

ИССЛЕДОВАНИЕ ПРОИЗВОДИТЕЛЬНОСТИ РАЗЛИЧНЫХ АЛГОРИТМОВ РАСЧЕТА ВАН-ДЕР-ВААЛЬСОВЫХ ВЗАИМОДЕЙСТВИЙ В ЗАДАЧАХ МОЛЕКУЛЯРНОЙ ДИНАМИКИ НА ГРАФИЧЕСКИХ УСТРОЙСТВАХ

Д.О. МОСКВИН, В.В. СОЦКИЙ, Н.В. УСОЛЬЦЕВА
ФГБОУВПО «Ивановский государственный университет», НИИ Наноматериалов,
Иваново, Российская Федерация
E-mail: Moskvin1987@mail.ru

Авторское резюме

Состояние вопроса: В связи с тем, что жидкие кристаллы находят широкое применение во многих областях деятельности человека, перед учеными стоит проблема синтеза их с заданными свойствами. Химический синтез по ряду причин, как правило, предваряется компьютерным моделированием. Использование алгоритмов, работающих на графических устройствах, позволяет значительно ускорить процесс моделирования. В настоящее время вопрос использования графических устройств для ускорения компьютерного моделирования в химии изучен недостаточно хорошо.

Материалы и методы. Для расчета ван-дер-ваальсовых взаимодействий использован потенциал Ленард-Джонса. Тот факт, что силы ван-дер-ваальсовых взаимодействий убывают пропорционально шестой степени межатомного расстояния, позволяет использовать методы расчета, основанные на выделении области пространства, где силы ван-дер-ваальсовых взаимодействий существенны, и области, где ими можно пренебречь. Рассматриваются метод с присоединенными списками и метод со списками Верле, реализующие означенный подход. Кроме того, рассмотрен метод полного перебора. В ходе работы были использованы графическое устройство nVidia GeForce 580 GTX и центральный процессор Intel core i7.

Результаты. Реализованы три алгоритма расчета ван-дер-ваальсовых взаимодействий на графических контроллерах (GPU). Для каждого алгоритма рассмотрены две стратегии работы с памятью GPU: в первом случае чтение данных осуществляется из глобальной памяти, во втором – из текстурной памяти.

Выводы: В ходе серии вычислительных экспериментов было выявлено, что алгоритм со списками Верле и алгоритм с присоединенными списками, читающие данные из текстурной памяти, имеют эквивалентную производительность на системе, состоящей из большого количества частиц, а также работают быстрее алгоритмов, читающих из глобальной памяти. Алгоритм полного перебора, напротив, имеет меньшую производительность при работе с текстурной памятью (в 1,5 раза меньше, по сравнению с глобальной), что связано с особенностями архитектуры графического устройства.

Ключевые слова: ван-дер-ваальсовы взаимодействия, молекулярная динамика, списки Верле, присоединенные списки, графические устройства, текстурная память.

PRODUCTIVITY INVESTIGATION OF DIFFERENT ALGORITHMS FOR CALCULATING VAN DER WAALS INTERACTIONS IN MOLECULAR DYNAMICS PROBLEMS ON GRAPHICS UNITS

D.O. MOSKVIN, V.V. SOTSKY, N.V. USOL'TSEVA
Nanomaterials Research Institute of Ivanovo State University, Ivanovo, Russian Federation
E-mail: Moskvin1987@mail.ru

Abstract

Background: The researches consider the actual synthesis problem of the liquid crystals with the given characteristics because of the fact that the liquid crystals are widely used in a lot of human spheres. The chemical synthesis is premised on the computer simulation because of the several reasons. The usage of the algorithms which work on graphical units allows to accelerate the simulation process significantly. At the present time, the issue of graphical units application for accelerating the computer simulation in chemistry is not studied well.

Materials and methods: The Lennard-Jones potential is used for calculating the van-der-waals interactions. The fact that van-der-Waals forces diminish proportionally to the sixth degree of interatomic spacing allows to use the calculation methods based on finding the space area where van-der-waals interactions are substantial and the area where it can be left out of account. The authors consider the cell method and Verlet method of lists which realize the mentioned approach. The method of full surplus is also observed. During the research process the graphic device GeForce 580GTX and the CPU Intel core i7 are used.

Results: Three calculation algorithms of the van-der-waals interactions on graphic units (GPU) are tested. For each algorithm two operation strategies with GPU memory are considered. In the first instance the data reading is realized from the global memory, in the second instance it is made from the texture memory.

Conclusions: During a number of calculation experiments it is revealed that the cell method and the Verlet method of lists reading data from texture memory have the equivalent productivity on a system composed of a large amount of at-

oms and work faster than algorithms reading from the global memory. On the contrary the algorithm of full surplus has less productivity during the work with texture memory (half as less in comparison with global one).

Key words: van-der-waals interactions, molecular dynamics, Verlet lists, cell method, graphical units, texture memory.

Введение. В настоящее время широкое применение во многих областях человеческой деятельности находят жидкие кристаллы (ЖК). В связи с этим как перед теоретиками, так и перед экспериментаторами стоит очень важная задача синтеза ЖК-соединений, обладающих заданным набором свойств. Для этих целей наряду с химическим экспериментом, ввиду его большой трудоемкости, высокой стоимости и негативного влияния на экологию, применяется компьютерное моделирование, которое, в отличие от химического эксперимента, позволяет дать прогноз относительно надмолекулярной организации соединений.

Одним из наиболее распространенных методов компьютерного моделирования в задачах физической химии является метод молекулярной динамики, подробно изложенный в [1–7].

В большинстве случаев моделируемая система состоит из десятков тысяч атомов, моделирование поведения такой системы, даже в наносекундных интервалах времени, требует больших временных затрат. В связи с тем, что некоторые вычисления (например, расчет сил взаимодействия для каждой частицы) могут быть выполнены независимо друг от друга, целесообразно использование технологий параллельных вычислений. Графические устройства (GPU) являются одним из представителей данных технологий. Архитектура GPU, а также способы работы с ними описаны в большом количестве работ, например [8–14]. Основным достоинством графических контроллеров является низкая стоимость, по сравнению с аналогичными по производительности системами. В то же время графические ускорители обладают рядом особенностей, требующих модификации (либо адаптации) уже существующих алгоритмов расчетов для эффективного использования ресурсов.

Расчет ван-дер-ваальсовых взаимодействий. Для расчета ван-дер-ваальсовых взаимодействий был использован потенциал Леннард-Джонса:

$$U = \varepsilon \left(\left(\frac{\sigma}{r} \right)^{12} - 2 \left(\frac{\sigma}{r} \right)^6 \right), \quad (1)$$

где ε – глубина потенциальной ямы; σ – диаметр частицы; r – расстояние между центрами масс частиц.

Наиболее «дорогостоящей» операцией при решении задач молекулярной динамики является расчет сил межмолекулярных взаимодействий. Полный перебор всех взаимодействий требует N^2 операций вычисления (где N – количество частиц). Однако силы, отвечающие ван-дер-ваальсовым взаимодействиям, убывают обратно пропорционально r^6 и на

некотором расстоянии ($\approx 2,5 \sigma$) становятся равными нулю. Это обстоятельство позволяет применять алгоритмы, основанные на выделении области, где силы отличны от нуля, и области, где силами можно пренебречь (можно не рассчитывать). Существует два основных алгоритма, реализующих данный принцип: списки Верле (когда для каждой частицы составляется список ближайших соседей) и присоединенные списки (когда пространство разбивается на отдельные ячейки и суммирование проводится внутри текущей ячейки и между соседними ячейками). При этом объем вычислений растет уже пропорционально N , а не N^2 , как в случае алгоритма с полным перебором всех сил [1, 4].

Указанные алгоритмы были реализованы следующим образом: в процессе работы алгоритма с присоединенными списками обновление списков выполнял центральный процессор (CPU), затем данные копировались в память GPU для последующего расчета сил межчастичных взаимодействий; алгоритм со списками Верле реализован так, что обновление списков и расчет сил взаимодействий выполняло исключительно графическое устройство; реализация алгоритма полного перебора также предусматривала вычисление сил взаимодействий частиц на GPU. Расчет скоростей и координат частиц (интегрирование уравнений движения) во всех трех случаях осуществлялся на CPU.

Некоторые сведения об архитектуре графических контроллеров nVidia. На текущий момент времени существует два поколения графических контроллеров данной фирмы. При этом у обоих поколений присутствует несколько типов памяти, существенно отличающихся по объему и скорости доступа (см. таблицу).

Типы памяти GPU

Тип памяти	Доступ для GPU	Скорость работы
Регистровая	Чтение/запись	Высокая
Локальная	Чтение/запись	Низкая
Общая	Чтение/запись	Высокая
Глобальная	Чтение/запись	Низкая
Константная	Только чтение	Высокая
Тектурная	Только чтение	Высокая

Первое поколение этих устройств характеризовалось более простой архитектурой, по сравнению с центральными процессорами. Ранее нами были реализованы алгоритмы, позволяющие эффективно использовать особенности графических устройств первого поколения для алгоритма полного перебора: создавался программный кэш (на основе общей памяти) и проверялось соответствие записанных в нем данных [14]. При этом удалось дос-

тигнуть 4-кратного повышения производительности.

Второе поколение графических контроллеров фирмы nVidia существенно отличается от первого. Например, у них имеется 768 КБ общей кэш-памяти второго уровня (nVidia parallel data cache) [8], что устраняет необходимость использования программного кэша. В то же время использование кэш-памяти на основе общей памяти является лишь частью стратегии по повышению быстродействия программного обеспечения. При моделировании методом молекулярной динамики использование текстурной памяти приводит к существенному увеличению скорости выполнения программы в случае алгоритмов со списками Верле и присоединенными списками (поскольку в этих алгоритмах трудно осуществить последовательное чтение). Разработчики фирмы nVidia заявляют, что текстурная память для второго поколения графических устройств дает меньший прирост производительности, чем это было для первого поколения [8]. Это делает актуальной проверку выполнения алгоритмов, учитывающих данные особенности.

Схема эксперимента. Каждый из трех алгоритмов вычисления сил ван-дер-ваальсовых взаимодействий был реализован в двух вариантах. В первой реализации чтение данных производилось из глобальной памяти, во второй – из текстурной. Эксперименты выполнены при помощи графического контроллера nVidia GeForce 580 GTX (второе поколение GPU) и центрального процессора Intel core i7.

Схема эксперимента составлена следующим образом: по каждому из шести алгоритмов (полный перебор, присоединенные списки и списки Верле с чтением данных из глобальной памяти, полный перебор, присоединенные списки и списки Верле с чтением данных из текстурной памяти) было выполнено 10000 итераций над одной и той же начальной конфигурацией. Количество частиц в начальной конфигурации варьировали от 8192 до 98304 частиц с шагом 8192 (данные цифры кратны 256, по количеству потоков на один блок графического контроллера). По завершении работы алгоритма рассчитывалось время, затраченное на вычисления. Плотность частиц моделируемой системы поддерживалась постоянной за счет использования зеркальных граничных условий [3, 4, 6] и расчета объема фазового пространства по количеству частиц моделируемой системы. Эксперименты были последовательно выполнены на двух начальных конфигурациях системы.

В первой стартовой конфигурации элементы массивов координат заполнялись таким образом, что координаты имели одномерную упорядоченность по оси z, во второй – частицы располагались случайным образом. Вторая конфигурация была выбрана с учетом особен-

ностей архитектуры графического устройства GeForce GTX 580 (наличие аппаратного кэша второго уровня) и особенностей алгоритмов, имеющих привязку номеров частиц к пространственному расположению.

Результаты эксперимента при наличии пространственной упорядоченности. При такой начальной конфигурации частицы, имеющие соседние пространственные координаты имеют последовательные номера в массивах координат, что делает возможным их кэширование при работе алгоритмов со списками Верле и присоединенными списками (при работе алгоритмов полного перебора чтение выполняется последовательно). Было выдвинуто предположение, что в рамках этой конфигурации можно будет получить больший выигрыш от аппаратного кэша.

Зависимости времени вычисления от количества частиц для различных алгоритмов представлены на рис. 1, 2.

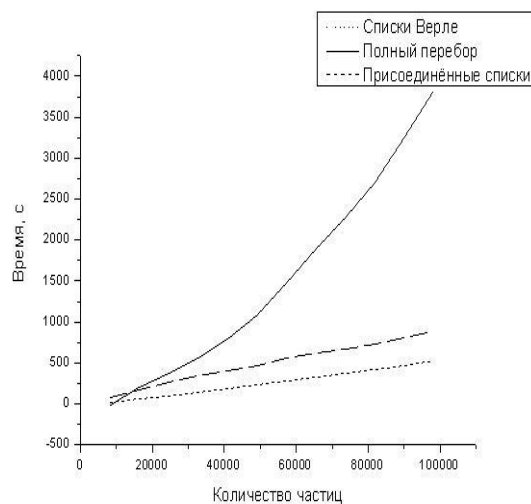


Рис. 1. Производительность алгоритмов, использующих глобальную память, на упорядоченной конфигурации

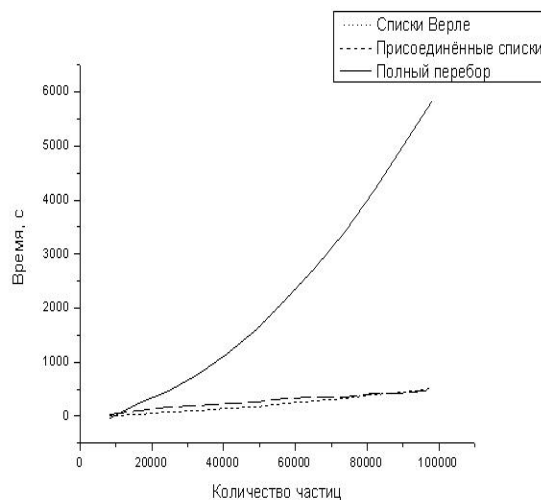


Рис. 2. Производительность алгоритмов, использующих текстурную память, на упорядоченной конфигурации

Анализ полученных зависимостей (рис. 1, 2) показывает, что версии алгоритмов с присоединенными списками и со списками Верле, где чтение данных производится из текстурной памяти, работают несколько быстрее алгоритмов, в которых чтение осуществляется из глобальной памяти. Что касается алгоритма полного перебора, то алгоритм с чтением данных из текстурной памяти работает примерно в 1,5 раз медленнее алгоритма с чтением из глобальной памяти.

Результаты эксперимента при отсутствии пространственной упорядоченности. При такой начальной конфигурации частицы, имеющие соседние пространственные координаты, не имеют последовательных номеров в массивах координат, что делает невозможным их кэширование при работе алгоритмов со списками Верле и присоединенными списками. Было выдвинуто предположение, что в рамках этой конфигурации можно будет получить большой выигрыш от текстурного кэша.

Зависимости времени вычисления от количества частиц для различных алгоритмов представлены на рис. 3, 4.

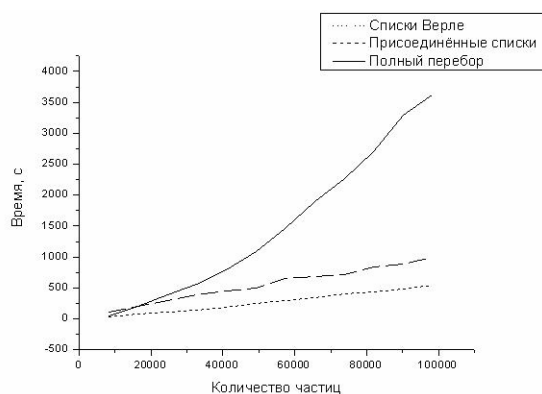


Рис. 3. Производительность алгоритмов, использующих глобальную память, на неупорядоченной конфигурации

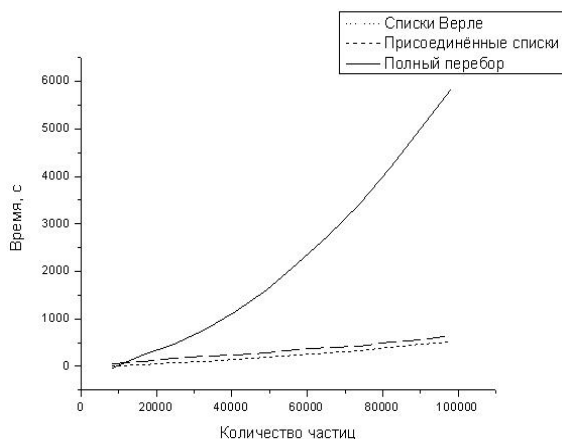


Рис. 4. Производительность алгоритмов, использующих текстурную память, на неупорядоченной конфигурации

Анализ полученных зависимостей (рис. 3, 4) показывает, что версии алгоритмов с присоединенными списками и со списками Верле да-

ют аналогичные результаты, по сравнению с системой, в которой присутствует пространственная упорядоченность.

Таким образом, гипотеза о том, что алгоритмы со списками Верле и с присоединенными списками, выполняющие чтение из текстурной памяти, в случае отсутствия упорядоченности будут работать быстрее, чем в случае упорядоченной конфигурации, не подтвердилась (см. рис. 5 и 6). Алгоритмы полного перебора на обеих конфигурациях (с чтением из текстурной памяти и чтением из глобальной памяти) показали идентичную производительность, что объясняется последовательным чтением данных из памяти (в этом случае порядок заполнения массивов координат роли не играет).

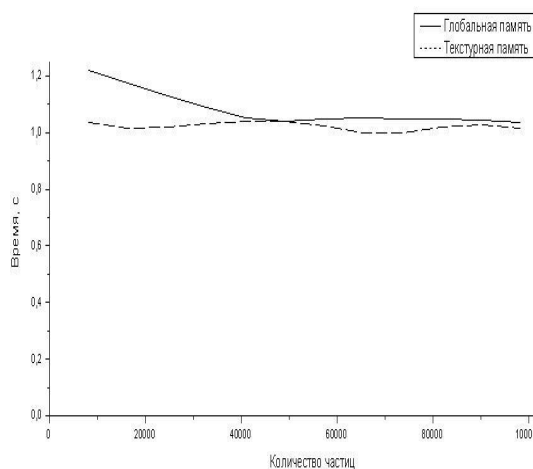


Рис. 5. Отношение времени неупорядоченной конфигурации к времени упорядоченной для алгоритма со списками Верле

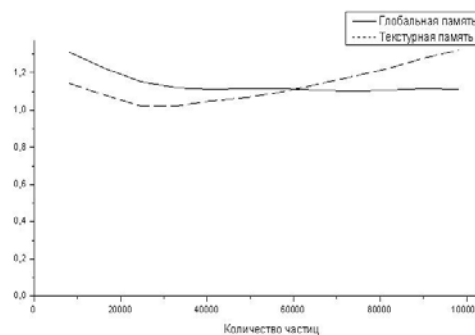


Рис. 6. Отношение времени неупорядоченной конфигурации к времени упорядоченной для алгоритма с присоединенными списками

Заключение

Таким образом, версии алгоритмов с присоединенными списками и со списками Верле, где чтение данных производится из текстурной памяти, работают несколько быстрее алгоритмов, в которых чтение осуществляется из глобальной памяти. Работа аппаратного кэша наиболее эффективно реализуется только для алгоритмов полного перебора, где

чение из глобальной памяти осуществляется в 1,5 раза быстрее, чем из текстурной памяти. Это связано с тем, что в процессе работы алгоритма полного перебора данные из памяти считываются последовательно. В то же время применение текстурной памяти делает приложение переносимым с одной архитектуры на другую, что актуально, например, для гетерогенных вычислительных систем.

Работа поддержана грантами: РФФИ 10-03-00881 и ФЦП 16.740.11.0206

Список литературы

1. Alen M.P., Tildesley D.J. Computer simulations of liquids. – Oxford: Clarendon press, 1991. – 385 p.
2. Demus D., Goolby J., Gray G. W., Spiess H.- W., Vill V. Handbook of liquid crystals. Vol. 1. – Wiley-VCH, 1998. – 2591 p.
3. Соцкий В.В., Усольцева Н.В. Методы моделирования мезоморфных систем. Деп. В ВИНТИ. 03.12.2007 № 1116. 2007. 58 с.
4. Методы компьютерного моделирования для исследования полимеров и биополимеров / под ред. В.А. Иванова, А.П. Рабиновича, А.Р. Хохлова. – М.: КД Либрком, 2009. – 696 с.
5. Аксёнова Е.В., Кшевецкий М.С. Вычислительные методы исследования молекулярной динамики. – СПб.: СПбГУ, 2009. – 50 с.
6. Получение стереографических изображений в задачах молекулярной динамики / Н.В. Усольцева, Ф.Н. Ясинский, В.В. Соцкий, М.С. Костин // Вестник ИГЭУ. – 2009. – Вып. 4. – С. 1–5.
7. Ибрагимов И.М., Ковшов А.Н., Назаров Ю.Ф. Основы компьютерного моделирования наносистем: учеб. пособие. – СПб.: Лань, 2010. – 386 с.
8. Документация по интерфейсу программирования CUDA, CUDA Documentation [Электронный ресурс]. – URL: <http://developer.nvidia.com/category/zone/cuda-zone>.
9. Giupponi G., Harvey M.J., De Fabritis G. The impact of accelerator processors for high-throughput molecular modeling and simulation // Drug discovery today. – 2008. – Vol. 13. – P. 1052–1058.
10. Anderson J.A., Lorenz C.D., Travesset A.A. General purpose molecular dynamics simulations fully implemented on graphics processing units // Journal of Computational Physics. – 2008. – Vol. 227. – P. 5342–5359.
11. Боярченко А.С., Поташников С.И. Использование графических процессоров и технологии CUDA для задач молекулярной динамики // Вычислительные методы и программирование. – 2009. – Т. 10. – С. 9–23.
12. Боярченко А.С., Поташников С.И. Параллельная молекулярная динамика с суммированием Эвальда и интегрированием на графических процессорах // Вычислительные методы и программирование. – 2009. – Т. 10. – С. 158–175.

Усольцева Надежда Васильевна,
Ивановский государственный университет,
доктор химических наук, профессор, директор НИИ Наноматериалов,
телефон (4932) 37-08-08.

Соцкий Валентин Викторович,
Ивановский государственный университет,
кандидат химических наук, научный сотрудник НИИ Наноматериалов,
телефон (4932) 37-08-08.

Москвин Дмитрий Олегович,
Ивановский государственный университет,
аспирант НИИ Наноматериалов,
телефон (4932) 37-08-08.

13. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. – М.: ДМК Пресс, 2010. – 232 с.
14. Соцкий В.В. Опыт применения графических контроллеров к решению задач молекулярной динамики // Жидкие кристаллы и их практическое применение. – Иваново: Иван. гос. ун-т, 2011. – Вып. 3. – С. 77–84.

References

1. Alen, M.P., Tildesley, D.J. Computer simulations of liquids, Oxford: Clarendon press, 1991, 385 p.
2. Demus, D., Goolby, J., Gray, G. W., Spiess, H.- W., Vill, V. Handbook of liquid crystals, vol. 1, Wiley-VCH, 1998, 2591 p.
3. Sotskiy, V.V., Usol'tseva, N.V. *Metody modelirovaniya mezomorfnykh sistem* [Simulation Methods of Mesomorphic Systems], Dep. v VINITI, 03.12.2007, № 1116. 2007, 58 p.
4. Ivanov, V.A., Rabinovich, A.L., Khokhlov, A.R. *Metody komp'yuternogo modelirovaniya dlya issledovaniya polimerov i biopolimerov* [Computer Simulation Methods of Polymer and Biopolymer Researches], Moscow: KD Librkom, 2009, 696 p.
5. Aksenova, E.V., Kshevetsiy, M.S. *Vychislitel'nye metody issledovaniya molekulyarnoy dinamiki* [Research Calculation Methods of Molecular Dynamics], Sankt Peterburg: SPbGU, 2009, 50 p.
6. Usol'tseva, N.V., Yasinskiy, F.N., Sotskiy, V.V., Kostin, M.S. *Vestnik IGEU*, 2009, issue 4, pp. 1–5.
7. Ibragimov, I.M., Kovshov, A.N., Nazarov, Yu.F. *Osnovy komp'yuternogo modelirovaniya nanosistem* [Foundations of Nanosystems Computer Simulation], Sankt Peterburg: Lan', 2010, 386 p.
8. *Dokumentatsiya po interfeysu programmirovaniya CUDA, CUDA Documentation* [Documentation for CUDA Programming Interface], URL: <http://developer.nvidia.com/category/zone/cuda-zone>.
9. Giupponi, G., Harvey, M.J., De Fabritis, G. The impact of accelerator processors for high-throughput molecular modeling and simulation, in Drug discovery today, 2008, vol. 13, pp. 1052–1058.
10. Anderson, J.A., Lorenz, C.D., Travesset, A.A. General purpose molecular dynamics simulations fully implemented on graphics processing units. in Journal of Computational Physics, 2008, vol. 227, pp. 5342–5359.
11. Boyarchenkov, A.S., Potashnikov, S.I. *Vychislitel'nye metody i programmirovaniye*, 2009, vol. 10, pp. 9–23.
12. Boyarchenkov, A.S., Potashnikov, S.I. *Vychislitel'nye metody i programmirovaniye*, 2009, vol. 10, pp. 158–175.
13. Boreskov, A.V., Kharlamov, A.A. *Osnovy raboty s tekhnologiyey CUDA* [Work Foundations with CUDA], Moscow: DMK Press, 2010, 232 p.
14. Sotskiy, V.V. Opyt primeneniya graficheskikh kontrollerov k resheniyu zadach molekulyarnoy dinamiki [Application Experience of Graphical Units for Solving Molecular Dynamics Problems], in *Zhidkie kristally i ikh prakticheskoe primeneniye*, Ivanovo: Ivanovskiy gosudarstvennyy universitet, 2011, issue 3, pp. 77–84.