

УДК 532.529

РЕШЕНИЕ ЗАДАЧ ГАЗОВОЙ ДИНАМИКИ И ТЕПЛООБМЕНА НА ГРАФИЧЕСКИХ ПРОЦЕССОРАХ

К. Н. Волков, Ю. Н. Дерюгин, В. Н. Емельянов, А. Г. Карпенко,
А. С. Козелков, И. В. Тетерина, А. В. Ялозо
(Балтийский ГТУ, СПбГУ, г. Санкт-Петербург; РФЯЦ-ВНИИЭФ, г. Саров)

Обсуждаются возможности использования графических процессоров общего назначения для моделирования течений вязкого сжимаемого газа и теплообмена. Для дискретизации уравнений Навье—Стокса используется метод конечных объемов на неструктурированных сетках, реализованный в рамках пакета ЛОГОС. Для программной реализации параллельных вычислительных алгоритмов применяется технология CUDA. Приводятся решения ряда модельных задач газовой динамики и теплообмена на графических процессорах и обсуждаются подходы к оптимизации программного кода, связанные с использованием различных типов памяти. Сравняется ускорение счета на графических процессорах по отношению к расчетам на центральном процессоре при использовании сеток разных размеров и различных вычислительных ресурсов.

Ключевые слова: графический процессор, параллельный алгоритм, газовая динамика, метод конечных объемов, неструктурированная сетка, CUDA, пакет ЛОГОС.

Введение

При моделировании течений жидкости и газа, конвективного теплообмена в технических и технологических приложениях широкое применение находят методы вычислительной газовой динамики (CFD — Computational Fluid Dynamics). К настоящему времени накоплен обширный фонд вычислительных алгоритмов, предназначенных для численного моделирования течений жидкости и газа и реализованных в виде универсальных коммерческих CFD-пакетов (например ANSYS Fluent и ANSYS CFX) и CFD-пакетов с открытым исходным кодом (например OpenFOAM). Имеется также ряд пакетов, развиваемых и поддерживаемых отдельными научными коллективами, например, VP2/3 [1], SINF [2], NOISETTE [3] и др., обладающих той или иной степенью универсальности в своей предметной области. К универсальным CFD-пакетам относится пакет ЛОГОС [4], разработанный в РФЯЦ-ВНИИЭФ и предназначенный для решения широкого круга задач газовой динамики и теплообмена на компьютерах с массовым параллелизмом.

Смешанный тип уравнений вычислительной гидродинамики и разномасштабность физических процессов влекут за собой определенные требования к алгоритмам дискретизации и методам математического моделирования.

Течения вязкого несжимаемого газа описываются алгоритмами, основанными на процедуре коррекции давления или принципе расщепления неизвестных, общей чертой которых являются формулировка разностной схемы относительно приращений неизвестных и решение уравнения Пуассона для поправки давления [5]. Для моделирования течений сжимаемого газа используются методики, в основе которых лежит решение полной системы уравнений Навье—Стокса [6]. В этих методиках предъявляются особые требования к алгоритмам расчета потоков и алгоритмам решения систем линейных алгебраических уравнений.

Необходимость обоснования проектных решений в режиме реального времени, оценки погрешности расчета и внедрение методов анализа неопределенностей в инженерную практику приводят к

увеличению количества расчетов. Многие пакеты допускают эксплуатацию не только на персональных компьютерах и рабочих станциях (CPU — Central Processor Unit), но и на многопроцессорных вычислительных системах.

Дополнительные возможности CFD-пакетов включают поддержку параллельных вычислений на графических процессорах (GPU — Graphics Processor Unit) и их кластерах, а также на гибридных вычислительных системах, состоящих из центральных и графических процессоров. Область использования GPU расширяется благодаря появлению новых технологий программирования и усовершенствованию аппаратного обеспечения. Основным критерием эффективности применения GPU является скорость вычислений и ускорение решения задачи на GPU по сравнению с CPU.

Некоторые возможности использования графических процессоров общего назначения для численного решения задач динамики вязкой несжимаемой жидкости и сжимаемого газа обсуждаются в работах [7, 8].

Решение двух- и трехмерных уравнений Эйлера рассматривается в работе [9]. Для программной реализации используется язык BrookGPU (видеокарта ATI 1950XT) и технология CUDA (видеокарта NVIDIA GeForce 8800 GTX). При использовании языка BrookGPU в случае двумерных уравнений достигается ускорение примерно в 29 раз относительно расчетов на CPU, а для трехмерных — примерно в 3 раза. Применение технологии CUDA для распараллеливания *трехмерного* кода позволяет увеличить производительность вычислений до 16 раз за счет эффективного использования различных типов памяти.

На неструктурированной сетке с ячейками в виде тетраэдров трехмерные уравнения Эйлера решаются в работах [10, 11]. Возможности разных подходов демонстрируются на примере моделирования сверхзвукового обтекания профиля NASA0012 при нулевом угле атаки. Ускорение вычислений на GPU составляет $\sim 9,5$ раза относительно ускорения, достигнутого при использовании *параллельного* кода на четырехъядерном процессоре (применяется технология OpenMP), и ~ 33 раза относительно расчетов на основе последовательной версии кода. Расчеты проводятся на платформе NVIDIA Tesla C2050 и центральном процессоре Intel Core 2 Q9450.

Расчеты в трехмерной постановке на блочно-структурированных сетках рассматриваются в работе [12] применительно к моделированию турбулентных течений в межлопаточных каналах газовых турбин на основе метода конечных объемов и модели турбулентности Прандтля. Для ускорения сходимости к стационарному решению используются локальный шаг по времени и многосеточный метод. Расчеты с использованием явной схемы второго порядка по времени и различного числа узлов GPU-кластера (максимальное число узлов 16) показывают практически линейный прирост ускорения при увеличении числа узлов. Относительно расчетов на двухъядерном CPU (используется видеокарта NVIDIA GT200) достигается ускорение примерно в 19 раз.

Уравнения Навье—Стокса для течений сжимаемой среды, замкнутые при помощи модели турбулентности Спаларта—Аллмареса, решаются в работе [13] на двух- и трехмерных неструктурированных сетках (ячейки сетки представляют собой треугольники и тетраэдры). Дискретизация по времени проводится при помощи неявной схемы. Для дискретизации невязких потоков применяется схема Роу. Дискретизация вязких потоков основывается на использовании примитивных переменных и их градиентов. Расчеты проводятся с одинарной (с ускорением в 28 раз), двойной (с ускорением в 19,6 раза) и смешанной (с ускорением в 25,2 раза) точностью. Для расчетов используется видеокарта NVIDIA GeForce GTX 280 и технология CUDA. Последовательный код выполняется на процессоре Intel Core 2 Duo с тактовой частотой 2,8 ГГц.

Возможности применения GPU имеются, в частности, при решении некоторого класса задач в пакете ANSYS (графические ускорители используются для решения систем разностных уравнений, порожденных конечно-элементной дискретизацией уравнений механики деформируемого твердого тела, но не для операций их сборки или обработки результатов расчетов), а также в ряде других узкоспециализированных пакетов и кодов, допускающих использование блочно-структурированных и неструктурированных сеток [7, 8, 10—13], что позволяет производить расчеты в областях сложной геометрической конфигурации.

В данной работе рассматривается реализация метода конечных объемов, предназначенного для моделирования нестационарных течений вязкого сжимаемого газа и теплообмена на неструктурированных сетках, при использовании графических процессоров общего назначения. Реализованные

средства распараллеливания вычислений на графических процессорах дополняют новыми возможностями вышеуказанный пакет программ ЛОГОС, что обеспечивается благодаря модульной структуре этого пакета. Обсуждается решение ряда модельных задач газовой динамики, имеющиеся возможности по оптимизации программного кода, а также ускорение вычислительной процедуры, полученное на сетках разных размеров, по сравнению с расчетами на центральном процессоре.

Метод конечных объемов

Для дискретизации уравнений Навье—Стокса на неструктурированной сетке используется метод конечных объемов.

Основные уравнения. В декартовой системе координат (x, y, z) нестационарное течение вязкого сжимаемого газа описывается следующим уравнением, записанным в консервативных переменных¹:

$$\frac{\partial Q}{\partial t} + \frac{\partial F_i}{\partial x_i} = 0.$$

Уравнение состояния совершенного газа имеет вид

$$p = (\gamma - 1)\rho \left[e - \frac{1}{2} (v_x^2 + v_y^2 + v_z^2) \right].$$

Вектор консервативных переменных Q и вектор потока F_i имеют следующий вид:

$$Q = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \rho e \end{pmatrix}; \quad F_i = \begin{pmatrix} \rho v_i \\ \rho v_i v_x - \tau_{ix} + p\delta_{ix} \\ \rho v_i v_y - \tau_{iy} + p\delta_{iy} \\ \rho v_i v_z - \tau_{iz} + p\delta_{iz} \\ (\rho e + p)v_i - \tau_{ix}v_x - \tau_{iy}v_y - \tau_{iz}v_z + q_i \end{pmatrix}.$$

Здесь t — время; ρ — плотность; v_x, v_y, v_z — составляющие скорости в координатных направлениях x, y и z ; e — полная энергия единицы массы; p — давление; γ — отношение удельных теплоемкостей; δ_{ij} — символ Кронекера. Компоненты тензора *вязких* напряжений и вектора теплового потока находятся из соотношений

$$\tau_{ij} = \mu \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} - \frac{2}{3} \frac{\partial v_k}{\partial x_k} \delta_{ij} \right); \quad q_i = -\lambda \frac{\partial T}{\partial x_i},$$

где T — температура. Вязкость зависит от температуры и определяется из закона Сазерленда. Теплопроводность выражается через вязкость и число Прандтля, которому присваивается постоянное значение.

При моделировании турбулентных течений молекулярные коэффициенты переноса заменяются их эффективными значениями [14]. В качестве модели турбулентности используется модель пути смешения Прандтля с демпфирующей функцией Ван Дрифта [15]. Выбор модели турбулентности Прандтля обуславливается простотой и удобством ее программной реализации в рамках существующего кода.

Особенности дискретизации. Для дискретизации по времени применяется явный метод Эйлера или явный двухшаговый метод Рунге—Кутты (модифицированный метод Эйлера). Расчет потоков основывается на решении задачи о распаде произвольного разрыва (применяются метод

¹Здесь и далее по повторяющемуся в одночленных выражениях индексу предполагается суммирование.

Годунова и схема Роу с энтропийной поправкой Хартена). Для повышения порядка точности по пространству используется кусочно-непрерывное распределение искомых функций внутри контрольного объема² и производится интерполяция решения на грань контрольного объема. В разностную схему вводятся переключатели (minmod, van Leer, SuperBee), которые осуществляют переход на схему первого порядка в тех областях, где существуют большие градиенты решения, и на схему второго порядка в области гладкого решения. Для расчета градиентов в центрах граней контрольных объемов используется линейная интерполяция градиентов на грани [16]. Особенности реализации вычислительной процедуры обсуждаются в работах [7, 8].

Применение модели турбулентности с демпфирующей функцией Ван Дрифта требует расчета пристеночной координаты. При использовании неструктурированной сетки для каждой грани имеется информация о соседних ячейках, поэтому напряжения трения на стенке рассчитываются заранее. В разработанном подходе применяется двойная процедура интерполяции. Параметры потока интерполируются с неструктурированной сетки на вспомогательную структурированную, на которой производится расчет турбулентной вязкости. Рассчитанные значения турбулентной вязкости затем переносятся обратно на исходную сетку.

Программная реализация

Для GPU требуется разработка алгоритмов, имеющих высокую степень параллелизма на уровне данных. При этом одна операция выполняется над всеми элементами массива данных (под элементом массива понимается структура данных или несколько чисел, хранящихся во внешней памяти).

Схема решения задачи (рис. 1). Формирование контрольных объемов и расчет их геометрических характеристик производятся на CPU. На GPU выполняются операции, связанные с вычислением потоков и расчетом шага интегрирования по времени.

Ядро 1 рассчитывает потоки через грани контрольных объемов (расчеты подобного типа основываются на использовании двух узлов i и j , соединенных гранью). Расчет локального шага интегрирования по времени выполняется в ядре 2 для каждого контрольного объема. Поиск минимального

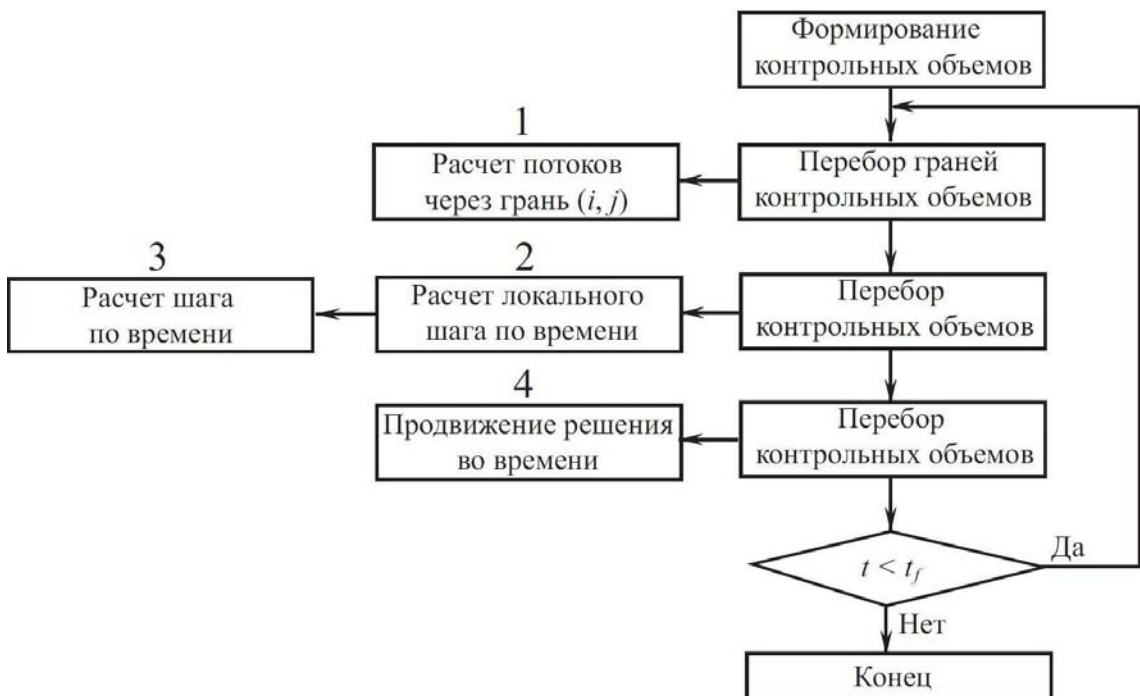


Рис. 1. Схема решения задачи

²В данном случае контрольный объем отождествляется с ячейкой.

шага по времени осуществляется в ядре 3, которое производит перебор всех контрольных объемов сетки. Продвижение решения во времени выполняет ядро 4. Ядра 1, 2 и 4 обладают высокой степенью параллелизма, а ядро 3 требует коммуникаций между блоками, снижая общую производительность вычислений. Расчет проводится до достижения заданного момента времени.

Декомпозиция. При решении задач газовой динамики на многопроцессорных вычислительных системах используется принцип геометрического параллелизма, который предполагает геометрическую декомпозицию расчетной области на подобласти, содержащие приблизительно равное число узлов сетки. На практике находят применение различные подходы к декомпозиции, предполагающие разбиение расчетной области вдоль одного, двух или трех координатных направлений. При использовании различных методов декомпозиции эффективность вычислительной процедуры зависит от схемы обменов и числа процессоров.

Явная схема обладает высокой степенью параллелизма, поскольку значения на новом временном слое рассчитываются в каждом узле независимо друг от друга. При этом отпадает необходимость в использовании циклических конструкций. Каждому узлу сетки соответствует своя *нить* GPU, которая производит расчеты параллельно с другими нитями.

При расчете течений совершенного газа каждому контрольному объему соответствует 5 значений физических переменных (плотность, 3 компоненты скорости, давление). На неструктурированной сетке данные обычно развертываются в линейный массив путем записи n_e блоков по 5 значений (под n_e понимается число контрольных объемов), а адресация переменной j для контрольного объема i имеет вид $u[5i + j]$. Для архитектуры GPU эффективнее размещать данные в 5 блоках по n_e значений, что дает адресацию $u[i + jn_e]$, которая соответствует модели объединения запросов (coalescing).

Расчет потоков. С точки зрения программной реализации вычислительная процедура представляет собой множество однотипных заданий по расчету потоков через грани контрольных объемов. Для вычисления потока через внутренние грани используются газодинамические переменные в контрольных объемах, которым эта грань принадлежит, а также геометрические параметры граней (площади, нормали).

Для расчета потоков по методу Годунова или по схеме Роу на каждой грани контрольного объема решается задача о распаде произвольного разрыва. В подходе Годунова для ее решения используется итерационный метод (например, метод Ньютона), скорость сходимости которого меняется от грани к грани. При сходимости итерационного процесса с разной скоростью в разных нитях одного *варпа* (warp) время выполнения варпа определяется временем выполнения самой медленной нити (грань с наиболее медленной скоростью сходимости итераций). Увеличение числа итераций не требует дополнительных операций чтения из глобальной памяти, а все вычисления производятся с данными из регистровой памяти, поэтому разная скорость сходимости метода Ньютона для различных граней оказывает слабое влияние на общее время счета.

Поток для контрольного объема определяется суммированием потоков через грани, рассчитанных по методу Годунова или по схеме Роу, и сохраняется в массиве потоков для ячеек, чтобы затем использоваться для расчета газодинамических переменных на новом слое по времени.

Одна из проблем, возникающих при реализации параллельных вычислений на GPU, связана с пересечением нитей при суммировании потоков [17]. При параллельном вычислении потоков через грани между контрольным объемом i и соседними объемами j и k возникает ошибка вследствие одновременного суммирования потоков для контрольного объема i двумя разными нитями.

Для исключения ошибки, возникающей в результате обработки различными нитями одного контрольного объема, при вычислении потоков через грани контрольных объемов используются различные подходы (рис. 2):

1. Последовательная обработка граничных элементов (см. рис. 2, *a*). Каждая нить обрабатывает только свои внутренние элементы. Обработка граничных элементов производится одной нитью после перебора внутренних элементов.
2. Повторение вычислений (см. рис. 2, *b*). Каждая нить производит обработку как внутренних

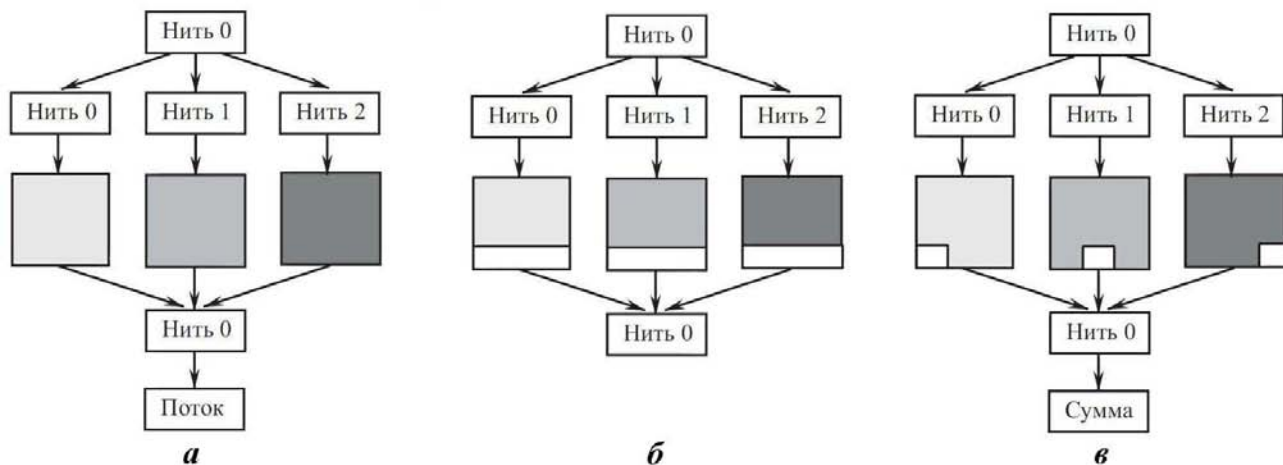


Рис. 2. Различные способы расчета потоков: *a* — с последовательной обработкой граничных элементов; *b* — с повторением вычислений; *v* — с репликацией данных

элементов, так и граничных, имеющих узлы, принадлежащие нити. Каждая нить осуществляет запись только для своих узлов.

3. Репликация данных (см. рис. 2, *v*). Производится распределение граничных элементов между нитями. Каждая нить сохраняет результаты расчетов в своем отдельном массиве данных по узлам. Суммирование результатов производится отдельной нитью.
4. Потактовое вычисление. Элементы делятся на группы, не имеющие пересечений. Каждая группа обрабатывается отдельной нитью.
5. Использование фиктивных ячеек. К границам области добавляются фиктивные ячейки, параметры в которых рассчитываются таким образом, чтобы удовлетворить граничным условиям при расчете потоков через грани.

В подходах 1—4 для вычисления потоков через граничные грани помимо геометрических параметров используются газодинамические переменные только в одном контрольном объеме. Поскольку газодинамические переменные полагаются постоянными на шаге интегрирования по времени, задания по вычислению потоков являются независимыми по входным данным и допускают параллельную обработку.

При вычислении потоков с репликацией данных (метод 3) для исключения конфликтов по доступу к памяти на этапе вычисления потоков создается дополнительный массив для записи потоков по граням ячеек, что несколько увеличивает потребление оперативной памяти [17]. Для каждой грани записывается два набора из 5 значений (плотность, 3 компоненты скорости и давление). После вычисления потоков через грани производится их суммирование, чтобы получить потоки для контрольных объемов. Для этого необходимо хранить в памяти дуальный граф связей контрольных объемов (узлы графа — ячейки сетки, ребра — связи ячеек через общие грани), который содержит $2n_s + n_e + 1$ целочисленных значений, где n_e , n_s — число контрольных объемов и их внутренних граней. Преимущество данного подхода состоит в том, что все задания по обработке внутренних граней контрольных объемов запускаются одновременно.

При потактовом вычислении потоков (метод 4) к дуальному графу сетки применяется алгоритм раскраски ребер. В множестве ребер графа выделяется подмножество, так чтобы в нем каждый узел встречался не более одного раза. Обработка запущенными на GPU нитями такого подмножества ребер происходит параллельно и исключает возможность конфликтов нитей по модификации одних и тех же ячеек памяти. Процедура вычисления потоков разбивается на такты, на каждом из которых обрабатывается одно из выделенных подмножеств ребер сетки. Число тактов по вычислению потоков не может быть меньше максимального числа граней для контрольных объемов. При этом число граней, одновременно обрабатываемых на каждом такте, не может быть больше чем $n_e/2$. Необходимость многократного запуска процедуры вычисления потоков для каждого такта и

снижение числа обрабатываемых одновременно граней — отрицательные стороны данного метода. Однако по сравнению с методом, использующим репликацию данных, в потактовом алгоритме нет необходимости в выделении дополнительных объемов оперативной памяти и лишнем суммировании вычисленных по граням потоков в ячейках.

В разработанном коде применяется подход с использованием фиктивных ячеек (метод 5). При этом отпадает необходимость учета типа грани контрольного объема (внутренняя или граничная). После копирования данных в память GPU запускается цикл по времени. На каждой итерации определяется шаг интегрирования по времени, исходя из критерия устойчивости разностной схемы. Для каждого граничного условия производится добавление необходимых параметров в фиктивные ячейки, после чего выполняется расчет градиентов в центрах контрольных объемов и добавление в фиктивные ячейки градиентов, соответствующих граничным условиям. Потоки рассчитываются для всех граней контрольных объемов, а затем для каждого внутреннего контрольного объема выполняются сложение потоков через грани и расчет параметров на следующем слое по времени.

Оптимизация производительности. Реализацию метода конечных объемов с использованием глобальной памяти и файла регистров поясняет рис. 3. На каждом слое по времени расчеты выполняются в два этапа. Для параллельной реализации метода конечных объемов на GPU используются два ядра, одно из которых производит вычисление потоков через грани контрольных объемов (этап 1, см. рис. 3, а), а другое осуществляет продвижение решения во времени (этап 2, см. рис. 3, б). На этапе 1 в глобальной памяти в массиве Q находятся значения переменных в центре контрольного объема. Расчет потока через каждую грань контрольного объема выполняется своей

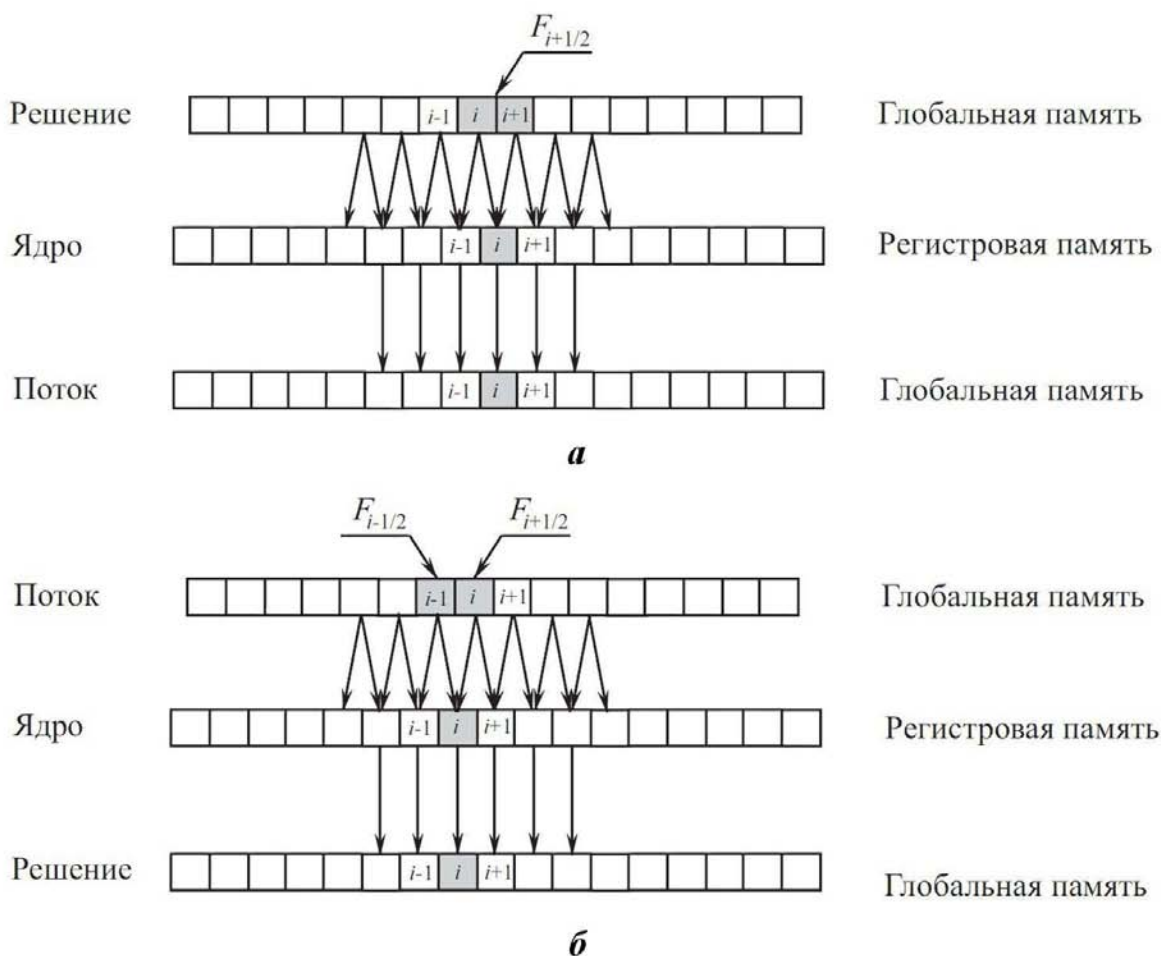


Рис. 3. Расчет потоков (а) и значений искомой функции на новом слое по времени (б)

нитью. Каждая нить использует значения переменных в контрольных объемах i и $i + 1$. Потоки через грани контрольного объема сохраняются в массиве F . На этапе 2 каждому контрольному объему соответствует своя нить. Для нахождения искомых функций на новом слое по времени используются потоки через грани $i - 1/2$ и $i + 1/2$ и решение в контрольном объеме i . Решение сохраняется в массиве Q .

Оптимизация производительности осуществляется за счет уменьшения числа обращений к глобальной памяти в каждой нити. Расчет левой и правой частей дискретного уравнения в каждом узле сетки основывается на переборе всех граней, связанных с данным узлом, и суммировании потоков через эти грани. Минимизация числа обращений к глобальной памяти достигается использованием разделяемой памяти и при помощи обновления значений переменных, хранящихся в глобальной памяти, по окончании всех расчетов.

Использование разделяемой памяти при расчете потоков поясняет рис. 4, на котором показан порядок копирования данных из глобальной памяти в разделяемую. При использовании трех контрольных объемов для расчета потоков и их ограничителей на шаге 1 производится копирование векторов решения, соответствующих центрированному расположению (см. рис. 4, а), а на шагах 2 и 3 — векторов решения, находящихся слева и справа (см. рис. 4, б, в). Каждая нить производит обработку трех векторов решения, находящихся в разделяемой памяти (см. рис. 4, г).

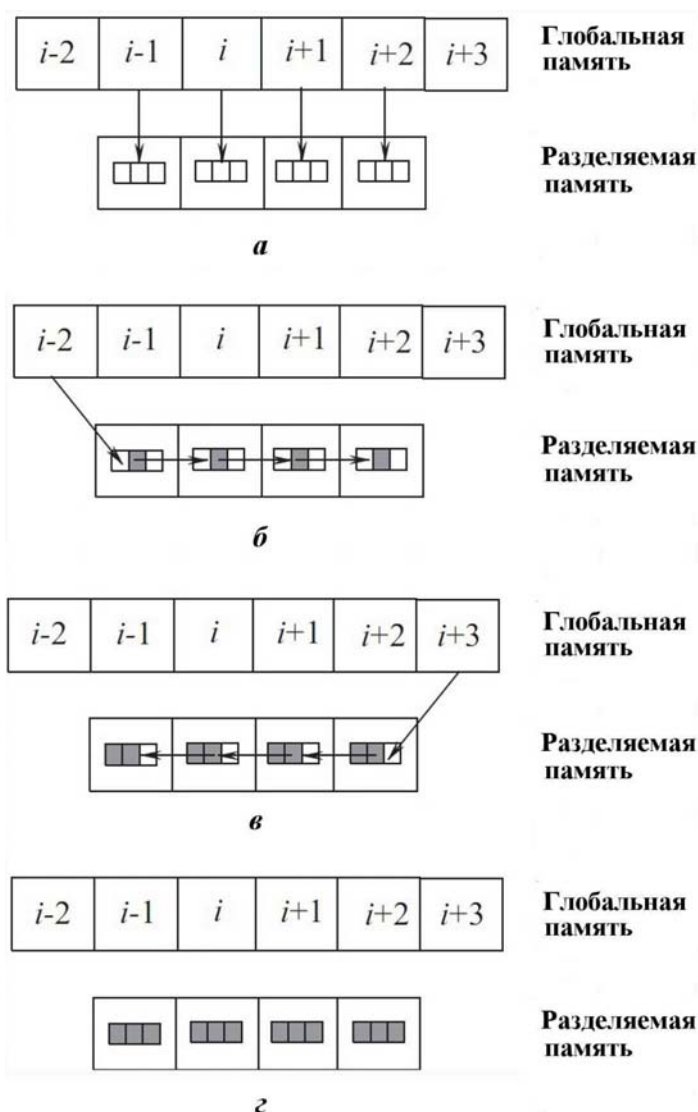


Рис. 4. Использование разделяемой памяти при расчете потоков

При реализации разностных схем расчета потоков данные, требуемые на шаге эволюции, рассчитываются перед началом шага по времени и хранятся в текстурной памяти. Результаты, полученные на шаге по времени, хранятся в глобальной памяти. Вместо обменов данными, соответствующими фиктивным ячейкам, между соседними блоками проводится расчет параметров в фиктивных ячейках каждого блока.

Решение тестовых задач

Задача Сода. Для сравнения различных схем расчета потоков рассмотрим решение задачи Сода [18]. В результате распада произвольного разрыва влево распространяется волна разрежения, вправо — ударная волна и контактный разрыв.

Дискретизация уравнений Эйлера проводится при помощи метода конечных объемов. Для явной дискретизации по времени используются схема Эйлера и двухшаговая схема Рунге—Кутты. Расчет потоков через грани контрольных объемов выполняется при помощи схем первого или второго порядка. Задача о распаде произвольного разрыва решается точным итерационным методом Годунова или приближенным методом Роу. В качестве рабочей среды рассматривается так называемый *вычислительный газ* [18].

Слева от разделяющей перегородки полагается $\rho_L = 1$, $u_L = 0,75$, $p_L = 1$, справа — $\rho_R = 0,125$, $u_R = 0$, $p_R = 1$. Расчеты проводятся на сетках с различным числом ячеек. При переходе от одной сетки к другой число ячеек увеличивается в несколько раз (1024 ячеек для сетки 1, 30 720 ячеек для сетки 2 и 307 200 ячеек для сетки 3). Сетка максимального размера (сетка 4) содержит около 3 млн ячеек. Шаг по времени составляет $1,52 \cdot 10^{-5}$ с, общее время счета — $7,63 \cdot 10^{-3}$ с. Число Куранта равняется 0,85. Расчеты проводятся на одном модуле платформы Tesla S1070 с тактовой частотой ядра 1,44 ГГц (количество ядер 256), а также на одном ядре центрального процессора AMD Phenom 2 с тактовой частотой 3 ГГц.

Время, необходимое для расчета одного шага по времени, а также ускорение решения задачи приводятся в табл. 1. Вариант 1 соответствует расчету по схеме Годунова, вариант 2 — расчету по схеме Роу. В обоих вариантах наблюдается хороший прирост ускорения S при увеличении числа контрольных объемов сетки.

Метод Годунова не является идеальным с точки зрения распараллеливания на GPU, поскольку точное решение задачи Римана предполагает большое количество ветвлений алгоритма и пересылок данных, снижающих производительность видеокарты по сравнению с приближенным подходом, основанным на схеме Роу.

Таблица 1

Время (мс) и ускорение решения задачи Сода

Номер варианта	Сетка 1			Сетка 2			Сетка 3			Сетка 4		
	CPU	GPU	S	CPU	GPU	S	CPU	GPU	S	CPU	GPU	S
1	1,63	0,13	12,43	47,70	0,20	245,25	460,64	0,92	502,50	4 627,61	8,06	574,39
2	0,14	0,07	1,87	5,51	0,17	33,17	43,58	0,57	76,00	436,09	5,22	83,48

Ударная труба. Расчеты течения в ударной трубе проводятся в трехмерной постановке. Дискретизация уравнений Эйлера или Навье—Стокса осуществляется на неструктурированной сетке с кубическими ячейками при помощи метода конечных объемов. Для явной дискретизации по времени используются схема Эйлера или двухшаговая схема Рунге—Кутты. Расчет потоков производится при помощи схем первого (схема Годунова) или второго порядка (используется ограничитель *minmod*). Задача о распаде произвольного разрыва решается точным итерационным методом Годунова или приближенным методом Роу. При решении задачи в вязкой постановке потоки дискретизируются при помощи схем второго порядка точности.

Длина расчетной области составляет $L = 10$ м. Расчеты проводятся с использованием сеток разного размера. Шаг сетки полагается постоянным. Самая грубая сетка содержит около 10^4

ячеек (сетка 1), самая подробная — 10^7 ячеек (сетка 4), промежуточные сетки — 10^5 (сетка 2) и 10^6 (сетка 3) ячеек. Шаг по времени составляет $1,52 \cdot 10^{-5}$ с, общее время счета — $7,63 \cdot 10^{-3}$ с. Число Куранта полагается равным 0,85. Расчеты выполняются на одном модуле платформы Tesla S1070 с тактовой частотой ядер 1,44 ГГц (количество ядер составляет 256), а также на одном ядре центрального процессора AMD Phenom 2 с тактовой частотой 3 ГГц.

Время счета, а также ускорение решения задачи приводятся в табл. 2 (время, необходимое для расчета 1000 шагов, указывается в секундах). Для обозначения варианта расчета используются три индекса. Первый индекс указывает на решение *невязкой* или *вязкой* задачи (1 — решаются уравнения Эйлера; 2 — уравнения Навье—Стокса). Вторым индексом обозначается способ дискретизации по времени (А — схема Эйлера; Б — двухшаговая схема Рунге—Кутты). Третьим индексом указывается на способ дискретизации невязких потоков (1, 2 — расчеты по схеме первого порядка с использованием точного и приближенного методов решения задачи о распаде произвольного разрыва; 3, 4 — схемы второго порядка с использованием точного и приближенного методов решения задачи о распаде произвольного разрыва).

При моделировании трехмерного течения в ударной трубе на подробной сетке, содержащей 10 млн ячеек (сетка 4), с использованием схемы Годунова (схема первого порядка) счет ускоряется примерно в 42 раза (вариант 1А2). Решение вязкой задачи требует увеличения порядка дискретизации по пространству до второго, и ускорение падает до 22 раз (вариант 2А2).

Время, необходимое для расчета 1000 шагов на сетке, содержащей 10^7 ячеек, и используемая память для различных реализаций алгоритма расчета приводятся в табл. 3. Вариант 1 соответствует расчету по схеме Годунова на одном модуле GPU, вариант 2 — расчету по аналогичному алгоритму на CPU, вариант 3 — расчету с использованием коммерческого пакета ANSYS Fluent (расчеты проводятся на кластере, содержащем 8 процессорных узлов). Для вариантов 2 и 3 указано также, во сколько раз превышаются время счета (S) и память (M) по сравнению с вариантом 1 (т. е. во сколько раз вариант с GPU выигрывает).

Таблица 2

Время (с) и ускорение решения задачи с ударной трубой

Номер расчета	Сетка 1			Сетка 2			Сетка 3			Сетка 4		
	CPU	GPU	S	CPU	GPU	S	CPU	GPU	S	CPU	GPU	S
1А1	6,63	0,47	13,96	69,99	2,64	26,50	671,22	22,67	29,61	6 329,61	198,63	31,87
1А2	14,72	0,62	23,68	121,83	3,73	32,68	934,05	31,60	29,56	8 787,27	207,82	42,28
1А3	12,03	1,29	9,30	130,33	10,41	12,52	1 262,17	94,27	13,39	11 018,60	826,18	13,34
1А4	20,66	1,55	13,33	193,47	12,81	15,11	1 630,47	115,92	14,06	13 934,40	872,42	15,97
2А1	18,79	1,36	13,83	198,96	10,98	18,12	1 918,32	99,60	19,26	17 485,10	874,99	19,98
2А2	27,79	1,60	17,35	261,63	13,31	19,65	2 285,81	120,88	18,91	20 542,40	914,83	22,46
1Б1	12,63	0,91	13,95	133,88	5,07	26,39	1 283,90	44,05	29,14	12 163,30	382,90	31,77
1Б2	29,14	1,20	24,30	242,48	7,29	33,26	1 909,59	61,83	30,89	16 989,20	406,65	41,78
1Б3	23,47	2,50	9,37	254,6	20,39	12,49	1 630,47	186,18	8,76	22 484,20	1 636,73	13,74
1Б4	41,57	3,04	13,67	380,87	25,60	14,88	3 227,67	230,86	13,98	27 978,30	1 734,85	16,127
2Б1	36,83	2,63	13,98	388,74	21,25	18,29	3 776,11	197,58	19,11	35 257,70	1 736,47	20,30
2Б2	56,28	3,14	17,90	515,97	26,28	19,63	4 672,10	240,95	19,39	40 595,40	1 822,97	22,27

Таблица 3

Время расчета и используемая память при решении задачи с ударной трубой

Вариант	Время, с	Память, Мб	S	M
1	305,29	2 582,28	1	1
2	14 916,60	2 696,72	48,86	1,04
3	7 662,00	8 210,16	25,10	3,18

Другие задачи. Разработанные средства численного моделирования реализуются в рамках пакета программ ЛОГОС и применяются для решения ряда тестовых задач, для которых существуют надежные экспериментальные и расчетные данные. Обзор тестовых задач пакета ЛОГОС дается в [4].

В настоящей работе рассматриваются задачи моделирования:

- ламинарного и турбулентного пограничных слоев на плоской пластине (A1);
- течения невязкого сжимаемого газа в плоском канале с клином при $M = 2$ (A2);
- невязкого обтекания профиля NACA0012 под углом атаки (A3);
- невязкого обтекания профиля RAЕ2822 (A4);
- невязкого течения около конуса при $M = 1,6$ (A5);
- теплообмена при обтекании обратной ступеньки (T1);
- теплообмена при обтекании препятствия на стенке плоского канала (T2);
- теплообмена при течении в квадратной каверне (T3).

Расчеты выполняются на основе явной схемы Эйлера первого порядка при помощи метода установления. Дискретизация невязких потоков проводится при помощи схемы второго порядка точности с ограничителем `minmod` (за исключением задачи T2, в которой применяется схема первого порядка). Для расчета вязких потоков используется интерполяция градиента из центра ячейки на грань, что позволяет получить схему второго порядка точности. Для решения задачи о распаде произвольного разрыва (задача Римана) применяется метод Годунова (все задачи, за исключением A1 и T3) или схема Роу с энтропийной поправкой Хартена (задачи A1 и T3).

Задачи A1 и T3 решаются в вязкой постановке, остальные задачи — в невязкой. При решении задачи A1 выполняется также моделирование турбулентного пограничного слоя на плоской пластине (подключается модель турбулентности).

Результаты решения тестовых задач не приводятся, основное внимание уделяется ускорению выполнения программы. Для тестовой реализации алгоритма измеряется производительность программного кода на CPU и GPU как при выполнении всего шага по времени, так и отдельных операций. Все вычисления выполняются для чисел с плавающей точкой одинарной точности.

Вычисления полностью выполняются на GPU (исключаются подготовка данных и обработка результатов), что не требует передачи данных между CPU и GPU. Если выполнять вычисления на CPU, то это приведет к необходимости обменов данными (операции чтения и записи) по $5n_e$ значений и увеличит общее время вычислений на GPU.

Результаты сравнения времени, необходимого на расчет одной итерации на CPU и GPU при решении различных тестовых задач, а также ускорение счета представлены в табл. 4. Расчеты производятся на одном ядре центрального процессора AMD Phenom 2,3 ГГц и на одном модуле графического процессора Tesla S1070, состоящем из 240 ядер, с тактовой частотой ядра 1,44 ГГц. Ускорение решения тестовых задач изменяется от 22 (задачи A4 и A5) до 60 раз (задача T1).

Для решения задачи A1 в турбулентной постановке использовались различные сетки и вычислительные системы. Расчеты выполнялись на одном ядре центрального процессора Xeon X5670 с

Таблица 4

Время и ускорение решения тестовых задач

Задача	Число узлов	Время счета на CPU, с	Время счета на GPU, с	Ускорение
A1	111 670	462,6	11,5	40,2
A2	95 000	288,8	12,3	23,5
A3	30 836	68,1	1,6	42,6
A4	38 265	141,9	6,3	22,5
A5	36 333	130,8	5,8	22,4
T1	138 003	295,3	4,9	59,7
T3	9 801	39,9	1,8	22,6

тактовой частотой 2,93 ГГц и на одном модуле графической системы Tesla S2050. Время счета одной итерации приводится в табл. 5 для сеток разного размера. При увеличении числа узлов сетки с 10^5 до 10^7 ускорение счета изменяется приблизительно на 10 %.

Таблица 5

Время и ускорение решения задачи A1

Число узлов	Время счета на CPU, с	Время счета на GPU, с	Ускорение
$1,3 \cdot 10^5$	0,140	0,003	46,67
$1,3 \cdot 10^6$	1,406	0,026	54,08
$6,6 \cdot 10^6$	7,091	0,126	56,28
$1,3 \cdot 10^7$	14,06	0,251	56,02

Заключение

В рамках пакета программ ЛОГОС, разработанного в РФЯЦ-ВНИИЭФ и предназначенного для решения широкого круга задач газовой динамики и теплообмена, создана вычислительная процедура моделирования трехмерных течений вязкого сжимаемого газа на неструктурированных сетках и обеспечена поддержка параллельных вычислений на графических процессорах общего назначения. Для программной реализации кода используется технология CUDA.

Приведены решения ряда модельных задач газовой динамики и теплообмена, на основе которых сравниваются ускорения счета на сетках разных размеров. Использование GPU позволяет ускорить расчеты в 10–50 раз (в зависимости от постановки задачи, используемых вычислительных алгоритмов и ресурсов).

Направления дальнейших работ и исследований связаны с реализацией дифференциальных моделей турбулентности, неявных разностных схем и методов ускорения сходимости (многосеточный метод, сглаживание невязки), а также с применением и реализацией разностных схем высокого порядка точности.

Работа выполнена при финансовой поддержке Российского фонда фундаментальных исследований (РФФИ) в рамках научного проекта № 13-07-12079, а также при частичной поддержке РФФИ проекта № 13-08-12033.

Список литературы

1. *Исаев С. А., Судаков А. Г., Баранов П. А. и др.* Разработка, верификация и применение основанного на многоблочных вычислительных технологиях распараллеленного пакета открытого типа VP2/3 для решения фундаментальных, прикладных и эксплуатационных задач аэромеханики и теплофизики // Вестник ЮУрГУ. Математическое моделирование и программирование. 2009. № 17(150). Вып. 3. С. 59–72.
2. *Смирнов Е. М., Зайцев Д. К.* Метод конечных объемов в приложении к задачам гидродинамики и теплообмена в областях сложной геометрии // Научно-технические ведомости СПбГПУ. 2004. № 2. С. 70–81.
3. *Абалакин И. В., Бахвалов П. А., Горобец А. В. и др.* Параллельный программный комплекс NOISETTE для крупномасштабных расчетов задач аэродинамики и аэроакустики // Вычислительные методы и программирование. 2012. Т. 13, № 1. С. 110–125.
4. *Козелков А. С., Дерюгин Ю. Н., Зеленский Д. К. и др.* Многофункциональный пакет программ ЛОГОС для расчета задач гидродинамики и теплопереноса на суперЭВМ. Базовые технологии и алгоритмы // Сб. трудов XII Межд. семинара "Супервычисления и математическое моделирование". 11–15 октября 2010 г. Саров: РФЯЦ-ВНИИЭФ, 2011. С. 215–230.

5. *Patankar S. V., Spalding D. B.* A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows // *Int. J. of Heat and Mass Transfer.* 1972. Vol. 15, No 10. P. 1787—1806.
6. *Волков К. Н., Емельянов В. Н.* Вычислительные технологии в задачах механики жидкости и газа. М.: Физматлит, 2012.
7. *Волков К. Н., Емельянов В. Н., Карпенко А. Г. и др.* Численное решение задач механики жидкости и газа на графических процессорах общего назначения // *Вычислительные методы и программирование.* 2013. Т. 14, № 1. С. 82—90.
8. *Волков К. Н., Емельянов В. Н., Карпенко А. Г. и др.* Реализация метода конечных объемов и расчет течений вязкого сжимаемого газа на графических процессорах // *Там же.* С. 183—194.
9. *Brandvik T., Pullan G.* Acceleration of a 3D Euler solver using commodity graphics hardware // *AIAA Paper.* 2008. No 2008-607.
10. *Corrigan A., Camelli F., Löhner R., Wallin J.* Running unstructured grid-based CFD solvers on modern graphics hardware // *Ibid.* 2009. No 2009-4001.
11. *Corrigan A., Camelli F., Löhner R., Mut F.* Semi-automatic porting of a large-scale Fortran CFD code to GPUs // *Int. J. for Numerical Methods in Fluids.* 2012. Vol. 69, No 2. P. 314—331.
12. *Brandvik T., Pullan G.* An accelerated 3D Navier–Stokes solver for flows in turbomachines // *ASME Paper.* 2009. No GT2009-60052.
13. *Kampolis I. C., Trompoukis X. S., Asouti V. G., Giannakoglou K. C.* CFD-based analysis and two-level aerodynamic optimization on graphics processing units // *Comp. Meth. in Appl. Mech. and Eng.* 2010. Vol. 199, No 9—12. P. 712—722.
14. *Волков К. Н., Емельянов В. Н.* Моделирование крупных вихрей в расчетах турбулентных течений. М.: Физматлит, 2008.
15. *Liu C. H., Li Y.* Turbulence modeling for computing viscous high-Reynolds-number flows on unstructured meshes // *Comp. Meth. in Appl. Mech. and Eng.* 2001. Vol. 190, No 40. P. 5325—5339.
16. *Mavriplis D. J.* Unstructured mesh discretizations and solvers for computational aerodynamics // *AIAA Paper.* 2007. No 2007-3955.
17. *Горобец А. В., Суков С. А., Железняков А. О. и др.* Применение GPU в рамках гибридного двухуровневого распараллеливания MPI+OpenMP на гетерогенных вычислительных системах // *Параллельные вычислительные технологии.* Челябинск: ЮУрГУ, 2011. С. 452—460.
18. *Sod G. A.* A survey of several finite difference methods of systems of nonlinear hyperbolic conservation laws // *J. of Comp. Phys.* 1978. Vol. 27, No 1. P. 1—31.

Статья поступила в редакцию 05.12.13.
