

УДК 519.6

**ОЦЕНКА ВОЗМОЖНОСТЕЙ ГРАФИЧЕСКОГО ПРОЦЕССОРА ВИДЕОКАРТЫ  
ПРИ РЕШЕНИИ ЗАДАЧИ КОНВЕКТИВНОГО ТЕПЛОПЕРЕНОСА**

© 2014 г.

**Ю.С. Цивинская, М.Ю. Цивинский, В.Н. Попов**

Институт теоретической и прикладной механики им. С.А. Христиановича СО РАН, Новосибирск

popov@itam.nsc.ru

*Поступила в редакцию 30.01.2014*

На примере решения нестационарной двумерной тестовой задачи конвективного теплопереноса рассмотрены возможности параллельных вычислений на персональном компьютере с видеокартой NVIDIA при реализации систем линейных алгебраических уравнений, полученных в случае неявной аппроксимации исходной модели на структурированных пространственных сетках.

*Ключевые слова:* СЛАУ, итерационные методы, параллельные вычисления, графические процессоры.

**Введение**

Для решения современных задач механики жидкости и тепломассопереноса ресурсов персонального компьютера часто оказывается недостаточно. В настоящее время всё более широко используется новое направление в ускорении вычислений – применение графических процессорных устройств (GPU – graphics processing units). Для расчетов на GPU компанией NVIDIA представлена унифицированная архитектура компьютерных вычислений CUDA (Compute Unified Device Architecture), которую можно применять при решении широкого круга задач с организацией параллельных вычислений. В связи с этим разработка алгоритмов для быстрой параллельной реализации систем линейных алгебраических уравнений (СЛАУ) является актуальной проблемой, так как доля таких расчетов, как правило, значительно преобладает по сравнению с остальными вычислениями при решении нестационарных задач распределения тепла и движения жидкости [1–3].

Применительно к задачам линейной алгебры для операций над матрицами и векторами уже существуют алгоритмы вычислений с использованием GPU, позволяющие ускорить расчеты с плотными матрицами в десятки и сотни раз [4]. Наряду с этим разрабатываются алгоритмы для решения итерационными методами разреженных систем линейных алгебраических уравнений [5–9], получаемых при численной реализации широкого класса задач математической физики, которые описываются дифференциальными уравнениями в частных производных. В одной из первых работ, посвященных реализации алгоритмов решения СЛАУ с разреженной матрицей на GPU [5], была предложена концепция

использования итерационных методов. Позднее в [9] сделан вывод, что ввиду отсутствия единого метода, подходящего для любого класса задач, поскольку один и тот же алгоритм может сходиться или не сходиться к решению в зависимости от типа матрицы, необходимо иметь библиотеку программ с реализацией различных итерационных методов.

В настоящей работе на примере решения нестационарной двумерной тестовой задачи конвективного теплопереноса рассмотрены возможности параллельных вычислений для реализации СЛАУ, полученных при неявной аппроксимации исходной модели методом конечных разностей. При расчетах использовался персональный компьютер, оснащенный дополнительной видеокартой NVIDIA GeForce GTX 285, являющейся 240-ядерным процессором GPU. При численной реализации исходной задачи сравнивалось использование CPU и CPU+GPU при решении систем уравнений различными итерационными методами для оценки общего затраченного времени. Вычисления проводились с одинарной точностью, что, с одной стороны, увеличивало число требуемых для сходимости итераций, а с другой – уменьшало суммарное время работы программы за счёт более эффективного использования вычислительных блоков GPU. Дополнительным доводом было и то, что даже на последних версиях ускорителей NVIDIA Tesla переход от одинарной точности к двойной заметно уменьшает итоговую производительность расчетов [10].

**Модельная задача и её аппроксимация**

С целью разработки и анализа алгоритмов, позволяющих применять графический процес-

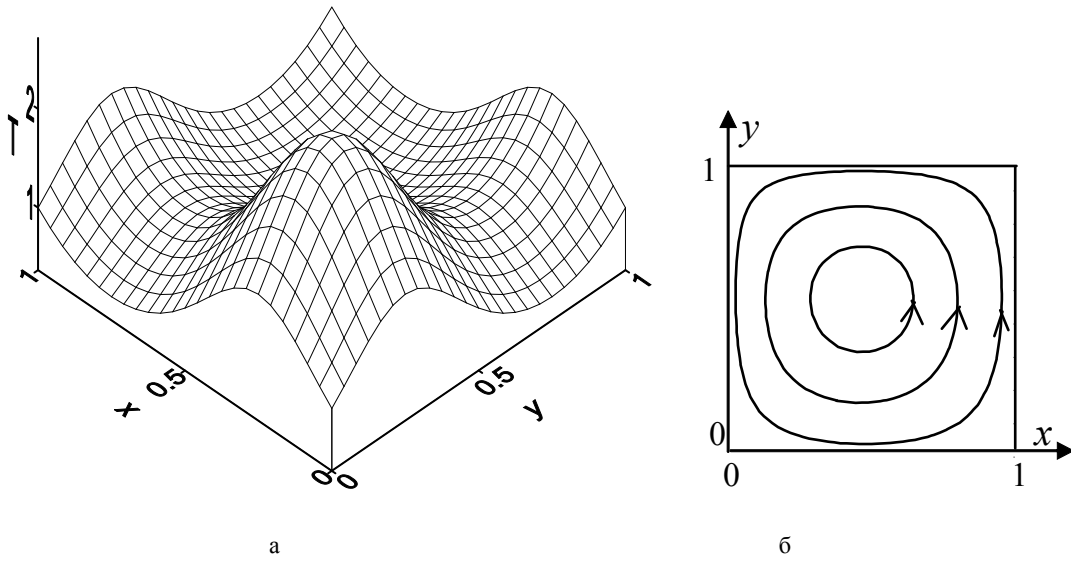


Рис. 1. Распределение температурного поля  $T$  при  $t=1$ ,  $\alpha=0.5$  (а) и линии тока в расчетной области (б)

сор видеокарты для параллельных вычислений при реализации моделей, описываемых уравнениями конвективной теплопроводности, была рассмотрена следующая тестовая задача:

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = a \Delta T + f, \quad t, x, y \in [0,1], \quad (1)$$

где

$$f(t, x, y) = \frac{\partial \theta}{\partial t} + u \frac{\partial \theta}{\partial x} + v \frac{\partial \theta}{\partial y} - a \frac{\partial^2 \theta}{\partial x^2} - a \frac{\partial^2 \theta}{\partial y^2},$$

$$\theta(t, x, y) = \exp \{ \alpha t [\sin(2\pi x) + \sin(2\pi y)] \},$$

$$u(x, y) = 8(x^4 - 2x^3 + x^2)(-16y^3 + 30y^2 - 12y),$$

$$v(x, y) = -8(4x^3 - 6x^2 + 2x)(-4y^4 + 10y^3 - 6y^2),$$

$$\max \{ |u|, |v| \} = 1.$$

Начальные условия

$$T|_{t=0} = \theta(0, x, y). \quad (2)$$

Граничные условия

$$\left. \frac{\partial T}{\partial x} \right|_{x=0,1} = 2\alpha t \pi \theta, \quad \left. \frac{\partial T}{\partial y} \right|_{y=0,1} = 2\alpha t \pi \theta. \quad (3)$$

Рисунок 1 иллюстрирует функцию  $T$  и структуру течения в расчетной области при  $t=1$  и значении  $\alpha=0.5$ , используемом в вычислениях.

Для решения задачи (1)–(3) применялся конечно-разностный алгоритм. В расчетной области использовалась равномерная пространственная сетка вида

$$x_i = h_x(i-1), \quad i = 1, \dots, I, \quad h_x = 1/(I-1),$$

$$y_j = h_y j, \quad j = 1, \dots, J, \quad h_y = 1/(J-1).$$

Вдоль временной переменной использовался постоянный шаг  $\tau$ . Распределение температуры

на  $n$ -м временном шаге в момент времени  $t_n$  в расчетной области описывалось в узлах сетки значениями  $T_{i,j}^n$ .

Разностные уравнения строились посредством аппроксимации (1) и граничных условий (3). Так как для конвективных членов используется противопоточная схема, то погрешность аппроксимации задачи имеет порядок  $O(\tau, h_x, h_y)$ . Разностное уравнение для узла  $(i, j)$  при  $t = t_n$  запишем в виде

$$e_{i,j} T_{i,j}^n - a_{i,j} T_{i-1,j}^n - b_{i,j} T_{i,j-1}^n - c_{i,j} T_{i+1,j}^n - d_{i,j} T_{i,j+1}^n = f_{i,j}^n,$$

где

$$a_{i,j} = 0, \quad c_{i,j} = 0, \quad j = 1, \dots, J;$$

$$b_{i,1} = 0, \quad d_{i,J} = 0, \quad i = 1, \dots, I.$$

Получаемую систему линейных алгебраических уравнений можно представить как

$$A \mathbf{T} = \mathbf{f}, \quad (4)$$

где матрица  $A$  является пятидиагональной, монотонной, обладает строгим диагональным преобладанием ( $e_{i,j} > a_{i,j} + b_{i,j} + c_{i,j} + d_{i,j}$ ), а  $\mathbf{T} = \{T_{i,j}^n\}$ ,  $\mathbf{f} = \{f_{i,j}^n\}$  – векторы.

Таким образом, при численной реализации задачи на каждом временном шаге необходимо находить решение системы (4), что и осуществлялось различными итерационными методами [11, 12] до получения на  $k$ -й итерации результата с заданной точностью  $\max_{i,j} |T_{i,j}^{n,k} - T_{i,j}^{n,k-1}| \leq \varepsilon$ ,

$\varepsilon = 10^{-6}$ .

Таблица 1.1

$I \times J$	51 × 51	101 × 101	201 × 201	401 × 401
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	6.0 (49–61) (1.16·10 <sup>-2</sup> )	51.0 (136–182) (6.64·10 <sup>-3</sup> )	524.0 (328–565) (1.23·10 <sup>-2</sup> )	5036.0 (521–1473) (3.55·10 <sup>-2</sup> )
ЦП+видеокарта	7.9 (49–61) (1.15·10 <sup>-2</sup> )	30.6 (136–185) (7.67·10 <sup>-3</sup> )	170.5 (329–559) (1.46·10 <sup>-2</sup> )	1443.9 (515–1489) (3.79·10 <sup>-2</sup> )

Таблица 1.2

$I \times J$	51 × 51	101 × 101	201 × 201	401 × 401
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	1.0 (4) (9.57·10 <sup>-2</sup> )	4.0 (5) (4.92·10 <sup>-2</sup> )	18.0 (6–7) (2.49·10 <sup>-2</sup> )	96.0 (11–13) (1.25·10 <sup>-2</sup> )
ЦП+видеокарта	0.7 (4) (9.58·10 <sup>-2</sup> )	1.0 (5) (4.92·10 <sup>-2</sup> )	2.8 (6–7) (2.47·10 <sup>-2</sup> )	15.9 (11–13) (1.23·10 <sup>-2</sup> )

Таблица 2.1

$I \times J$	51 × 51	101 × 101	201 × 201	401 × 401
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	2.0 (24–34) (1.18·10 <sup>-2</sup> )	16.2 (65–104) (7.36·10 <sup>-3</sup> )	291.5 (329–421) (1.22·10 <sup>-2</sup> )	3214.5 (762–1229) (4.15·10 <sup>-2</sup> )
ЦП+видеокарта	7.0 (24–31) (1.19·10 <sup>-2</sup> )	20.5 (63–100) (6.36·10 <sup>-3</sup> )	124.3 (329–443) (8.05·10 <sup>-3</sup> )	600.5 (613–1257) (2.72·10 <sup>-2</sup> )

Таблица 2.2

$I \times J$	51 × 51	101 × 101	201 × 201	401 × 401
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	1.0 (3) (9.57·10 <sup>-2</sup> )	3.7 (3) (4.92·10 <sup>-2</sup> )	13.9 (3–4) (2.48·10 <sup>-2</sup> )	57.1 (4–5) (1.22·10 <sup>-2</sup> )
ЦП+видеокарта	1.0 (3) (9.58·10 <sup>-2</sup> )	1.0 (3) (4.91·10 <sup>-2</sup> )	1.6 (3–4) (2.46·10 <sup>-2</sup> )	3.1 (4–5) (1.20·10 <sup>-2</sup> )

### Численные результаты

В данном разделе представлены результаты решения описанной задачи с использованием различных итерационных методов. Тестирование проводилось на персональном компьютере, оснащённом CPU 2.67 ГГц и дополнительной видеокартой NVIDIA GeForce GTX 285. Время решения на CPU сравнивалось с временем решения задачи на CPU+GPU. Расчёты проводились на сетках с различным количеством узлов при постоянном временном шаге  $\tau = 10^{-3}$ . В таблицах приведены полученные результаты при  $a = 1$  (N, 1) и  $a = 0.001$  (N, 2), где N – номер таблицы. Основной целью было определить время в секундах, необходимое для расчёта каждого варианта решения задачи, и проанализировать полученные данные. В скобках указано количество итераций, затрачиваемых на различных временных шагах для достижения требуемой точности получаемого результата, либо их минимальное и максимальное количество на различных временных шагах. Наряду с этим определялась погрешность решения задачи сравнением рассчитанных результатов с точными зна-

чениями при  $t_n = 1$  ( $\max_{i,j} |T_{i,j}^n - T(x_i, y_j, 1)|$ ). Программы реализованы на C, использовался компилятор для 64-битной операционной системы.

#### Метод Якоби (метод простой итерации)

$$\mathbf{T}^k = D^{-1}(L\mathbf{T}^{k-1} + U\mathbf{T}^{k-1} + \mathbf{f}).$$

Здесь  $A = -L + D - U$ ,  $D = \{e_{i,j}\}$  – диагональная матрица,  $L = \{a_{i,j}, b_{i,j}\}$ ,  $U = \{c_{i,j}, d_{i,j}\}$  – неотрицательные нижняя и верхняя треугольные матрицы,  $k$  – номер итерации. Результаты вычислений представлены в табл. 1.1, 1.2.

#### Метод наискорейшего спуска

$$\mathbf{r}^k = \mathbf{f} - A\mathbf{T}^k, \quad \tau_k = \frac{(\mathbf{r}^{k-1}, \mathbf{r}^{k-1})}{(A\mathbf{r}^{k-1}, \mathbf{r}^{k-1})},$$

$$\mathbf{T}^k = \mathbf{T}^{k-1} + \tau_k \mathbf{r}^{k-1}.$$

Результаты вычислений представлены в табл. 2.1, 2.2.

#### Метод сопряженных градиентов

$$\mathbf{r}^0 = \mathbf{f} - A\mathbf{T}^0, \quad \mathbf{p}^0 = \mathbf{r}^0, \quad \mathbf{T}^k = \mathbf{T}^{k-1} - \alpha_{k-1} \mathbf{p}^{k-1},$$

$$\alpha_k = \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(\mathbf{p}^k, A\mathbf{p}^k)},$$

Таблица 3.1

$I \times J$	$51 \times 51$	$101 \times 101$	$201 \times 201$	$401 \times 401$
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	1.5 (13–15) ( $1.19 \cdot 10^{-2}$ )	7.3 (23–28) ( $5.46 \cdot 10^{-3}$ )	51.5 (42–49) ( $2.60 \cdot 10^{-3}$ )	443.4 (73–84) ( $1.38 \cdot 10^{-3}$ )
ЦП+видеокарта	4.6 (13–15) ( $1.21 \cdot 10^{-2}$ )	8.6 (23–28) ( $6.43 \cdot 10^{-3}$ )	18.4 (42–49) ( $6.41 \cdot 10^{-3}$ )	57.1 (73–78) ( $1.70 \cdot 10^{-2}$ )

Таблица 3.2

$I \times J$	$51 \times 51$	$101 \times 101$	$201 \times 201$	$401 \times 401$
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	1.0 (3) ( $9.57 \cdot 10^{-2}$ )	3.8 (3) ( $4.92 \cdot 10^{-2}$ )	14.0 (3) ( $2.48 \cdot 10^{-2}$ )	63.9 (3–4) ( $1.22 \cdot 10^{-2}$ )
ЦП+видеокарта	1.2 (3) ( $9.58 \cdot 10^{-2}$ )	1.2 (3) ( $4.91 \cdot 10^{-2}$ )	1.6 (3) ( $2.46 \cdot 10^{-2}$ )	3.7 (3–4) ( $1.20 \cdot 10^{-2}$ )

Таблица 4.1

$I \times J$	$51 \times 51$	$101 \times 101$	$201 \times 201$	$401 \times 401$
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	1.5 (13–17) ( $1.19 \cdot 10^{-2}$ )	11.3 (23–26) ( $5.49 \cdot 10^{-3}$ )	70.6 (40–47) ( $2.63 \cdot 10^{-3}$ )	709.6 (71–85) ( $1.46 \cdot 10^{-3}$ )
ЦП+видеокарта	5.4 (13–17) ( $1.21 \cdot 10^{-2}$ )	9.9 (23–26) ( $6.45 \cdot 10^{-3}$ )	21.5 (40–47) ( $6.44 \cdot 10^{-3}$ )	73.2 (71–80) ( $1.71 \cdot 10^{-2}$ )

Таблица 4.2

$I \times J$	$51 \times 51$	$101 \times 101$	$201 \times 201$	$401 \times 401$
	Время расчетов в с (Кол-во итераций) (Погрешность)			
ЦП	1.0 (3) ( $9.56 \cdot 10^{-2}$ )	4.1 (3) ( $4.92 \cdot 10^{-2}$ )	15.6 (3) ( $2.48 \cdot 10^{-2}$ )	78.2 (4) ( $1.23 \cdot 10^{-2}$ )
ЦП+видеокарта	1.2 (3) ( $9.58 \cdot 10^{-2}$ )	1.3 (3) ( $4.91 \cdot 10^{-2}$ )	1.8 (3) ( $2.46 \cdot 10^{-2}$ )	4.8 (4) ( $1.20 \cdot 10^{-2}$ )

$$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha_{k-1} A \mathbf{p}^{k-1}, \quad \mathbf{p}^k = \mathbf{r}^{k-1} + \beta_{k-1} \mathbf{p}^{k-1},$$

$$\beta_k = \frac{(\mathbf{r}^k, \mathbf{r}^k)}{(\mathbf{r}^{k-1}, \mathbf{r}^{k-1})}.$$

Полученные результаты представлены в табл. 3.1, 3.2.

#### Метод сопряженных градиентов

$$\mathbf{r}^0 = \mathbf{f} - A \mathbf{T}^0, \quad \bar{\mathbf{r}}^0 = \mathbf{r}^0, \quad \mathbf{p}^0 = \mathbf{r}^0, \quad \bar{\mathbf{p}}^0 = \bar{\mathbf{r}}^0,$$

$$\alpha_k = \frac{(\mathbf{r}^k, \bar{\mathbf{r}}^k)}{(A \mathbf{p}^k, \bar{\mathbf{p}}^k)}, \quad \mathbf{T}^k = \mathbf{T}^{k-1} + \alpha_{k-1} \mathbf{p}^{k-1},$$

$$\mathbf{r}^k = \mathbf{r}^{k-1} - \alpha_{k-1} A \mathbf{p}^{k-1}, \quad \bar{\mathbf{r}}^k = \bar{\mathbf{r}}^{k-1} - \alpha_{k-1} A^T \bar{\mathbf{p}}^{k-1},$$

$$\beta_k = \frac{(\mathbf{r}^k, \bar{\mathbf{r}}^k)}{(\mathbf{r}^{k-1}, \bar{\mathbf{r}}^{k-1})},$$

$$\mathbf{p}^k = \mathbf{r}^{k-1} + \beta_{k-1} \mathbf{p}^{k-1}, \quad \bar{\mathbf{p}}^k = \bar{\mathbf{r}}^{k-1} + \beta_{k-1} \bar{\mathbf{p}}^{k-1}.$$

Результаты представлены в табл. 4.1, 4.2.

На рис. 2 представлены величины ускорения вычислений, которые подсчитывались делением времени решения задачи на CPU на время решения с использованием GPU.

Из представленных результатов следует, что в случаях небольшого количества узлов пространственной сетки ( $51 \times 51$ ) выигрыш при применении параллельных расчетов фактически отсутствует. Эффективность GPU начинает проявляться и возрастает с увеличением коли-

чества неизвестных. В ходе численных экспериментов при использовании видеокарты NVIDIA GeForce было получено ускорение расчетов в пределах 3–18 раз в зависимости от количества точек аппроксимации, решаемой СЛАУ и применяемого итерационного метода.

Надо отметить, что с увеличением количества узлов пространственной сетки выявлено расхождение в точности получаемых результатов (не превышающее 3%) при расчетах на ЦП и при использовании видеокарты. Были предприняты попытки улучшить результаты либо уменьшением временного шага, либо повышением точности реализации итерационного процесса. В табл. 5 представлены результаты, полученные при решении задачи методом сопряженных градиентов.

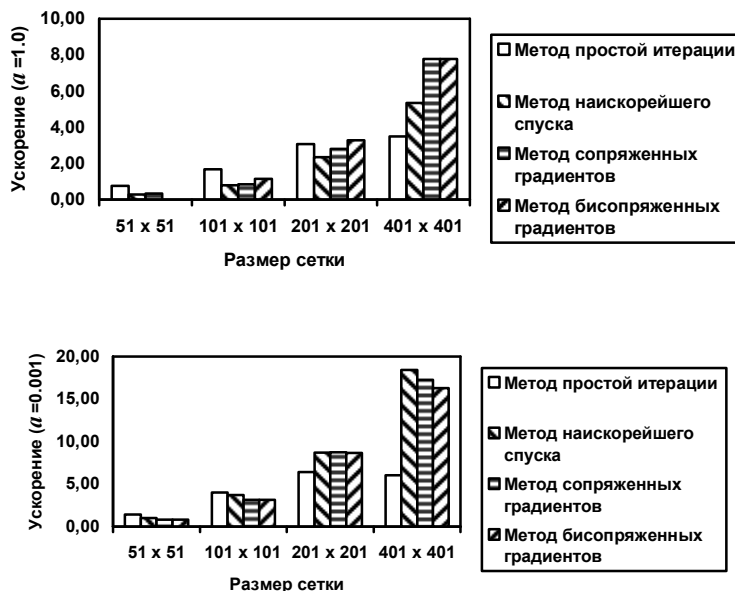
Из полученных данных следует, что такое изменение условий расчётов не привело к улучшению решения и, возможно, объясняется тем, что CUDA-совместимые устройства не полностью реализуют стандарт IEEE-754 представления чисел с десятичной точкой.

#### Заключение

Представлены результаты решения разреженных СЛАУ, полученных на структуриро-

Таблица 5

$I \times J = 401 \times 401$	$\tau = 10^{-4}, \varepsilon = 10^{-6}$	$\tau = 10^{-3}, \varepsilon = 10^{-7}$
	Время расчетов в с (погрешность)	
ЦП	963 ( $2.12 \cdot 10^{-3}$ )	1150 ( $1.19 \cdot 10^{-3}$ )
ЦП+видеокарта	79 ( $2.45 \cdot 10^{-2}$ )	99 ( $1.75 \cdot 10^{-2}$ )

Рис. 2. Величины ускорения расчетов при различных значениях параметра  $a$ 

ванных пространственных сетках, различными итерационными методами на ПК с графическим процессором NVIDIA. Определено, что такой подход при реализации задач позволяет значительно повысить производительность расчетов при относительно невысоких затратах, а эффективность применения графического процессора растёт с увеличением размерности СЛАУ.

*Работа выполнена в рамках проекта РФФИ № 10-01-00575-а.*

#### Список литературы

1. Максимов Д.Ю., Кудряшов И.Ю., Марченко Н.А. Суперускорение гидродинамических расчетов с помощью применения графических процессоров NVIDIA и технологий программирования CUDA // Вестник ЦКР Роснедра. 2010. № 1. С. 67–69.
2. Коньшин В.Н. Применение программного комплекса FlowVision для решения прикладных задач на терафлопных вычислительных системах // Параллельные вычислительные технологии (ПаВТ2008): Труды Международной научной конференции (Санкт-Петербург, 28 января–1 февраля 2008 г.). Челябинск: Изд-во ЮУрГУ, 2008. С. 124–129.
3. Попов В.Н. Моделирование затвердевания металлической капли на холодной подложке // Математическое моделирование. 2001. Т. 13. № 9. С. 119–127.
4. Volkov V., Demmel J.W. Benchmarking GPUs to tune dense linear algebra // [http://mc.stanford.edu/cgi-bin/images/6/65/SC08\\_Volkov\\_GPU.pdf](http://mc.stanford.edu/cgi-bin/images/6/65/SC08_Volkov_GPU.pdf).
5. Bell N., Garland M. Efficient Sparse Matrix Vector Multiplication on Cuda // NVIDIA Technical Report NVR-2008-004. December 11, 2008.
6. Baskaran M., Bordawekar R. Optimising sparse matrix-vector multiplication on GPUs // IBM Tech. Rep. 2009. (<http://www.nvidia.com/docs/IO/66889/nvr-2008-004.pdf>).
7. Buatois L., Cauman G., Levy B. Concurrent Number Cruncher: An Efficient Sparse Linear Solver on GPU // High Performance Computation Conference (HPCC). Springer Lecture Notes in Computer Sciences. 2008.
8. Чадов С.Н. Реализация алгоритма решения несимметричных систем линейных уравнений на графических процессорах // Вычислительные методы и программирование. 2009. Т.10. С. 321–326.
9. Губайдуллин Д.А., Никифоров А.И., Садовников Р.В. Использование графических процессоров для решения разреженных СЛАУ итерационными методами подпространств Крылова с предобуславливанием на примере задач теории фильтрации // Вестник Нижегородского университета им. Н.И. Лобачевского. 2011. № 1. С. 205–212.
10. Кривов М.А., Казеннов А.М. Сравнение вычислительных возможностей графических ускорителей при решении различных классов задач // Труды Всероссийской научно-практической конференции «Применение гибридных высокопроизводительных вычислительных систем для решения научных и инженерных задач». Н. Новгород. 2011. С. 18–24.
11. Самарский А.А., Николаев Т.С. Методы решения сеточных уравнений. М.: Наука, 1978. 592 с.
12. Ильин В.П. Методы неполной факторизации для решения алгебраических систем. М.: Физматлит, 1995. 288 с.

**ASSESSMENT OF VIDEO CARD'S GPU PERFORMANCE CAPABILITIES WHEN SOLVING THE PROBLEM OF CONVECTIVE HEAT TRANSFER***Yu.S. Tsivinskaya, M.Yu. Tsivinsky, V.N. Popov*

Performance capabilities of parallel computing on a PC with NVIDIA graphics card are considered for SLAEs obtained in implicit approximation of the original model on structured spatial grids, an example test case being the solution of a non-stationary two-dimensional convective heat transfer problem.

*Keywords:* system of linear algebraic equations (SLAE), iterative methods, parallel computing, graphics processors.

*References*

1. Maksimov D.Yu., Kudryashov I.Yu., Marchenko N.A. Superuskorenie gidrodinamicheskikh raschetov s pomoshch'yu primeneniya graficheskikh processorov NVIDIA i tekhnologij programmirovaniya SUDA // Vestnik CKR Rosnedra. 2010. № 1. S. 67–69.
2. Kon'shin V.N. Primenenie programmnoogo kompleksa FlowVision dlya resheniya prikladnykh zadach na teraflopnnykh vychislitel'nykh sistemah // Parallelnyye vychislitel'nye tekhnologii (PaVT'2008): Trudy Mezhdunarodnoy nauchnoy konferencii (Sankt-Peterburg, 28 yanvarya–1 fevralya 2008 g.). Chelyabinsk: Izd-vo YUUGU, 2008. S. 124–129.
3. Popov V.N. Modelirovanie zatverdevaniya metallicheskoj kapli na holodnoj podlozhke // Matematicheskoe modelirovanie. 2001. T. 13. № 9. C. 119–127.
4. Volkov V., Demmel J.W. Benchmarking GPUs to tune dense linear algebra // [http://mc.stanford.edu/cgi-bin/images/6/65/SC08\\_Volkov\\_GPU.pdf](http://mc.stanford.edu/cgi-bin/images/6/65/SC08_Volkov_GPU.pdf).
5. Bell N., Garland M. Efficient Sparse Matrix Vector Multiplication on Cuda // NVIDIA Technical Report NVR-2008-004. December 11, 2008.
6. Baskaran M., Bordawekar R. Optimising sparse matrix-vector multiplication on GPUs // IBM Tech. Rep. 2009. (<http://www.nvidia.com/docs/IO/66889/nvr-2008-004.pdf>).
7. Buatois L., Cauman G., Levy B. Concurrent Number Cruncher: An Efficient Sparse Linear Solver on GPU // High Performance Computation Conference (HPCC). Springer Lecture Notes in Computer Sciences. 2008.
8. Chadov S.N. Realizaciya algoritma resheniya nesimmetrichnykh sistem linejnykh uravnenij na graficheskikh processorah // Vychislitel'nye metody i programmirovaniye. 2009. T.10. S. 321–326.
9. Gubajdullin D.A., Nikiforov A.I., Sadovnikov R.V. Ispol'zovanie graficheskikh processorov dlya resheniya razrezhennykh SLAU iteracionnymi metodami podprostranstv Krylova s predobuslovlivaniem na primere zadach teorii fil'tracii // Vestnik Nizhegorodskogo universiteta im. N.I. Lobachevskogo. 2011. № 1. S. 205–212.
10. Krivov M.A., Kazennov A.M. Sravnenie vychislitel'nykh vozmozhnostej graficheskikh uskoritelej pri reshenii razlichnykh klassov zadach // Trudy Vserossijskoj nauchno-prakticheskoy konferencii «Primenenie gibridnykh vysokoproizvoditel'nykh vychislitel'nykh sistem dlya resheniya nauchnykh i inzhenernykh zadach». N. Novgorod. 2011. S. 18–24.
11. Samarskij A.A., Nikolaev T.C. Metody reshenij setochnykh uravnenij. M.: Nauka, 1978. 592 c.
12. Il'in V.P. Metody nepolnoj faktorizacii dlya resheniya algebraicheskikh sistem. M.: Fizmatlit, 1995. 288 s.