

Особенности параллельной реализации алгоритма расчета нестационарных течений газа через пористые среды с использованием графических ускорителей

Д.В. Леонтьев, Г.В. Тарасов, Д.И. Харитонов

Институт автоматики и процессов управления ДВО РАН

Аннотация. Рассматриваются вопросы реализации на графических ускорителях (GPU) алгоритма моделирования нестационарных течений газа через пористые среды с очагами гетерогенного горения. Численный метод решения задачи основан на комбинации явных и неявных разностных схем. Разработанный программный код имеет классическую структуру вычислительной задачи, включающую стадию подготовки счета, стадию вычислений и стадию сохранения результатов. В работе предпринята попытка переноса всей вычислительной стадии на GPU. Использование связки CPU+GPU для проведения расчетов оказывается неэффективным для данной задачи по двум причинам. Во-первых, наличие связности по данным между величинами явной и неявной схемы. В связи с этим невозможно распараллелить вычисления на разные процессоры. Во-вторых, наличие больших накладных расходов на передачу данных между памятью CPU и памятью GPU не позволяет организовать расчет заданных уравнений и граничных условий с высокой производительностью. Для расчета явной схемы используется классическое отображение сетки нитей на вычислительную сетку. Каждая нить вычисляет одну ячейку расчетной области. Размер блока подбирается вручную экспериментально для получения оптимальной производительности. Для нахождения решения неявной схемы используется готовый решатель из библиотеки cuSPARSE. С помощью профилировщика NVidia Visual Profiler получены результаты анализа производительности разработанных расчетных кодов на графический ускоритель. Дается оценка полученных результатов, описываются последующие шаги по изменению кода программы и увеличение производительности расчетов.

Ключевые слова: механика жидкости и газа, математическое моделирование, параллельное программирование, графический ускоритель, CUDA

The features of parallel program of computation of unsteady gas flows through a porous media using graphical accelerator

D.V. Leontev, G.V. Tarasov, D.I. Kharitonov

Institute of automation and control processes FEB RAS

Abstract. The questions of development on graphical accelerator (GPU) of an algorithm of modeling an unsteady gas flows through porous media with a sources of chemical combustion are considered. Numerical method for solving considered problem is based on combination of explicit and implicit finite-differences methods (FDM). The developed program's source code has a classical internal structure, including stage of input data preparing, stage of calculations, and stage of writing final results. By this article the migration attempt of calculation stage on GPU is initiated. It is found that for the considered problem the utilization of both CPU and GPU for computations is not effective through a two reasons. Firstly, there is a data dependency between values of explicit and implicit methods. In concerning this it is no possibility to parallelize methods' computations on different processors. Secondly, the large overheads on data transmission between CPU memory and GPU memory do not allow performing computations to solve equations and boundary conditions with a high performance. The classical mapping method of kernel threads on computational domain is used for solving explicit FDM. Each kernel thread computes the single cell of computational grid. Block size is matched experimentally for optimal performance values. The implicit FDM is solved using predefined solver from cuSPARSE library specially optimized for GPU. The performance analysis is fulfilled using NVidia Visual Profiler. The evaluation of the performance result is given; the future steps of code modification and performance improvement are described.

Keywords: fluid and gas mechanics, mathematical modeling, parallel programming, graphical processing unit, CUDA

Изучение процесса распространения газа через пористую среду имеет важное практическое значение. Приведем несколько примеров объектов реального мира, про которые можно сказать, что они, в той или иной степени приближения, являются пористыми средами. Свалки, торфяники, завалы в шахтах, разрушенные производственные и промышленные объекты и другие подобные объекты, все это является примером пористой среды. Исследование газодинамических процессов, происходящих в подобных объектах, важно с точки зрения обеспечения безопасности. Например, процессы горения и тления в торфяниках могут протекать годами, распространяться на огромные территории и наносить непоправимый вред человеку и окружающей среде [1]. Эффективных мер против подобных явлений практически не существует [2]. Распространение огня в разрушенных промышленных объектах тоже является

опасным явлением. Детальное понимание происходящих газодинамических и химических процессов внутри подобных объектов является основой для выработки успешных мер по противодействию распространения огня и его тушению.

В данной работе описываются особенности реализации параллельного алгоритма моделирования газодинамических и химических процессов, протекающих в пористом объекте с очагами гетерогенного горения. Математическая модель задачи и оригинальный численный метод ее решения детально представлены в работах [3-4]. Для решения задачи применяется комбинация явной и неявной разностных схем. Для ускорения расчетов все ресурсоемкие вычисления переносятся на графический ускоритель. Данная работа является продолжением работ [5-6], в которых исследовались методы распараллеливания аналогичных задач с использованием технологий параллельного программирования MPI и OpenMP. Особенностью данной работы является необходимость сочетания вычислений явной и неявной схемы на специализированном графическом процессоре, где не всякие методы решения систем линейных уравнений имеют оптимальную производительность.

Материал в статье будет представлен в следующем порядке. В первой части дано краткое описание математической модели и численного метода решения задачи. Далее кратко описан последовательный алгоритм решения задачи и выполнен анализ ресурсоемких процедур, перенос которых на графический ускоритель может дать ускорение выполнения расчетов. Далее приведены некоторые детали реализации численного метода решения задачи на графическом ускорителе. В заключительной части дано сравнение производительности высокооптимизированной CPU-версии программы и разработанной GPU-версии для сеток различной размерности.

1. Математическая постановка задачи

Будем рассматривать неподвижный пористый объект в форме прямоугольного параллелепипеда, состоящий из горючего компонента, инертного компонента и твердых продуктов реакции, и имеющий открытые границы сверху и снизу, и закрытые непроницаемые и нетеплопроводные боковые границы. Через пористый объект протекает газ, поступающий в объект через нижнюю границу, и вытекающий из него через верхнюю границу. Твердый горючий материал превращается в газовые и твердые продукты реакции в результате взаимодействия с газообразным окислителем. Математическая модель строится в предположении взаимодействующих взаимопроникающих континуумов [7] с использованием классических подходов теории фильтрационного горения [8] и включает в себя уравнения состояния, неразрывности, движения и энергии для каждого компонента (твердого и газообразного). Для описания динамики газа используется уравнение сохранения импульса для пористых сред. Процессы горения

описываются одностадийной химической реакцией первого порядка по отношению к каждому компоненту. В модели учитываются изменения объема и массы фаз при взаимодействии, наличие диффузии окислителя и зависимость динамической вязкости газа от температуры. Система уравнений рассматриваемой задачи моделирования нестационарных течений газа в пористых средах с очагами гетерогенного горения записывается следующим образом [3-4]:

$$\begin{aligned}
 (\rho_{cf}c_{cf} + \rho_{ci}c_{ci} + \rho_{cp}c_{cp}) \frac{\partial T_c}{\partial t} &= -\alpha(T_c - T_g) + Q\rho_{cf0}W + (1 - a_g)\lambda_c\Delta T_c, \\
 \rho_g c_g \left(\frac{\partial T_g}{\partial t} + (v_g \cdot \nabla)T_g \right) &= \alpha(T_c - T_g) + a_g\lambda_g\Delta T_g, \quad p = \frac{\rho_g R T_g}{a_g M}, \\
 \rho_g \left(1 + \chi(1 - a_g) \right) \left(\frac{\partial v_g}{\partial t} + (v_g \cdot \nabla)v_g \right) &= -a_g \nabla p + \rho_g g - a_g^2 \frac{\mu}{k_1} v_g - (1 - \\
 \mu_p) \rho_{cf0} W v_g, \\
 \frac{\partial \rho_g}{\partial t} + \nabla \cdot (\rho_g v_g) &= (1 - \mu_p) \rho_{cf0} W, \\
 \rho_g \left(\frac{\partial C}{\partial t} + (v_g \cdot \nabla)C \right) &= \nabla \cdot (\rho_g D_g \nabla C) - \mu_p \rho_{cf0} W - (1 - \mu_p) \rho_{cf0} W C, \\
 W = (1 - \eta) C k e^{-\frac{E}{RT_c}}, \frac{\partial \eta}{\partial t} &= W, \quad \rho_{cf} = (1 - \eta) \rho_{cf0}, \quad \rho_{cp} = \mu_p \rho_{cf0} \eta, \\
 a_g = a_{g0} + (a_{cf0} - a_{cfEnd}) \eta, \quad D_g &= D_g \left(\frac{T_g}{273} \right)^b, \quad \mu = c_{s1} \frac{T_g^{1.5}}{c_{s2} + T_g}.
 \end{aligned} \tag{1}$$

Здесь: a - объемная концентрация, C - массовая концентрация окислителя, c - удельная теплоемкость, c_{s1} и c_{s2} - константы в формуле Сазерленда, D_g - коэффициент диффузии газа, E - энергия активации, g - ускорение силы тяжести, k_1 - коэффициент проницаемости, M - молярная масса газа, p - давление газа, Q - тепловыделение, R - универсальная газовая постоянная, t - время, T - температура, v_g - скорость газа, W - скорость химической реакции, α - константа, определяющая интенсивность межфазного теплообмена, η - степень превращения горючего компонента, λ - коэффициент теплопроводности, μ - динамическая вязкость газа, ρ - эффективная плотность, χ - коэффициент, учитывающий инерционное взаимодействие фаз [7]. Индексы: 0 - начальный момент, *End* - конечный момент, c - конденсированная фаза, i - инертный компонент, f - горючий компонент, g - газ, p - продукт горения.

На нижней (открытой) границе пористого объекта известны температура и давление газа, а также массовая концентрация окислителя. На верхней (открытой) границе также известно давление. Известны условия теплообмена на границах пористого объекта. Скорость газа на открытой границе пористого объекта неизвестна и должна определяться в ходе решения задачи.

2. Численный метод решения задачи

Для решения системы уравнений (1) с описанными условиями в двумерной постановке разработан численный метод, основанный на комбинации явных и неявных конечно-разностных схем. Согласно разработанному методу уравнения энергии, сохранения импульса и концентрации окислителя преобразуются в явные конечно-разностные уравнения, а уравнения неразрывности преобразуется в неявное конечно-разностное уравнение. Подобный метод использовался для расчета нестационарных течений газа через пористые объекты с источниками энерговыделения и подробно описан в [9].

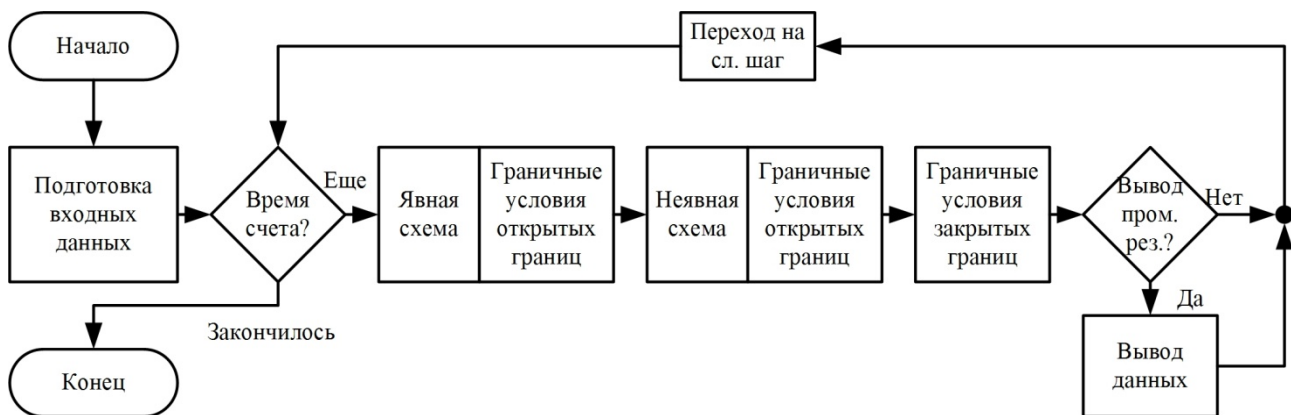


Рис. 1. Последовательный алгоритм решения задачи

Последовательный алгоритм решения задачи (см. рис. 1) представляет собой классический вычислительный алгоритм с главным циклом по времени моделирования. На каждом шаге моделирования сначала ищется решение уравнений явной схемы с вычислением граничных условий величин явной схемы на открытых границах, потом решаются уравнения неявной схемы, также с вычислением граничных условий величин неявной схемы. На заключительной стадии решения каждого шага вычисляются граничные условия на закрытых границах, а через определенные интервалы времени производится запись промежуточных результатов моделирования.

Реализации данного алгоритма на GPU основывается на следующих положениях. Обращения к памяти GPU является дорогой операций, соответственно нужно свести к минимуму операции выделения памяти и копирования данных между CPU-GPU. Также следует отметить сложность задачи и сильную взаимную связанность всех величин. А именно, невозможно запустить решение уравнений неявной схемы без получения результата явной схемы. Аналогичная ситуация с граничными условиями. Все эти факторы приводят к тому, что наиболее оптимальным вариантом использования GPU для решения этой задачи – это перенос всех вычислительных процедур на графический ускоритель с проведением соответствующих оптимизационных процедур и подбором оптимальных параметров сетки. Реализация смешанных

конфигураций расчета, когда только часть вычислительных процедур переносится на GPU (например, явная схема считается на GPU, а неявная – на CPU), вряд ли будет эффективной в виду упомянутых высоких накладных расходов при передаче данных между CPU и GPU. Таким образом, общий алгоритм расчета реализован следующим образом. Перед началом расчетов на ускорителе выделяется память и загружаются начальные данные задачи. Далее запускается цикл по времени моделирования, на каждом шаге которого вызываются соответствующие вычислительные функции для расчета величин явной и неявной схем. На CPU остается вычисление только счетчика итераций и записи промежуточных результатов, которые предварительно выгружаются из памяти ускорителя. Учитывая, что запись промежуточных результатов происходит редко, задержки при передаче данных можно считать минимальными по сравнению с общим временем счета. На рис. 2 показана схема проведения расчета с разбивкой по соответствующим вычислительным устройствам.

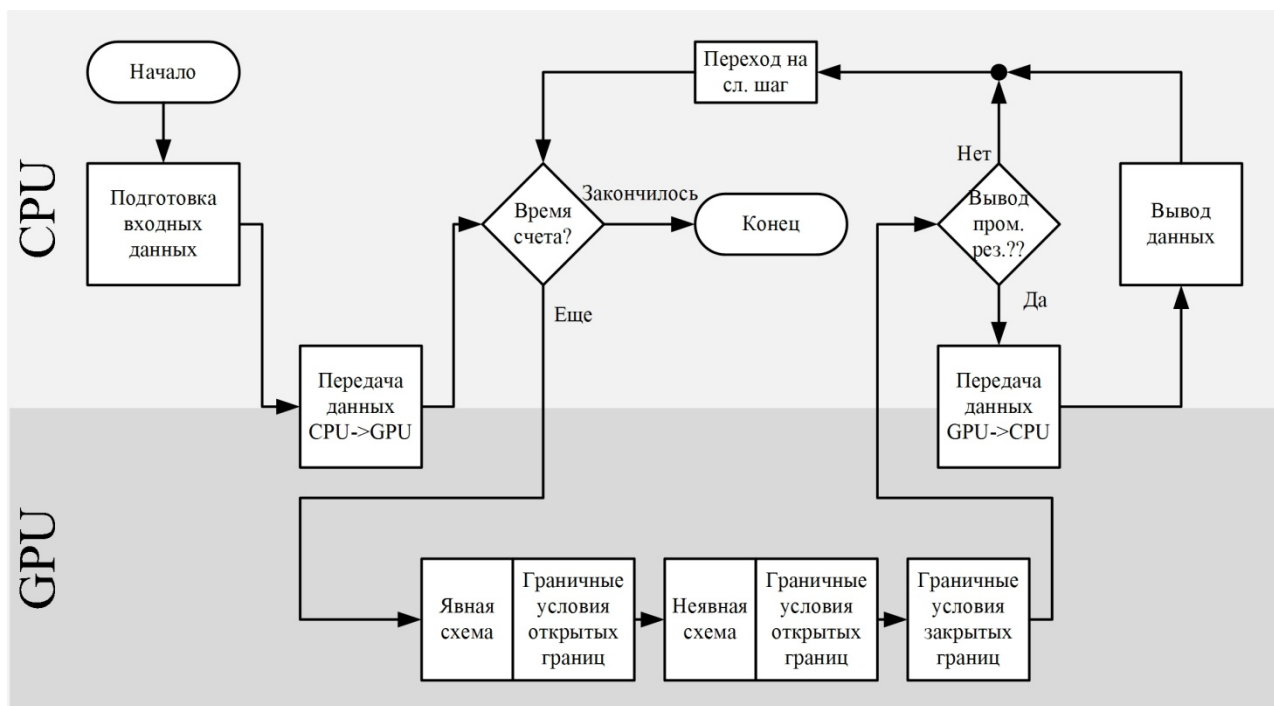


Рис. 2. Схема расчета с использованием GPU

Опишем некоторые детали реализации.

3. Реализация явной схемы

Явная схема наиболее просто реализуется на графическом ускорителе, так как значение каждой ячейки вычислительной сетки определяется на основании только соседних ячеек. Такая схема естественным образом определяет геометрическую декомпозицию вычислительной области задачи на независимые блоки для проведения вычислений большим количеством ядер

графического процессора. По условиям системы программирования CUDA вычислительные функции графического ускорителя выполняются на виртуальной сетке, которая описывается блоками одинакового размера. При реализации явной схемы основная задача состоит в определении геометрии виртуальной сетки, которая в конечном итоге определит количество нитей (потоков), которые будет выполнять графический процессор. Оптимальное разбиение на блоки зависит от многих факторов и чаще всего подбирается под конкретную задачу. Опишем геометрию виртуальной сетки для данной задачи. Схема разбиения показана на рис. 3(а).

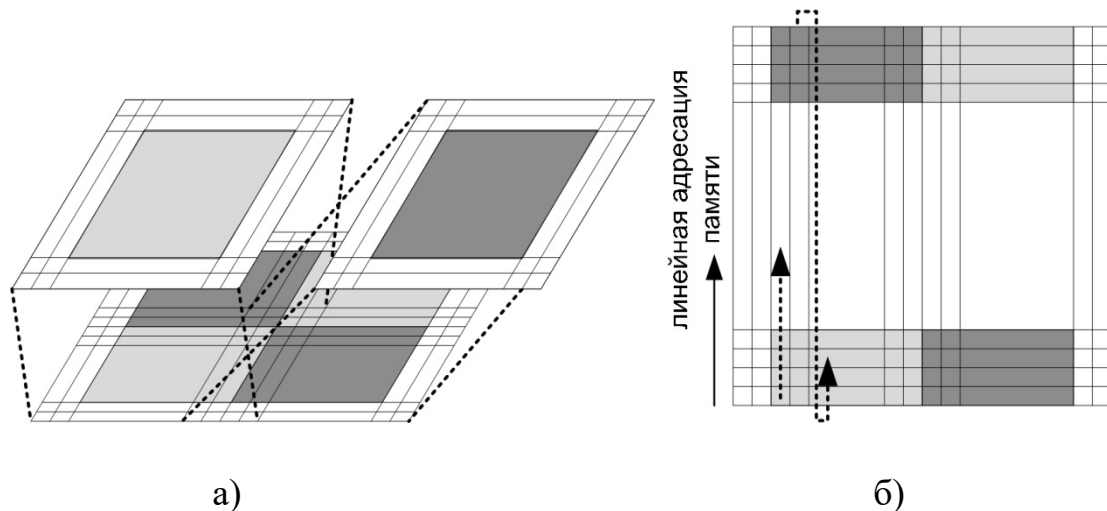


Рис. 3. Геометрическая декомпозиция вычислительной сетки для решения явной схемы, (а) – решение уравнений, (б) – граничные условия

Вся область расчета разбивается на равные блоки небольшого размера. Аппаратные ограничения графических ускорителей не позволяют создавать блоки с количеством нитей больше 1024, поэтому выбирается небольшой размер блока (до 1024 ячеек). Экспериментально был подобран размер блока 4×32 , для которого были получены наиболее оптимальные результаты производительности расчетов. Данный размер включает только ячейки, непосредственно представляющие область расчетов. Учитывая, что схема имеет 4-ый порядок аппроксимации, то к основному размеру необходимо добавить соседние ячейки. Таким образом, размер блока составил 8×36 ячеек. Вычислительная функция строилась из расчета – по одному ядру на ячейку. Количество блоков в виртуальной сетке подбиралось кратным размеру блока. Рассматривались сгущающиеся сетки: 32×32 , 64×64 , 96×96 и т.д. до 1024×1024 . Соответственно количество блоков для данных видов сеток составило: 8×1 , 16×2 , 24×3 и т.д. до 256×32 . Для каждого блока выделяется блок разделяемой памяти, куда потоки загружают соответствующие значения вычислительной сетки. Граничные потоки блока не производят вычислений, по окончании загрузки данных они завершают выполнения. Потоки, которые соответствуют основному размеру блока, решают уравнения явной схемы и определяют

значения пористости, скорости химической реакции, скорости превращения горючего вещества, температуры тела, концентрации горючего вещества, температуры и скорости газа.

Для вычисления граничных условий явной схемы использовался фиксированный размер блока по координате ОУ. Учитывая 2-ой порядок аппроксимации на границах, для вычисления верхней и нижней границ исследуемого объекта требовался размер блока 8 ячеек по высоте и по длине, равный размеру объекта (исключая левую и правую границы). Схема разбиения показана на рис. 3(б). Так как в памяти значения ячеек вычислительной сетки хранятся по столбцам (column-major), данный способ разбиения позволил сохранить линейный способ обращения к памяти потоков для загрузки значений вычислительной сетки и обеспечить загруженность вычислительных ядер графического процессора.

4. Реализация неявной схемы

Неявная схема в реализации имеет более сложную структуру, требуется находить решения систем линейных алгебраических уравнений. Метод прогонки, который использовался в исходном, последовательном, варианте программы не может быть эффективно реализован на графическом ускорителе в виду явной зависимости вычислительных выражений (на прямом и обратном ходе прогонки). Для решения неявной схемы требовалось найти другой метод, который бы был более адаптирован для графических ускорителей и имел бы достаточно эффективную реализацию. Поскольку исходная система уравнений является трехдиагональной, то в качестве альтернативных методов рассматривались алгоритм блочной прогонки и решатели, которые реализованы в библиотеке cuSPARSE. В данной работе предложено использовать решатель трехдиагональных систем линейных уравнений из библиотеки cuSPARSE, который строит решение группы систем линейных уравнений

$$A^{(i)} * y^{(i)} = x^{(i)}$$

Предлагаемые библиотекой функции строят решение относительно матрицы коэффициентов A и правой части x для группы систем линейных уравнений $i = 0..N$, где N – число систем уравнений. Следует отметить, что решение систем линейных уравнений относительно y в результате перезаписывает коэффициенты правой части x , поэтому общая формула проводимых вычислений записана не в традиционном математическом виде $Ax = b$. Библиотека cuSPARSE предлагает несколько функций для решения уравнений. Все перечисленные ниже функции имеют общий одинаковый набор входных параметров, определяющих матрицы коэффициентов. Отличие состоит только в дополнительном параметре, задающем вспомогательную память для вычислений:

а) `cusparsesolve<T>gtsvStridedBatch()` – полностью решает группу систем линейных уравнений; делает дополнительные вызовы для выделения вспомогательной памяти, необходимой для построения решения (`cudaMalloc` и `cudaFree`);

б) `cusparsesolve<T>gtsv2StridedBatch()` – только строит решение, выделить вспомогательную память разработчик обязан самостоятельно, необходимый размер памяти определяется функцией `cusparsesolve<T>gtsv2StridedBatch_bufferSize()`.

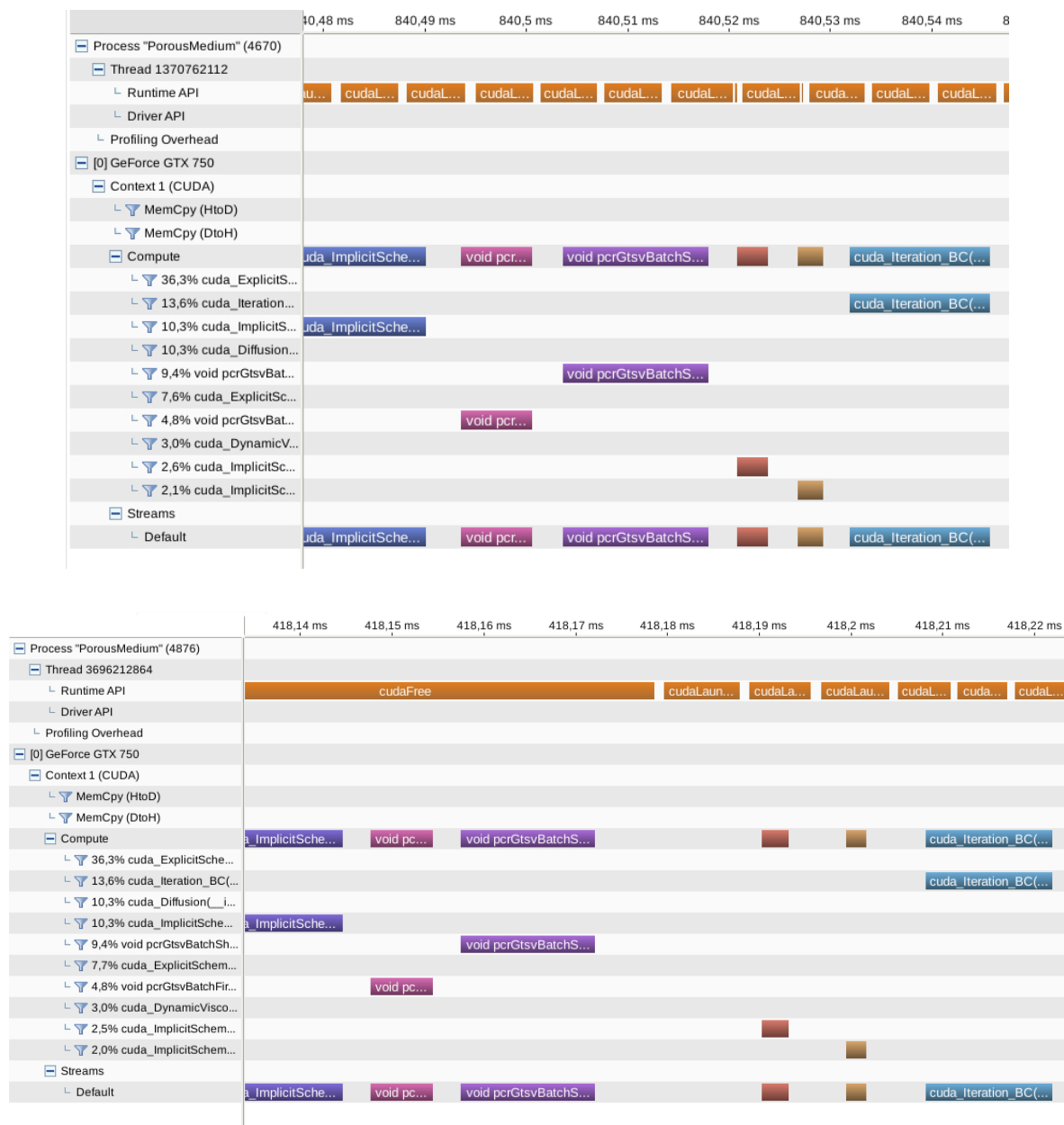


Рис. 4. Анализ временных задержек на выделение дополнительной памяти для решателя трехдиагональных систем библиотеки cuSPARSE

Очевидно, что использование функции из пункта а) является менее выгодным ввиду дополнительных задержек на выделение памяти. Учитывая,

что решения уравнений нужно строить на каждом шаге расчета, небольшие задержки на выделение памяти превращаются в значительные задержки в процессе длительного счета. Поскольку в процессе счета конфигурация вычислительной сетки задачи, размерность системы линейных уравнений не меняется, то гораздо эффективнее один раз выделить вспомогательную память для решателя в начале счета, дальше просто использовать ее для построения решения. С использованием профилировщика NVIDIA Visual Profiler были определены затраты на выделение вспомогательной памяти для неявной схемы. Они составили около 50% от общего времени решения уравнений неявной схемы. На рис. 4 показаны замеры времени для конфигураций программы для обоих случаев.

В результате решения уравнений определяются значения давления газа по всей расчетной области. По известному давлению и температуре газа явно определяется плотность газа с помощью вычислительных функций аналогичных явной схеме, но с более простой конфигурацией сетки потоков. Граничные условия для давления и плотности вычисляются подобным образом, как это делается для явной схемы.

5. Оценка производительности

Для оценки эффективности применения GPU для решения подобных задач проведено тестирование и сравнение скорости работы высокооптимизированной последовательной CPU-версии программы (максимально использующей возможности векторизации вычислений) и разработанной GPU-версии программы. Результаты представлены на графике на рис. 5.

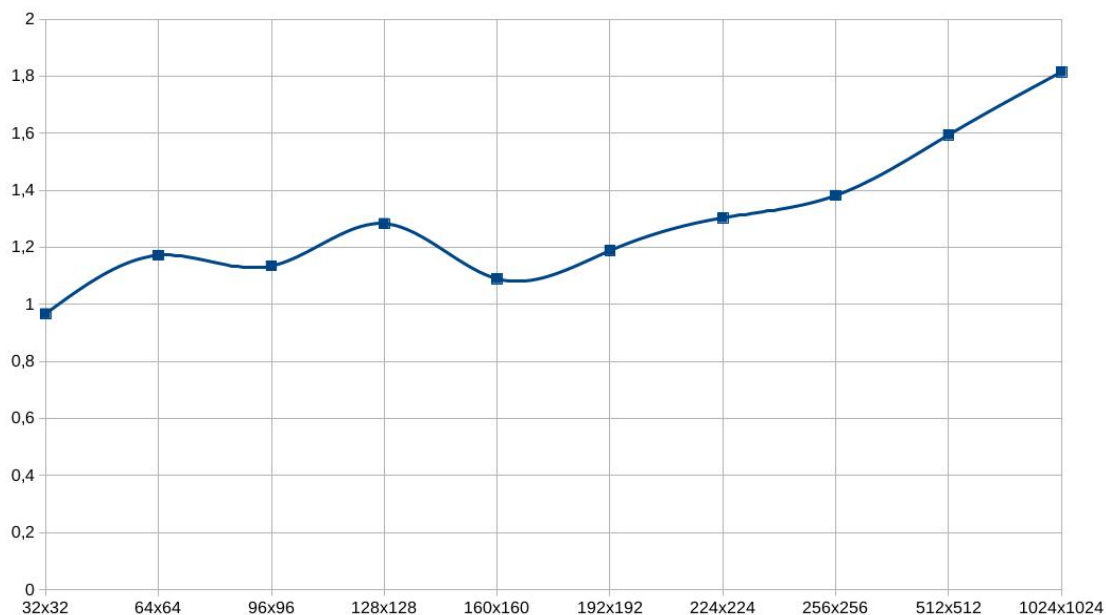


Рис. 5. График ускорения вычисления на GPU в сравнении с высокооптимизированной CPU-версией программы для сеток различных размерностей

Сравнение проводилось при одинаковых входных данных с уменьшением шага сетки. Тестовые запуски CPU-версии проводились на процессорах Core i5 третье поколения с поддержкой AVX (микроархитектура Ivy Bridge). GPU-версия программы тестировалась на картах NVidia GeForce 700 серии (512 ядер, CUDA compute capability 5.0). Видно, что для небольших объемов данных GPU немного отстает от CPU. С увеличением числа узлов сетки ускорение достигает примерно 2 раз.

6. Заключение

Разработана программа расчета нестационарных течений газа через пористые среды, адаптированная для графических ускорителей. Не смотря на то, что получено небольшое ускорение расчетов по сравнению с высокооптимизированной CPU-версией, наличие первой версии расчетных кодов позволит провести более глубокий анализ производительности и определить узкие места в реализации для последующей оптимизации. В настоящее время можно сделать вывод, что низкая производительность, в первую очередь, обусловлена значительными затратами на обращения к памяти вследствие несбалансированности объемов вычислений и данных, используемых в процессе счета. Часть этих затрат, например, расчет граничных условий скорее всего вообще не удастся сократить из-за специфики проводимых вычислений: простые и небольшие вычисления наряду со значительными обращениями к памяти. Часть же затрат можно сократить за счет балансировки и использования более новых поколений ускорителей, что позволит ускорить проведение расчетов.

Работа выполнена при частичной финансовой поддержке РФФИ (грант 16-01-00103-а) и программы фундаментальных исследований ДВО РАН «Дальний Восток» 2018-2020 (проект 18-5-100).

Литература

1. Ложкин В.Н., Тимофеев В.Д., Ложкина О.В., Иванов А.Н. Методика прогнозирования чрезвычайно опасного загрязнения воздуха в окрестности автомагистрали торфяным пожаром // Проблемы управления рисками в техносфере. – 2016. – № 3 (39). – С. 89-97.
2. Чвѣрткин А.Г. Подземно-защитная стена как новый метод повышения безопасности населѣнного пункта при торфяном пожаре // Актуальные проблемы гуманитарных и естественных наук. – 2013. – № 9. – С. 52-55.
3. Левин В.А., Луценко Н.А. Двумерные течения газа при гетерогенном горении твѣрдых пористых сред // Доклады Академии наук. - 2017. - Т. 476. № 1. - С. 30-34.
4. Lutsenko N.A. Numerical model of two-dimensional heterogeneous combustion in porous media under natural convection or forced filtration // Combustion Theory and Modelling. – 2018. – Vol. 22, Is. 2. – P. 359-377.

5. Луценко Н.А., Тарасов Г.В., Гырник К.А. OpenMP-версия параллельного алгоритма расчета нестационарных течений газа через пористые объекты с источниками энерговыделения: анализ и применение // Сибирский журнал вычислительной математики. - 2014. - Т. 17. № 3. - С. 229-244.
6. Луценко Н.А., Мирошниченко Т.П., Одякова Д.С., Харитонов Д.И. параллельная реализация алгоритма для расчета двумерных нестационарных течений газа через пористые объекты с источниками тепловыделения // Вычислительные технологии. - 2011. - Т. 16. № 2. - С. 98-110.
7. Нигматулин Р.И. Основы механики гетерогенных сред. – М.: Наука, 1978. – 336 с.
8. Алдушин А.П., Мержанов А.Г. Теория фильтрационного горения: общие представления и состояние исследований // Распространение тепловых волн в гетерогенных средах. – Новосибирск: Наука, 1988. – С. 9-52.
9. Луценко Н.А. Численное моделирование трехмерных нестационарных течений газа через пористые объекты с источниками энерговыделения // Вычислительная механика сплошных сред. – 2016. – Т. 9, № 3. – С. 331-344.

References

1. Lozhkin V.N., Timofeev V.D., Lozhkina O.V., Ivanov A.N. Metodika prognozirovaniia chrezvychaino opasnogo zagriazneniia vozdukha v okrestnosti avtomagistrali torfianym pozharom // Problemy upravleniia riskami v tekhnosfere. – 2016. – Vol. 3 (39). – P. 89-97.
2. Chvertkin A.G. Podzemno-zashchitnaia stena kak novyi metod povysheniia bezopasnosti naselennogo punkta pri torfianom pozhare // Aktualnye problemy gumanitarnykh i estestvennykh nauk. – 2013. – Vol. 9. – P. 52-55.
3. Levin V.A., Lutsenko N.A. Dvumernye techeniia gaza pri geterogennom gorenii tverdykh poristykh sred // Doklady Akademii nauk. - 2017. - Vol. 476. Is. 1. - P. 30-34.
4. Lutsenko N.A. Numerical model of two-dimensional heterogeneous combustion in porous media under natural convection or forced filtration // Combustion Theory and Modelling. – 2018. – Vol. 22, Is. 2. – P. 359-377.
5. N.A. Lutsenko, G.V. Tarasov, K.A. Gyrnik. An OpenMP Version of the Parallel Algorithm for Calculation of Unsteady Gas Flow through Porous Objects with Energy Sources: Analysis and Application // Numerical Analysis and Applications, 2014, Vol. 7, Is. 3. P. 191-203.
6. Lutsenko N.A., Miroshnichenko T.P., Odiakova D.S., Kharitonov D.I. parallelnaia realizatsiia algoritma dlia rascheta dvumernykh nestatsionarnykh techenii gaza cherez poristye obieekty s istochnikami teplovydeleniia // Vychislitelnye tekhnologii. - 2011. - Vol. 16. Is. 2. - S. 98-110.
7. Nigmatulin R.I. Osnovy mekhaniki geterogennykh sred. – М.: Nauka, 1978. – 336 p.

8. Aldushin A.P., Merzhanov A.G. Teoriia filtratsionnogo gorenii: obshchie predstavleniia i sostoianie issledovaniia // Rasprostranenie teplovykh voln v geterogennykh sredakh. – Novosibirsk: Nauka, 1988. – P. 9-52.
9. Lutsenko N.A. Chislennoe modelirovanie trekhmernykh nestatsionarnykh techenii gaza cherez poristyie obiekty s istochnikami energovydeleniia // Vychislitelnaia mekhanika sploshnykh sred. – 2016. – Vol. 9, Is. 3. – P. 331-344.