

## Моделирование механики жидкости и газа с применением технологии CUDA

В.В. Неткачев  
ФГБОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,  
г. Иваново, Российская Федерация  
E-mail: vovanok.net@gmail.com

### Авторское резюме

**Состояние вопроса:** Существующие методы решения задач механики жидкостей и газов на ЭВМ требуют больших затрат машинного времени, использования больших дорогостоящих суперкомпьютеров и часто не позволяют решать эти задачи в реальном времени. В связи с этим актуальной является разработка методов и способов их распараллеливания на суперкомпьютерах с графическими ускорителями.

**Материалы и методы:** Используются известные методы численного решения дифференциальных уравнений и технология CUDA.

**Результаты:** Описаны подходы к распараллеливанию численных методов решения дифференциальных уравнений при имеющихся входных данных: программная модель, данные вычислительной модели и алгоритм вычисления. В их числе подходы к распараллеливанию явных и неявных схем с использованием технологии CUDA. Приведены описание программного решения в виде класса-декоратора для пользовательских решателей метода прогонки и результаты практического применения подходов для решения задачи развития лесных пожаров.

**Выводы:** Разработанные быстрые алгоритмы универсальны, эффективны и удобны, поэтому могут быть использованы во многих сферах науки и техники, в том числе для моделирования процессов в реальном времени, а также в задачах, где требуется перебор большого количества вариантов.

**Ключевые слова:** численное моделирование, подход к распараллеливанию, явная схема, метод прогонки, модель массового параллелизма, CUDA, декоратор, прогнозирование лесных пожаров, энергетика.

## Liquid and Gas Mechanics Modeling with CUDA Technology

V.V. Netkachev  
Ivanovo State Power Engineering University, Ivanovo, Russian Federation  
E-mail: vovanok.net@gmail.com

### Abstract

**Background:** The existing methods for solving the mechanics problems of liquid and gases on computers require large resources, usage of large and expensive supercomputers and often it is impossible to solve the problem in real time. Thus, the development of the methods and ways of their paralleling on the supercomputers with graphic accelerators is very urgent.

**Materials and methods:** The author uses the famous methods of numerical solution to differential equations and CUBA technologies in the article.

**Results:** The article describes a new approach for parallelization of differential equations with certain beginning data: program model, calculation model data and calculation algorithm. It includes the approaches for parallelization of explicit and implicit schemes with the CUDA technology usage. The program solution description is given as class-decorator for solvers of sweep method. The results of practical usage of the approaches were represented for forest fire forecast.

**Conclusion:** The developed fast algorithms are universal, effective and easy to use. Therefore, they can be used in many areas of science and technology, including the simulation processes in real-time, as well as in applications where a large number of options are required.

**Key words:** numerical simulation, approach to parallelization, explicit scheme, sweep method, massively parallelization model, CUDA, decorator, forest fires forecast, power engineering.

Моделирование задач механики жидкости и газа важно в различных сферах науки и техники. В качестве целей моделирования таких задач можно выделить две:

1. *Прогнозирование.* Например, прогнозирование развития чрезвычайных ситуаций (ЧС) (лесного пожара, разлива нефти, лавы, наводнений и т.д.).

2. *Оптимизация.* Например, оптимизация вентиляционной системы помещения для обеспечения наиболее комфортной для людей и оборудования циркуляции воздуха, оптимизация форм

трубопроводов для обеспечения наилучшей производительности жидкостей и газов и т.д.

Проблема оптимизации воздушных потоков остро стоит в области энергетики. Часто случаются проблемы с вентиляцией машинных отделений на электростанции, когда из-за неравномерной циркуляции воздуха возникает перепад температур в разных частях генератора, что приводит к его заклиниванию. Эту ошибку, приводящую к выводу из строя очень дорогой техники, возможно исправить с помощью моделирования.

Для того чтобы добиться полученных целей, необходимы быстрые вычисления с использованием суперкомпьютерных технологий. В случае если вычисления будут происходить медленно, к моменту окончания расчета результаты потеряют свою ценность. Например, если расчет прогнозирования лесного пожара будет происходить несколько часов, то к окончанию расчета лес либо сгорит, либо его уже потушат, и результат не будет иметь ценности. В случае с оптимизацией воздушных потоков в помещении нужно перебрать очень большое количество вариантов расположения вентиляционных каналов. Причем в каждом варианте моделирование будет происходить в виде расчета большого количества итераций. В лучшем случае расчеты будут происходить сутками, в худшем – результат так и не будет получен.

Рассмотрим эффективные методики распараллеливания вычислений на графических процессорах.

Графические процессоры в качестве аппаратного инструмента выбраны неслучайно. Последнее время они получили большое распространение в связи с легкостью их распространения и низкой стоимостью. Аналогичные с графическими ускорителями по производительности суперкомпьютеры стоят в сотни раз дороже. Суперкомпьютеры занимают большие залы и доступны только для специализированных организаций: образовательных учреждений, крупных коммерческих организаций и др. Графические процессоры, напротив, присутствуют сейчас почти в каждом домашнем компьютере (по крайней мере, в игровом точно).

В качестве технологии программирования выбрана технология CUDA, обеспечивающая удобный интерфейс к графическим ускорителям NVIDIA. На сегодняшний день она самая популярная, ввиду своего удобства, совместимости с различными языками и средами программирования, постоянного развития и поддержки.

Технология CUDA имеет специфическую программную модель, которая накладывает массу ограничений и допущений при ее применении к распараллеливанию. Наложение программной модели CUDA на данные и алгоритм решения задачи назовем подходом к распараллеливанию.

Разработка эффективного подхода к распараллеливанию является сложной наукоёмкой задачей. Подходы могут быть очень простыми или очень сложными, использующими тонкости абстракции, которую предоставляет технология.

Перед разработкой подходов к распараллеливанию различных схем рассмотрим, что из себя представляет программная модель CUDA [3].

CUDA предоставляет программную абстракцию для работы с графическим процессором (GPU) как с сопроцессором, который выступает в качестве мясорубки. При этом загрузка мяса в мясорубку, а также операции по ее настройке и

включению выполняются со стороны центрального процессора (CPU).

Абстракция CUDA представляет собой сетку большого количества виртуальных потоков, соизмеримого с размерностью задачи. При этом от пользователя скрыто, каким образом виртуальные потоки выполняются физически. Потоки используют общую память, называемую глобальной. Только с ней может взаимодействовать CPU, передавая в нее мясо для прокрутки. Потоки разбиты на блоки потоков, каждый из которых использует свою разделяемую память в качестве кэша (работа с разделяемой памятью гораздо быстрее, чем с глобальной). Также у каждого потока есть в распоряжении регистры и локальная память, доступные только им. Существуют и другие виды памяти, доступные потокам (рис. 1). Каждый вид имеет определенную специфику использования. Эффективным использованием можно считать гибкое сочетание разных видов памяти.

Описанную программную модель требуется применить для решения задач моделирования, которое представляет собой численное решение системы дифференциальных уравнений, описывающих тот или иной процесс.

Для решения будем применять явные и неявные разностные схемы. Балансирование между этими двумя схемами – это балансирование между вычислительной устойчивостью и сложностью наложения программной модели. Наложение программной модели на явную схему – задача простая. Наложение на неявную схему преследует собой ряд трудностей. Для каждого метода, использующего неявную схему, трудности разные.

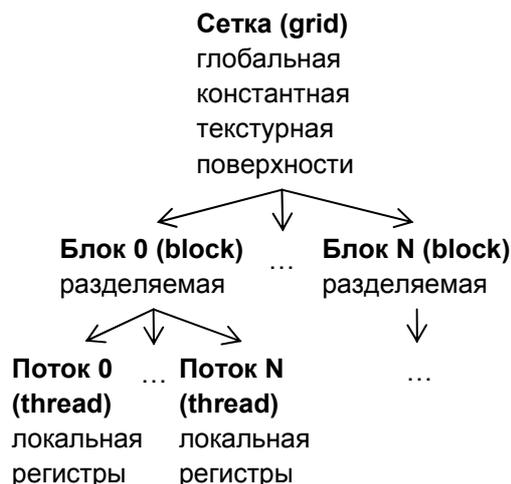


Рис. 1. Схематическая интерпретация памяти CUDA

Наложение программной модели на явную схему [5] представляет собой выделение сетки потоков, равной размерности вычисляемого поля, и ассоциирование каждого потока с одной точкой поля. Один поток рассчитывает свою точку и пишет результат в глобальную память. Алгоритм выполнения ядра (функции, выполняющейся на

устройстве) можно представить следующим образом (повествование от лица потока):

1. Мы с потоками моего блока пишем в разделяемую память из глобальной свои точки, причем если я нахожусь на границе подобласти, на которую попадает мой блок, то я пишу в разделяемую память также точку, прилегающую к этой подобласти (это необходимо, потому что для расчета текущей точки необходимы значения соседних точек, а на границах их попросту нет) (рис. 2).

2. Используя только данные из разделяемой памяти (это гораздо быстрее, чем напрямую из глобальной), я рассчитываю свою точку и записываю ее в регистр или локальную память.

3. Записываю свою точку в глобальную память.

Предлагаемый подход прост, и любые попытки оптимизации в большинстве своем приведут только к замедлению вычисления.

В качестве исследований наложения программной модели на неявную схему численного решения дифференциальных уравнений возьмем метод прогонки [2].

Метод прогонки состоит из следующих этапов:

- наложение разностной сетки;
- разложение уравнения по соответствующему измерению в виде

$$\alpha_{\text{текущий}} Q_{\text{предыдущий}}^{\text{новый}} + \beta_{\text{текущий}} Q_{\text{текущий}}^{\text{новый}} + \gamma_{\text{текущий}} Q_{\text{следующий}}^{\text{новый}} = Q_{\text{текущий}}^{\text{старый}};$$

- прямая и обратная прогонка по каждому измерению. Прямая прогонка заключается в вычислении прогоночных коэффициентов  $L$  и  $M$ :

$$L_{\text{следующий}} = -\frac{\gamma_{\text{текущий}}}{\beta_{\text{текущий}} + \alpha_{\text{текущий}} L_{\text{текущий}}};$$

$$M_{\text{следующий}} = \frac{Q_{\text{текущий}}^n - \alpha_{\text{текущий}} M_{\text{текущий}}}{\beta_{\text{текущий}} + \alpha_{\text{текущий}} L_{\text{текущий}}}.$$

Обратная прогонка заключается в вычислении нового значения величины по прогоночным коэффициентам:

$$U_{\text{текущий}}^{\text{новый}} = L_{\text{текущий}} U_{\text{следующий}}^{\text{старый}} + M_{\text{текущий}}.$$

Количество пар «прямая прогонка – обратная прогонка» зависит от количества измерений вычисляемого поля.

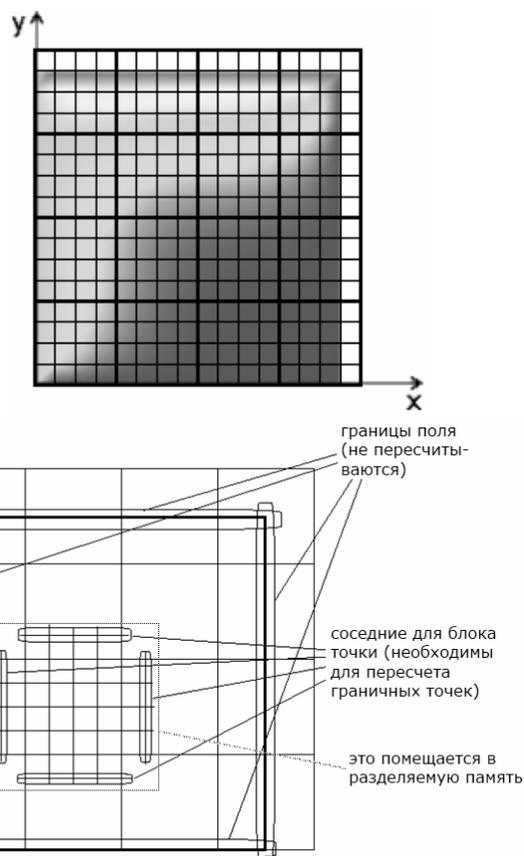


Рис. 2. Наложение программной модели CUDA на явную разностную схему

Согласно описанному подходу, простая ассоциация поток-точка невозможна, потому что значение вычисляется в два этапа, причем оно зависит от следующего по соответствующей координатной оси значения и прогоночных коэффициентов, значения которых зависят от предыдущего значения.

Рассмотрим прямой подход к распараллеливанию данного метода численного решения дифференциальных уравнений. Для этого воспользуемся одномерной сеткой потоков программной модели CUDA. Она разделена на одномерные блоки. Для двумерной задачи каждому потоку соответствует одна строка поля при прогонке вдоль оси  $X$  и один столбец при прогонке вдоль оси  $Y$ . Таким образом, при размере расчетного поля  $M \times N$  размер сетки при прогонке по  $X$  будет равен  $N \times 1$ , а по  $Y$  –  $M \times 1$ .

Каждый поток рассчитывает свою строку/столбец (рис. 3).

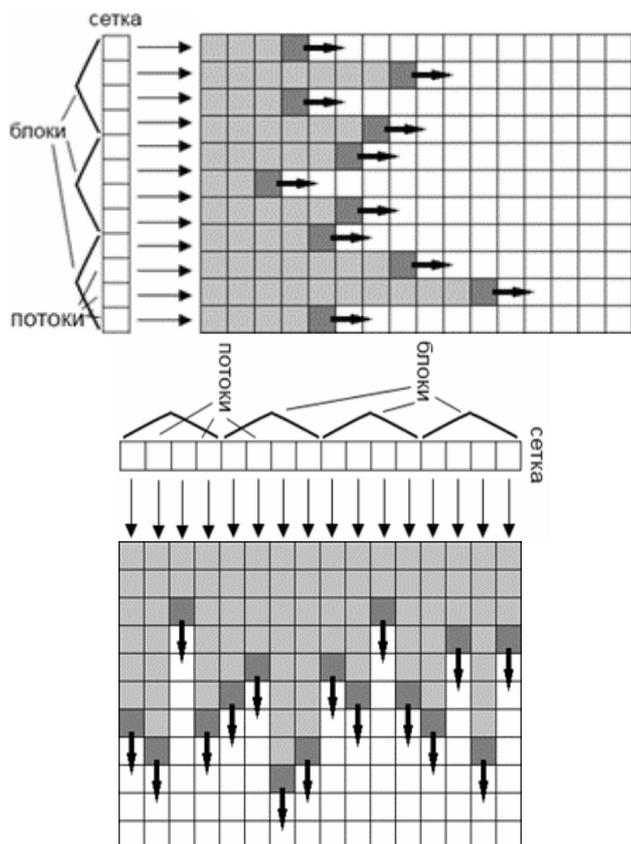


Рис. 3. Прямой подход к наложению программной модели CUDA на метод прогонки

К недостатком такого подхода можно отнести следующие:

- требуется реализация и запуск отдельных ядер при прогонке по  $X$  и по  $Y$ ;
- степень параллелизма невысокая. Количество потоков при каждом запуске ядер прогонок по  $X$  и по  $Y$  зависит только от длины и ширины поля соответственно;
- нагрузка на потоковые процессоры переменная. При прогонке по  $X$  потоки обрабатывают  $M$  точек, а при прогонке по  $Y$  –  $N$  точек (при размерности поля  $M \times N$ ).

Рассмотренный подход несложен в реализации, но не позволяет в полной мере задействовать всю мощь графических ускорителей.

Следующий подход лишен таких недостатков, но наделен другими. Он заключается в ассоциации потоку не целой строки/столбца, а ее части фиксированного размера, равного размеру блока потоков CUDA. Таким образом, одну строку/столбец обрабатывает не один поток, а  $M / \langle \text{размер блока} \rangle$  или  $N / \langle \text{размер блока} \rangle$  для строк и столбцов соответственно (рис. 4).

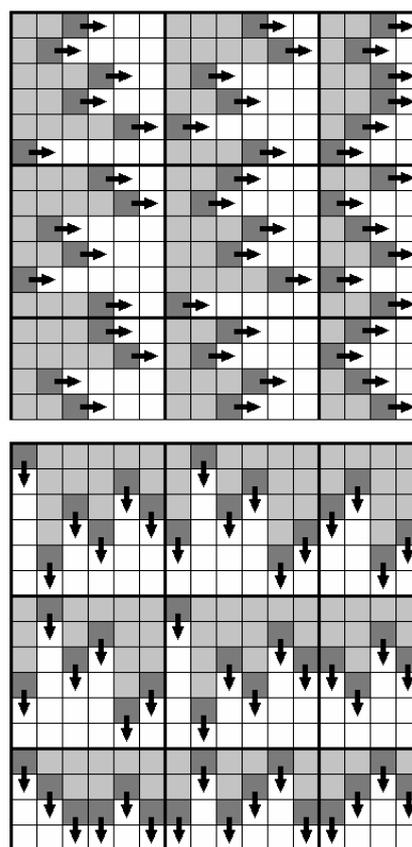


Рис. 4. Улучшенный подход к наложению программной модели CUDA на метод прогонки

Этот подход убирает недостатки предыдущего подхода: нагрузка на потоки теперь одинакова; можно выделить большее количество потоков, что повышает степень параллелизма; прогонки по  $X$  и по  $Y$  можно объединить в одно вычислительное ядро.

Недостатки этого подхода:

- необходима синхронизация блоков. Так как для вычисления следующего значения в точке поля нужны соседние, то будут проблемы с вычислениями на границах. Для этого можно воспользоваться явными схемами для вычислений на границах, схожими по сходимости с методом прогонки (например, схема Головичева или схема Дюфорта-Франкеля);

• схемы Головичева и Дюфорта-Франкеля [1] используют для расчета позавчерашние значения точек, и в результате нужно хранить в памяти еще одно поле позавчерашних значений.

Прямой подход был реализован в виде универсальной объектной модели решателей (рис. 5).

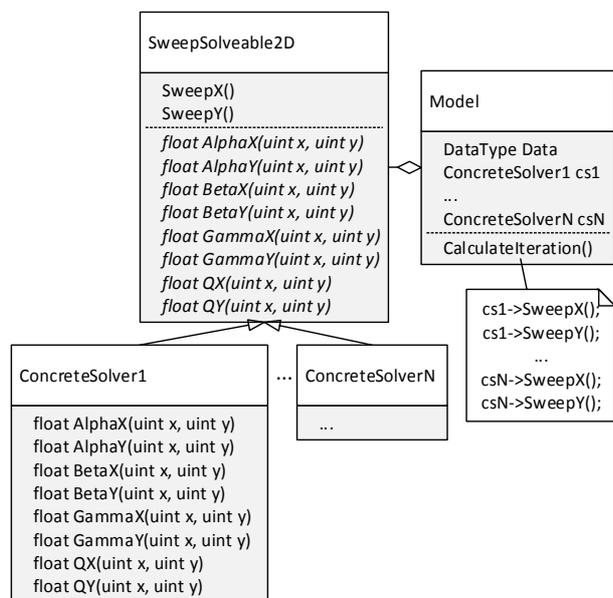


Рис. 5. Диаграмма классов решателей

Создан класс SweepSolveable2D, который умеет эффективно выполнять прогонку для той или иной задачи. У него есть методы для выполнения соответствующих прогонок: SweepX(), SweepY(), а также виртуальные абстрактные методы для получения соответствующих частей уравнения  $\alpha$ ,  $\beta$ ,  $\gamma$  и правой части Q: AlphaX(), AlphaY(), BetaX(), BetaY(), GammaX(), GammaY(), QX(), QY() и другие служебные методы. Все эти методы возвращают тип float и принимают на вход два параметра x и y целочисленного типа. Для того чтобы воспользоваться SweepSolveable2D, надо создать класс решателя, который будет содержать или иметь доступ к некоторому полю данных, унаследовать его от SweepSolveable2D и переопределить виртуальные методы получения  $\alpha$ ,  $\beta$ ,  $\gamma$ , Q, а также несколько служебных методов. При этом SweepSolveable2D будет являться декоратором для решателей, создаваемых пользователем [4]. После этого у решателя можно вызывать соответствующие методы SweepX(), SweepY(), которые будут эффективным способом выполнять прогонку над данными решателя.

Полученная разработка была применена для решения задачи прогнозирования лесных пожаров [6, 8]. Использование графических ускорителей для ускорения вычислений позволило повысить скорость расчета более чем в 5 раз (использовался прямой подход) (рис. 5).

Входные и выходные данные [7] представляются в виде kml файлов, которые впоследствии могут быть открыты, например, программой Google Earth (рис. 7).

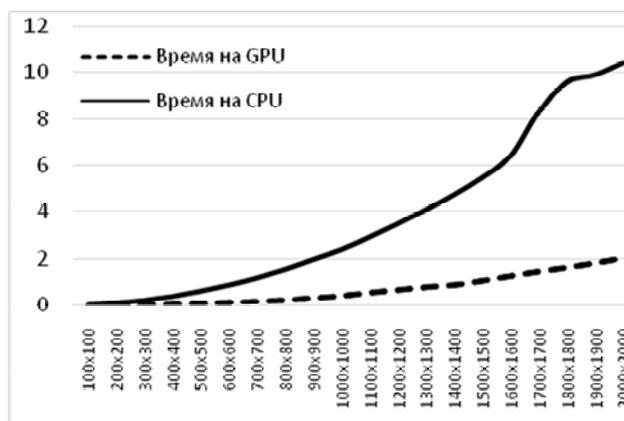


Рис. 6. Зависимость времени расчета от размерности поля

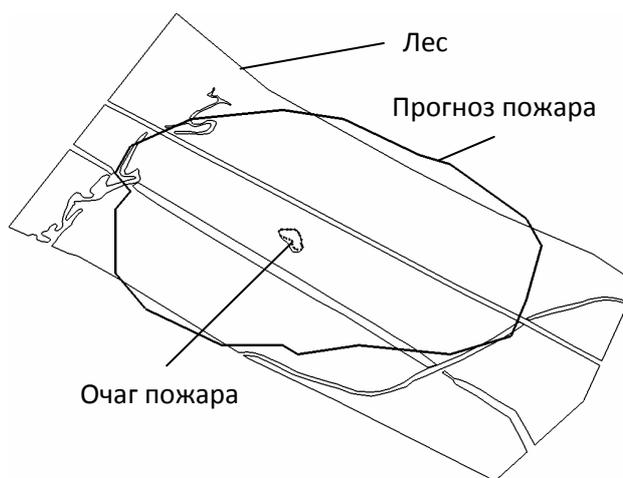


Рис. 7. Визуализация прогноза пожара

Расчет выполнен при западном легком ветре на 1 час.

#### Список литературы

1. Численные методы и параллельные вычисления для задач механики жидкости, газа и плазмы: учеб. пособие / Э.Ф. Балаев, Н.В. Нурдин, В.В. Пекунов и др.; Иван. гос. энерг. ун-т. – Иваново, 2003.
2. Самарский А.А. Введение в теорию разностных схем. – М.: Наука, 1971
3. CUDA C Programming guide, PG-02829-001\_v5.0 | October 2012 Design Guide.
4. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Д. Приемы объектно-ориентированного проектирования. Паттерны проектирования. – СПб.: Питер, 2012.
5. Вержицкий В.М. Основы численных методов: учеб. для вузов. – М.: Высш. шк., 2002.
6. Неткачев В.В. Моделирование распространения лесного пожара на компьютерах с графическими ускорителями: сб. тр. межвуз. науч.-техн. конф. аспирантов и студентов «Молодые ученые – развитию текстильной и легкой промышленности» (Поиск – 2012). – Иваново, 2012.
7. Мочалов А.С. Проектирование на географическую карту модели развития лесного пожара: сб. тр. межвуз. науч.-техн. конф. аспирантов и студентов «Молодые ученые – развитию текстильной и легкой промышленности» (Поиск – 2012). – Иваново, 2012.
8. Prediction and modeling of the forest using neural networks and supercomputers / I.A. Maliy, O.V. Potemkina, I.F. Yasinskiy, F.N. Yasinskiy, S.G. Sidorov, A.S. Mochalov, V.V. Netkachev & L.P. Chernysheva // Third International Conference on Modeling, Monitoring and Modeling of Forest Fires «Forest Fires 2012».

## References

1. Balaev, E.F., Nuzhdin, N.V., Pekunov, V.V., Sidorov, S.G., Chernysheva, L.P., Yasinskiy, I.F., Yasinskiy, F.N. *Chislennyye metody i parallel'nye vychisleniya dlya zadach mekhaniki zhidkosti, gaza i plazmy* [Numerical Methods and Parallel Calculations for Solving the Mechanics Problems of Liquid, Gas and Plasma]. Ivanovo, 2003.
2. Samarskiy, A.A. *Vvedenie v teoriyu raznostnykh skhem* [Introduction into the Theory of Difference Scheme]. Moscow, Nauka, 1971.
3. CUDA C Programming guide, PG-02829-001\_v5.0 | October 2012 Design Guide.
4. Gamma, E., Khelm, R., Dzhonson, R., Vlissides, D. *Priemy ob'ektno-orientirovannogo proektirovaniya. Patterny proektirovaniya* [Ways of Object-oriented Design. Designing Patterns]. Saint-Petersburg, Piter, 2012.
5. Verzhbitskiy, V.M. *Osnovy chislennykh metodov* [Foundations of Numerical Methods]. Moscow, Vysshaya shkola, 2002.
6. Netkachev, V.V. *Modelirovanie rasprostraneniya lesnogo pozhara na komp'yuterakh s graficheskimi uskoritelyami* [Modelling of Forest Fire Spread on the Computers with Graphic Accelerators]. *Sbornik trudov mezhvuzovskoy nauchno-tekhnicheskoy konferentsii aspirantov i studentov «Molodye uchenye – razvitiyu tekstil'noy i legkoy promyshlennosti» (Poisk – 2012)* [Collected Works of Universities Scientific and Technical Conference of Post Graduate Students and Students «Young scientists for developing the Textile and Light Industry»]. Ivanovo, 2012.
7. Mochalov, A.S. *Proektirovanie na geograficheskuyu kartu modeli razvitija lesnogo pozhara* [Designing the Model of Forest Fire Spread on the Geographical Map]. *Sbornik trudov mezhvuzovskoy nauchno-tekhnicheskoy konferentsii aspirantov i studentov «Molodye uchenye – razvitiyu tekstil'noy i legkoy promyshlennosti» (Poisk – 2012)* [Collected Works of Universities Scientific and Technical Conference of Post Graduate Students and Students «Young scientists for developing the Textile and Light Industry»]. Ivanovo, 2012.
8. Maliy, I.A., Potemkina, O.V., Yasinskiy, I.F., Yasinskiy, F.N., Sidorov, S.G., Mochalov, A.S., Netkachev, V.V., Chernysheva, L.P. *Prediction and modeling of the forest using neural networks and supercomputers. Third International Conference on Modeling, Monitoring and Modeling of Forest Fires «Forest Fires 2012»*.

*Неткачев Владимир Владимирович*

ФГБОУ ВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,  
аспирант кафедры высокопроизводительных вычислительных систем,  
e-mail: vovanok.net@gmail.com