

МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ ГИДРОДИНАМИКИ И ТЕПЛООБМЕНА В СФЕРИЧЕСКИХ СЛОЯХ С ИСПОЛЬЗОВАНИЕМ ВЫЧИСЛЕНИЙ НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ

И.В. Бычин, А.В. Гореликов

Численное моделирование теплообмена и течения жидкости в сферических слоях является одним из основных методов исследования процессов конвекции в жидких и газовых оболочках Земли, звезд и планет. В частности, вычислительный эксперимент широко используется при изучении мантийной конвекции и конвекции во внешнем жидком ядре Земли. Кроме изучения природных процессов численное моделирование конвекции в областях со сферической геометрией необходимо для решения ряда инженерно-технических задач теплообмена. Примером являются задачи конвекции в топливных баках и отсеках космических аппаратов сферической формы.

На кафедре прикладной математики Сургутского государственного университета разработан комплекс программ для численного моделирования течений вязкой несжимаемой теплопроводящей жидкости в сферических оболочках. В качестве математической модели используются: уравнение Навье – Стокса, уравнение неразрывности и уравнение переноса тепла. Программный комплекс предназначен для решения двух (в осесимметричной постановке) и трехмерных задач гидродинамики и теплообмена в сфере или сферическом слое. Программы позволяют использовать размерную или любую допустимую безразмерную формулировку решаемой задачи. В вычислительных программах реализован метод контрольного объема [1] в сферических координатах. Поле скорости течения жидкости и давление рассчитывается по алгоритму PISO (Pressure-Implicit with Splitting of Operators) [2, 3].

Численное моделирование задач гидродинамики и теплообмена, зачастую требует большого количества вычислительных ресурсов. Таким образом, возникает необходимость использования высокопроизводительных вычислительных систем с параллельной архитектурой. Первоначально программный код был написан на языке Fortran 90 и распараллелен с использованием стандарта OpenMP для систем с общей памятью и интерфейса MPI для вычислительных систем с распределенной памятью. В настоящее время большие возможности по увеличению скорости вычислений предоставляют гетерогенные системы, в состав которых наряду с центральными процессорами (ЦПУ) входят графические ускорители (ГПУ). Для использования вычислительных возможностей систем с графическими ускорителями при моделировании задач гидродинамики и теплообмена, код программного комплекса был портирован с использованием стандарта OpenCL [4, 5].

В исходном коде вычислительной программы, разработанной на языках программирования C и OpenCL C, можно выделить три логических части:

1. последовательная часть;
2. управляющая программа;
3. вычислительные ядра.

В последовательной части исходного кода реализованы функции задания геометрии расчетной области, построения расчетной сетки, задания начальных и граничных условий, сохранения данных в файлы и т.д. Последовательная часть кода выполняется на ЦПУ. Управляющая часть программы разработана с использованием OpenCL API (application programming interface) и организует взаимодействие хост-системы (одного ядра ЦПУ) с OpenCL-устройством (ГПУ). В управляющей программе реализованы следующие основные функции:

1. создание контекста для исполнения программного кода вычислительных функций на OpenCL-устройстве;
2. выбор графической карты в качестве OpenCL-устройства;
3. инициализация графической карты в созданном контексте;
4. создание очереди команд на основе ID графической карты и контекста;
5. создание OpenCL-программы на основе исходного кода вычислительных функций и контекста;
6. сборка OpenCL-программы;
7. создание OpenCL-объектов «ядер» на основе программного кода вычислительных функций;
8. создание объектов памяти для организации хранения входных и выходных данных в буферах памяти ГПУ;
9. чтение/запись данных;
10. исполнение созданных «ядер» на процессорах графической карты;
11. освобождение ресурсов.

Вычислительные ядра являются массивно-параллельными частями исходного кода, исполняемыми на графическом процессоре. Процедуры исходного кода ЦПУ-версии программ, допускающие хотя бы минимальный прирост производительности при выполнении на ГПУ, были портированы в «OpenCL-ядра». В

основном такая адаптация вычислительного кода представляла сугубо техническую работу по переписыванию процедур выполняющих циклическую обработку массивов независимых данных. В вычислительных ядрах реализованы: процедуры расчета коэффициентов дискретных аналогов уравнений моделирующих гидродинамику и теплообмен; процедура решения СЛАУ.

В ряде случаев, при портировании вычислительного кода, возникали затруднения связанные со спецификой стандарта OpenCL и архитектурой ГПУ. Стандарт OpenCL поддерживает два вида объектов памяти: буферы глобальной памяти и текстурную память (image). На данный момент, только буферы глобальной памяти позволяют хранить данные с двойной точностью (fp64). Такие буферы ГПУ предназначены только для работы со структурами данных, имеющими линейное представление. Поэтому все двумерные и трехмерные массивы исходной ЦПУ-версии были заменены на одномерные массивы соответствующей длины. Для удобства в использовании таких буферов были введены с помощью директивы препроцессора #define следующие макроподстановки:

- #define DEPLOY3(i, j, k) (dimY * (i) + (j) + dimX * dimY * (k));
- #define DEPLOY2(j, k) (dimY * (k) + (j));

позволяющие работать с одномерными массивами как с многомерными. Профилирование вычислительных ядер, реализующих в качестве решателя СЛАУ метод переменных направлений показало, что выполнение процедуры прогонки вдоль оси OY занимает в два раза больше времени чем прогонка вдоль других направлений. При выполнении такой прогонки, каждый поток обращается к своему непрерывному участку глобальной памяти и такие обращения не объединяются в один запрос, что приводит к существенному замедлению расчёта в следствии высокой латентности глобальной памяти и малого процента кэш-попаданий. В статье [6] было предложено для решения данной проблемы использовать предварительное транспонирование массивов участвующих в прогонке. Применив предложенный способ, была проведена оптимизация процедуры прогонки вдоль оси OY. Время выполнения оптимизированной версии прогонки, с учетом накладных расходов на транспонирование стало сопоставимо со временем выполнения прогонок вдоль других направлений.

Далее представлены результаты тестирования OpenCL-версии программного комплекса. Тестирование производилось на вычислительной системе следующей аппаратной и программной конфигурации:

- четырехъядерный центральный процессор Intel Core i5-3570 с тактовой частой 3400 МГц и 8-ю Гб оперативной памяти типа DDR3;
- графическая карта Nvidia GeForce GTX 560 Ti с 2 Гб оперативной памяти типа GDDR5;
- операционная система Linux Mint 14 (ядро Linux 3.5.0-17-generic);
- компилятор Intel® C++ Composer XE 2013;
- реализации стандарта OpenCL 1.1 от компании Nvidia (cuda toolkit 4.2).

В данной конфигурации производительность процессора Intel Core i5-3570 3,4 GHz при расчетах с двойной точностью составляет 50,62 Gflops (по результатам арифметического теста SiSoftware Sandra Lite 2013). Пиковая производительность графического процессора Nvidia GeForce GTX 560Ti при расчетах с двойной точностью составляет 158 Gflops. Таким образом, можно ожидать примерно трехкратное ускорение при использовании данного графического процессора, поскольку при выполнении OpenCL-версии многопоточного кода основная арифметическая нагрузка ложится на ГПУ.

При проведении тестовых расчетов сравнивалось время выполнения OpenMP-версии программного кода на четырех ядрах ЦПУ со временем выполнения OpenCL-версии на одном ядре ЦПУ и графическом процессоре. В рассмотренных тестовых задачах (Тест 1,2) в зависимости от количества расчетных точек получено 2,1 - 2,5 кратное ускорение при использовании ГПУ, что с учетом накладных расходов достаточно близко к максимально возможному ускорению для вычислительной системы данной конфигурации.

Тест 1 (3D задача о нестационарном течении изотермической жидкости в сферическом слое G_L).

Безразмерные переменные: $\{e_R, e_\theta, e_\phi\}$ – базис в сферических координатах (R, θ, ϕ) ; τ – время; $U = U e_R + V e_\theta + W e_\phi$ – скорость жидкости.

$$G_L = \{ (R, \theta, \phi) : 0.5 < R < 1; 0 \leq \theta \leq \pi; 0 \leq \phi < 2\pi \} ;$$

$$\frac{DU}{D\tau} = -\nabla P + \text{div } \hat{S} + \cos \tau F(R, \theta, \phi) , \quad (1)$$

$$\text{div } U = 0 , \quad (2)$$

где $F(R, \theta, \phi) = 0,5 \cdot ([R \sin \theta \cos \phi]^2 - 1) (\cos \theta e_R - \sin \theta e_\theta)$.

$$\text{Граничные условия: } U|_{R=0.5} = \sin \tau F|_{R=0.5} , \quad U|_{R=1} = \sin \tau F|_{R=1} . \quad (3)$$

$$\text{Начальные условия: } U|_{\tau=0} = 0 . \quad (4)$$

Аналитическое решение: $U_0 = F(R, \theta, \phi) \sin \tau$, $P_0 = R \cos \theta \sin \tau + const$.

Точность численного решения оценивалась по средней в области относительной ошибке, определяемой по следующей формуле:

$$\delta f' = \frac{1}{V_L \bar{G}_i} \int \left| \frac{f - f_0}{f_0} \right| dv \cdot 100\% . \quad (5)$$

В таблице 1 приведены максимальные значения относительных ошибок для компонент скорости на временном отрезке [1, 3];

На рисунках 1-2 представлены результаты численного моделирования в виде изоповерхностей радиальной U и меридиональной V компоненты скорости при $\tau = 3$.

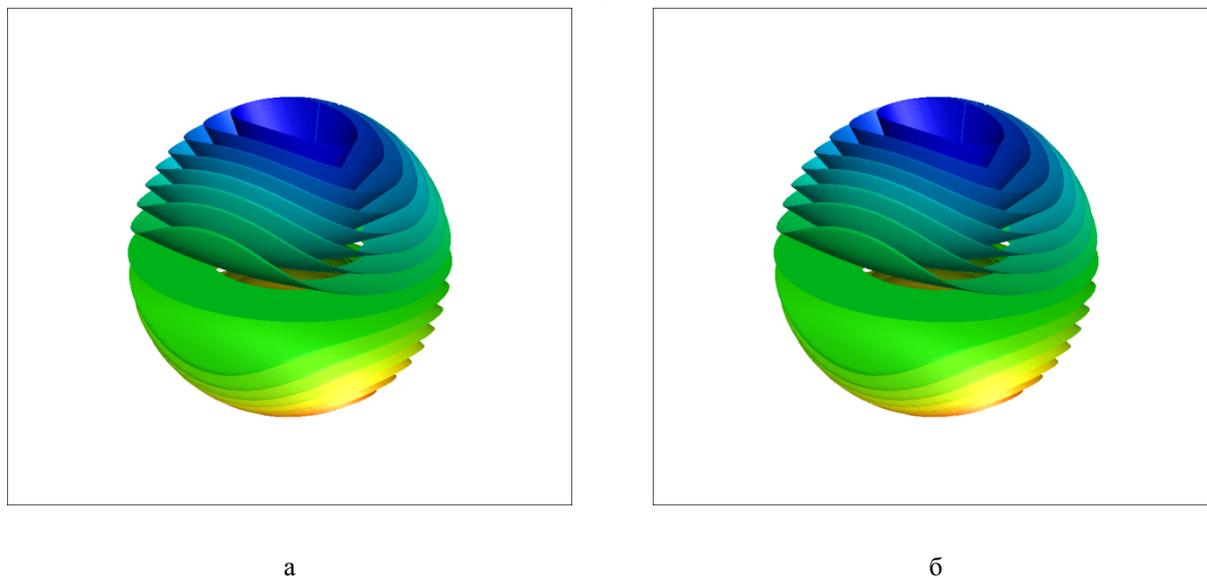


Рис.1. Изоповерхности радиальной компоненты скорости, а – аналитическое решение, б – численное, полученное на расчетной сетке 34x98x194.

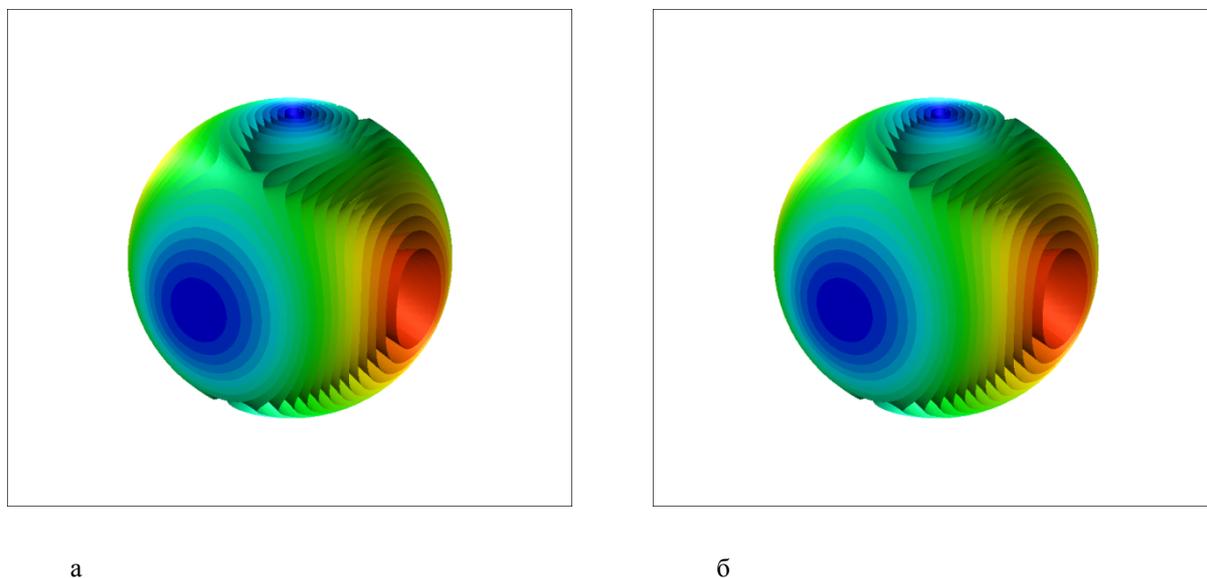


Рис.2. Изоповерхности меридиональной компоненты скорости V , а – аналитическое решение, б – численное, полученное на расчетной сетке 34x98x194.

Таблица 1. Результаты тестовых расчетов

$n_R \times n_\theta \times n_\phi$	$\delta U, \%$	$\delta V, \%$
10×26×50	0,1029	0,2767
34×34×34	0,0478	0,1576
18×50×98	0,0252	0,0710
34×66×130	0,0125	0,0428
34×98×194	0,0062	0,0181

В таблице 2 отображено время, затраченное на тестовые расчёты, проводимые на ЦПУ и ГПУ.

Таблица 2. Сравнение времени расчёта на ЦПУ и ГПУ

$n_R \times n_\theta \times n_\phi$	Время расчёта на ЦПУ, с	Время расчёта на ГПУ + одно ядро ЦПУ, с	Ускорение
10×26×50	98	80	1,22
34×34×34	1380	750	1,84
18×50×98	8052	3523	2,29
34×66×130	23290	9756	2,39
34×98×194	53850	21714	2,48

На рисунке 3 представлен график ускорения расчётов на ГПУ по сравнению с многопоточной версией кода, выполняемой на ЦПУ.

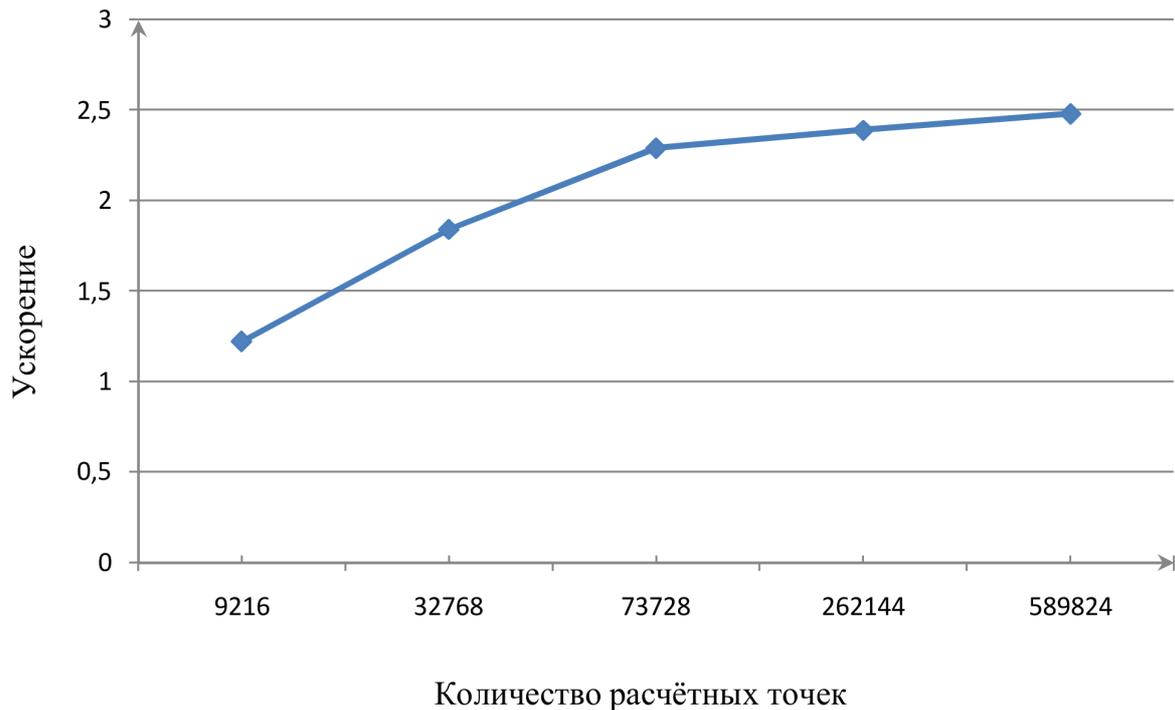


Рис.3. График ускорения расчётов на ГПУ по сравнению с расчётами на ЦПУ.

Тест 2 (задача о естественной конвекции в центральном поле тяжести во вращающейся сферической оболочке [7]).

Следующий тест – это одна из трех задач рассмотренных в работе [7]. В [7] представлены результаты, полученные различными численными методами с использованием разных вычислительных программ шестью группами исследователей из Германии, Франции, Великобритании, США и Японии. Результаты численных исследований хорошо согласуются между собой, что позволяет говорить о высокой степени их достоверности.

Задача рассматривается в следующей постановке. Вращающийся вокруг оси z с угловой скоростью $\Omega = \Omega e_z$ сферический слой заполнен вязкой несжимаемой жидкостью. Отношение внутреннего радиуса сферического слоя r_i к внешнему радиусу r_o равно 0,35. Температуры на границах фиксированы, ΔT

– разность температур на внутренней и внешней границах слоя ($\Delta T > 0$). Используется приближение Буссинеска, причем сила тяжести направлена к центру слоя и линейно зависит от радиуса. В качестве характерного размера выбирается толщина слоя $D = r_o - r_i$, что делает безразмерные радиусы равными $r_o = 20/13$ и $r_i = 7/13$. Характерные масштабы: времени D^2/ν , где ν – кинематическая вязкость; скорости ν/D температуры ΔT давления $\rho \nu \Omega$. Безразмерная температура внешней границы равна нулю. Система уравнений естественной конвекции в равномерно вращающейся системе отсчета, в безразмерной форме:

$$E \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \nabla^2 \mathbf{u} \right) + 2 \mathbf{e}_z \times \mathbf{u} + \nabla P = Ra \frac{\mathbf{r}}{r_o} T \quad (6)$$

$$\nabla \cdot \mathbf{u} = 0 \quad (7)$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T = \frac{1}{Pr} \nabla^2 T \quad (8)$$

Граничные условия:

$$r = r_i : \mathbf{u} = 0 ; T = 1 ; \quad (9)$$

$$r = r_o : \mathbf{u} = 0 ; T = 0 . \quad (10)$$

Начальная скорость равна нулю, а начальная температура:

$$T = \frac{r_o r_i}{r} - r_i + \frac{210 A}{\sqrt{17920} \pi} (1 - 3x^2 + 3x^4 - x^6) \sin^4 \theta \cos 4\phi , \quad (11)$$

где θ – широта, ϕ – долгота, и $x = 2r - r_i - r_o$.

Безразмерные параметры:

$$Ra = \frac{\alpha g_o \Delta T D}{\nu \Omega} \quad \text{– модифицированное число Релея, где } \alpha \text{ – коэффициент теплового расширения,}$$

g_o – ускорение свободного падения на внешнем радиусе; $E = \frac{\nu}{\Omega D^2}$ – число Экмана; $Pr = \frac{\nu}{k}$ – число Прандтля, где k – коэффициент теплопроводности.

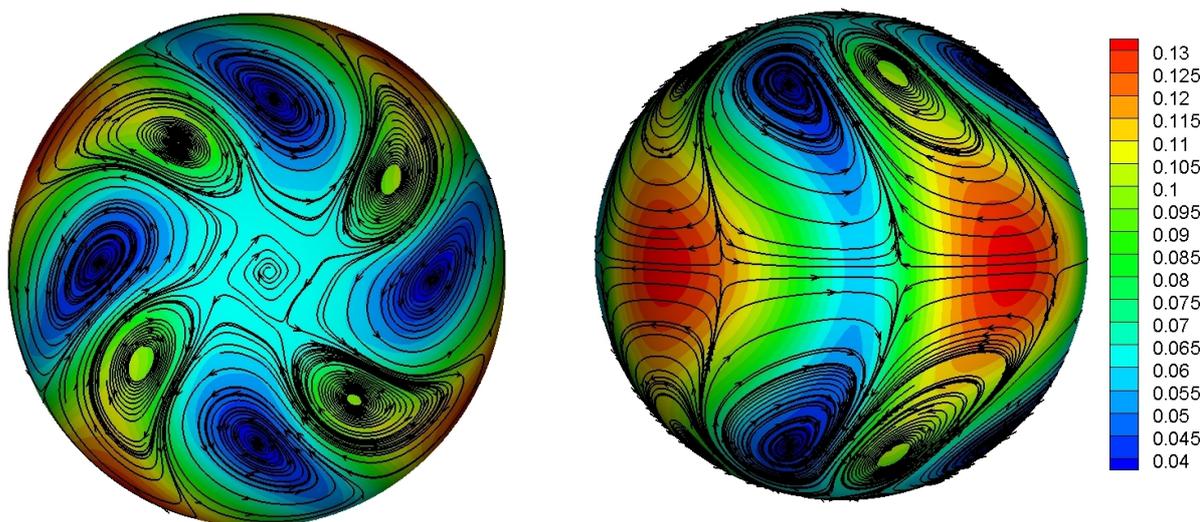
Значения безразмерных параметров тестовой задачи: $Ra = 100$, $E = 10^{-3}$; $Pr = 1$. Квазистационарное решение (рис.4) достигается примерно за 1,2 единиц безразмерного времени. Решение симметрично относительно экватора ($\theta = \pi/2$) и обладает четырехлучевой симметрией по азимутальному направлению. Решение устойчиво к возмущениям несимметричным относительно плоскости экватора [7]. Конвективная структура решения, не меняя размеров и формы, равномерно вращается вокруг оси z . Зависимость от времени в таком квазистационарном решении может быть представлена в виде: $\mathbf{u} = \mathbf{u}(r, \theta, \phi - \omega t)$, $T = T(r, \theta, \phi - \omega t)$, где ω – частота дрейфа в азимутальном направлении.

В работе [7] представлены результаты четырнадцати численных экспериментов по решению данной задачи. Численное моделирование проводилось с использованием различных вариантов комбинаций спектральных методов (разложение по сферическим гармоникам) с методами конечных разностей. Рассчитывалась средняя плотность кинетической энергии жидкости по формуле:

$$E_{kin} = \frac{1}{2V_L V_L} \int \mathbf{u}^2 dV , \quad (12)$$

где V_L – объём жидкой оболочки.

Кроме того, вычислялись локальные характеристики T и u_ϕ в точке с координатами (r, θ, ϕ) : $r = (r_o + r_i)/2$; $\theta = \pi/2$ – экваториальная плоскость; ϕ – координата определяется условиями: $u_r = 0$ и $(\partial u_r / \partial \phi) > 0$. В таблице 3 представлены значения E_{kin} , T , u_ϕ полученные в данной работе на сетке $n_r = 122$, $n_\theta = 162$, $n_\phi = 192$, и значения предложенные в [7], как стандартные для рассматриваемой задачи.



а

б

Рис. 4. Распределение температуры и линии тока тангенциальных составляющих скорости на сферической поверхности ($R=1,3693$).

Таблица 3.

	E_{kin}	T	u_{ϕ}
Результаты теста 2	58,419	0,4282	-10,099
Результаты работы [7]	58,348±0,050	0,42812±0,00012	-10,1571±0,0020

Заключение

С использованием стандарта OpenCL разработан и протестирован программный комплекс для численного моделирования 3D задач гидродинамики и теплообмена в сферических слоях на гетерогенных вычислительных системах с графическими процессорами. Если экстраполировать полученные данные о ускорении на графическую карту Nvidia Tesla K20x (пиковая производительность при расчетах с двойной точностью - 1,31 Tflops), то получается примерно двадцатикратное ускорение по сравнению с четырехпоточным кодом исполняемым на Intel Core i5-3570 на задачах вычислительной гидродинамики. Предполагается использовать разработанный программный комплекс для численного исследования естественной конвекции в сферических оболочках.

Работа выполнена при финансовой поддержке РФФИ в рамках научного проекта № 13-01-12051 офи_м.

ЛИТЕРАТУРА:

1. Патанкар С. Численные методы решения задач теплообмена и динамики жидкости – М.: Энергоатомиздат, 1984. 152 с.
2. Issa R.I. Solution on the implicitly discretised fluid flow equations by operator-splitting // Journal of Computational Physics, 61, 1985, P. 40-65.
3. Issa R.I., Gosman A.D. and Watkins A.P. The Computation of compressible and incompressible recirculating flows by a non-iterative implicit scheme // Journal of Computational Physics, 62, 1986, P. 66-82.
4. OpenCL : сайт. – URL: <http://www.khronos.org/ocl/> (дата обращения 10.05.2013).
5. Munshi, A. OpenCL Programming Guide // Addison-Wesley Professional, 2011. 603 p.
6. Моделирование течений в акватории Белого моря: применение массивно-параллельной архитектуры GPU / Н.А. Сахарных, С.Б. Березин, В.М. Пасконов // Журнал "Суперкомпьютеры". – N 2(6).– 2011. – с. 51-55.
7. Christensen, U.R., et al. A numerical dynamo benchmark // Physics of the Earth and Planetary Interiors, 128, 2001, 25–34.