

Barochkin Evgeny Vitalyevich,  
Ivanovo State Power Engineering University,  
Doctor of Engineering Sciences (Post-Doctoral degree), Professor, Head of the Heat Power Plants Department,  
address: Ivanovo, Rabfakovskaya St., 34, Building «V» (B), Room 417,  
telephone (4932) 26-99-13,  
e-mail: admin@tes.ispu.ru

*Ледуховский Григорий Васильевич,*  
ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина»,  
кандидат технических наук, доцент кафедры тепловых электрических станций,  
адрес: г. Иваново, ул. Рабфаковская, д. 34, кор. В, ауд. 408,  
телефон (4932) 26-99-31,  
e-mail: lgv83@yandex.ru

*Ledukhovsky Grigory Vasilievich,*  
Ivanovo State Power Engineering University,  
Candidate of Engineering Sciences (PhD), Associate Professor of the Heat Power Plants Department,  
address: Ivanovo, Rabfakovskays St, 34, Building B (V), Room 408,  
telephone (4932) 26-99-31,  
e-mail: lgv83@yandex.ru

УДК 004.942

## Методы ускорения расчетов и повышения устойчивости полевой модели гидродинамических турбулентных процессов на CUDA

А.А. Гудухина, И.Ф. Ясинский  
ФГБОУВО «Ивановский государственный энергетический университет имени В.И.Ленина»,  
г. Иваново, Российская Федерация  
E-mail: igor.yasinskiy@gmail.com

### Авторское резюме

**Состояние вопроса.** Одной из главных проблем при построении математических моделей турбулентных гидродинамических сред остается высокая вычислительная сложность компьютерных расчетов, поскольку требуется получить решение нестационарной задачи с сеткой такого пространственного шага, чтобы он соответствовал размерам самых малых вихревых структур. Существуют рекомендации по использованию разностных вычислительных схем для турбулентной динамики жидкости, полученные при моделировании розлива нефтепродуктов по водной поверхности. В связи с этим актуальным является получение устойчивого расчета математической модели динамики вязкой несжимаемой жидкости согласно описанию Эйлера с учетом влияния турбуликации и ускорение расчетов в параллельном интерфейсе CUDA.

**Материалы и методы.** При проведении вычислительных экспериментов применяются методы математического моделирования физических объектов, метод численного интегрирования дифференциальных уравнений, метод противоточных производных для повышения устойчивости разностных схем. Для оценки турбулентной вязкости сплошной среды используется метод Секундова.

**Результаты.** Создана устойчивая параллельная реализация математической модели, описывающей процессы в сплошной среде с учетом влияния турбулизирующих гидродинамических структур, устойчивость которой достигается за счет применения метода противоточных производных и замены традиционных разностных схем при расчете полей скорости и давления на четырехточечные аналоги. Предложен метод равномерного распределения вычислительных ресурсов графического ускорителя при больших размерах сетки. Получен устойчивый расчет модели динамики сплошной среды с вычислением турбулентной вязкости по модели Секундова на значительном временном отрезке.

**Выводы.** Построенное решение на основе интерфейса CUDA позволяет достичь ускорения вычислений от 2 до 8 раз в зависимости от возможностей аппаратной части. Разработанная система может являться инструментом исследования объектов энергетической отрасли, для которых требуется ускорение принятия управляющих решений по сравнению со стандартным программным обеспечением, не предусматривающим распараллеливания расчетов. Достоверность результатов вызвана соответствием системных условий конфигурациям двухфазных систем питания энергетических установок.

**Ключевые слова:** сплошные среды, гидрогазодинамика, турбулентность, параллельные вычисления, параллельный интерфейс CUDA

## Calculations acceleration and stability improvement methods for hydrodynamic turbulent processes model on CUDA

A.A. Gudukhina, I.F. Yasinskiy  
Ivanovo State Power Engineering University, Ivanovo, Russian Federation  
E-mail: igor.yasinskiy@gmail.com

### Abstract

**Background.** High computational complexity of computer calculations remains one of the main problems in the construction of mathematical models of turbulent hydro and gas dynamic environments, since it is necessary to obtain a solution to a non-stationary problem with a grid of such a spatial step, which corresponds to the size of the smallest vortex structures. The authors took into account the recommendations on the use of differential computational schemes for the turbulent dynamics of a liquid, obtained in the work on modeling the bottling of oil products on the water surface. The purpose of this work is to obtain the mathematical model stable calculation of the viscous incompressible fluid dynamics according to the Euler description, taking into account the influence of turbulence, and the acceleration of calculations in a parallel CUDA interface.

**Materials and methods.** The method of mathematical modeling of physical objects and the method of differential equations numerical integration are used. The method of countercurrent derivatives is used to increase the stability of differential schemes. The Sekundov's method of turbulent viscosity estimation of a continuous medium is used, as well as the analysis of the computational experiments results.

**Results.** A stable parallel realization has been created for numerical simulation of processes in a continuous medium. It takes into account the influence of turbulizing hydro- and gas-dynamic structures. Stability is achieved by using the counter-current derivative method and replacing traditional difference schemes, when calculating the velocity and pressure fields, with four-point counterparts. The method of uniform distribution of the graphics accelerator computing resources with large sizes of the computational grid is proposed. A stable calculation created of a mathematical model of continuous medium dynamics with the Sekundov's turbulent viscosity estimation at a considerable time interval.

**Conclusion.** Based on the NVidia CUDA interface, this solution allows acceleration of calculations to be achieved from 2 to 8 times, depending on the capabilities of the hardware configuration. The developed system can be a tool for the study of objects in the energy industry, for which acceleration of control decisions is required, compared with standard software, that does not provide for calculations parallelization. Reliability of the results is caused by the compliance of the system conditions with the configuration of two-phase supply systems of power installations.

**Key words:** continuous media, hydro-gas dynamics, turbulence, parallel computing, CUDA parallel interface

**DOI:** 10.17588/2072-2672.2018.6.072-080

**Введение.** Для современной науки представляют интерес природные и технологические процессы механики жидкости и газа, осложненные вихревыми турбулентными течениями. Турбулизация приводит к масштабному воздействию на все характеристики поля течения. Особые свойства таких течений находят применение в энергетическом, теплообменном, технологическом оборудовании ядерной энергетики, аэрокосмической техники, нефтеперерабатывающей промышленности, промышленной теплоэнергетики [1]. Большие перспективы имеет использование вихревого закрученного потока для комплексного решения задач повышения безопасности и эффективности работы ядерно-энергетических установок, конструктивного усовершенствования МГД-генераторов, ускорения протекания химико-технологических процессов, таких как обжиг, плавление, коксование, пиролиз, интенсификации процессов тепло- и массообмена в условиях вынужденной и естественной конвекции [2]. Особый интерес вызывает моделирование осложненной динамики жидкости в контексте проектирования аппаратов для производства высокодисперсных сред [3].

Моделирование вихревой турбулентной динамики является проблемой, которая до сих

пор решена не полностью [4]. При математическом моделировании механики сплошной среды используются следующие способы описания турбулентных режимов:

- уравнения Рейнольдса, в которых мгновенные значения скорости представлены в виде суммы осредненной по времени и пульсационной составляющих;
- модели, описывающие турбулентную вязкость в зависимости от поля скоростей, расстояния от твердых поверхностей (модель Секундова);
- параметрические модели турбулентности, в которых выполняется расчет кинетической энергии пульсаций и энергии диссипации (К-Е модель).

Рейнольдсом установлено, что переход от ламинарного течения к турбулентному происходит при достижении безразмерным комплексом  $Re$ , названным впоследствии числом Рейнольдса, некоторого предельного значения  $Re_{кр}$ . Число  $Re$  включает в себя среднерасходную скорость среды  $u$ , характерный размер  $d$  и кинематический коэффициент вязкости потока  $\nu$  [5]. Сначала с увеличением значения числа  $Re$ , отражающего склонность сплошной среды к возникновению турбулентных пульсаций в данных условиях, наблюдается смена режима

течения от ламинарного к переходному. Для последнего характерна перемежающаяся форма движения, когда в данной точке чередуются ламинарная и турбулентная динамика. По мере дальнейшего роста числа  $Re$ , когда силы инерции начинают в достаточной степени превосходить силы трения, появляется выраженное хаотическое движение пульсирующих молей среды, накладывающееся на основное движение потока. Эти пульсации скорости являются источником пульсации других параметров среды – температуры, давления, плотности, концентрации.

Обычно с возрастанием  $Re$  типичные размеры вихревых образований в потоке становятся меньше. В этом случае требуется получить вычислительное решение нестационарной задачи с расчетной сеткой такого пространственного шага, чтобы он соответствовал размерам самых малых вихревых структур. Поскольку в моделируемой среде могут одновременно находиться турбулентные фрагменты разных размеров, то для их оценки применяют различные группы масштабов. Малые вихревые образования характеризуются микромасштабом и масштабом диссипирующих вихрей. На основании изложенного можно утверждать, что математическое моделирование гидрогазодинамики относится к вычислительно сложным задачам. При этом распараллеливание программ на многопроцессорных суперкомпьютерах естественным образом оказывается необходимым инструментом для их решения.

Эволюция идей гидродинамики на суперкомпьютерах имеет долгую историю. В качестве заметных итераций можно отметить гибридную параллельную реализацию сеточных уравнений Больцмана в [6]. В скором времени после появления системы CUDA, дающей возможность реализации задач общего назначения на мультимасштабных видеоускорителях Nvidia, эта система приобрела популярность в области моделирования задач механики сплошной среды. В 2010 году вышла работа Д.Е. Демидова и соавторов [7], в которой обозначены подходы к получению ускорения различных моделей гидродинамических процессов на интерфейсе CUDA.

Известны результаты, изложенные в публикации D. Nishiura и соавторов, 2015 [8], где рассматривается распараллеливание не сеточных моделей гидродинамики в парадигме Лагранжа на гетерогенных суперкомпьютерах с общей памятью, включая системы OpenMP и GPGPU. Решение подобных задач уже на массивно-параллельном суперкомпьютере Archer осветили в 2018 году X. Guo и соавторов в [9].

Ниже рассматривается метод ускорения гидродинамической задачи о течении вязкой несжимаемой жидкости с турбулентными процессами на базе платформы CUDA.

**Математическое описание задачи о вихревом турбулентном движении вязкой несжимаемой жидкости.** Как было отмечено, с ростом числа Рейнольдса размеры вихревых образований в потоке уменьшаются. В этом случае расчетная сетка должна соответствовать размерам самых малых турбулентных структур, что значительно повысит затраты машинного времени на расчет модели. Для оценки возможности ускорения работы была взята область с такими начальными и граничными условиями, которые способствуют образованию турбулентных структур (рис. 1). В ней имеется два боковых впускных отверстия  $U_{вх1}$ ,  $U_{вх2}$  и одно донное выходное  $U_{вых}$ . Геометрия источников и стоков хорошо соответствует конфигурации двухфазных систем питания энергетических установок.

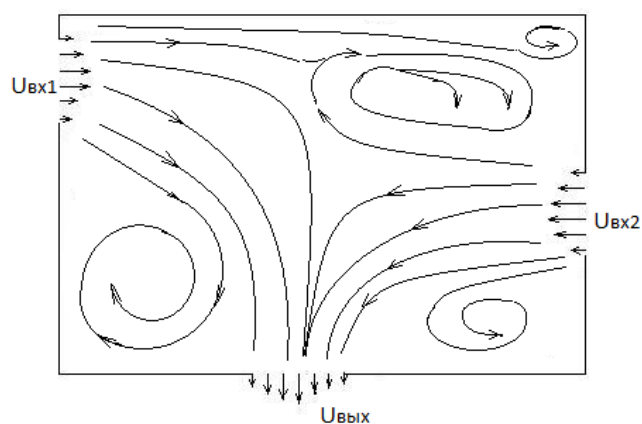


Рис. 1. Конфигурация области моделирования динамики сплошной среды

Для описания как ламинарных, так и турбулентных течений применяется система уравнений Рейнольдса [4]:

$$\frac{\partial U_i}{\partial t} + \sum_{j=1}^3 U_j \frac{\partial U_i}{\partial x_j} = \sum_{j=1}^3 \frac{\partial S_{ij}}{\partial x_j} - \frac{1}{\rho} \frac{\partial P}{\partial x_i}; \quad i = 1, 2, 3; \quad (1)$$

$$\sum_{i=1}^3 \frac{\partial U_i}{\partial x_i} = 0; \quad (2)$$

$$S_{ij} = \nu_{eff} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right). \quad (3)$$

где  $U_i$  – составляющие вектора скорости по осям  $x, y, z$ ;  $\nu_{eff}$  – эффективная кинематическая вязкость.

Кинематическая вязкость вычисляется суммой молекулярной вязкости  $\nu_{mol}$  и турбулентной вязкости  $\nu_{turb}$ . В случае выраженного ламинарного течения с учетом постоянства турбулентной вязкости уравнения (1)–(3) упрощаются до формы Навье-Стокса:

$$\frac{\partial U_i}{\partial t} + \sum_{j=1}^3 U_j \frac{\partial U_i}{\partial x_j} = \nu_{eff} \sum_{j=1}^3 \frac{\partial^2 U_i}{\partial x_j^2} - \frac{1}{\rho} \frac{\partial P}{\partial x_i}; \quad i = 1, 2, 3. \quad (4)$$

Для расчета турбулентной вязкости в данном случае была взята модель А.Н. Секундова [10]:

$$\frac{\partial v_{turb}}{\partial t} + \sum_{j=1}^3 U_j \frac{\partial v_{turb}}{\partial x_j} = \sum_{j=1}^3 \frac{\partial}{\partial x_j} \left( (v_{mol} + \kappa v_{turb}) \frac{\partial v_{turb}}{\partial x_j} \right) + v_{turb} f \left( \frac{v_{turb}}{8v_{mol}} \right) D - \gamma L_{min}^{-2} (v_{mol} + \beta v_{turb}) v_{turb} \quad (5)$$

где  $f()$  – функция:

$$f(z) = 0,2 \frac{z^2 + 1,47 \cdot z + 0,2}{z^2 - 1,47 \cdot z + 1} \quad (6)$$

$\kappa$ ,  $\beta$ ,  $\gamma$  – константы, подобранные эмпирическим путем ( $\kappa = 2$ ;  $\beta = 0,06$ ;  $\gamma = 50$ );  $D$  – величина, описывающая силу деформации среды. Ее получение подразумевает использование формы

$$D = \left( \sum_{i,j} \frac{\partial U_i}{\partial x_j} \left( \frac{\partial U_i}{\partial x_j} + \frac{\partial U_j}{\partial x_i} \right) \right)^{1/2} \quad (7)$$

По модели Кармана длина пути перемешивания  $L$  вычисляется согласно выражению

$$L = \kappa \left| \frac{\partial U}{\partial y} / \frac{\partial^2 U}{\partial y^2} \right| \quad (8)$$

Расчеты учитывают наличие неподвижных поверхностей, для которых значение турбулентной вязкости всегда равно нулю, т.е.  $v_{turb}|_r = 0$ . Для формирования поля турбулентности необходимо установить начальные значения. Данные могут быть получены эмпирическим путем или с использованием модели Прандтля–Кармана.

Модель А.Н. Секундова (5)–(6) показывает достаточно точные результаты, удобна для компьютерного моделирования и применима для описания турбулентных течений в несжимаемых средах. Эту модель удобно сочетать с уравнениями Навье–Стокса, проинтегрировать дифференциальные уравнения (5) можно аналогичным образом (явная схема, метод прогонки и др.).

**Распараллеливание задачи.** Распараллеливание по точкам системы хорошо ложится на сетку процессов параллельного интерфейса CUDA [11]. Каждый процесс может просчитывать одну точку или группу точек, поставленную ему в соответствии (рис. 2).

Алгоритм работы модели, реализованный в виде главной функции  $main()$ , представлен на рис. 3.

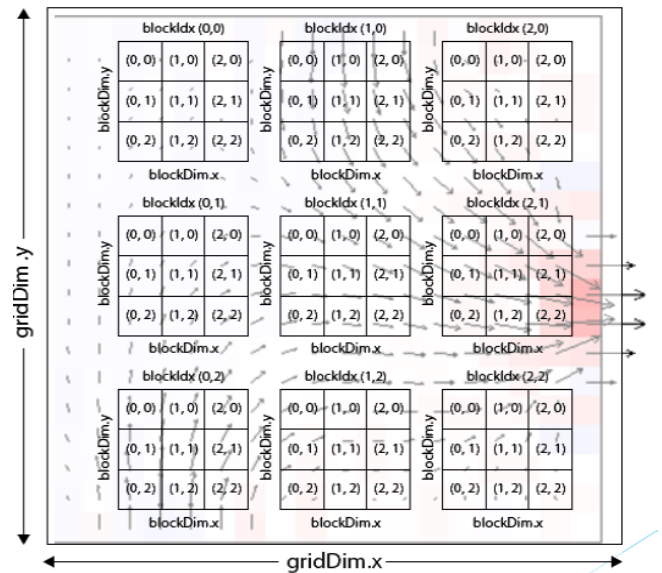


Рис. 2. Наложение сетки CUDA на сетку узлов полевой модели

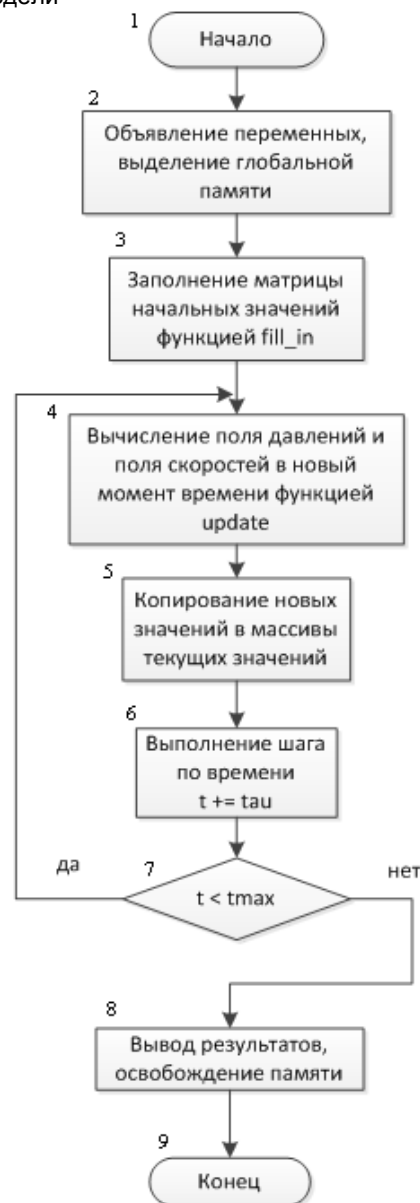


Рис. 3. Алгоритм главной функции

**Блок 1.** Начало алгоритма.

**Блок 2.** Сначала создаются одномерный массивы в памяти CPU для удобства копирования значений в глобальную память. Массивы необходимы для хранения информации об актуальном состоянии моделируемой сплошной среды, а также для хранения значений предыдущих шагов для вычисления новых. Поле давлений сохраняется при помощи единственного массива:

```
Ux = newfloat[Nx * Ny];
Ux_old = newfloat[Nx * Ny];
Uy = newfloat[Nx * Ny];
Uy_old = newfloat[Nx * Ny];
P = newfloat[Nx * Ny];
```

Соответствующие им массивы создаются в глобальной памяти видеокарты:

```
cudaMalloc((void**)&UxDev, size);
cudaMalloc((void**)&Ux_oldDev, size);
cudaMalloc((void**)&UyDev, size);
cudaMalloc((void**)&Uy_oldDev, size);
cudaMalloc((void**)&P_Dev, size);
```

**Блок 3.** Массивы рассчитываемых параметров заполняются начальными значениями с помощью функции `fill_in()`, которая выполняется на видеокarte и использует параллелизм:

```
fill_in<<<dim3((Nx / block_size) + 1, (Ny /
block_size) + 1), dim3(block_size, block_size)
>>> (UxDev, UyDev, P_Dev);
```

**Блоки 4–6.** В этом блоке начинается цикл по времени, в рамках которого вызывается функция `update()`, рассчитывающая обновленные значения полей скорости и давления сплошной среды, которые затем копируются в массив текущих значений и выполняется переход к следующему шагу по времени:

```
do{
    update <<< dim3((Nx / block_size) + 1,
(Ny / block_size) + 1), dim3(block_size,
block_size) >>> (UxDev, Ux_oldDev, UyDev,
Uy_oldDev, P_Dev);
    cudaThreadSynchronize();
    cudaMemcpy(Ux_oldDev, UxDev, size,
cudaMemcpyDeviceToDevice);
    cudaMemcpy(Uy_oldDev, UyDev, size,
cudaMemcpyDeviceToDevice);
    t += tau;
} while (t < tmax);
```

**Блоки 7, 8.** Цикл заканчивает работу по достижении максимального времени моделирования. После этого выводятся результаты расчетов и освобождается память, выделенная под массивы.

Параллелизм данной реализации заключается в функциях `fill_in()` и `update()`.

В общем случае нить с координатами  $i, j$  вычисляет параметры точки целевой системы с координатами  $i$  и  $j$  или если точек в системе больше, чем доступных нитей, то любую точку, которая будет поставлена в соответствие данной нити. В CUDA координаты нити находятся следующим образом [11]:

```
inti = blockIdx.x*blockDim.x + threadIdx.x;
int j = blockIdx.y*blockDim.y + threadIdx.y;
```

где `blockIdx.x` и `blockIdx.y` – координаты блока нитей по осям  $x$  и  $y$  соответственно; `blockDim.x` и `blockDim.y` – размерности блока нитей по осям  $x$  и  $y$  соответственно; `threadIdx.x` и `threadIdx.y` – координаты нитей в блоке по осям  $x$  и  $y$  соответственно.

Для равномерного распределения большого количества узлов расчетной сетки среди меньшего количества нитей предлагается определять номер нити для расчета данного узла через операцию остатка от деления номера данного узла на общее число доступных нитей:

$$N = N_c \bmod N_t, \quad (9)$$

где  $N$  – номер нити, которая будет обчислять данный узел;  $N_t$  – общее число доступных нитей;  $N_c$  – номер узла, рассчитываемый по формуле

$$N_c = i \cdot N_x + j, \quad (10)$$

где  $N_x$  – размер сетки по оси  $X$ ;  $i, j$  – координаты ячейки в сетке.

*Параллелизм инициализации начальными значениями (функция `fill_in()`).* Данная функция заполняет массивы параметров начальными значениями. Каждой нити ставится в соответствие узел, значения параметров которого она рассчитывает. Если узел попадает на входные или выходные отверстия системы, то начальные значения вычисляются по граничным условиям. В противном случае значения заполняются в соответствии с принятыми начальными условиями (в рассматриваемом случае – нулевыми). Алгоритм выполнения функции представлен на рис. 4.

Начальные скорости и давления в рассматриваемом случае приняты нулевыми:

```
Ux[j*Nx + i] = 0.0;
Uy[j*Nx + i] = 0.0;
P[j*Nx + i] = p0;
```

Далее каждая нить вычисляет координаты нижних входных и выходных отверстий. Если эти координаты поставлены в соответствие нити, то они вычисляют скорости в данной точке:

```
if ((i >= i1) && (i < i2) && (j == Ny - 1))
    Uy[j * Nx + i] = F(x);
    Ux[j * Nx + i] = F(x);
```

Здесь  $F(x)$  – функция, по которой вычисляется скорость движения жидкости.



Рис. 4. Блок-схема функции fill-in

Параллелизм расчета новых значений параметров (функция update()). Алгоритм функции представлен на рис. 5. Пересчитываются скорости и давление каждого отдельного узла. Все нити, кроме тех, что находятся на границах, пересчитывают давление по методу слабой сжимаемости [10]:

$$-c^2 \cdot \text{div} \vec{U} = \frac{\partial P}{\partial t}, \quad (11)$$

где  $c$  – константа, влияющая на характеристики жидкости в системе.

Эта форма вычисляется следующим образом:

```

if ((i < Nx - 1) && (j < Ny - 1) && (i > 0) && (j > 0))
{
    divU = f1(i, j, Ux, Uy);
    P[j * Nx + i] -= c * tau * divU;
}
  
```

В целях повышения устойчивости дивергенция скорости  $D$  берется по схеме

$$D_{i,j}^{k+1} = \frac{U_{xi+1,j+1}^k + U_{xi+1,j-1}^k - U_{xi-1,j+1}^k - U_{xi-1,j-1}^k}{4h} + \frac{U_{yi+1,j+1}^k + U_{yi-1,j+1}^k - U_{yi+1,j-1}^k - U_{yi-1,j-1}^k}{4h}. \quad (12)$$

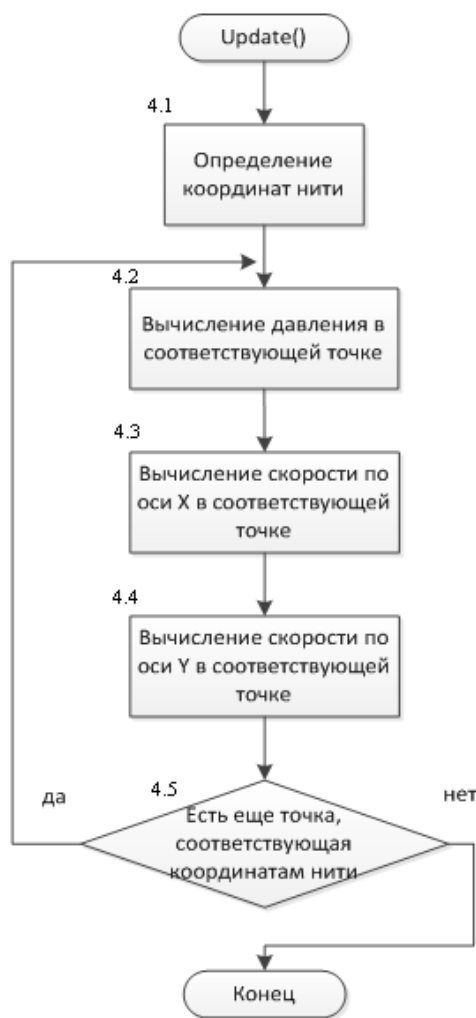


Рис. 5. Алгоритм функции update()

Ее реализует функция f1(), которая выглядит следующим образом:

```

__device__ float f1(int i, int j, float* Ux, float* Uy)
{
    return ((Ux[(j + 1) * Nx + i + 1] + Ux[(j - 1) * Nx + i + 1]) - (Ux[(j - 1) * Nx + i - 1] + Ux[(j + 1) * Nx + i - 1]) + (Uy[(j + 1) * Nx + i - 1] + Uy[(j - 1) * Nx + i + 1]) - (Uy[(j - 1) * Nx + i - 1] + Uy[(j + 1) * Nx + i + 1])) / 4.0 / h;
}
  
```

В функцию передаются координаты  $i$  и  $j$  точки, так что при вызове функции на каждой нити просчитывается отдельно взятый узел системы. Функция будет вызываться в ядре, а выполняться на видеокарте, так как перед ней стоит спецификатор `__device__`.

«Граничные» нити находят значения давления копированием значений из соседних клеток.

Для поля давления на всех границах использованы граничные условия II рода, т. е. задается производная давления (плотности) по нормали к границе. В целях упрощения производная давления взята равной нулю на всех

твердых границах. На входных и выходных границах принято граничное условие, при котором вторая производная равна нулю.

Проекции вектора скорости среды рассчитываются согласно выражениям:

$$U_{xi,j}^{k+1} = U_{xi,j}^k + \tau \left( -\frac{U_{xi,j}^k + |U_{xi,j}^k|}{2} \frac{U_{xi,j}^k - U_{xi-1,j}^k}{h} - \frac{U_{xi,j}^k - |U_{xi,j}^k|}{2} \frac{U_{xi+1,j}^k - U_{xi,j}^k}{h} - \frac{U_{yi,j}^k + |U_{yi,j}^k|}{2} \frac{U_{xi,j}^k - U_{xi,j-1}^k}{h} - \frac{U_{yi,j}^k - |U_{yi,j}^k|}{2} \frac{U_{xi,j+1}^k - U_{xi,j}^k}{h} + v \frac{U_{xi+1,j}^k + U_{xi-1,j}^k + U_{xi,j+1}^k + U_{xi,j-1}^k - 4U_{xi,j}^k}{h^2} - \frac{1}{\rho} \frac{P_{i+1,j+1}^{k+1} + P_{i-1,j+1}^{k+1} - P_{i+1,j-1}^{k+1} - P_{i-1,j-1}^{k+1}}{4h} + F_{xi,j}^{k+1} \right) \quad (13)$$

$$U_{yi,j}^{k+1} = U_{yi,j}^k + \tau \left( -\frac{U_{xi,j}^k + |U_{xi,j}^k|}{2} \frac{U_{yi,j}^k - U_{yi-1,j}^k}{h} - \frac{U_{xi,j}^k - |U_{xi,j}^k|}{2} \frac{U_{yi+1,j}^k - U_{yi,j}^k}{h} - \frac{U_{yi,j}^k + |U_{yi,j}^k|}{2} \frac{U_{yi,j}^k - U_{yi,j-1}^k}{h} - \frac{U_{yi,j}^k - |U_{yi,j}^k|}{2} \frac{U_{yi,j+1}^k - U_{yi,j}^k}{h} + v \frac{U_{yi+1,j}^k + U_{yi-1,j}^k + U_{yi,j+1}^k + U_{yi,j-1}^k - 4U_{yi,j}^k}{h^2} - \frac{1}{\rho} \frac{P_{i+1,j+1}^{k+1} + P_{i-1,j+1}^{k+1} - P_{i+1,j-1}^{k+1} - P_{i-1,j-1}^{k+1}}{4h} + F_{yi,j}^{k+1} \right) \quad (14)$$

Сначала для всех точек, кроме граничных, вычисляется новое значение скорости по оси X с помощью функции f2():

```
if ((i < Nx - 1) && (j < Ny - 1) && (i > 0) && (j > 0))
{
    Ux[j * Nx + i] = f2(i, j, Ux_old, Uy_old, P);
}
```

Сама функция f2() при этом выполняется на девайсе и имеет вид

```
__device__ float f2(inti, intj, float* Ux, float* Uy, float* P)
{
    return Ux[j * Nx + i] + tau * (-((Ux[j * Nx + i] + abs(Ux[j * Nx + i])) / 2.0 * (Ux[j * Nx + i] -
```

```
Ux[j * Nx + i - 1]) / h + (Ux[j * Nx + i] - abs(Ux[j * Nx + i])) / 2.0 * (Ux[j * Nx + i + 1] - Ux[j * Nx + i]) / h) - ((Uy[j * Nx + i] + abs(Uy[j * Nx + i])) / 2.0 * (Ux[j * Nx + i] - Ux[(j - 1) * Nx + i]) / h + (Uy[j * Nx + i] - abs(Uy[j * Nx + i])) / 2.0 * (Ux[(j + 1) * Nx + i] - Ux[j * Nx + i]) / h) - (P[(j + 1) * Nx + i + 1] + P[(j - 1) * Nx + i - 1]) / 4.0 / h / ro + nu * (Ux[j * Nx + i + 1] + Ux[j * Nx + i - 1] + Ux[(j + 1) * Nx + i] + Ux[(j - 1) * Nx + i] - 4 * Ux[j * Nx + i]) / h / h);
}
```

Аналогично с помощью функции f3() рассчитывается поле скоростей по оси Y, функция вызывается в ядре, а выполняется на видекарте. В соответствующей функции программы также выполняется обновление поля турбулентной вязкости согласно модели А.Н. Секундова (5).

При таком подходе к распараллеливанию приобретает значительный выигрыш по времени. Вместо того, чтобы пересчитывать значения каждой ячейки одна за другой, каждая нить выбирает себе задачу из предложенных в соответствии с ее координатами и рассчитывает по три значения для каждой точки.

**Вычислительные эксперименты по ускорению параллельной реализации математической модели турбулентной динамики сплошной среды.** Вычислительные эксперименты проводились на расчетной сетке размерностей 100x100, 200x200, ..., 1000x1000.

В процессе расчета модели поле скоростей сплошной среды приняло вид, отображенный на рис. 6. Можно наблюдать теоретически ожидаемые вихревые образования и турбулентные пульсации скорости в правом верхнем и нижнем левом углах расчетной области.

В таблице представлены результаты по затратам машинного времени для последовательного и параллельного алгоритмов с использованием технологии CUDA.

Последовательная программа запускалась на Intel Core i7 частотой 2,2 ГГц. Параллельные расчеты CUDA проводились на NVIDIA GEFORCE 630M и NVIDIA GTX 1050.

**Обсуждение.** Сравнительный анализ затрат машинного времени при последовательной и параллельной реализациях программы моделирования механики жидкости иллюстрирует рис. 7.

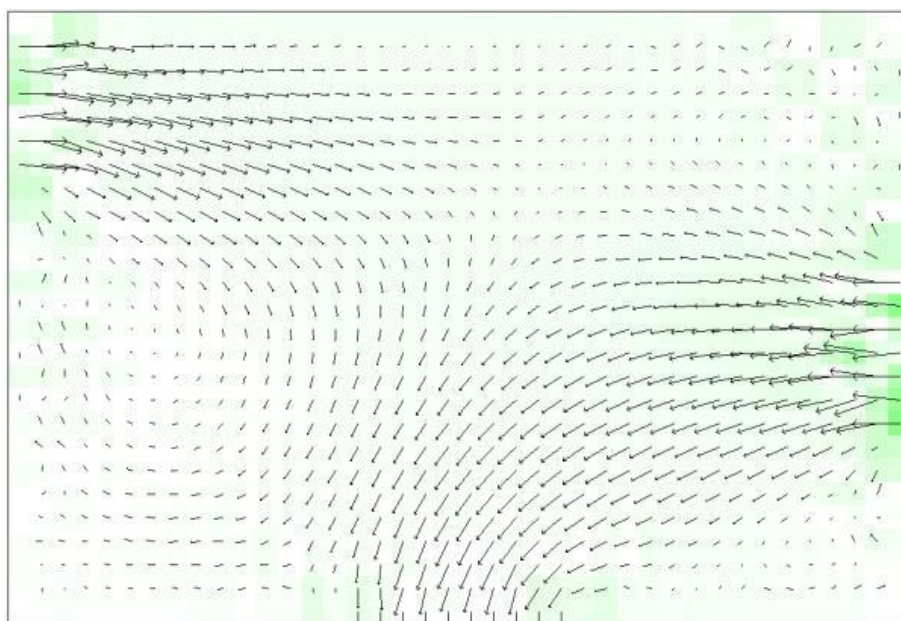


Рис. 6. Поле скоростей при расчете математической модели движения вязкой несжимаемой жидкости

### Запуск последовательного и CUDA алгоритмов

Размерность системы	100	200	300	400	500	600	700	800	900	1000
Последовательная реализация	1,10	4,05	7,50	13,58	21,67	30,58	41,87	54,1	68,22	85,94
Nvidia GeForce GT630M	0,64	1,77	3,85	6,53	10,48	14,46	20,03	25,8	32,96	40,15
Nvidia GeForce GTX 1050	0,31	0,69	1,34	2,25	3,46	4,89	6,52	8,51	11,04	13,44

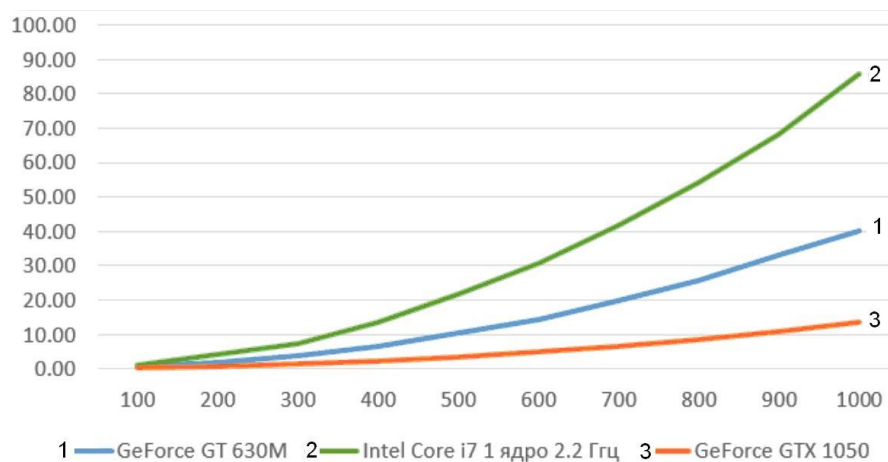


Рис. 7. Графики затрат машинного времени при последовательной и параллельной реализациях программы моделирования механики жидкости

Параллельный алгоритм с использованием технологии CUDA ускорил вычисления в 8 раз на современной видеокарте NVIDIA GTX 1050 с computer capability 6.1. Менее производительная видеокарта Nvidia Geforce 630M с computer capability 2.1 дала ускорение вычислений в среднем в 2 раза.

Таким образом, построено устойчивое высокопроизводительное решение вычислительно сложной задачи о моделировании вязкой несжимаемой жидкости с включением турбулентных образований на базе интерфейса CUDA графического ускорителя NVidia. Иссле-

дование показало возможности значительного ускорения при расчете описанной математической модели (от 2 до 8 раз) по сравнению с последовательной реализацией в зависимости от производительности аппаратной конфигурации.

Предложенные схемы получения устойчивых разностных решений и распределения мощностей графического вычислителя актуальны как для ученых, занимающихся исследованиями механики сплошных сред, так и для инженеров, проектирующих энергетические установки с жидкостным питанием.



## Список литературы

1. Митрофанова О.В. Гидродинамика и теплообмен закрученных потоков в каналах ядерно-энергетических установок. – М.: ФИЗМАТЛИТ, 2010. – 288 с.
2. Лойцянский Л.Г. Механика жидкости и газа: учеб. для вузов. – 7-е изд., испр. – М.: Дрофа, 2003. – 840 с.
3. Кожевников С.О., Калинин Е.Н. К постановке задачи моделирования движения жидкой среды в перемешивающем аппарате // Вестник ЧГУ. – 2017. – № 1. – С. 40–47.
4. Моделирование процессов розлива нефтепродуктов по водной поверхности с использованием суперкомпьютеров с графическими ускорителями / Ф.Н. Ясинский, С.Г. Сидоров, И.А. Малый и др. // Вестник ИГЭУ. – 2013. – Вып. 6. – С. 90–94.
5. Ковальцов Н.Н. Прикладная механика жидкости и газа. – Ульяновск: УлГТУ, 2010. – 219 с.
6. Heuveline V., Krause M.J., Latt J. Towards a hybrid parallelization of lattice Boltzmann methods // Computers and Mathematics with Applications. – 2009. – № 58. – P. 1071–1080.
7. Демидов Г.Е., Егоров А.Г., Нуриев А.Н. Решение задач вычислительной гидродинамики с применением технологии Nvidia CUDA // Ученые записки КГУ. – 2010. – Т. 152 (1). – С. 142–154.
8. Nishiura D., Furuichi M., Sakaguchi H. Computational performance of a smoothed particle hydrodynamics simulation for shared-memory parallel computing // Computer Physics Communications. – 2015. – № 194. – P. 18–32.
9. New massively parallel scheme for Incompressible Smoothed Particle Hydrodynamics (ISPH) for highly nonlinear and distorted flow / X. Guo, B.D. Rogers, S. Lind, P.K. Stansby // Computer Physics Communications. – 2018. – № 233. – P. 16–28.
10. Ясинский Ф.Н., Кокорин А.С. Математическое моделирование процессов вентиляции и отопления в больших производственных, культурных и спортивных помещениях // Вестник ИГЭУ. – 2010. – Вып. 3. – С. 90–92.
11. Боресков А.В., Харламов А.А. Основы работы с технологией CUDA. – М.: ДМК Пресс, 2010. – 232 с.

## References

1. Mitrofanova, O.V. *Gidrodinamika i teploobmen zakruchennykh potokov v kanalakh yaderno-energeticheskikh ustanovok* [Hydrodynamics and heat

transfer of swirling flows in channels of nuclear power plants]. Moscow: FIZMATLIT, 2010. 288 p.

2. Loytsyanskiy, L.G. *Mekhanika zhidkosti i gaza* [Fluid and gas mechanics]. Moscow: Drofa, 2003. 840 p.

3. Kozhevnikov, S.O., Kalinin, E.N. K postanovke zadachi modelirovaniya dvizheniya zhidkoy sredy v peremeshivayushchem apparate [On the formulation of the problem of modeling the motion of a liquid medium in a mixing apparatus]. *Vestnik ChGU*, 2017, no. 1, pp. 40–47.

4. Yasinsky, F.N., Sidorov, S.G., Malyi, I.A., Potemkina, O.V., Yasinsky, I.F., Mochalov, A.S., Netkachev, V.V. Modelirovanie protsessov rozлива nefteproduktov po vodnoy poverkhnosti s ispol'zovaniem superkomp'yutero v s graficheskimi uskoritel'yami [Modeling of the processes of bottling oil products on the water surface using supercomputers with graphic accelerators]. *Vestnik IGEU*, 2013, issue 6, pp. 90–94.

5. Koval'nogov, N.N. *Prikladnaya mekhanika zhidkosti i gaza* [Applied fluid and gas mechanics]. Ulyanovsk: UIGTU, 2010. 219 p.

6. Heuveline, V., Krause, M.J., Latt, J. Towards a hybrid parallelization of lattice Boltzmann methods. *Computers and Mathematics with Applications*, 2009, no. 58, pp. 1071–1080.

7. Demidov, G.E., Egorov, A.G., Nuriev, A.N. Reshenie zadach vychislitel'noy gidrodinamiki s primeneniem tekhnologii Nvidia CUDA [Solution of the computational hydrodynamics problems by means of Nvidia CUDA technology]. *Uchenye zapiski KGU*, 2010, vol. 152(1), pp. 142–154.

8. Nishiura, D., Furuichi, M., Sakaguchi, H. Computational performance of a smoothed particle hydrodynamics simulation for shared-memory parallel computing. *Computer Physics Communications*, 2015, no. 194, pp. 18–32.

9. Guo, X., Rogers, B.D., Lind, S., Stansby, P.K. New massively parallel scheme for Incompressible Smoothed Particle Hydrodynamics (ISPH) for highly nonlinear and distorted flow. *Computer Physics Communications*, 2018, no. 233, pp. 16–28.

10. Yasinskiy, F.N., Kokorin, A.S. Matematicheskoe modelirovanie protsessov ventilyatsii i otopleniya v bol'shikh proizvodstvennykh, kul'turnykh i sportivnykh pomeshcheniyakh [Mathematical modeling of the processes of ventilation and heating in large industrial, cultural and sports facilities]. *Vestnik IGEU*, 2010, issue 3, pp. 90–92.

11. Boreskov, A.V., Kharlamov, A.A. *Osnovy raboty s tekhnologiyey CUDA* [Basics of working with CUDA technology]. Moscow: DMK Press, 2010. 232 p.

Ясинский Игорь Федорович,

ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина»,  
кандидат технических наук, доцент кафедры высокопроизводительных вычислительных систем,  
e-mail: igor2266@yandex.ru.com

Yasinskiy Igor Fedorovich,

Ivanovo State Power Engineering University

Candidate of Engineering Sciences (PhD), Associate Professor of the Department of High-Performance Computer Systems,

e-mail: igor2266@yandex.ru

Гудухина Александра Антоновна,

ФГБОУВО «Ивановский государственный энергетический университет имени В.И. Ленина»,

магистрант кафедры высокопроизводительных вычислительных систем,

e-mail: esqulapic@mail.ru

Gudukhina Alexandra Antonovna,

Ivanovo State Power Engineering University,

A Master course student of the Department of High-Performance Computer Systems,

e-mail: esqulapic@mail.ru