно выявлять более эффективные алгоритмы распараллеливания вычислений, что, в свою очередь, может снизить временные издержки или позволит выполнять расчеты на оборудовании с более низкими характеристиками производительности.

В настоящей работе рассмотрена задача разработки численных методов и алгоритмов решения модели синтеза углеродных наноструктур (УНС) методом термической возгонки графита плазмой дугового разряда.

Цель работы — разработка эффективных численных методов и алгоритмов решения задачи синтеза УНС с учетом обработки больших объемов данных и ее программная реализация, ориентированная на использование технологии распараллеливания вычислений, основанной на *CPU* и *GPU*.

Для достижения поставленной цели необходимо решить следующие задачи:

- разработать параллельный алгоритм ускорения численного решения задачи;
- разработать программу для моделирования процесса плазменного синтеза УНС с использованием гибридных вычислительных систем;
- провести анализ эффективности предлагаемого алгоритма.

Обладая специфическими характеристиками, УНС (фуллерены и углеродные нанотрубки) рассматриваются как перспективный материал в современной промышленности. В связи с этим получение наноструктур с заданными параметрами является важной технологической задачей. Рост промышленного производства УНС сдерживает низкая производительность и эффективность существующих технологий синтеза вследствие недостаточной изученности механизмов и условий образования УНС.

Один из наиболее известных методов синтеза УНС — процесс термического распыления графита в плазме дугового разряда в среде инертного газа (He, Ar). Указанный метод можно охарактеризовать как наиболее эффективный и производительный, так как он позволяет вовлекать в технологический процесс значительные объемы сырья, отличается высокой скоростью протекания и качеством синтезируемого материала.

Некоторые особенности электродугового синтеза УНС (наноразмеры получаемого продукта, быстротечность, малые размеры области реакции, необходимость поддержания стабильных параметров процесса) усложняют эмпирические исследования и повышают актуальность применения средств математического моделирования [1]. Исходя из этого, актуальным направлением изучения процессов при синтезе УНС является их теоретическое описание с применением методов моделирования. Необходимость использования технологии высокопроизводительных вычислений при моделировании процессов синтеза УНС обусловлена высокой размерностью математических моделей, состоящих из большого числа взаимодействующих частиц.

Межэлектродное пространство электродугового разряда заполняет многокомпонентная плазма [2], состоящая из электронов, нейтральных и возбужденных

молекул буферного газа и вещества электродов. Начальное распределение для всех типов частиц предполагается максвелловским. Требуется найти параметры электромагнитного поля, задать начальное распределение частиц, определить их перемещение в трехмерном пространстве, диагностировать взаимодействия между частицами с учетом упругих и неупругих столкновений, рассчитать с учетом энергетических и пространственных условий вероятностное образование устойчивых связей между частицами углерода, формирующих объемные УНС.

Для решения поставленной задачи необходимо учитывать, что число только однотипных частиц в рассматриваемой области составляет $10^{15} \dots 10^{17}$ (для углеродных электродов диаметром 12 мм), что требует использования больших объемов данных, которые должны эффективно и корректно обрабатываться во время численных расчетов. Это делает задачу очень ресурсоемкой, решение которой без использования параллельных вычислений требует значительных временных затрат.

Один из возможных вариантов решения указанной проблемы — использование перспективного и интенсивно развивающегося направления в области параллельного программирования — технологии параллельных вычислений *GPGPU* (*General-Purpose Graphics Processing Units*) [3]. Эта технология позволяет графический процессор видеокарты эффективно применять не только для обработки графической информации, но и для вычислений общего назначения [4].

Метод решения задачи. В основе разработанной модели синтеза УНС лежит система уравнений Больцмана

$$\frac{\partial f_{\alpha}}{\partial t} + \vec{\mathfrak{V}} \frac{\partial f_{\alpha}}{\partial \vec{r}} - \frac{q_{\alpha}}{m_{\alpha}} \left(\vec{E} + \frac{1}{C} [\vec{\mathfrak{V}}, \vec{B}] \right) \frac{\partial f_{\alpha}}{\partial \vec{\mathfrak{V}}} = \frac{\partial f_{\alpha}}{\partial t} \Big|_{CT}, \quad \alpha = e, h, c,$$

для каждой компоненты плазмы, дополненная системой уравнений Максвелла, описывающей электромагнитное поле [5, 6]:

$$\text{rot } \vec{H} = \frac{4\pi \vec{j}}{C} + \frac{1}{C} \frac{\partial \vec{D}}{\partial t};$$

$$\text{rot } \vec{E} = -\frac{1}{C} \frac{\partial \vec{B}}{\partial t};$$

$$\text{div } \vec{B} = 0;$$

$$\text{div } \vec{D} = 4\pi \rho.$$

Здесь f_{α} — функции распределения компоненты плазмы; e — электрон; h — ион буферного газа; c — ион углерода; \vec{E} , \vec{H} — напряженность электрического и магнитного полей; \vec{D} , \vec{B} — электрическая и магнитная индукции; q_{α} , m_{α} — заряд и масса частицы; ρ — плотность заряда; \vec{j} — плотность тока; C — скорость света; $\vec{\mathfrak{V}}$ — поле скоростей частицы; \vec{r} — координаты частицы.

Допуская, что в плазме взаимодействия происходят только между электронами, ионами буферного газа и частицами углерода во всем межэлектродном пространстве, интегралы парных столкновений можно представить как

$$\left. \frac{\partial f_{\alpha}}{\partial t} \right|_{CT} = \sum_{\beta=e,c,h} \iint_V (f'_{\alpha} f'_{\beta} - f_{\alpha} f_{\beta}) |\bar{\mathfrak{Q}} - \bar{\mathfrak{Q}}'| d\sigma d\bar{\mathfrak{Q}}', \quad \alpha = e, c, h,$$

где f_{α} , f_{β} , f'_{α} , f'_{β} — функции распределения частиц α и β до и после столкновения; $\bar{\mathfrak{Q}}$, $\bar{\mathfrak{Q}}'$ — векторы скоростей до и после столкновения частиц; $d\sigma$ — дифференциальное эффективное сечение рассеяния.

Общий алгоритм расчета движения и взаимодействия частиц в многокомпонентной плазме (рис. 1) включает в себя последовательное выполнение семи основных шагов.

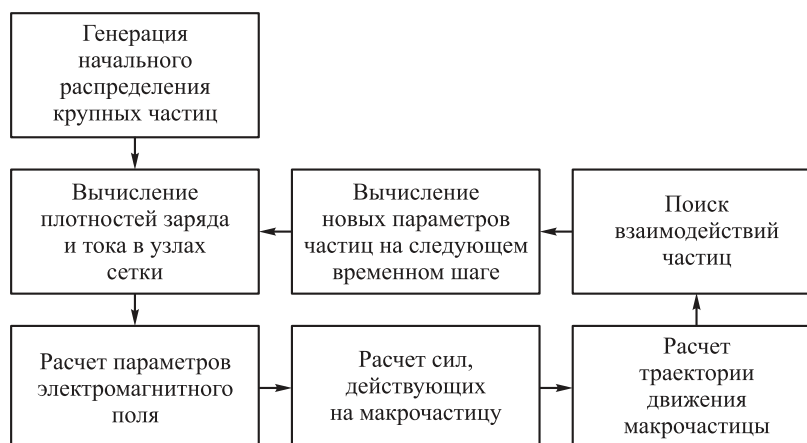


Рис. 1. Блок-схема алгоритма расчета движения и взаимодействия частиц в многокомпонентной плазме

Для снижения объема вычислений за счет уменьшения числа однотипных частиц в расчете путем их группировки до обоснованного уровня, не влияющего на точность расчета, используется метод конечных частиц (МКЧ). Далее описана краткая характеристика вычислительной схемы МКЧ.

Область моделирования имеет сеточную форму прямоугольного параллелепипеда со сторонами, которые параллельны осям декартовой системы координат. При использовании МКЧ в расчетных ячейках сетки плазма моделируется набором из N заряженных макрочастиц, каждая из которых характеризуется переменными $\bar{\mathfrak{Q}}$, \vec{r} , а также постоянными q_{α} , m_{α} . На основе распределения большого числа частиц в расчетном пространстве вычисляются значения плотности заряда в узлах сетки, по которым затем из решения уравнения Пуассона находятся значения потенциала и силы, действующие на частицы в ячейках сетки. Следующий шаг — определение новых координат частиц из уравнений их движения. Тем самым распределение частиц в пространстве изменяется под

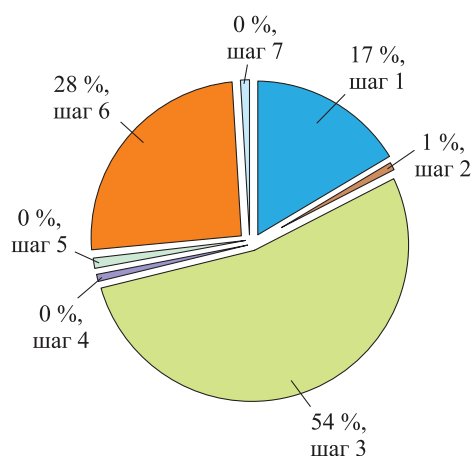


Рис. 2. Круговая диаграмма распределения времени, затрачиваемого на выполнение шагов

действием самосогласованного поля, а также вследствие взаимодействия частиц между собой.

Круговая диаграмма распределения времени, затрачиваемого на выполнение шагов, приведена на рис. 2.

Анализ показывает, что основное время работы программы численного расчета модели занимают шаги 1, 3 и 6. Если первый шаг можно решать один раз для каждого набора исходных начальных параметров моделирования путем предварительной генерации необходимых данных в отдельный файл, а затем лишь использовать полученное распределение, то шаги 3 и 6 имеют большее число изменяемых

входных аргументов, что делает подход подготовительного расчета несостоятельным. Решение, полученное именно на шаге 3 и 6, требует наибольших затрат вычислительных ресурсов (процессорного времени, оперативной памяти) в процессе моделирования кинетики плазмы. Одним из способов снижения общего времени расчета рассматриваемой трудоемкой вычислительной задачи является организация параллельных и (или) распределенных вычислений.

Подробное рассмотрение обозначенных задач с позиции алгоритмизации вычислительного процесса показывает, что основная характеристика, влияющая на вычислительную сложность, — размерность сетки моделируемой области. На основе сравнения вычислительной сложности алгоритмов, а также достоинств и недостатков технологий параллельного программирования для реализации поставленной задачи выбраны технологии распараллеливания *CPU* и *GPU*. Конечно-разностная схема для нахождения сеточных значений потенциалов и соответствующих сил, действующих на частицы, решаемая методом установления, который имеет широкие возможности для разработки параллельных вычислительных алгоритмов (шаг 3), приведена в работе [7]. Такая задача имеет большой ресурс параллелизма — вычисление значения для каждого узла двумерной сетки на новом временном слое не зависит от других узлов сетки на новом временном слое (при использовании двух массивов). Необходима только синхронизация при переходе от предыдущего временного слоя к следующему. Поскольку при расчете параметров электромагнитного поля фазовой области в каждой точке расчетной сетки выполняются одинаковые операции над большим объемом данных для нахождения значений переменных в последующий момент времени для каждого шага, целесообразно применить технологию распараллеливания взаимодействующих вычислительных процессов.

Указанный способ организации компьютерных вычислений можно осуществить с помощью технологии расчета *GPU*, позволяющей выполнять общие

расчеты, которые обычно проводит центральный процессор на видеокарте. Эта технология дает возможность на одном ПК с использованием видеоадаптера с архитектурой *SIMD* достичь достаточно высокого уровня параллелизма, при этом не получить значительных временных затрат на передачу данных между вычислительными узлами и синхронизацию результатов вычислений.

В качестве средства для программирования графического процессора была выбрана технология *Nvidia CUDA*. Видеоадаптеры компании *Nvidia* с архитектурой *CUDA (Compute Unified Device Architecture)* занимают большую часть рынка графических решений как в России, так и за рубежом [8]. С учетом особенностей выбранной технологии и методов построения вычислительных алгоритмов подобных задач [9] был сделан следующий вывод: наилучшим решением по использованию распараллеливания потоков является деление исходной расчетной сетки на заданное число блоков, каждый из которых содержит одинаковое число узлов, и один параллельный поток рассчитывает значение в одном узле сетки. Этот вывод сделан с позиции простоты декомпозиции и аппаратных ограничений [10].

В целях минимизации общего времени расчетов были рассмотрены различные версии организации алгоритмов распараллеливания и в результате был выбран алгоритм, комбинирующий параллельное и однопоточное выполнение. Инициализация исходных данных, представляющих собой сетку, на предыдущем и текущем временном шаге выполняется в однопоточном режиме, а вычисление потенциала в узлах сетки и точности решения — в параллельном. Для каждого момента времени синхронизация вычислений организовывается описанным ниже способом. После того как вычислены значения всех узлов на графическом адаптере, каждый блок параллельных потоков синхронизируется и вычисляет точность своих узлов, а получение общего значения точности осуществляется в последовательном режиме. В зависимости от значения точности расчет либо завершается, либо запускается следующая итерация алгоритма.

Следующая ресурсоемкая проблема — задачи моделирования перемещения и поиска взаимодействующих друг с другом частиц. Задача обнаружения столкновений заключается в проверке факта пересечения траекторий объектов в пространстве. Существует несколько вариантов построения алгоритмов решения этой проблемы. Первый самый простой и, соответственно, самый медленный способ решения — полный перебор всех возможных пар частиц. Учитывая, что на каждом шаге участвует большое число крупных частиц, можно утверждать, что вычисление столкновений с помощью указанного алгоритма является вычислительно сложной и ресурсозатратной задачей. Еще один способ решения — использование алгоритма *Sweep and Prune* [11], суть которого заключается в том, что все частицы первоначально необходимо отсортировать по их левой границе, а затем проверить взаимодействия только тех частиц, у которых левая граница меньше правой границы текущей частицы. Основное время в этом алгоритме занимает сортировка частиц, поэтому его целесообразно применять, если перемещение частиц в каждый момент времени небольшое.

Другой возможный метод поиска столкновений — разбиение расчетной области на множество непересекающихся областей с дальнейшим поиском взаимодействий только внутри них. Поскольку используемая в работе область моделирования имеет прямоугольное сечение, наиболее простым является деление ее на прямоугольные ячейки, размер которых сопоставим или превосходит размер крупной частицы. Каждая такая ячейка представляется в памяти как массив крупных частиц, в нее вошедших частично или полностью, а также частиц, которые могли побывать в данной ячейке в течение рассматриваемого такта. В связи с этим для моделирования большого числа частиц необходима дополнительная память. Далее описана работа алгоритма, следует отметить, что область разбивалась на три отрезка по каждой пространственной оси, поэтому итоговая разбивка состоит из 27 ячеек.

Поиск столкновений включает в себя два основных этапа. На первом этапе в качестве входных данных используется информация о начальном и конечном положении частиц. Каждый вычислительный поток на основе этих данных определяет ячейку, в которой находилась частица изначально, ячейку, куда частица переместилась, и список ячеек, в которых потенциально могло произойти взаимодействие частиц.

На втором этапе проводится поиск взаимодействий исключительно внутри каждой ячейки, что ввиду уменьшения числа частиц значительно снижает вычислительную сложность расчета. Далее необходимо учесть, что если частица взаимодействовала внутри нескольких ячеек, то действительным является только первое столкновение. Следует отметить, что уместно использование числа потоков по числу выделенных ячеек.

Каждый описанный шаг можно выполнять как в однопоточном, так и в многопоточном режиме.

Результаты. Тестирование предложенных алгоритмов проводилось на компьютере, оснащенный процессором *Intel® Core™2 Quad Q 8200 2.3 GHz* и видеокартой *Nvidia GeForce GTX 660 Ti*. Для анализа эффективности применения приведенных версий параллельных алгоритмов численные эксперименты были выполнены для двух различных сеток расчетной области с числом узлов $25 \times 25 \times 25$ и $50 \times 50 \times 50$.

Сравнение общего времени выполнения параллельной версии алгоритма с последовательным для задачи расчета электрического поля методом установления с учетом передач данных между *CPU* и *GPU* показало, что для сетки $25 \times 25 \times 25$ общее время вычислений уменьшается в 132 раза, а для сетки $50 \times 50 \times 50$ — в 152 раза.

Далее была проведена оценка времени, необходимого для выполнения алгоритма поиска взаимодействий частиц. Рассмотрено четыре различные модификации алгоритма:

- 1) простой однопоточный перебор всех крупных частиц на предмет обнаружения их столкновений;
- 2) однопоточный поиск столкновений с использованием оптимизации на основе ячеек, описанный выше;

3) простой многопоточный перебор всех крупных частиц для обнаружения их столкновений;

4) многопоточный поиск столкновений с помощью оптимизации на основе ячеек.

Результаты оценки времени исполнения различных версий алгоритма приведены в таблице.

Сравнение расчетного времени исполнения различных версий алгоритма

Версия алгоритма	Время расчета, с, для числа частиц	
	15 625	125 000
Алгоритм 1	146,22	4682,93
Алгоритм 2	87,19	1837,5
Алгоритм 3	56,59	643,38
Алгоритм 4	12,18	90,73

Заключение. Разработанное алгоритмическое и программное обеспечение на основе МКЧ позволяет моделировать кинетику и взаимодействие частиц в многокомпонентной плазме при синтезе УНС в допустимое время на персональных компьютерах с использованием графических процессоров. Предложены подходы к распараллеливанию вычислений для ресурсоемких задач, учитывающие специфику обработки больших объемов данных. Перечислены трудности, которые возникают при программной реализации, и предложены методы их преодоления. Исследовано ускорение параллельной программы. Исходя из полученных результатов по времени расчетов отдельных задач, можно отметить следующее. Технологии распараллеливания вычислений *CPU* и *GPU* дают ускорение расчетов в случае громоздких вычислений с минимумом передач данных между *CPU* и *GPU*. При использовании комбинации приведенных технологий можно добиться высокой производительности решения поставленной задачи.

ЛИТЕРАТУРА

1. *Абрамов Г.В., Миронченко Е.А., Толстова И.С.* Математическое моделирование процессов при электродуговом синтезе углеродных нанотрубок // Вестник ВГУИТ. 2013. № 4. С. 106–110.
2. *Цветков И.В.* Применение численных методов для моделирования процессов в плазме. М.: МИФИ, 2007. 84 с.
3. *Использование видеокарт для вычислений* // GPGPU.ru: веб-сайт. URL: <http://www.gpgpu.ru> (дата обращения: 25.11.2017).
4. *Nvidia CUDA* — неграфические вычисления на графических процессорах // ixbt.com: веб-сайт. URL: <http://www.ixbt.com/video3/cuda-1.shtml> (дата обращения: 25.11.2017).
5. *Абрамов Г.В., Гаврилов А.Н.* Математическое моделирование движения взаимодействующих частиц на основе функций распределения в плазме электродугового синтеза УНС // Вестник ВГУИТ. 2012. № 2. С. 71–75.
6. *Черчиньяни К.* Теория и приложения уравнения Больцмана. М.: Мир, 1978. 485 с.

7. *Abramov G., Gavrilov A., Tolstova I.* The use of technology for parallelization method of large particles using cloud computing // *British Journal of Science, Education and Culture*. 2014. No. 2 (6). P. 380–387.
8. *Graphics add-in board shipments decline 15.2 % from last quarter.*
URL: <https://www.jonpeddie.com/press-releases/graphics-add-in-board-shipments-decline-15.2-from-last-quarter> (дата обращения: 25.11.2017).
9. *Деммель Дж.* Вычислительная линейная алгебра. Теория и приложения. М.: Мир, 2001. 430 с.
10. *Усков Р.В.* О некоторых особенностях применения технологии *CUDA* для моделирования переноса излучения // *Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки*. 2011. № 3. С. 71–83.
11. *Baraff D.* Dynamic simulation of non-penetrating rigid bodies. Ph. D. thesis. Cornell University, Computer Science Department, 1992. P. 52–56.

Абрамов Геннадий Владимирович — д-р техн. наук, профессор кафедры математического и прикладного анализа Воронежского государственного университета (Российская Федерация, 394018, Воронеж, Университетская площадь, д. 1).

Гаврилов Александр Николаевич — канд. техн. наук, доцент кафедры информационных и управляющих систем Воронежского государственного университета инженерных технологий (Российская Федерация, 394036, Воронеж, пр-т Революции, д. 19).

Ивашин Алексей Леонидович — канд. техн. наук, доцент кафедры высшей математики и информационных технологий Воронежского государственного университета инженерных технологий (Российская Федерация, 394036, Воронеж, пр-т Революции, д. 19).

Толстова Ирина Сергеевна — старший преподаватель кафедры высшей математики и информационных технологий Воронежского государственного университета инженерных технологий (Российская Федерация, 394036, Воронеж, пр-т Революции, д. 19).

Просьба ссылаться на эту статью следующим образом:

Абрамов Г.В., Гаврилов А.Н., Ивашин А.Л., Толстова И.С. Использование параллельных вычислений в ресурсоемких задачах моделирования процессов движения и взаимодействия частиц в плазме при синтезе углеродных наноструктур // *Вестник МГТУ им. Н.Э. Баумана. Сер. Естественные науки*. 2018. № 5. С. 4–14. DOI: 10.18698/1812-3368-2018-5-4-14

USING PARALLEL COMPUTING IN COMPUTATIONALLY INTENSIVE PROBLEMS OF SIMULATING PARTICLE MOTION AND INTERACTION IN PLASMA DURING CARBON NANOSTRUCTURE SYNTHESIS

G.V. Abramov¹

agwl@yandex.ru

A.N. Gavrilov²

ganinvrn@yandex.ru

A.L. Ivashin²

ivashin.alexei@gmail.com

I.S. Tolstova²

irin2102ka@mail.ru

¹ **Voronezh State University, Voronezh, Russian Federation**

² **Voronezh State University of Engineering Technologies, Voronezh, Russian Federation**

Abstract

Modern parallel computing technology makes it possible to solve complex computationally intensive problems of simulating physical processes using a personal computer. The paper considers a numerical solution to the equations forming a mathematical model of multicomponent plasma particle motion and interaction during arc discharge synthesis of carbon nanostructures. A large number of interacting particles ($10^{16} \dots 10^{17}$) to be taken into account at each time step means considerable computational expense and time. Conventional numerical simulations of this type involve supercomputers or cloud computing. However, parallel computing technologies such as *GPGPU* and *Nvidia CUDA* make it possible to perform general purpose computations on graphics processing unit cores. We present software algorithms based on a modified particle-in-cell method which enable numerical simulations of the model under consideration to be run on a personal computer in a reasonable timeframe. We propose approaches to decomposing computation problems accounting for result synchronisation. We list those difficulties that arise while implementing collision detection algorithms for particles in plasma in actual software code and propose ways of overcoming these difficulties. We calculated the efficiency of various parallel algorithms

Keywords

Carbon nanostructure synthesis, modified particle-in-cell method, parallel programming, graphics processing unit, GPU, CPU, Nvidia CUDA

Received 22.01.2018

© BMSTU, 2018

REFERENCES

- [1] Abramov G.V., Mironchenko E.A., Tolstova I.S. Mathematical modeling of processes in the electric arc synthesis of carbon nanotubes. *Vestnik VGUIT* [Proceedings of the Voronezh State University of Engineering Technologies], 2013, no. 4, pp. 106–110 (in Russ.).
- [2] Tsvetkov I.V. *Primenenie chislennykh metodov dlya modelirovaniya protsessov v plazme* [Application of numerical methods for simulation plasma processes]. Moscow, MEPhI Publ., 2007. 84 p.
- [3] *Ispol'zovanie videokart dlya vychisleniy* [Using video-card for calculations]. *GPGPU.ru*: website (in Russ.). Available at: <http://www.gpgpu.ru> (accessed: 25.11.2017).
- [4] *Nvidia CUDA — negraficheskie vychisleniya na graficheskikh protsessorakh* [Nvidia CUDA: Non-graphic calculations on graphic processors]. *ixbt.com*: website (in Russ.). Available at: <http://www.ixbt.com/video3/cuda-1.shtml> (accessed: 25.11.2017).
- [5] Abramov G.V., Gavrilov A.N. Mathematical modeling of motion of interacting particles on the basis of the distribution functions in the plasma arc synthesis of ONS. *Vestnik VGUIT* [Proceedings of the Voronezh State University of Engineering Technologies], 2012, no. 2, pp. 71–75 (in Russ.).
- [6] Cercignani C. *Theory and application of the Boltzmann equation*. Chatto and Windus, 1975. 415 p.
- [7] Abramov G., Gavrilov A., Tolstova I. The use of technology for parallelization method of large particles using cloud computing. *British Journal of Science, Education and Culture*, 2014, no. 2 (6), pp. 380–387.

- [8] Graphics add-in board shipments decline 15.2 % from last quarter. Available at: <https://www.jonpeddie.com/press-releases/graphics-add-in-board-shipments-decline-15.2-from-last-quarter> (accessed: 25.11.2017).
- [9] Demmel J.W. Applied numerical linear algebra. SIAM, 1997. 184 p.
- [10] Uskov R.V. On some peculiarities of application of *CUDA* technology for simulation of transport of radiation. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Estestv. Nauki* [Herald of the Bauman Moscow State Tech. Univ., Nat. Sci.], 2011, no. 3, pp. 71–83 (in Russ.).
- [11] Baraff D. Dynamic simulation of non-penetrating rigid bodies. Ph. D. thesis. Cornell University, Computer Science Department, 1992, pp. 52–56.

Abramov G.V. — Dr. Sc. (Eng.), Professor, Department of Mathematical and Applied Analysis, Voronezh State University (Universitetskaya ploschad 1, Voronezh, 394018 Russian Federation).

Gavrilov A.N. — Cand. Sc. (Eng.), Assoc. Professor, Department of Data Processing and Control Systems, Voronezh State University of Engineering Technologies (Revolyutsii prospekt 19, Voronezh, 394036 Russian Federation).

Ivashin A.L. — Cand. Sc. (Eng.), Assoc. Professor, Department of Higher Mathematics and Information Technology, Voronezh State University of Engineering Technologies (Revolyutsii prospekt 19, Voronezh, 394036 Russian Federation).

Tolstova I.S. — Assist. Professor, Department of Higher Mathematics and Information Technology, Voronezh State University of Engineering Technologies (Revolyutsii prospekt 19, Voronezh, 394036 Russian Federation).

Please cite this article in English as:

Abramov G.V., Gavrilov A.N., Ivashin A.L., Tolstova I.S. Using Parallel Computing in Computationally Intensive Problems of Simulating Particle Motion and Interaction in Plasma during Carbon Nanostructure Synthesis. *Vestn. Mosk. Gos. Tekh. Univ. im. N.E. Baumana, Estestv. Nauki* [Herald of the Bauman Moscow State Tech. Univ., Nat. Sci.], 2018, no. 5, pp. 4–14 (in Russ.). DOI: 10.18698/1812-3368-2018-5-4-14