УДК 519.688

# Б. А. Корнеев, В. Д. Левченко

# <sup>D</sup><sub>T</sub><sup>G</sup>: алгоритм и программный комплекс для расчёта сверхбольших задач газовой динамики в трёхмерной постановке на гетерогенных (супер)компьютерах

Аннотация. В данной работе представляется программный комплекс для численного моделирования трехмерных задач газовой динамики. В качестве математической модели выбраны уравнения Эйлера невязкого газа, дополненные уравнениями, описывающими мультикомпонентность среды. Рунге-Кутты разрывный метод Галеркина (RKDG метод) используется для численного решения получаемой системы уравнений. Алгоритмом реализации рассматриваемой численной схемы для расчёта на мультипроцессорных гетерогенных компьютеров служит алгоритм DiamondTorre, принадлежащий к семейству локально-рекурсивных нелокально-асинхронных (LRnLA) алгоритмов. В их основе лежит идея пространственно-временной декомпозиции расчётной области с учётом графа зависимостей численного шаблона. Алгоритм обладает свойствами повышенной интенсивности вычислений, а также эффективной параллельностью. Разработанный программный комплекс, получивший название  $\mathbb{D}_{\mathbb{T}}^{\mathbb{G}}$ , открывает доступ к решению задач с размерами области порядка  $10^8 - 10^9$  ячеек на одном ПК с видеокартой, и в несколько порядков раз больше при использовании суперкомпьютерных вычислительных мощностей, так как алгоритм имеет хорошую «слабую масштабируемость».

Kлючевые слова и фразы: вычислительная газодинамика, метод RKDG, вычисления на GPU, CUDA программирование, MPI, эффективные реализации численных методов, алгоритмы LRnLA, алгоритм DiamondTorre.

Работа поддержана грантом «УМНИК» Фонда содействия развитию малых форм предприятий в научно-технической сфере<sup>(1)</sup>

<sup>©</sup> Б. А. Корнеев<sup>(1)</sup>, В. Д. Левченко<sup>(2)</sup>, 2015

<sup>©</sup> Московский физико-технический институт, Долгопрудный, Российская Федерация (1) 2015

С Институт прикладной математики имени М.В. Келдыша РАН, Москва, Российская Федерация $^{(2)}$  2015

<sup>©</sup> Программные системы: теория и приложения, 2015

## Введение

Данная работа посвящена математическому моделированию сверхбольших задач газовой динамики с использованием современных численных методов, суперкомпьютерных вычислительных мощностей и специальных алгоритмов реализации численных методов на таких ЭВМ. Под «сверхбольшими» задачами мы понимаем задачи в полностью трехмерных постановках и с числом узлов по каждой пространственной оси вычислительной области порядка 10<sup>3</sup> и более. Сетки с адаптивным разбиением в данной работе не рассматриваются. Такие размеры вычислительной области необходимы (и не всегда достаточны) для адекватного моделирования актуальных задач газовой динамики с учетом ударных волн в мультикомпонентных средах. Разрабатываемый комплекс планируется использовать для решения именно таких задач, например, задач о взаимодействии ударных волн с неоднородностью в газе, но специально не ограничиваясь ими.

Задачи такого типа исследованы со стороны вычислительной математики, и для таких задач разработаны эффективные численные методы: явные методы высокого порядка аппроксимации, не осциллирующие на особенностях решений — ударных волнах и контактных разрывах. В данной работе используется разрывный метод Галёркина (RKDG метод [1]), авторов привлекает его достойные вычислительные характеристики вкупе с достаточно простой для понимания численной схемой.

Акцент на использовании суперкомпьютеров сделан по причине того, что задачи заявленных размеров (порядка 10<sup>8</sup>–10<sup>9</sup> и более) практически не поддаются расчёту на менее мощных машинах. Тем не менее, бурный рост производительности процессоров, а особенно графических процессоров, до значений 10<sup>12</sup>-10<sup>13</sup>FLOPs по пиковой производительности передовых GPU, даёт право как минимум сделать такого рода расчёты менее «эксклюзивными», приблизив, таким образом, инженерную применимость численного моделирования в рассматриваемой постановке. Авторами проведено подобное исследование в [2], одним из основных результатов которого была серия расчётов задачи о взаимодействии пузырька с ударной волной на сетке  $1024 \times 512 \times 512$  на персональном компьютере с, как говорят, «игровыми» характеристиками (достаточными для запуска современных видеоигр и комфортной игры в них), оборудованным 32Gb оперативной памяти и видеокартой NVidia GTX Titan. Расчёт одной задачи занимал несколько суток машинного времени. В данной работе рассматривается развитие данной методики, а именно, строится суперкомпьютерная реализация вычислительного алгоритма на основе [2].

Алгоритм, имеющий название DiamondTorre, использует расчёт на GPU в «вычислительном окне», когда как вся задача хранится в оперативной памяти CPU, которая обычно в несколько раз более вместительная. Используется пространственно-временная декомпозиция на «башни» (torres, ит.), каждая из которых имеет в основании ромб (diamond, англ.). В трехмерном пространстве выделяются векторизованая ось (Z), синхронная ось (X) и асинхронная ось (Y). «Распараллеливание» по вычислительным узлам кластера может происходить по осям X и Y, причём по оси X строится также пространственновременная декомпозиция, а по Y — стандартный геометрический параллелизм. Используются технологии CUDA для программирования GPU и MPI для работы с распределенной вычислительной системой. Предлагаемое рабочее название программного комплекса  $\mathbb{D}_{\mathbb{T}}^{\mathbb{G}}$  представляет собой совокупность первых букв DG в численном методе и DT в алгоритме.

Проводятся результаты тестирования производительности алгоритма. Особый интерес представляет эффективность решения больших и сверхбольших задач, масштабируемость алгоритма, а также доля реально достигнутой производительности в сравнении с теоретически возможной, исходя из пиковой производительности вычислительных устройств.

## 1. Численный метод

Поведение газа в процессах, когда влияние вязкости пренебрежимо мало, описывается системой уравнений Эйлера. Запишем её в так называемом дивергентном виде

(1) 
$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_j(\mathbf{U})}{\partial x_j} = 0,$$

 $\mathbf{U} = (\rho, \rho u_i, E)^T$  — консервативные переменные (плотность, количество движения в единице объема, энергия в единице объема).  $\mathbf{F}_j(\mathbf{U}) = (\rho u_j, \rho u_i u_j + p \delta_{ij}, (E + p) u_j)^T$  — эйлеровы потоки. Система становится замкнутой при добавлении уравнения состояния газа, функционально связывающего внутреннюю энергию, давление и плотность, например,  $\varepsilon = \varepsilon(p, \rho)$ .

Будем решать эту систему в трёхмерной области  $\Pi \in \mathbb{R}^3$  с заданными начальными и граничными условиями.

#### 1.1. Пространственная дискретизация

Вводим разбиение области П ячейками  $S_p, p = \overline{1, ..., N}, \bigcup_{p=1}^N S_p = \Pi, S_p \cap S_q = \emptyset, p \neq q$ . Приближенное решение  $\mathbf{U}_h$  в каждой  $S_p$  представим в виде разложения

(2) 
$$\mathbf{U}_h(x,t) = u_n(t)\varphi_n(x)$$

по базисным функциям конечномерного пространства  $V_p = \{\varphi_n(x_i)\}_{n=1}^k$  размерности k. Коэффициенты  $u_n(t) = [u_n^1(t), \ldots, u_n^5(t)]^T$  зависят только от времени.

Подставив (2) в (1), поставив для невязки условие ортогональности базисным функциям, и воспользовавшись теоремой Гаусса, можно получить следующую систему обыкновенных дифференциальных уравнений для определения неизвестных  $u_n(t)$  в каждой ячейке

(3) 
$$\dot{u}_n \int_{S_p} \varphi_n \varphi_m dV + \int_{\partial S_p} \mathbf{F}_j(\mathbf{U}_h) \varphi_m n_j d\Sigma - \int_{S_p} \mathbf{F}_j(\mathbf{U}_h) \frac{\partial \varphi_m}{\partial x_j} dV = 0,$$

Функция  $\mathbf{F}_{j}(\mathbf{U}_{h})$ , вообще говоря, не определенная при  $x \in \partial S_{p}$ , доопределяется как функция  $h_{j}(x_{int}, x_{ext})$ , зависящая от предельных значений функции  $\mathbf{U}_{h}$  слева и справа от грани, по которой соприкасаются ячейки, и определяется тем или иным способом, например, как точное или приближённое решение задачи Римана [3]. После этого получим

(4) 
$$\dot{u}_n \int_{S_p} \varphi_n \varphi_m dV + \int_{\partial S_p} h_j (\mathbf{U}_h(x_{int}), \mathbf{U}_h(x_{ext})) \varphi_m n_j d\Sigma - \int_{S_p} \mathbf{F}_j (\mathbf{U}_h) \frac{\partial \varphi_m}{\partial x_j} dV = 0,$$

Интегралы в (4) заменяем квадратурными формулами. Данная система представима в виде, разрешенном относительно старшей производной

(5) 
$$\dot{u} = Lu$$
  
 $u(0) = u_0$ 

 ${}^{\mathbb{D}}{}_{\mathbb{T}}{}^{\mathbb{G}}$ : алгоритм и программный комплекс для сверхбольших CFD задач

## 1.2. Временная дискретизация

Систему (5) будем решать явным методом Рунге-Кутты, применяя на каждой стадии специальный ограничитель-монотонизатор.

(1) 
$$u^{0} = \Lambda \Pi_{h} P(u_{0})$$
  
(2)  $\forall n = 0, \dots, M - 1$ :  
•  $u^{(0)} = u^{n}$   
•  $\forall i = 1, \dots, K + 1 : u^{(i)} = \Lambda \Pi_{h} \Big\{ \sum_{l=0}^{i-1} \alpha_{il} u^{(l)} + \beta_{il} \Delta t^{n} L(u^{(l)}) \Big\};$   
•  $u^{n+1} = u^{K+1}$ 

 $\Lambda \Pi_h$  — специальный ограничитель-монотонизатор, необходимый при расчёте разрывных течений для устранения нефизических осцилляций на разрывах. Результат действия ограничителя зависит от значений базисных коэффициентов в рассматриваемой ячейке и значений её соседей по граням.

В данной работе используется численный метод RKDG второго порядка по пространству и времени, численный поток HLLC [3] и ограничитель градиента типа *minmod* [1].

#### 2. Вычислительный алгоритм

В данной работе применяется алгоритм DiamondTorre [2]. Расчёт задачи с помощью него удобно представить себе как заполнение вычислительной области башнями. Схема такого заполнения представлена на рисунках 1(a) и 1(b). Суть каждого кирпичика башни с координатами  $(x_i, y_i, t_k)$  — выполнение части расчёта по численной схеме по всему отрезку  $(x_i, y_i, \alpha L_z, t_k), \alpha \in [0, 1]$ , векторизованному с помощью CUDA-тредов, — это может быть выполнение стадий Рунге-Кутты или ограничения градиента. На рисунке 1(a) они обозначены R1, R2, L1 и L2 соответственно. Будем считать координатами башни координаты её левого нижнего кирпичика в плоскости xt. Параметр  $F \ge 1$  определяет, сколько шагов по времени рассчитывается при одной итерации алгоритма. В плоскости xt каждая башня строится снизу вверх по оси t, а по x — в сторону её наклона (для конфигурации, как на рисунке 1, — по возрастанию x). На каждой итерации алгоритма заполнение области в плоскости xy (рисунок 1(b)) идёт справа налево, начиная от правой границы по x, и на каждом  $x_i, i \in \overline{0, \dots, Nx - 1}$  башни строятся через одну по оси y, при этом если при  $x = x_i$  строились башни только на чётных y, то при  $x = x_{i-1}$  строятся башни только на нечетных y. В результате все

башни, имеющие одну координату x (на рисунке они раскрашены в один цвет), во время каждой итерации могут выполняются асинхронно в CUDA-блоках. Во время расчёта с помощью данного алгоритма,



Рис. 1. Алгоритм DiamondTorre: (а) отображение в плоскости xt, (b) отображение в плоскости xy, (c) — организация обменов данными между GPU и CPU

для хранения дополнительных данных при вычислении R1, R2, L1 и L2 требуется массив размером не более 5F + 1 по координате x. Эти данные гранятся на GPU. В силу того, что ширина массива гораздо меньше ширины расчётной области, память используется более эффективно. Особо хотелось бы подчеркнуть, что объем обмена данными по построению независим от высоты башни. Таким образом, варьируя этот размер, удается понизить негативное влияние такого обмена на темп счёта.

В частности, с помощью разработанного программного комплекса на основе численного метода RKDG и алгоритма DiamondTorre оказалось способным рассчитывать задачи типа "bubble-shock interaction" [4] размером  $512 \times 512 \times 1024$  ячеек на персональном компьютере [2]. Результаты вычислений иллюстрированы на рисунке 2.

#### 3. Тестирование производительности

#### 3.1. Эффективность использования GPU

Сперва проведем оценку производительности алгоритма на одном узле или ПК с разными GPU. В следующем тесте исследуется,

 ${}^{\mathbb{D}}{}_{\mathbb{T}}{}^{\mathbb{G}}:$ алгоритм и программный комплекс для сверхбольших CFD задач



Рис. 2. Примеры численных решений

как зависит производительность вычислений от размера окна вычислений. Разамер области фиксирован и составляет  $512 \times 512 \times 1024$  ячеек. Варьируется количество «этажей» в «башне», а вместе с ним и размер вычислительного окна, и строится зависимость производительности счёта в миллионах ячеек в секунду на одну итерацию по времени от размера окна вычислений. На рисунке 3 показаны результаты такого тестирования. Можно отметить несколько тенденций: во-



Рис. 3. Зависимость производительности вычислений от размера «окна» для разных GPU

первых, скорость вычислений растет с размером окна. Это легко объяснить, если вспомнить, как организован обмен данными между GPU и CPU (рисунок 1(c)): вне зависимости от размеров окна происходит лишь пара обменов между памятью видеокарты и оперативной памятью компьютера, то есть потери на обмен — фиксированы, а доля полезных вычислений растет с ростом размеров окна. К сожалению, размер окна ограничен сверху величиной глобальной памяти видеокарты. Видно, что для некоторых карт размера глобальной памяти видеокарты достаточно, чтобы рассматриваемся зависимость вышла на асимптоту, то есть достигается максимальный уровень производительности для алгоритма при данных условиях.



Рис. 4. Максимально достигнутый темп счёта для разных GPU

На рисунке 4 отложена максимально достигнутая в тестировании производительность счёта. Учитывая тот факт, что на одну ячейку требуется около  $10^5$  операций с плавающей точкой, как следует из данных профилировщика nvprof и аналитических соображений, можно вычислить абсолютную эффективность алгоритма как отношение реально достигнутого темпа счёта к теоретически возможному. Оказывается, что этот показатель составляет в лучших случаях до 30-40%, что, однако, является весьма и весьма высоким результатом.

## 3.2. Эффективность использования суперкомпьютера

Приступим к исследованию эффективности масштабируемости данного алгоритма на множество вычислительных узлов. Далее все тестирования проводятся на суперкомпьютере К100 Института прикладной математики им. М.В. Келдыша. Каждый узел кластера К-100 содержит три видеокарты Tesla C2050 с 448 ядрами каждая [5]. Всего доступно для 64 узла кластера. В тестах  $N_{nodes} \leq 32$ .

Рассматриваются два стандартных теста — тест на «слабую» масштабируемость, при котором проявляется поведение алгоритма при расчёте задачи очень больших размеров на большом числе узлов, и тест на «сильную» масштабируемость, когда для задачи заданного размера проверяется, какой выигрыш в производительности может дать её решение на суперкопмьютере.

Сперва рассмотрим «слабую» масштабируемость алгоритма, то есть вычислим зависимость производительности вычислений от числа узлов при запуске задачи пропорционального числа узлов размера. Возьмем вычислительную область размером  $1024 \times 64N_{nodes} \times 1024$ , где  $N_{nodes}$  — число используемых узлов кластера. На рисунке 5(a) от-



Рис. 5. Результаты weak (слева) и strong (справа) scaling алгоримта на K-100

ложен график зависимости производительности счёта в зависимости от используемого числа узлов в рассматриваемой постановке. «Идеальный» случай линейного масштабирования без потерь отложен зеленой линией, результаты реальных тестов — линией с точками. Видно, что результаты теста показывают очень хорошую «слабую» масштабируемость алгоритма.

Следующий тест — это так называемая сильная масштабируемость, когда мы запускаем задачу одних и тех же размеров на разном количестве узлов. В качестве исходной задачи возьмем постановку, на которой мы исследовали производительность GPU на одном узле, а именно  $1024 \times 512 \times 512$  узлов. На рисунке 5(b) отложен график зависимости производительности счёта в зависимости от используемого числа узлов в рассматриваемой постановке. «Идеальный» случай линейного масштабирования без потерь также отложен зеленой линией, результаты реальных тестов — линией с точками. Обыкновенно, рост производительности не является линейным, и претерпевает насыщение начиная с некоторого числа узлов. Можно заметить, что в данном тесте насыщение наступает при  $N_{nodes} \approx 32$ .

## 4. Заключение

Рассмотрен программный комплекс для численного моделирования задач газовой динамики на гетерогенных суперкомпьютерах на основе численного метода RKDG и алгоритма DiamondTorre. Приведены результаты тестирования производительности алгоритма на ПК и суперкомпьютере К-100, показывающие, что возможен расчёт с производительностью до 10<sup>8</sup> ячеек в секунду на ПК и более 10<sup>9</sup> ячеек в секунду на К-100. Такие показатели дают возможность полагать, что с помощью данного солвера можно получать численные решения задач газовой динамики с ударными волнами с недоступной ранее точностью.

#### Благодарности

Работа поддержана грантом «УМНИК» фонда содействия развитию малых форм предприятий в научно-технической сфере, а также частично грантами РФФИ 14-01-00787 и 15-01-05052.

#### Список литературы

- Bernardo Cockburn and Chi-Wang Shu. Runge-Kutta discontinuous Galerkin methods for convection-dominated problems // Journal of scientific computing, 2001. Vol. 16, no. 3, p. 173-261. ↑ 2, 5.
- [2] Boris A. Korneev and Vadim D. Levchenko. DiamondTorre GPU Implementation Algorithm of the RKDG Solver for Fluid Dynamics and its Using for the Numerical Simulation of the Bubble-shock Interaction Problem // Procedia Computer Science, 2015. Vol. 51, p. 1292 - 1302, URL http://www.sciencedirect. com/science/article/pii/S1877050915011229. ↑ 2, 3, 5, 6.
- [3] E. F. Toro. Riemann Solvers And Numerical Methods for Fluid Dynamics: A Practical Introduction: Springer, 2009.  $\uparrow$  4, 5.
- [4] J. H. J. a. G. Niederhaus J.A. and Oakley. A computational parameter study for the three-dimensional shock-bubble interaction // Journal of Fluid Mechanics, 2008. Vol. 594, p. 85–124. ↑ 6.
- [5] Вычислительные ресурсы K-100, URL http://www.kiam.ru/MVS/resourses/. ↑ 8.

Об авторах:

#### Борис Азаматович Корнеев

Московский физико-технический институт, Долгопрудный, Российская <br/> Федерация

e-mail:

boris.korneev@phystech.edu

#### Вадим Дмитриевич Левченко

Институт прикладной математики имени М.В. Келдыша РАН, Москва, Российская Федерация

e-mail:

lev@keldysh.ru

Пример ссылки на эту публикацию:

Б. А. Корнеев, В. Д. Левченко. «<sup>Д</sup><sub>Т</sub><sup>G</sup>: алгоритм и программный комплекс для расчёта сверхбольших задач газовой динамики в трёхмерной постановке на гетерогенных (супер)компьютерах», Программные системы: теория и приложения, 2015, ??:?, с. ??-??.

URL

http://psta.psiras.ru/read/

Boris Korneev, Vadim Levchenko.  $\mathbb{D}_{\mathbb{T}}^{\mathbb{G}}$ : the algorithm and software for the calculation of large-scale fluid dynamic problems in three-dimensional statement on heterogeneous (super)computers.

ABSTRACT. In this paper the software tool for the numerical simulation of threedimensional fluid dynamics problems is considered. Euler equations of inviscious gas dynamics are chosen as the mathematical model, in addition with the equations describing the multicomponent flow. Runge Kutta discontinous Galerkin (RKDG) method is used as a numerical method of solving the obtained system of equations. The special algorithm called 'DiamondTorre' to implement the considered numerical scheme for heterogenous multi-processing computers is developed. It belongs to the family of locally recursive non-locally asynchronous algorithms (LRnLA), based on the idea of space-temporal tiling of the numerical domain with respect to the stencil dependency graph. The algorithm has increased intencity of computing as well as efficient parallelization. Created software package named  ${}^{\mathbb{D}}_{\mathrm{T}}{}^{\mathrm{G}}$  makes possible to simulate the problems with the domain size of about  $10^8 - 10^9$  cells on a singe PC with GPU, and several dozens of times larger with the help of the supercomputing facilities due to good weak scaling of the algorithm.

(in Russian).

Key Words and Phrases: computational fluid dynamics, RKDG method, GPU computing, CUDA programming, effective stencil computations, LRnLA algorithms, DiamondTorre algorithm,

#### Sample citation of this publication

Boris Korneev, Vadim Levchenko " ${}^{\mathbb{D}}_{\mathbb{T}}{}^{\mathbb{G}}$ : the algorithm and software for the calculation of large-scale fluid dynamic problems in three-dimensional statement on heterogeneous (super)computers", Program systems: theory and ap*plications*, 2015, ??:?, pp. ??-??. (In Russian.)

URL

http://psta.psiras.ru/read/

<sup>©</sup> B. A. Korneev<sup>(1)</sup>, V. D. Levchenko<sup>(2)</sup>, 2015

Moscow Institute of Physics and Technology, Dolgoprudny, Russian Federation <sup>(1)</sup> 2015
 Keldysh Institute of Applied Mathematics RAS, Moscow, Russia <sup>(2)</sup> 2015

<sup>©</sup> Program systems: Theory and Applications, 2015