

17. *Pershin I.M.* Analiz i sintez sistem s raspredelennymi parametrami [Analysis and synthesis of systems with distributed parameters]. Pyatigorsk: Reklamno-informatsionnoe agentstvo na KMV, 2007, 244 p.
18. *Rapoport E.Ya.* Strukturnoe modelirovanie ob"ektov i sistem upravleniya s raspredelennymi parametrami [Structural modeling of objects and control systems with distributed parameters]. Moscow: Vysshaya shkola, 2003, 299 p.
19. *Stanislavski V., Imaev D.Kh.* Dinamicheskie modeli pryamotochnykh parogeneratorov kak ob"ektov upravleniya ierarkhicheskoy struktury [Dynamic models of direct-flow steam generators as control objects of hierarchical structure], *Tr. Mezhdunar. nauch.-praktich. konf. "Teoreticheskie i prakticheskie problemy razvitiya elektroenergetiki Rossii" (27-28 iyunya 2002)* [Proceedings of the International. Scientific and practical conference "Theoretical and practical problems of the development of the Russian electric power industry" (June 27-28, 2002)]. Saint Petersburg: SPbGTU, 2002, pp. 203-213.
20. *Tikhonov A.N., Samarskiy A.A.* Uravneniya matematicheskoy fiziki [Equations of mathematical physics]. Moscow: Nauka, 1977.
21. *Tsvetkov F.F., Grigor'ev B.A.* Teplomassoobmen [Heat and mass transfer]. Moscow: Izd-vo MEI, 2005, 550 p.
22. *Chubov A.B., Sokolova E.G.* Osnovy teorii i protsessa skleivaniya drevesiny: ucheb. posobie [Fundamentals of the theory and process of gluing wood: textbook]. Saint Petersburg: SPbGLTU, 2014, 66 p.
23. *Chubinskiy A.N.* Tekhnologiya fanery: ucheb. posobie [Plywood technology: textbook]. Saint Petersburg: SPbGLTU 2019, 81 p.

Статью рекомендовал к опубликованию д.т.н., профессор А.Р. Бестугин.

Ляшенко Александр Леонидович – Санкт-Петербургский государственный университет аэрокосмического приборостроения; e-mail: akuna_matata_kmv@mail.ru; Санкт-Петербург, Россия; тел.: +79111049232; кафедра управления в технических системах; доцент.

Liashenko Alexander Leonidovich – Saint-Petersburg State University of Aerospace Instrumentation; e-mail: akuna_matata_kmv@mail.ru; Saint-Petersburg, Russia; phone: +79111049232; the department of control in technical systems; associate professor.

УДК 519.6

DOI 10.18522/2311-3103-2021-5-93-105

Н.Н. Грачева, В.Н. Литвинов, А.В. Никитина, Н.Б. Руденко, А.Е. Чистяков
ВЫЧИСЛИТЕЛЬНЫЕ АСПЕКТЫ РЕШЕНИЯ СЕТОЧНЫХ УРАВНЕНИЙ
НА ГРАФИЧЕСКИХ УСКОРИТЕЛЯХ

Для прогнозирования чрезвычайных ситуаций (ЧС) и необратимых последствий деятельности человека ученые широко применяется математическое моделирование. При возникновении ЧС очень важно минимизировать время принятия решения. Разработка проекта решения может основываться на прогнозе изменения моделируемого процесса. При численном решении задач гидрофизики и биологической кинетики возникает необходимость в разработке эффективных методов решения систем сеточных уравнений большой размерности с несамосопряженным оператором. Большой объем обрабатываемой информации и сложность вычислений приводят к необходимости использования вычислительных кластеров, в состав которых добавляются видеоадаптеры для увеличения производительности вычислительной системы и скорости обработки информации. Целью исследования является разработка программного модуля, реализующего алгоритм решения системы линейных алгебраических уравнений (СЛАУ) модифицированным попеременно-треугольным итерационным методом (МПТМ) (самосопряженный и несамосопряженный случаи) с использованием технологии NVIDIA CUDA. Описан способ декомпозиции расчетной области в трехмерном случае. Предложена графовая модель организации параллельного конвейерного вычислительного процесса, ориентированная на графический ускоритель GPU

(Graphics Processing Unit). Для двух видеоадаптеров с различными характеристиками были проведены экспериментальные исследования для определения оптимальной двумерной конфигурации потоков в вычислительном блоке, реализуемом на одном потоковом мультипроцессоре, при которой время реализации на GPU одного шага МПТМ является минимальным. Проведенные исследования показали, что выбор способа декомпозиции расчетной области в виде параллелепипедов необходимо выполнять с учетом архитектуры видеоадаптера. Разработанные алгоритм и программный модуль позволяют более эффективно задействовать вычислительные ресурсы GPU, используемой для решения вычислительно-трудоемких задач гидрофизики.

Математическое моделирование; параллельный алгоритм; графический ускоритель.

N.N. Gracheva, V.N. Litvinov, A.V. Nikitina, N.B. Rudenko, A.E. Chistyakov

COMPUTATIONAL ASPECTS OF SOLVING GRID EQUATIONS ON GRAPHICS ACCELERATORS

To predict emergencies and irreversible consequences of human activities, scientists use mathematical modeling. When an emergency occurs, it is very important to minimize the decision-making time. The development of the project solution can be based on the forecast of changes in the modeled process. In the numerical solution of problems of hydrophysics and biological kinetics, it becomes necessary to develop effective methods for solving systemic equations of large dimension with a non-self-adjoint operator. The large volume of processed information and the complexity of computations necessitate the use of computational clusters, which include video adapters to increase the performance of the computing system and the speed of information processing. The aim of the research is to develop a solution for a module that implements the algorithm of the system of linear algebraic equations (SLAE) by the modified alternative triangular iterative method (MATM) (self-adjoint and non-self-adjoint case) using NVIDIA CUDA technology. A method for decomposition of the computational domain in a three-dimensional case is described. A graph model of a parallel pipeline computational process is proposed, focused on the GPU (Graphics Processing Unit). To determine the two-dimensional configuration of flows in the computational unit, when performing one step of one step, the MATM is minimal. The studies have shown that the choice of the method of decomposition of the computational domain in the form of parallelepipeds must be performed taking into account the architecture of the video adapter. The developed algorithm and software module make it possible to more effectively use the computational resources of the GPU used to solve computationally laborious problems of hydrophysics.

Mathematical modeling; parallel algorithm; graphics accelerator.

Введение. Прогнозирование экологических рисков позволяет снизить ущерб от неблагоприятных и чрезвычайных ситуаций в природе, в частности в прибрежной зоне водоемов. Исследования в этой области требуют построения математических моделей и изучения влияния различных факторов на объект исследования.

В настоящее время большую актуальность приобретает компьютерное моделирование, которое заменяет собой сложные системы и физические модели, а так же позволяет прогнозировать различные явления и процессы, происходящие в природе. В основе компьютерного моделирования, как правило, лежат математические модели, дискретизация которых приводит к СЛАУ большой размерности с самосопряженным и несамосопряженным операторами. Для решения таких систем сеточных уравнений требуются большие вычислительные мощности.

Исследованием и прогнозированием водных экосистем занимаются многие российские и зарубежные учёные. Ученые Института вычислительной математики им. Г.И. Марчука РАН [1, 2], Института прикладной математики им. М.В. Келдыша [3, 4] занимаются анализом и моделированием сложных систем (в экологии, окружающей среде и т.д.), моделированием гидродинамических процессов, прогнозированием климатических изменений мирового океана. Под руководством Матишова Г.Г. проводятся работы по моделированию гидрофизических процессов

на примере Азовского моря [5]. В работах Bonaduce A., Staneva J. описаны математические модели динамики уровней морей [6]. Совершенствованием океанических моделей занимаются ученые Marchesiello P., Androsov A. и др. [7, 8]. Существующие стандартные программные комплексы зачастую включают упрощенные математические модели, не учитывающие пространственно-неоднородное движение водной среды, имеют недостаточную точность моделирования вихревых структур течений водного потока, рельефов берега и дна [9]. Актуальным направлением совершенствования программных комплексов является разработка параллельных алгоритмов, исполняемых как на CPU (Central Processing Unit), так и на GPU (Graphics Processing Unit) [10]. Ученые Weicheng Xue и Christopher J. Roy занимаются исследованиями, связанными с оптимизацией производительности вычислений при решении задач гидродинамики на нескольких графических процессорах, улучшением производительности мульти-GPU на структурированных сетках. В своих работах они показывают, что использование графических процессоров улучшает производительность в 30-70 раз [11, 12]. Исследователи Taku Nagatake и Tomoaki Kuniagi анализировали возможность применения GPU для ускорения вычисления многофазных потоков. Ими было установлено, что время вычислений на графическом процессоре (одиночном GTX280) было примерно в 4 раза быстрее, чем время вычислений на центральном процессоре (Xeon 5040, 4 распараллеленных потока) [13]. David J. Munk и Timoleon Kirougos в своих работах описывают ускорение процесса оптимизации мультифизической топологии на GPU-архитектуре [14].

Целью исследования является разработка алгоритмов, позволяющих решить СЛАУ большой размерности за ограниченное время, и их программная реализация в среде гетерогенных вычислительных систем.

1. Метод решения сеточных уравнений. Пусть A – линейный положительно определенный оператор ($A > 0$), и в конечномерном Гильбертовом пространстве H необходимо решить операторное уравнение

$$Ax = f, A: H \rightarrow H. \quad (1)$$

Для сеточного уравнения (1) используются итерационные методы, которые в канонической форме могут быть представлены уравнением [15–18]

$$B \frac{x^{m+1} - x^m}{\tau_{m+1}} + Ax^m = f, B: H \rightarrow H, \quad (2)$$

где m – номер итерации, $\tau_{m+1} > 0$ – параметр итерации, B – предобуславливатель. Оператор B строится исходя из аддитивного представления оператора A_0 – симметричной части оператора A

$$A_0 = R_1 + R_2, \quad R_1 = R_2^*, \quad (3)$$

где $A = A_0 + A_1$, $A_0 = A_0^*$, $A_1 = -A_1^*$.

Предобуславливатель формируется следующим образом

$$B = (D + \omega R_1) D^{-1} (D + \omega R_2), \quad D = D^* > 0, \quad \omega > 0, \quad (4)$$

где D – диагональный оператор, R_1, R_2 являются нижними и верхними треугольными операторами соответственно.

Алгоритм вычисления сеточных уравнений модифицированным попеременно-треугольным методом вариационного типа записывается в следующем виде

$$\begin{aligned}
r^m &= Ax^m - f, \quad B(\omega_m)w^m = r^m, \quad \tilde{\omega}_m = \sqrt{\frac{(Dw^m, w^m)}{(D^{-1}R_2w^m, R_2w^m)}}, \\
s_m^2 &= 1 - \frac{(A_0w^m, w^m)^2}{(B^{-1}A_0w^m, A_0w^m)(Bw^m, w^m)}, \quad k_m^2 = \frac{(B^{-1}A_1w^m, A_1w^m)}{(B^{-1}A_0w^m, A_0w^m)}, \\
\theta_m &= \frac{1 - \sqrt{\frac{s_m^2 k_m^2}{(1+k_m^2)}}}{1+k_m^2(1-s_m^2)}, \quad \tau_{m+1} = \theta_m \frac{(A_0w^m, w^m)}{(B^{-1}A_0w^m, A_0w^m)}, \quad x^{m+1} = x^m - \tau_{m+1}w^m, \quad \omega_{m+1} = \tilde{\omega}_m,
\end{aligned} \tag{5}$$

где r^m – вектор невязки, w^m – вектор коррекции, s_m – параметр, описывающий скорость сходимости метода, k_m – описывает отношение нормы кососимметричной части оператора к норме симметричной части.

2. Программная реализация метода решений сеточных уравнений. Решение сеточных уравнений с помощью МПТМ (модифицированного попеременно-треугольного итерационного метода) предполагает использование трёхмерной равномерной расчетной сетки [19]

$$\begin{aligned}
\bar{w}_h &= \{t^n = n\tau, x_i = ih_x, y_j = jh_y, z_k = kh_z; n = \overline{0, n_1 - 1}, i = \overline{0, n_1 - 1}, \\
j &= \overline{0, n_2 - 1}, k = \overline{0, n_3 - 1}; (n_1 - 1)\tau = T, (n_1 - 1)h_x = l_x, (n_2 - 1)h_y = l_y, (n_3 - 1)h_z = l_z\},
\end{aligned}$$

где τ – шаг по времени, h_x, h_y, h_z – пространственные шаги, n_1 – количество временных слоев, T – верхняя граница по временной координате, n_1, n_2, n_3 – количество узлов по пространственным координатам, l_x, l_y, l_z – пространственные размеры прямоугольного параллелепипеда, в который вписана расчетная область.

При дискретизации моделей математической физики, в частности гидродинамики, получим систему сеточных уравнений. Каждое уравнение системы может быть представлено в канонической форме, при этом будем использовать семиточечный шаблон (рис. 1):

$$c(m_0)u(m_0) - \sum_{i=1}^6 c(m_0, m_i)u(m_i) = F(m_0),$$

где $m_0(x_i, y_j, z_k)$ – центр шаблона, $M'(P) = \{m_1(x_{i+1}, y_j, z_k), m_2(x_{i-1}, y_j, z_k), m_3(x_i, y_{j+1}, z_k), m_4(x_i, y_{j-1}, z_k), m_5(x_i, y_j, z_{k+1}), m_6(x_i, y_j, z_{k-1})\}$ – окрестность центра, $c_0 \equiv c(m_0)$ – коэффициент центра шаблона, $c_i \equiv c(m_0, m_i)$ – коэффициенты окрестности центра шаблона, F – вектор правых частей, u – рассчитываемый вектор.

В разработанном программном модуле использованы одномерные массивы. Переход от трехмерного представления узла сетки (i, j, k) к одномерной записи (номеру узла) осуществляется по следующей формуле:

$$m_0 = i + jn_1 + kn_1n_2.$$

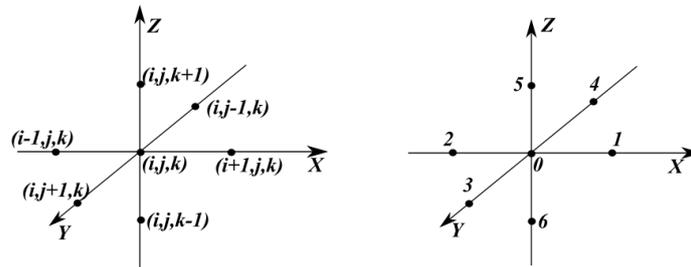


Рис. 1. Сеточный шаблон для решения уравнений гидродинамики

Номера узлов, стоящих в окрестности центра шаблона $m_i, i = \overline{1,6}$, рассчитываются по формулам:

$$m_1 = m_0 + 1, \quad m_2 = m_0 - 1, \quad m_3 = m_0 + n_1, \\ m_4 = m_0 - n_1, \quad m_5 = m_0 + n_1 n_2, \quad m_6 = m_0 - n_1 n_2.$$

Алгоритм МПТМ состоит из четырёх этапов:

- ◆ вычисление значений вектора невязки r^m и его равномерной нормы;
- ◆ расчет вектора поправки w^m ;
- ◆ нахождение скалярных произведений и расчет на их основе итерационных параметров τ_{m+1}, ω_{m+1} ;
- ◆ переход на следующий итерационный слой.

Итерационный процесс продолжается пока норма вектора невязки r^m не достигнет заданной точности.

Наиболее трудоемкой частью алгоритма является расчет вектора поправки из уравнения:

$$(D + \omega R_1) D^{-1} (D + \omega R_2) w^m = r^m.$$

Решение данной задачи сводится к решению двух СЛАУ с нижнетреугольной и верхнетреугольной матрицами:

$$(D + \omega R_1) y^m = r^m, \quad (D + \omega R_2) w^m = D y^m.$$

Расчет вектора невязки осуществляется за $14N$ арифметических действий. Трудоемкость расчета значений вектора поправки составляет $19N$ арифметических операций (по $9N$ и $10N$ для решения СЛАУ верхнетреугольного и нижнетреугольного видов соответственно). Переход на следующую итерацию потребует еще $2N$ арифметических действий. Итого, общее число арифметических действий, необходимое для решения СЛАУ с семидиагональной матрицей с помощью МПТМ в случае известных итерационных параметров τ_{m+1}, ω_{m+1} , равно $35N$.

Определим трудоемкость адаптивной оптимизации МПТМ минимальных поправок. Вычисление векторов $A_0 w^m$, $A_1 w^m$ и $R_2 w^m$ требует по $13N$, $11N$ и $7N$ операций. Умножение векторов на диагональные операторы D^{-1} и D потребует по N операций. Обращение B для нахождения векторов $B^{-1} A_0 w^m$ и $B^{-1} A_1 w^m$ потребует по $19N$ операций. Также необходимо вычислить 6 скалярных произведений, каждое из которых потребует еще по $2N$ операций. Таким образом, каждая адаптивная оп-

тимизации МПТМ минимальных поправок требует $83N$ арифметических операций в несамосопряженном случае и $49N$ в самосопряженном. Процесс расчета итерационных параметров τ_{m+1}, ω_{m+1} является трудоемким, но при решении сеточных уравнений в адаптивном случае достаточно быстро наблюдается его установление. Вследствие чего данные параметры не требуется вычислять на каждой итерации.

3. Параллельная реализация. Численная реализация МПТМ для решения СЛАУ высокой размерности основана на разработанных параллельных алгоритмах, реализующих конвейерный процесс вычислений. Использование данных алгоритмов позволяет полностью задействовать все доступные вычислительные ресурсы, включая высокопроизводительные графические ускорители. Предполагается, что каждый вычислительный узел системы может содержать от 1 до 2 центральных процессоров (CPU), содержащих от 1 до 32 ядер, и от 1 до 12 видеоускорителей NVIDIA с поддержкой технологии CUDA (GPU), обладающих от 192 (NVIDIA GeForce GT 710) до 5120 (NVIDIA Tesla V100) ядер CUDA. Обмен данными между узлами осуществляется с помощью технологии MPI (Message Passing Interface). Организацию вычислений на каждом узле выполняет алгоритм, управляющий всеми доступными потоками CPU и GPU (вычислителями). Каждый вычислитель выполняет расчеты только для своего фрагмента расчетной области. Для этого расчетная область разбивается на подобласти, назначаемые отдельным вычислительным узлам. Далее каждая подобласть разбивается на фрагменты, назначаемые каждому ядру CPU и каждому вычислительному блоку GPU.

Разбиение подобласти на фрагменты, сопоставляемые с каждым вычислителем внутри отдельного вычислительного узла, выполняется следующим образом: количество фрагментов расчетной области по оси Oz выбирается как наименьшее общее кратное оптимальных размерностей вычислительных блоков CUDA для всех задействованных в кластере видеоускорителей. Количество фрагментов расчетной области по оси Ox в блоке (Nx) выбирается таким образом, чтобы их количество было больше количества вычислителей в кластере и они были одинаковыми. Количество фрагментов расчетной области по оси Oy в блоке выбирается так, чтобы время расчета каждого блока разными вычислителями было примерно одинаковое. Для этого предварительно выполняется серия экспериментов по расчету производительности вычислителей, представляющей собой 95-й процентиль времени расчета в пересчете на 1000 узлов расчетной сетки.

Для описания связей между смежными фрагментами расчетной сетки и организации процесса конвейерного вычисления использовалась графовая модель (рис. 2). Каждый узел графа представляет собой объект класса, описывающего фрагмент расчетной области. Данный класс содержит следующие поля: размерности фрагмента по осям Ox , Oy и Oz ; индекс нулевого узла фрагмента в глобальной расчетной области, указатели на смежные фрагменты расчетной сетки, а также указатели на объекты, описывающие параметры вычислителей. Вычислительный процесс представляет собой обход графа от корневого узла с параллельным запуском вычислителей, обрабатывающих узлы графа, в соответствии со значением счетчика этапов вычислений $s = k \cdot i + j$.

Для повышения эффективности расчетов фрагментов расчетной сетки, закреплённых за графическими ускорителями, разработан алгоритм и его программная реализация на языке CUDA C [20, 21].

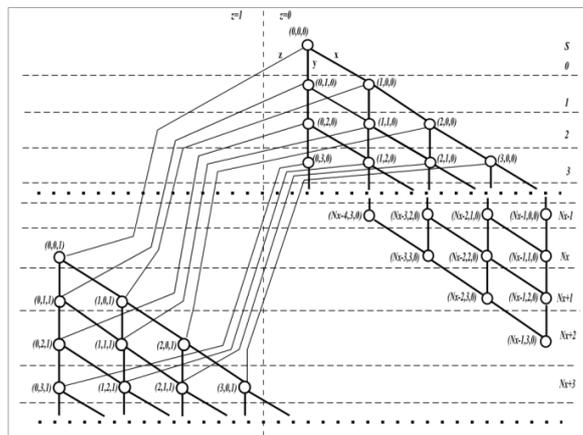


Рис. 2. Графовая модель, описывающая связи между смежными фрагментами расчетной сетки и процесс конвейерного вычисления

Приведем алгоритм поиска решения системы уравнений с нижнетреугольной матрицей (прямой ход) на CUDA C (рис. 3).

Algorithm 1 ptmKernel3(IN: $c_0, c_2, c_4, c_6, \omega$ IN/OUT: r ;)

```

1: threadX  $\leftarrow$  blockDim.x  $\cdot$  blockDim.x + threadIdx.x;
2: threadZ  $\leftarrow$  blockDim.z  $\cdot$  blockDim.z + threadIdx.z;
3: i  $\leftarrow$  threadX + 1;
4: j  $\leftarrow$  1;
5: k  $\leftarrow$  threadZ + 1;
6: for s  $\in$  [3; n1 + n2 + n3 - 3] do
7:   if (i + j + k = s)  $\wedge$  (s < i + n2 + k) then
8:     m0  $\leftarrow$  i + (blockDim.x + 1)  $\cdot$  j + n1  $\cdot$  n2  $\cdot$  k;
9:     if c0[m0] > 0 then
10:      m2  $\leftarrow$  m0 - 1;
11:      m4  $\leftarrow$  m0 - n1;
12:      m6  $\leftarrow$  m0 - n1  $\cdot$  n2;
13:      rm4  $\leftarrow$  0;
14:      if (s > 3 + threadX + threadZ) then
15:        rm4  $\leftarrow$  cache[threadX][threadZ];
16:      else
17:        rm4  $\leftarrow$  r[m4];
18:      rm2  $\leftarrow$  0;
19:      if (threadX  $\neq$  0)  $\wedge$  (s > 3 + threadX + threadZ) then
20:        rm2  $\leftarrow$  cache[threadX - 1][threadZ];
21:      else
22:        rm2  $\leftarrow$  r[m2];
23:      rm6  $\leftarrow$  0;
24:      if (threadZ  $\neq$  0)  $\wedge$  (s > 3 + threadX + threadZ) then
25:        rm6  $\leftarrow$  cache[threadX][threadZ - 1];
26:      else
27:        rm6  $\leftarrow$  r[m6];
28:      rm0  $\leftarrow$  ( $\omega \cdot$  (c2[m0]  $\cdot$  rm2 + c4[m0]  $\cdot$  rm4 + c6[m0]  $\cdot$  rm6) + r[m0]) / ((0.5  $\cdot$   $\omega$  + 1)  $\cdot$  c0[m0]);
29:      cache[threadX][threadZ]  $\leftarrow$  rm0;
30:      r[m0]  $\leftarrow$  rm0;
31:      j  $\leftarrow$  j + 1;

```

Рис. 3. Алгоритм поиска решения системы уравнений с нижнетреугольной матрицей

Входными параметрами алгоритма являются векторы коэффициентов сеточных уравнений C_0, C_2, C_4, C_6 и константа ω . Выходной параметр – вектор скорости водного потока r . Перед запуском алгоритма необходимо программно задать размерности вычислительного блока CUDA $blockDim.x, blockDim.z$ по про-

пространственным координатам x, z соответственно. Инфраструктура CUDA запускает данный алгоритм для каждой нити, при этом значения переменных $threadIdx.x, threadIdx.z, blockIdx.x, blockIdx.z$ автоматически инициализируются индексами соответствующих нитей и блоков. Глобальные индексы нитей вычисляются в строках 1 и 2.

Индекс строки i и индекс слоя k , которые обрабатывает текущий поток, вычисляются в строках 3 и 5. В строке 4 инициализируется переменная j , представляющая собой счетчик по координате y . Конвейер вычислений организован в виде цикла в строке 6. В строках 8, 10–12 вычисляются индексы центрального узла сеточного шаблона m_0 и окрестных узлов m_2, m_4, m_6 .

Двумерный массив *cache* расположен в разделяемой памяти GPU и предназначен для сохранения результатов вычислений на текущем слое по координате y . Это позволяет уменьшить количество считываний из медленной глобальной памяти и ускорить процесс вычислений на величину до 30 %.

Проведенные исследования показывают существенную зависимость времени реализации алгоритма вычисления преобразователя от соотношения нитей по пространственным координатам.

Коллективом авторов были проведены экспериментальные исследования, в которых использовались два видеоадаптера с различными характеристиками.

Первый видеоадаптер: GeForce MX 250, объём видеопамяти которого составляет 4 Гб, тактовая частота ядра 1518–1582 МГц, тактовая частота памяти 7000 МГц, разрядность шины видеопамяти – 64 бита, число ядер CUDA – 384.

Цель эксперимента – определить распределение потоков по осям Ox и Oz расчетной сетки при фиксированном значении узлов сетки по оси Oy , равным 10000 таким образом, чтобы время реализации на GPU одного шага ПТМ было минимальным.

В качестве факторов были приняты две величины: X – количество потоков по оси Ox , Z – количество потоков по оси Oz . Целевая функция: T_{GPU} – время реализации одного шага МПТМ на GPU, мс. Произведение потоков X и Z не должно превышать 1024. Такое ограничение накладывает CUDA, т.к. 1024 – это количество потоков в одном блоке. Поэтому уровни варьирования факторов X и Z были выбраны так, как показано в табл. 1.

Таблица 1

Результаты эксперимента

№ п/п	X	Z	T_{GPU} , мс
	16	64	64
	32	32	65
	64	16	81
	128	8	109
	256	4	100
	512	2	103

В результате анализа экспериментальных данных установлено:

♦ на время реализации одного шага МПТМ на GPU влияет только количество потоков по оси Oz . Время реализации одного шага МПТМ на GPU обратно пропорционально количеству узлов расчетной сетки по оси Oz , т.е. с увеличени-

ем числа узлов по оси Oz время вычисления уменьшается по логарифмическому закону. Поэтому целесообразно выполнять декомпозицию расчетной области в виде параллелепипедов, у которых размер по оси Oz максимален, а по оси Ox минимален;

- ◆ оптимальные значения X и Z , которые составили 16 и 64 соответственно.

Второй видеоадаптер: GeForce GTX 1650 имеет объем видеопамати 4 Гб, тактовые частоты ядра и памяти 1485 МГц и 8000 МГц соответственно, разрядность шины видеопамати – 128 бит. Вычислительная часть состоит из 56 текстурных процессорных кластеров (Texture Processor Cluster – TPC) по 2 мультипроцессора (SM) в каждом. Каждый мультипроцессор содержит 8 потоковых процессоров (Streaming Processor – SP) или ядер CUDA. Таким образом, число ядер CUDA для видеоадаптера GeForce GTX 1650 – 896.

Цель эксперимента – определить распределение потоков по осям Ox и Oz расчетной сетки при различных значениях ее узлов по оси Oy так, чтобы время реализации на GPU одного шага МПТМ было минимальным.

В качестве факторов приняты две величины: $k = X/Z$ – отношение количества потоков по оси Ox , (X) к количеству потоков по оси Oz , (Z), Y – количество потоков по оси Oy . Значения целевой функции: T_{GPU} – время реализации одного шага МПТМ на GPU в пересчете на 1000 узлов расчетной сетки, мс. Произведение потоков X и Z не должно превышать 640 – количества потоков в одном блоке. Поэтому уровни варьирования величин X и Z были выбраны с учетом ограничений CUDA. Так, например, количество потоков по оси Oy варьировалось в диапазоне [1000, 30000].

Анализ экспериментальных данных для значений факторов $X = 1, Z = 640$ и $X = 640, Z = 1$ показал, что при вычислении целевой функции в указанных точках не используется распределяемая память, поэтому при проведении регрессионного анализа эти точки были исключены.

В результате анализа экспериментальных данных установлено:

- ◆ при $k < 10$ и $Y < 1000$ скорость вычислений замедляется, т.к. в этом случае неэффективно используется распределяемая память графического ускорителя;
- ◆ наименьшее время реализации одного шага МПТМ в пересчете на 1000 узлов расчетной сетки на видеоадаптере GeForce GTX 1650 будет получено при наибольшем количестве потоков по оси Oy и наибольшем значении коэффициента k ;
- ◆ время реализации одного шага МПТМ в пересчете на 1000 узлов расчетной сетки на видеоадаптере GeForce GTX 1650 обратно пропорционально количеству узлов расчетной сетки по оси Oz , т.е. с увеличением числа узлов по оси Oz время вычисления уменьшается;
- ◆ наибольшее значение коэффициента k достигается при увеличении количества потоков по оси Ox и уменьшении количества потоков по оси Oz . Поэтому целесообразно выполнять декомпозицию расчетной области в виде параллелепипедов, у которых размер по оси Oz минимален, а по оси Ox максимален.

Заключение. В результате проведенных исследований разработаны алгоритм и программный модуль, его реализующий, предназначенные для решения СЛАУ (самосопряженный и несамосопряженный случаи), возникающей при дискретизации задачи гидродинамики мелководного водоема, МПТМ с использованием технологии NVIDIA CUDA. Описан способ декомпозиции расчетной сетки, применимый для гетерогенных вычислительных систем. Предложена графовая модель,

позволяющая организовать параллельный конвейерный вычислительный процесс на GPU, предназначенный для решения систем сеточных уравнений большой размерности.

Для каждого из двух видеоадаптеров с различными характеристиками были проведены экспериментальные исследования для определения оптимальной двумерной конфигурации потоков в вычислительном блоке, реализуемом на одном потоковом мультипроцессоре, при которой время реализации на GPU одного шага МПТМ является минимальным. Проведенные исследования показали, что выбор способа декомпозиции расчетной области в виде параллелепипедов необходимо выполнять с учетом архитектуры видеоадаптера.

Разработанный программный инструментарий позволяет более эффективно задействовать вычислительные ресурсы GPU, используемой для решения вычислительно-трудоемких пространственно-трехмерных задач гидрофизики.

Благодарности. Исследование выполнено при финансовой поддержке РФФИ в рамках научного проекта № 19-07-00623.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. *Vassilevski Y., Terekhov K., Nikitin K., Kapyrin I.* Parallel Finite Volume Computation on General Meshes. – Springer International Publishing, 2020. – 186 p.
2. *Дымников В.П., Тыртышиников Е.Е., Лысков В.Н., Залесный В.Б.* Математическое моделирование климата, динамики атмосферы и океана: К 95-летию Г.И. Марчука и 40-летию ИВМ РАН // Известия РАН. Физика атмосферы и океана. – 2020. – Т. 56, № 3. – С. 251-254.
3. *Четверушкин Б.Н., Мингалев И.В., Чететкин В.М., Орлов К.Г., Федотова Е.А., Мингалев В.С., Мингалев О.В.* Модели общей циркуляции атмосферы Земли. Достижения и направления развития // Математическое моделирование. – 2020. – Т. 32, № 11. – С. 29-46.
4. *Головизнин В.М., Четверушкин Б.Н.* Алгоритмы нового поколения в вычислительной гидродинамике // Журнал вычислительной математики и математической физики. – 2018. – Т. 58, № 8. – С. 20-29.
5. *Матишов Г.Г., Гаргона Ю.М., Бердников С.В., Дженюк С.Л.* Закономерности экосистемных процессов в Азовском море. – М.: Наука, 2006. – 304 с.
6. *Bonaduce A., Staneva J., Grayek S., Bidlot J.-R., Breivik O.* Sea-state contributions to sea-level variability in the European Seas // Ocean Dynamics. – 2020. – 70 (12). – P. 1547-1569. – DOI: 10.1007/s10236-020-01404-1.
7. *Marchesiello P., McWilliams J.C., Shchepetkin A.* Open boundary conditions for long-term integration of regional oceanic models // Oceanic Modelling Journal. – Netherlands: Elsevier BV. – 2001. – Vol. 3, No. 1-2. – P. 1-20. – DOI: 10.1016/s1463-5003(00)00013-5.
8. *Андросов А.А., Вольцингер Н.Е.* Проливы мирового океана. Общий подход к моделированию. – М.: Наука, 2005. – 172 с.
9. *Nieuwstadt F., Westerweel J., Boersma B.J.* Turbulence. Introduction to Theory and Applications of Turbulent Flows. – Springer, 2016. – 288 p.
10. *Воеводин В.В., Воеводин Вл.В.* Параллельные вычисления. – СПб.: БХВ-Петербург, 2002. – 608 с.
11. *Xue W., Roy C.J.* Multi-GPU performance optimization of a computational fluid dynamics code using OpenACC // Concurrency and Computation Practice and Experience. – 2020. – 33 (4). – DOI: 10.1002/cpe.6036.
12. *Xue W., Jackson C.W., Xue W., Roy C.J.* Multi-CPU/GPU Parallelization, Optimization and Machine Learning based autotuning of Structured Grid CFD Codes // AIAA Aerospace Sciences Meeting. – 2018. – P. 0362. – DOI: 10.2514/6.2018-0362.
13. *Nagatake T., Kunugi T.* Application of GPU to Computational Multiphase Fluid Dynamics // IOP Conference Series: Materials Science and Engineering. – 2010. – Vol. 10 (1). – P. 012024. – DOI: 10.1088/1757-899X/10/1/012024.

14. Munk D.J., Kipouros T., Vio G.A. Multi-physics bi-directional evolutionary topology optimization on GPU-architecture // *Engineering with Computers*. – 2019. – Vol. 35 (4). – P. 1059-1079. – DOI: 10.1007/s00366-018-0651-1.
15. Сухинов А.И., Атаян А.М., Белова Ю.В., Литвинов В.Н., Никитина А.В., Чистяков А.Е. Обработка данных натурных измерений экспедиционных исследований для математического моделирования гидродинамических процессов Азовского моря // *Вычислительная механика сплошных сред*. – 2020. – Т. 13, № 2. – С. 161-174.
16. Sukhinov A.I., Chistyakov A.E., Shishenya A.V., Timofeeva E.F. Predictive Modeling of Coastal Hydrophysical Processes in Multiple-Processor Systems Based on Explicit Schemes // *Mathematical Models and Computer Simulations*. – 2018. – 10 (5). – P. 648-658. – DOI: 10.1134/S2070048218050125.
17. Коновалов А.Н. Метод скорейшего спуска с адаптивным попеременнотреугольным переусловливателем // *Дифференциальные уравнения*. – 2004. – Т. 40, № 7. – С. 953-963.
18. Сухинов А.И., Чистяков А.Е., Литвинов В.Н., Никитина А.В., Белова Ю.В., Филина А.А. Вычислительные аспекты математического моделирования гидробиологических процессов в мелководном водоеме // *Вычислительные методы и программирование*. – 2020. – Т. 21, № 4. – С. 452-469. – DOI: <https://doi.org/10.26089/NumMet.v21r436>.
19. Самарский А.А., Вабищевич П.Н. Численные методы решения задач конвекции-диффузии. Изд. стереотип. – М.: Книжный дом «ЛИБРОКОМ», 2015. – 248 с.
20. Oyarzun G., Borrell R., Gorobets A., Oliva A. MPI-CUDA sparse matrix-vector multiplication for the conjugate gradient method with an approximate inverse preconditioner // *Computers and Fluids*. – 2014. – Vol. 92. – P. 244-252. – DOI: 10.1016/j.compfluid.2013.10.035.
21. Zheng Liang, Gerya Taras, Knepley Matthew, Yuen David, Zhang Huai, Shi Yaolin. Implementation of a multigrid solver on a GPU for Stokes equations with strongly variable viscosity based on Matlab and CUDA // *International Journal of High Performance Computing Applications*. – 2014. – 28 (1). – P. 50-60. – DOI: 10.1007/978-3-642-16405-7_21.

REFERENCES

1. Vassilevski Y., Terekhov K., Nikitin K., Kapryin I. *Parallel Finite Volume Computation on General Meshes*. Springer International Publishing, 2020, 186 p.
2. Dymnikov V.P., Tyrtshnikov E.E., Lykosov V.N., Zalesnyy V.B. Matematicheskoe modelirovanie klimata, dinamiki atmosfery i okeana: K 95-letiyu G.I. Marchuka i 40-letiyu IVM RAN [Mathematical modeling of climate, atmospheric and ocean dynamics: To the 95th anniversary of G.I. Marchuk and the 40th anniversary of the IVM RAS], *Izvestiya RAN. Fizika atmosfery i okeana* [News of the Russian Academy of Sciences. Physics of the atmosphere and ocean], 2020, Vol. 56, No. 3, pp. 251-254.
3. Chetverushkin B.N., Mingalev I.V., Chechetkin V.M., Orlov K.G., Fedotova E.A., Mingalev V.S., Mingalev O.V. Modeli obshchey tsirkulyatsii atmosfery Zemli. Dostizheniya i napravleniya razvitiya [Models of the general circulation of the Earth's atmosphere. Achievements and development goals], *Matematicheskoe modelirovanie* [Mathematical modeling], 2020, Vol. 32, No. 11, pp. 29-46.
4. Goloviznin V.M., Chetverushkin B.N. Algoritmy novogo pokoleniya v vychislitel'noy gidrodinamike [New generation algorithms in computational fluid dynamics], *Zhurnal vychislitel'noy matematiki i matematicheskoy fiziki* [Journal of Computational Mathematics and Mathematical Physics], 2018, Vol. 58, No. 8, pp. 20-29.
5. Matishov G.G., Gargopa Yu.M., Berdnikov S.V., Dzhenyuk S.L. Zakonomernosti ekosistemnykh protsessov v Azovskom more [Regularities of ecosystem processes in the Sea of Azov]. Moscow: Nauka, 2006, 304 p.
6. Bonaduce A., Staneva J., Grayek S., Bidlot J.-R., Breivik O. Sea-state contributions to sea-level variability in the European Seas, *Ocean Dynamics*, 2020, 70 (12), pp. 1547-1569. DOI: 10.1007/s10236-020-01404-1.
7. Marchesiello P., McWilliams J.C., Shchepetkin A. Open boundary conditions for long-term integration of regional oceanic models, *Oceanic Modelling Journal*. Netherlands: Elsevier BV, 2001, Vol. 3, No. 1-2, pp. 1-20. DOI: 10.1016/s1463-5003(00)00013-5.
8. Androssov A.A., Vol'tsinger N.E. Prolivy mirovogo okeana. Obshchiy podkhod k modelirovaniyu [Straits of the world ocean. General approach to modeling]. Moscow: Nauka, 2005, 172 p.

9. *Nieuwstadt F., Westerweel J., Boersma B.J.* Turbulence. Introduction to Theory and Applications of Turbulent Flows. Springer, 2016, 288 p.
10. *Voevodin V.V., Voevodin V.V.* Parallelnye vychisleniya [Parallel computing]. Saint Petersburg: BKhV-Peterburg, 2002, 608 p.
11. *Xue W., Roy C.J.* Multi-GPU performance optimization of a computational fluid dynamics code using OpenACC, *Concurrency and Computation Practice and Experience*, 2020, 33 (4). DOI: 10.1002/cpe.6036.
12. *Xue W., Jackson C.W., Xue W., Roy C.J.* Multi-CPU/GPU Parallelization, Optimization and Machine Learning based autotuning of Structured Grid CFD Codes, *AIAA Aerospace Sciences Meeting*, 2018, pp. 0362. DOI: 10.2514/6.2018-0362.
13. *Nagatake T., Kunugi T.* Application of GPU to Computational Multiphase Fluid Dynamics, *IOP Conference Series: Materials Science and Engineering*, 2010, Vol. 10 (1), pp. 012024. DOI: 10.1088/1757-899X/10/1/012024.
14. *Munk D.J., Kipouros T., Vio G.A.* Multi-physics bi-directional evolutionary topology optimization on GPU-architecture, *Engineering with Computers*, 2019, Vol. 35 (4), pp. 1059-1079. DOI: 10.1007/s00366-018-0651-1.
15. *Sukhinov A.I., Atayan A.M., Belova Yu.V., Litvinov V.N., Nikitina A.V., Chistyakov A.E.* Obrabotka dannykh naturnykh izmereniy ekspeditsionnykh issledovaniy dlya matematicheskogo modelirovaniya gidrodinamicheskikh protsessov Azovskogo morya [Processing of data of field measurements of expeditionary research for mathematical modeling of hydrodynamic processes of the Sea of Azov], *Vychislitel'naya mekhanika sploshnykh sred* [Computational mechanics of continuous media], 2020, Vol. 13, No. 2, pp. 161-174.
16. *Sukhinov A.I., Chistyakov A.E., Shishenya A.V., Timofeeva E.F.* Predictive Modeling of Coastal Hydrophysical Processes in Multiple-Processor Systems Based on Explicit Schemes, *Mathematical Models and Computer Simulations*, 2018, 10 (5), pp. 648-658. DOI: 10.1134/S2070048218050125.
17. *Konovalov A.N.* Metod skoreyshego spuska s adaptivnym poperemennotreugol'nym pereobuslovlivatelyem [The method of rapid descent with an adaptive alternating triangular re-conditionalist], *Differentsial'nye uravneniya* [Differential equations], 2004, Vol. 40, No. 7, pp. 953-963.
18. *Sukhinov A.I., Chistyakov A.E., Litvinov V.N., Nikitina A.V., Belova Yu.V., Filina A.A.* Vychislitel'nye aspekty matematicheskogo modelirovaniya gidrobiologicheskikh protsessov v melkovodnom vodoeme [Computational aspects of mathematical modeling of hydrobiological processes in a shallow reservoir], *Vychislitel'nye metody i programmirovaniye* [Computational methods and programming], 2020, Vol. 21, No. 4, pp. 452-469. DOI: <https://doi.org/10.26089/NumMet.v21r436>.
19. *Samarskiy A.A., Vabishchevich P.N.* Chislennyye metody resheniya zadach konveksii-diffuzii [Numerical methods for solving convection-diffusion problems. Stereotype]. Moscow: Knizhnyy dom «LIBROKOM», 2015, 248 p.
20. *Oyarzun G., Borrell R., Gorobets A., Oliva A.* MPI-CUDA sparse matrix–vector multiplication for the conjugate gradient method with an approximate inverse preconditioner, *Computers and Fluids*, 2014, Vol. 92, pp. 244-252. DOI: 10.1016/j.compfluid.2013.10.035.
21. *Zheng Liang, Gerya Taras, Knepley Matthew, Yuen David, Zhang Huai, Shi Yaolin.* Implementation of a multigrid solver on a GPU for Stokes equations with strongly variable viscosity based on Matlab and CUDA, *International Journal of High Performance Computing Applications*, 2014, 28 (1), pp. 50-60. DOI: 10.1007/978-3-642-16405-7_21.

Статью рекомендовал к опубликованию д.ф.-м.н. И.М. Губайдуллин.

Грачева Наталья Николаевна – Азово-Черноморский инженерный институт ФГБОУ ВО Донской ГАУ; e-mail: nata.grachewa2011@yandex.ru; г. Зерноград, Россия; тел.: 89081908999; кафедра математики и биоинформатики; к.т.н.; доцент.

Литвинов Владимир Николаевич – e-mail: litvinovvn@rambler.ru; тел.: 89185172138; кафедра математики и биоинформатики; к.т.н.; доцент.

Руденко Нелли Борисовна – e-mail: nelli-rud@yandex.ru; тел.: 89081908999; кафедра математики и биоинформатики; к.т.н.; доцент.

Никитина Алла Валерьевна – Южный федеральный университет, e-mail: nikitina.vm@gmail.com; г. Таганрог, Россия; тел.: 89896260069; кафедра интеллектуальных и многопроцессорных систем; д.т.н.; профессор.

Чистяков Александр Евгеньевич – Донской государственный технический университет; e-mail: cheese_05@mail.ru; г. Ростов-на-Дону, Россия; тел.: 89508593872; кафедра математики и информатики; д.т.н.

Gracheva Natalia Nikolaevna – Azov-Black Sea Engineering Institute of the Don State Agrarian University; e-mail: nata.grachewa2011@yandex.ru; Zernograd, Russia; phone: +79081908999; the department of mathematics and bioinformatics; cand. of eng. sc.; associate professor.

Litvinov Vladimir Nikolaevich – e-mail: litvinovvn@rambler.ru; phone: +79185172138; the department of mathematics and bioinformatics; cand. of eng. sc.; associate professor.

Rudenko Nelly Borisovna – e-mail: nelli-rud@yandex.ru; phone: +79081908999; the department of mathematics and bioinformatics; cand. of eng. sc.; associate professor.

Nikitina Alla Valerievna – Southern Federal University; e-mail: nikitina.vm@gmail.com; Taganrog, Russia; phone: +79896260069; the department of Intelligent and multiprocessor systems; dr. of eng. sc.; professor.

Chistyakov Alexander Evgen'evich – Don State Technical University; e-mail: cheese_05@mail.ru; Rostov-on-Don, Russia; phone: +79508593872; the department of mathematics and informatics; dr. of eng. sc.; professor.

УДК 004.75

DOI 10.18522/2311-3103-2021-5-105-119

И.Б. Сафроненкова, А.Б. Клименко

ОЦЕНКА ВЛИЯНИЯ ИЗМЕНЯЮЩИХСЯ УСЛОВИЙ ОКРУЖАЮЩЕЙ СРЕДЫ НА РАСПРЕДЕЛЕНИЕ ВЫЧИСЛИТЕЛЬНОЙ НАГРУЗКИ В ГРУППЕ БПЛА*

Рассмотрена проблема распределения вычислительной нагрузки в группе беспилотных летательных аппаратов (БПЛА) при осуществлении мониторинга некоторой области в изменяющихся условиях внешней среды, которая оказывает непосредственное влияние на потребление бортового энергоресурса. Описан один из этапов решения задачи мониторинга, осуществляемого гетерогенной группой БПЛА, заключающийся в распределении БПЛА по полосам сканирования. Отмечено, что при выполнении данного этапа, отсутствует возможность учета факторов влияния окружающей среды, что важно ввиду ограниченности бортовых энергоресурсов. В связи с этим, весьма вероятна ситуация, когда БПЛА не в состоянии выполнить назначенную на него подзадачу, что ставит под угрозу выполнения всей миссии группы. Во избежание данной ситуации, предложено использовать методику принятия решения о необходимости перераспределения нагрузки в группе мобильных роботов (МР). В основе принятия решения лежит процедура онтологического анализа, позволяющая ограничить число вариантов для переноса нагрузки. Разработана модель онтологии распределения вычислительной нагрузки в группе БПЛА, учитывающая возможность привлечения дополнительной производительности либо за счет ресурсов соседних БПЛА, либо за счет устройств «туманного» слоя. Приведены примеры продукционных правил, на основе которых принимается решение о необходимости переноса нагрузки. Показано, что при увеличении числа изменений условий окружающей среды, время использования дополнительных вычислительных ресурсов уменьшается, что, приводит к необходимости привлечения их большего объема для выполнения поставленной задачи. Проведена сравнительная оценка объема привлекаемых ресурсов при реализации двух методов-аналогов решения

* Исследование выполнено при поддержке гранта РФФИ №20-04-60485 и в рамках реализации ГЗ ЮНЦ РАН АААА-А19-119011190173-6.