

УДК 004.94

## Компьютерное моделирование движения воздуха в условиях городской застройки с использованием NVIDIA CUDA<sup>1</sup>

А.Л. Архипов

ФГБОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,  
Иваново, Российская Федерация  
E-mail: AIArMail999@mail.ru

### Авторское резюме

**Состояние вопроса:** Моделирование движения воздуха в условиях городской застройки является важной задачей, позволяющей определять направление распространения потоков воздуха в зависимости от расположения городских строений. Подобные задачи обычно решаются сеточными методами на многопроцессорных кластерах, позволяющих выполнять вычисления с большой скоростью. Однако стоимость такого компьютерного оборудования достаточно высока. Предлагается использовать в качестве вычислительного средства графический процессор. На сегодняшний день два основных игрока рынка графических процессоров – NVIDIA и AMD – предлагают технологии использования своих GPU для вычислений общего назначения (General-purpose computing on graphics processing units – GPGPU). Наиболее развитой технологией выполнения вычислений общего назначения на GPU является платформа NVIDIA CUDA.

**Материалы и методы:** Используется платформа NVIDIA CUDA, позволяющая выполнять вычисления на графических процессорах. Выбор этой платформы был обусловлен ее продуманным API и поддержкой самых распространенных на сегодняшний день видеокарт NVIDIA GeForce. Для использования графического процессора в качестве вычислительного средства адаптирована конечно-разностная схема вычислений.

**Результаты:** Предложен вариант решения задачи моделирования движения воздуха в условиях городской застройки с использованием графических процессоров современных видеокарт в качестве вычислительного устройства.

**Выводы:** Адаптированная конечно-разностная схема и алгоритм на ее основе позволили получить прирост скорости вычислений от использования графического процессора, по сравнению с вариантом алгоритма, ориентированным на работу с центральным процессором. В результате был получен значительный прирост скорости вычислений. Использование графических процессоров в качестве вычислительного средства и соответствующая адаптация алгоритмов позволяют снизить в 60–70 раз стоимость компьютерного оборудования, необходимого для вычислений, при решении задачи моделирования движения воздуха в условиях городской застройки.

**Ключевые слова:** численное моделирование, дифференциальные уравнения, уравнения Навье-Стокса, параллельные вычисления, NVIDIA CUDA.

## Computer Simulation of Air Flows in Urban Areas Using NVIDIA CUDA

A.L. Arkhipov

Ivanovo State Power Engineering University, Ivanovo, Russian Federation  
E-mail: AIArMail999@mail.ru

### Abstract

**Background:** Modelling air movement in urban areas is an important problem, which allows determining the air flows depending on the location of city buildings. Such kind of problems is usually solved by mesh based methods on the multiprocessor supercomputer that allows performing calculations with great efficiency. However, this approach requires expensive computer hardware. In this article the usage of the GPU as a computational device instead of expensive multiprocessor computer is suggested. Nowadays, two main market players of graphic processors, NVIDIA CUDA and AMD, suggest the technologies of GPU usage for general purposes calculations. The more developing technology of general purposes calculations on GPU is NVIDIA CUDA platform.

**Materials and methods:** NVIDIA CUDA is used to calculate on the graphic processors. This choice was made on the basis of API and support of the most widely-spread NVIDIA GeForce video-cards. The finite-difference calculation scheme is adapted for using the graphic processors as a calculation means.

**Results:** In this article a way to solve a problem of the modeling the movement of air in urban areas is proposed. The GPU is used as a computational device instead of expensive multiprocessor computer.

**Conclusions:** The finite-difference scheme and the algorithm based on it allowed to get the velocity growth of calculation due to their implementation on the NVIDIA CUDA platform comparing to the algorithm which works with the central processor. As a result, the outstanding calculation velocity growth is received. The usage of graphic processors as a calculation means and algorithm adaptation allows to decrease the cost of computer equipment which is necessary for calculating, for solving the tasks of the movement modelling of air in urban areas up to 60 - 70 times.

**Key words:** numerical simulation, differential equations, Navier–Stokes equations, parallel computing, NVIDIA CUDA.

<sup>1</sup> Работа выполнена при поддержке Министерства образования и науки РФ ГК№13 G25.31.0077.

Моделирование движения воздуха в условиях городской застройки может помочь найти ответы на вопросы, куда будут распространяться вредные выбросы того или иного промышленного объекта (например, электростанции или химического завода), где будет наибольшая концентрация вредного газа или дыма, исходящего из какого-либо источника техногенного или природного типа. Также такое моделирование может использоваться при решении вопроса о выборе места строительства экологически критичного промышленного объекта (например, АЭС или обычной тепловой электростанции).

Подобное моделирование обычно проводят с использованием параллельных вычислений для ускорения расчетов. В качестве вычислительных устройств применяются многопроцессорные компьютеры и вычислительные кластеры. Нами предлагается использовать в качестве вычислительных устройств графические процессоры (GPU), что, предположительно, позволит получить сравнимую с большими многопроцессорными компьютерами скорость вычислений при меньших материальных затратах.

На сегодняшний день два основных игрока рынка графических процессоров – NVIDIA и AMD – предлагают технологии использования своих GPU для вычислений общего назначения (General-purpose computing on graphics processing units (GPGPU)). Наиболее развитой технологией выполнения вычислений общего назначения на GPU является платформа NVIDIA CUDA [3].

В качестве дифференциальных уравнений, описывающих движение газа, в большинстве случаев используют уравнения Навье-Стокса:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} + \frac{\partial(\rho w)}{\partial z} &= 0; \\ \frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + (\gamma - 1)\rho e_m)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} + \frac{\partial(\rho uw)}{\partial z} - v \left( \frac{\partial^2(\rho u)}{\partial x^2} + \frac{\partial^2(\rho u)}{\partial y^2} + \frac{\partial^2(\rho u)}{\partial z^2} \right) &= Q_x; \\ \frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu)}{\partial x} + \frac{\partial(\rho v^2 + (\gamma - 1)\rho e_m)}{\partial y} + \frac{\partial(\rho vw)}{\partial z} - v \left( \frac{\partial^2(\rho v)}{\partial x^2} + \frac{\partial^2(\rho v)}{\partial y^2} + \frac{\partial^2(\rho v)}{\partial z^2} \right) &= Q_y; \\ \frac{\partial(\rho w)}{\partial t} + \frac{\partial(\rho wu)}{\partial x} + \frac{\partial(\rho wv)}{\partial y} + \frac{\partial(\rho w^2 + (\gamma - 1)\rho e_m)}{\partial z} - v \left( \frac{\partial^2(\rho w)}{\partial x^2} + \frac{\partial^2(\rho w)}{\partial y^2} + \frac{\partial^2(\rho w)}{\partial z^2} \right) &= Q_z; \\ \frac{\partial(\rho E)}{\partial t} + \frac{\partial(u\rho E + u(\gamma - 1)\rho e_m)}{\partial x} + \frac{\partial(v\rho E + v(\gamma - 1)\rho e_m)}{\partial y} + \frac{\partial(w\rho E + w(\gamma - 1)\rho e_m)}{\partial z} &= 0; \\ Q_x = \rho g_x + 2\omega_z \rho v; \quad Q_y = \rho g_y - 2\omega_z \rho u; \quad Q_z = \rho g_z; \\ e_m = E - \frac{u^2 + v^2 + w^2}{2}; \quad R = (\gamma - 1)c_v, \end{aligned}$$

где  $\rho$  – парциальная плотность;  $T$  – температура;  $E$  – удельная энергия;  $e_m$  – внутренняя энергия 1 кг воздуха;  $u, v, w$  – скорости воздуха по  $Ox, Oy, Oz$ ;  $R = 287$  Дж/(кг·К) – индивидуальная газовая постоянная для сухого воздуха;  $\omega_z$  – проекция вектора угловой скорости вращения Земли на вертикаль в данном месте;  $g_x, g_y, g_z$  – проекции ускорения свободного падения на оси  $Ox, Oy, Oz$ :  $g_x = 0, g_y = 0, g_z = -9,81$  м/с<sup>2</sup>;  $\nu$  – кинематическая вязкость воздуха;  $\gamma = 1,4$  – показатель адиабаты;  $c_v = 717$  Дж/(кг·К) – теплоемкость воздуха при постоянном объеме;  $c_p = 1006$  Дж/(кг·К) – теплоемкость воздуха при постоянном давлении.

Для удобства написания разностной схемы расчета следует представить дифференциальные уравнения Навье-Стокса в матричном виде:

$$\begin{aligned} \frac{\partial M}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} + \frac{\partial F_z}{\partial z} - v \left( \frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} \right) &= Q, \\ M &= \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E \end{pmatrix}, \quad F_x = \begin{pmatrix} \rho u \\ \rho u^2 + (\gamma - 1)\rho e_m \\ \rho uv \\ \rho uw \\ u(\rho E + (\gamma - 1)\rho e_m) \end{pmatrix}, \\ F_y &= \begin{pmatrix} \rho v \\ \rho v^2 + (\gamma - 1)\rho e_m \\ \rho vw \\ v(\rho E + (\gamma - 1)\rho e_m) \end{pmatrix}, \quad F_z = \begin{pmatrix} \rho w \\ \rho w^2 + (\gamma - 1)\rho e_m \\ \rho wv \\ w(\rho E + (\gamma - 1)\rho e_m) \end{pmatrix}, \\ V &= \begin{pmatrix} 0 \\ \rho u \\ \rho v \\ \rho w \\ 0 \end{pmatrix}, \quad Q = \begin{pmatrix} 0 \\ \rho g_x + 2\omega_z \rho v \\ \rho g_y - 2\omega_z \rho u \\ \rho g_z \\ 0 \end{pmatrix}, \\ E + (\gamma - 1)e_m = E + RT &= \frac{u^2 + v^2 + w^2}{2} + \gamma e_m = \\ &= \frac{u^2 + v^2 + w^2}{2} + c_p T. \end{aligned}$$

Если в качестве разностной схемы расчета выбрать обычную схему по методу конечных разностей, то придется использовать очень маленький шаг по времени, чтобы вычисления были устойчивыми. Лучше в качестве разностной схемы расчета выбрать такую схему, которая бы позволила использовать больший шаг по времени. Обычно это достигается за счет усложнения вычислений на каждом шаге интегрирования. Нами была выбрана сеточная схема, приведенная в [1]. Эта схема позволяет наиболее точно считать системы дифференциальных уравнений, содержащие в себе уравнения Навье-Стокса.

Схема вычислений использует дополнительные переменные и коэффициенты:  $\tau$  – шаг

по времени;  $n$  – момент времени;  $h$  – шаг сетки по осям  $Ox$ ,  $Oy$ ,  $Oz$ ;  $i$  – индекс по оси  $Ox$ ,  $i = 2, \dots, N_1-1$ ,  $N_1$  – количество точек по оси  $Ox$ ;  $j$  – индекс по оси  $Oy$ ,  $j = 2, \dots, N_2-1$ ,  $N_2$  – количество точек по оси  $Oy$ ;  $k$  – индекс по оси  $Oz$ ,  $k = 2, \dots, N_3-1$ ,  $N_3$  – количество точек по оси  $Oz$ ;  $\theta$  – сглаживающий коэффициент,  $\theta = 1,4$ .

Для получения значения матрицы  $M$  в следующей момент времени выполним вычисления:

$$M_{i,j,k}^{n+1} = M_{i,j,k}^n + \tau \cdot dM_{i,j,k}^n,$$

где  $M_{i,j,k}^n$  – состояние системы в точке сетки  $i,j,k$  в момент времени  $n$ .

Для получения коэффициента  $dM_{i,j,k}^n$  необходимо выполнить следующие вычисления [1]:

$$dM_{i,j,k}^n = \frac{F_x^n_{i-\frac{1}{2},j,k} - F_x^n_{i+\frac{1}{2},j,k}}{2h_x} + \frac{F_y^n_{i,j,k-\frac{1}{2}} - F_y^n_{i,j,k+\frac{1}{2}}}{2h_y} + \frac{F_z^n_{i,j,k-\frac{1}{2}} - F_z^n_{i,j,k+\frac{1}{2}}}{2h_z} + v \cdot dV_{i,j,k}^n + Q_{i,j,k}^n;$$

$$F_x^n_{i+\frac{1}{2},j,k} = F_x \left( M_{i+\frac{1}{2},j,k}^L \right) + F_x \left( M_{i+\frac{1}{2},j,k}^R \right) +$$

$$+ a_{i+\frac{1}{2},j,k} \left( M_{i+\frac{1}{2},j,k}^L - M_{i+\frac{1}{2},j,k}^R \right);$$

$$M_{i+\frac{1}{2},j,k}^L = M_{i,j,k} + \frac{(M_x)_{i,j,k}}{2};$$

$$M_{i+\frac{1}{2},j,k}^R = M_{i+1,j,k} - \frac{(M_x)_{i+1,j,k}}{2};$$

$$M_{i-\frac{1}{2},j,k}^L = M_{i-1+\frac{1}{2},j,k}^L = M_{i-1,j,k} + \frac{(M_x)_{i-1,j,k}}{2};$$

$$M_{i-\frac{1}{2},j,k}^R = M_{i-1+\frac{1}{2},j,k}^R = M_{i,j,k} - \frac{(M_x)_{i,j,k}}{2};$$

$$(M_x)_{i,j,k} = \min \text{mod} \left( \theta (M_{i+1,j,k} - M_{i,j,k}), \right.$$

$$\left. \frac{M_{i+1,j,k} - M_{i-1,j,k}}{2}, \theta (M_{i,j,k} - M_{i-1,j,k}) \right);$$

$$\min \text{mod}(\varphi_1, \varphi_2, \dots) = \begin{cases} \min_i \varphi_i, & \text{если } \varphi_i > 0 \forall i, \\ \max_i \varphi_i, & \text{если } \varphi_i < 0 \forall i, \\ 0, & \text{в противном случае;} \end{cases}$$

$$a_{i+\frac{1}{2},j,k} = \max \left( r \left( \frac{\partial F_x \left( M_{i+\frac{1}{2},j,k}^L \right)}{\partial M} \right), r \left( \frac{\partial F_x \left( M_{i+\frac{1}{2},j,k}^R \right)}{\partial M} \right) \right),$$

где  $r$  – радиус распространения возмущений.

Для расчета подобных аэродинамических моделей максимальная скорость распространения возмущений  $a$  в точке пространства определяется следующим образом:

$$a_{i+\frac{1}{2},j,k} = \max \left( \left| u_{i+\frac{1}{2},j,k}^L \right| + c_{i+\frac{1}{2},j,k}^L, \left| u_{i+\frac{1}{2},j,k}^R \right| + c_{i+\frac{1}{2},j,k}^R \right),$$

$$a_{i,j+\frac{1}{2},k} = \max \left( \left| v_{i,j+\frac{1}{2},k}^L \right| + c_{i,j+\frac{1}{2},k}^L, \left| v_{i,j+\frac{1}{2},k}^R \right| + c_{i,j+\frac{1}{2},k}^R \right),$$

$$a_{i,j,k+\frac{1}{2}} = \max \left( \left| w_{i,j,k+\frac{1}{2}}^L \right| + c_{i,j,k+\frac{1}{2}}^L, \left| w_{i,j,k+\frac{1}{2}}^R \right| + c_{i,j,k+\frac{1}{2}}^R \right),$$

где  $c$  – скорость звука в данной точке пространства:

$$c_{i,j,k} = \sqrt{\gamma(\gamma-1)e_m}.$$

Для расчета влияния вязкости можно применить явный метод конечных разностей:

$$dV_{i,j,k}^n = \frac{V_{i-1,j,k}^n - 2V_{i,j,k}^n + V_{i+1,j,k}^n}{h_x^2} + \frac{V_{i,j,k-1}^n - 2V_{i,j,k}^n + V_{i,j,k+1}^n}{h_y^2} + \frac{V_{i,j,k-1}^n - 2V_{i,j,k}^n + V_{i,j,k+1}^n}{h_z^2}.$$

Значения скоростей  $u$ ,  $v$ ,  $w$ , температура и давление в точке вычисляются по формулам:

$$u = \frac{\rho u}{\rho}, \quad v = \frac{\rho v}{\rho}, \quad w = \frac{\rho w}{\rho},$$

$$e_m = \frac{\rho E}{\rho} = \frac{(\rho u)^2 + (\rho v)^2 + (\rho w)^2}{2\rho^2}, \quad T = \frac{e_m}{c_v},$$

$$\rho = (\gamma-1) \left( \rho E - \frac{(\rho u)^2 + (\rho v)^2 + (\rho w)^2}{2\rho} \right).$$

После получения значений всех переменных на текущем шаге по времени можно переходить к вычислению следующего шага.

Условие устойчивости схемы:

$$\frac{\tau}{\min(h_x, h_y, h_z)} \max_{i,j,k} \left( \sqrt{u_{i,j,k}^2 + v_{i,j,k}^2 + w_{i,j,k}^2} + c_{i,j,k} \right) < \frac{1}{8}.$$

**Реализация.** Нами предлагается использовать вычисления на графическом процессоре (GPU) для реализации алгоритма расчета вышеописанной разностной схемы.

GPU представляет собой потоковый процессор, состоящий из большого числа однотипных ядер. Количество ядер достигает нескольких сотен, что много больше, чем количество ядер CPU (центрального процессора). Использование такой мультиядерной архитектуры позволяет достичь существенного возрастания скорости вычислений. Однако у GPU есть и ограничения: потоковый процессор обеспечивает только параллелизм данных, но не параллелизм задач. То есть можно выполнять одну и ту же операцию над большим количеством данных одновременно, но нельзя запустить одновременно несколько разных задач. При разработке алгоритмов для выполнения на GPU нужно учитывать эту особенность.

Рассмотренная выше вычислительная схема позволяет создать алгоритм с использованием параллелизма данных: в каждой точке сетки выполняются одинаковые вычисления для нахождения значений переменных в сле-

дующий момент времени. Таким образом, данная вычислительная схема может быть использована для выполнения расчетов на GPU.

Синхронизация вычислений для каждого момента времени была организована следующим образом: после вычисления значений всех переменных на GPU управление передавалось CPU, результаты вычисления обрабатывались, после чего происходил переход к следующей итерации алгоритма. Без такой синхронизации вычисления потеряли бы устойчивость. Понятно, что такая синхронизация, как и вообще сама организация вычислений на GPU, требует определенных издержек – часть вычислительного времени тратится на передачу данных от CPU к GPU и обратно, а также на обслуживание процессов, связанных с самими вычислениями на GPU. Поэтому прирост скорости вычислений относительно CPU заметен тем больше, чем большего размера сетка обрабатывается.

Каждый объект, находящийся в расчетной области, препятствует движению воздуха. Для учета этого на его гранях задаются соответствующие граничные условия. В качестве таких граничных условий был применен следующий прием: внутри объектов скорость воздуха обнуляется.

Вычисления расчетной схемы на CUDA были организованы таким образом, что каждое ядро GPU выполняло расчет состояния одной ячейки расчетной области. В него передавались переменные  $\rho$ ,  $\rho u$ ,  $\rho v$ ,  $\rho w$ ,  $\rho E$ . Для уменьшения задержек, связанных с обращением к глобальной памяти видеокарты, данные ячеек, необходимых для выполнения вычисления, кэшировались в регистрах. Шаг по времени  $\tau$ , показатель адиабаты  $\gamma$  и кинематическая вязкость воздуха  $\nu$  были заданы константами. На каждом шаге сначала вычислялись новые значения переменных, а затем накладывались граничные условия: в ячейках, примыкающих к границам расчетной области, и ячейках, принадлежащих объектам-препятствиям, значения переменных соответствующим образом корректировались. На данном этапе, чтобы излишне не усложнять алгоритм, границы объектов задавались непосредственно в коде программы.

Исходный код программы был составлен таким образом, что его можно было скомпилировать как для расчетов с использованием CUDA, так и для расчетов с использованием CPU. В варианте расчетов на CPU вычисления производились в цикле по всем ячейкам в один поток. Такая организация исходного кода позволила выполнить сравнение времени выполнения этих двух вариантов программы и оценить полученное ускорение.

**Выводы и результаты.** Были проведены вычисления 100000 шагов по времени на сетке разной размерности. Шаг сетки составлял 1 м, шаг по времени – 0,0001 с. Время работы программы в секундах занесено в таблицу:

Размер сетки	GPU	CPU
50x50x50	180	10000
64x64x64	344	20900
100x100x100	1331	70900

Описанная вычислительная схема в варианте для GPU обчисывалась в среднем в 60 раз быстрее однопоточкового варианта на CPU. Соответственно, чтобы получить такую же скорость вычислений на многопроцессорном кластере, он должен содержать не менее 60 процессорных ядер. Стоимость компьютера с видеокартой NVIDIA GeForce будет в 60–70 раз меньше стоимости такого многопроцессорного кластера. Однако производимые вычисления потребовали для себя существенного объема памяти видеокарты, поэтому размер расчетной области был ограничен – максимальный размер куба составил 100×100×100 ячеек. В расчетной области были заданы два препятствия в виде параллелепипедов: один высотой 50 ячеек (метров), другой – 32 ячейки (метра). Внутри них скорость воздуха обнулялась. На левой грани куба расчетной области была задана начальная скорость входящего потока воздуха по осям  $(x, y, z) = (5, 5, 0)$  м/с. Ось  $x$  направлена слева направо, ось  $y$  – сверху вниз. Программа позволяет просматривать срезы расчетной области по оси  $z$  (рис. 1, 2).

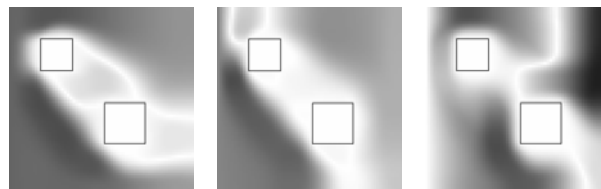


Рис. 1. Скорость воздуха по осям  $x$ ,  $y$  на уровне  $z = 20$

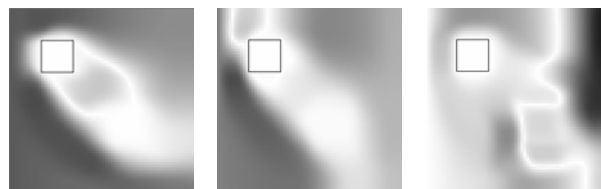


Рис. 2. Скорость воздуха по осям  $x$ ,  $y$  на уровне  $z = 33$

По осям  $x$  и  $y$  максимальная величина скорости составила 5,1, по оси  $z$  – 1,2.

Дальнейшее развитие предполагает введение дополнительной субстанции в состав воздуха (например, дыма), картину распространения которой надо будет определить. Также планируется задавать форму и расположение объектов в расчетной области динамически, а не непосредственно в коде программы, как это было сделано в данном слу-

чае. Также планируется изменить отображение результата вычислений: вместо горизонтального среза расчетной области показывать трехмерный вид расчетной области. Кроме того, планируется увеличить размерность сетки, реорганизовав код таким образом, чтобы сократить требуемый для вычислений объем памяти.

#### Список литературы

1. **Kurganov A., Tadmor E.** New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations // *J. Comput. Phys.* – 160 (2000). – P. 241–282.
2. **Kurganov A., Levy D.** A third-order semi-discrete central scheme for conservation laws and convection–diffusion equations // *SIAM J. Sci. Comput.* – 22 (2000). – P. 1461–1488.

3. **NVIDIA CUDA** — неграфические вычисления на графических процессорах [Электронный ресурс]. – Режим доступа: <http://www.ixbt.com/video3/cuda-1.shtml>
4. **Википедия: CUDA** [Электронный ресурс]. – Режим доступа: <http://ru.wikipedia.org/wiki/CUDA>

#### References

1. Kurganov, A., Tadmor, E. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *J. Comput. Phys.*, 160 (2000), pp. 241–282.
2. Kurganov, A., Levy, D. A third-order semi-discrete central scheme for conservation laws and convection–diffusion equations // *SIAM J. Sci. Comput.*, 22 (2000), pp. 1461–1488.
3. *NVIDIA CUDA – nefraficheskie vychisleniya na graficheskikh protsessorakh* [General-purpose computing on graphics processing units]. Available at: <http://www.ixbt.com/video3/cuda-1.shtml>
4. *Vikipediya: CUDA* [Electronic Resource]. Available at: <http://en.wikipedia.org/wiki/CUDA>

*Архипов Ален Леонидович*,  
ФГБОУВПО «Ивановский государственный энергетический университет имени В.И. Ленина»,  
программист ИВЦ, аспирант кафедры высокопроизводительных вычислительных систем,  
e-mail: AIArMail999@mail.ru