

ЖАЛНИН Р. В., ЛЕЩАНКИНА Т. М.

**МОДЕЛИРОВАНИЕ ГАЗОДИНАМИЧЕСКИХ ТЕЧЕНИЙ
С ИСПОЛЬЗОВАНИЕМ ТЕХНОЛОГИИ CUDA**

Аннотация. В статье описан численный алгоритм решения уравнений газовой динамики с использованием технологии CUDA. Для численного эксперимента были смоделированы условия развития неустойчивости при многократном прохождении ударной волны через контактный разрыв с начальными данными Погги. Результаты численного исследования демонстрируют возможности разработанной параллельной версии программы.

Ключевые слова: уравнения газовой динамики, параллельный алгоритм, технология CUDA.

ZHALNIN R. V., LESCHANKINA T. M.

MODELING OF GAS DYNAMIC FLOWS USING CUDA TECHNOLOGY

Abstract. This article describes a numerical algorithm for solving equations of gas dynamics using CUDA technology. The Poggie's conditions of instability by repeated passage of a shock wave through a contact discontinuity were simulated for the numerical experiment. The experiment results demonstrate the capabilities of the developed parallel version of the program.

Keywords: gas dynamics equations, parallel algorithm, CUDA technology.

Введение. В настоящее время большой практический интерес представляют решения задач газовой динамики. Данного вида задачи используются при проектировании самолетов и автомобилей, предсказании погоды, моделировании климата и т. п. Из-за нелинейности уравнений газовой динамики, для их решения необходимо использовать численные методы, которые на данный момент являются универсальными методами решения этих уравнений. В последние годы было проведено множество исследований в сфере разработки эффективных разностных схем высокого разрешения, которые позволяли бы осуществить математическое моделирование вышеуказанных процессов. Реализация данного рода задач на подобных сетках подразумевает использование больших вычислительных ресурсов. И появление новых технологий, ориентированных на использование в своей разработке графических ускорителей в дополнение к уже существующим позволяет использовать новые возможности для проведения интенсивных вычислений. Увеличение производительности видеокарт привело к тому, что количество вычислительных ядер в них достигло нескольких сотен. На сегодняшний день ведущей технологией разработки для работы на GPU (Graphical Processor Unit) является технология CUDA.

CUDA (Compute Unified Device Architecture) – это программная модель, включающая описание вычислительного параллелизма и иерархичной структуры памяти непосредственно в языке программирования [1]. Эта технология позволяет работать с языками программирования C и FORTRAN. С точки зрения программного обеспечения, реализация CUDA представляет собой кроссплатформенную систему компиляции и исполнения программ, части которых работают на CPU и GPU. CUDA предназначена для разработки GPGPU-приложений без привязки к графическим API и поддерживается всеми GPU NVIDIA.

1. Математическая модель и численная схема. Систему уравнений газовой динамики, записанную в переменных Эйлера, можно представить в виде:

$$\left\{ \begin{array}{l} \frac{\partial \rho}{\partial t} = \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y}, \\ \frac{\partial(\rho u)}{\partial t} = \frac{\partial(\rho u^2 + p)}{\partial x} + \frac{\partial(\rho uv)}{\partial y}, \\ \frac{\partial(\rho v)}{\partial t} = \frac{\partial(\rho uv)}{\partial x} + \frac{\partial(\rho v^2 + p)}{\partial y}, \\ \frac{\partial(\rho e)}{\partial t} = \frac{\partial((\rho e + p)u)}{\partial x} + \frac{\partial((\rho e + p)v)}{\partial y}. \end{array} \right. \quad (1)$$

За уравнение состояния принимается уравнение состояния идеального газа с показателем адиабаты γ : $p = (\gamma - 1)\varepsilon\rho$. В приведенной выше системе уравнений (1) $\rho = \rho(x, t)$ – плотность газа, $\vec{v} = (u, v)$ – вектор скорости частиц, $p = p(x, t)$ – давление, ε – удельная внутренняя энергия на единицу массы, $e = \varepsilon + \frac{u^2 + v^2}{2}$ – полная энергия, $\gamma = \frac{c_p}{c_v}$ – показатель адиабаты, c_p, c_v – теплоёмкость при постоянном давлении и постоянном объеме соответственно.

В векторной форме система уравнений, принимает вид:

$$\frac{\partial U}{\partial t} + \frac{\partial F(U)}{\partial x} + \frac{\partial G(U)}{\partial y} = 0, \quad (2)$$

где

$$U = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho e \end{pmatrix}, F(U) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho e + p)u \end{pmatrix}, G(U) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho e + p)v \end{pmatrix}. \quad (3)$$

Для аппроксимации системы уравнений газовой динамики (2) использовалась нелинейная консервативная квазимоноотонная дифференциально-разностная схема [3–5]:

$$\frac{U_{ij}^{n+1} - U_{ij}^n}{\tau} + \frac{F_{i+1/2j} - F_{i-1/2j}}{h_x} + \frac{G_{ij+1/2} - G_{ij-1/2}}{h_y} = 0, \quad (4)$$

где $F_{i+1/2j}, G_{ij+1/2}$ – дискретные потоки.

Дискретные потоки на гранях между ячейками вычислялись как функции двух переменных по интерполированным значениям газодинамических параметров на гранях между ячейками [4; 5]:

$$\begin{aligned} F_{i+\frac{1}{2}j} &= F\left(U_{i+\frac{1}{2}j}^l, U_{i+\frac{1}{2}j}^r\right), \\ G_{ij+\frac{1}{2}} &= Y\left(U_{ij+\frac{1}{2}}^l, U_{ij+\frac{1}{2}}^r\right), \end{aligned} \quad (5)$$

Здесь $U_{ij+1/2}^l, U_{ij+1/2}^r$ – "левые" и "правые" значения вектора U на грани между i и $i + 1$ ячейками, для которой вычисляется поток $F_{i+1/2j}$, $U_{ij+1/2}^a$, ($a = r, l$) "левые" и "правые" значения вектора U на грани между j и $j + 1$ ячейками соответственно. Для того чтобы вычислить значения вектора U , на указанных гранях между ячейками введем новый вектор переменных $Y = Y(U)$. Проведем его интерполяцию на грань между ячейками i , затем, пересчитаем искомое значение вектора $U = U(Y)$ на данной грани. В расчетах полагалось $Y = (\rho, p, u, v)$, потоки вычислялись по схеме Лакса-Фридрихса-Русанова:

$$\begin{aligned} F_{i+\frac{1}{2}j} &= \frac{1}{2} \left[\left(F\left(U_{i+\frac{1}{2}j}^r\right) + F\left(U_{i+\frac{1}{2}j}^l\right) \right) - \alpha_1 \left(U_{i+\frac{1}{2}j}^r - U_{i+\frac{1}{2}j}^l \right) \right], \\ G_{ij+\frac{1}{2}} &= \frac{1}{2} \left[\left(G\left(U_{ij+\frac{1}{2}}^r\right) + G\left(U_{ij+\frac{1}{2}}^l\right) \right) - \alpha_2 \left(U_{ij+\frac{1}{2}}^r - U_{ij+\frac{1}{2}}^l \right) \right], \end{aligned} \quad (6)$$

где

$$\begin{aligned} \alpha_1 &= \max\{|u_{ij}| + c_{ij}, |u_{i+1j}| + c_{i+1j}\}, \\ \alpha_2 &= \max\{|v_{ij}| + c_{ij}, |v_{ij+1}| + c_{ij+1}\}, \end{aligned} \quad (7)$$

2. Реализация алгоритма на языке CUDA C. Графический ускоритель использовался для проведения расчетов численных потоков на границах ячеек и вычисления ограничителей.

Для проведения расчетов на GPU необходимо выполнить следующие действия:

1. Выделить память для данных и потоков на GPU. Пример: выделяется память под газодинамические параметры:

```
cudaMalloc((void**)&r_dev, (ntri_new_m[0])*sizeof(float));
cudaMalloc((void**)&p_dev, (ntri_new_m[0])*sizeof(float));
cudaMalloc((void**)&u_dev, (ntri_new_m[0])*sizeof(float));
cudaMalloc((void**)&v_dev, (ntri_new_m[0])*sizeof(float));
```

где первым параметром идет указатель, в который записывается адрес выделенной памяти, а вторым – размер выделяемой памяти в байтах.

2. Скопировать данные на GPU. Пример: газодинамические параметры (r,p,u,v) копируем на GPU по средствам использования функции:

```
cudaMemcpy(r_dev, r, (ntri_new_m[0])*sizeof(float), cudaMemcpyHostToDevice);
cudaMemcpy(p_dev, p, (ntri_new_m[0])*sizeof(float), cudaMemcpyHostToDevice);
cudaMemcpy(u_dev, u, (ntri_new_m[0])*sizeof(float), cudaMemcpyHostToDevice);
cudaMemcpy(v_dev, v, (ntri_new_m[0])*sizeof(float), cudaMemcpyHostToDevice);
```

где первым параметром идет указатель, содержащий адрес места-назначения копирования; вторым – указатель, содержащий адрес источника копирования; следом – размер копируемого ресурса в байтах; далее – параметр, указывающий направление копирования.

3. Распределить ядра для вычисления.

Расчет газодинамических параметров состоит из последовательности вызываемых процедур, запускаемых на GPU на каждом шаге по времени:

```
while (t <= tmax) {
    boundcond<<<blocks, threads>>> (r_dev, p_dev, u_dev, v_dev,
        d1_dev, d2_dev, d3_dev, d4_dev, d5_dev, d6_dev,
        otr_u_dev, otr_u1_dev, otr_v_dev, otr_v1_dev,
        alpha1_dev, alpha2_dev, sootv_dev, cm);
    calcU<<<blocks, threads>>>(r_dev, p_dev, u_dev, v_dev,
        ro_dev, ru_dev, rv_dev, en_dev, ntri_dev);
    ENO_mass<<<blocks, threads>>>(ux_m_dev, uy_m_dev, vx_m_dev, vy_m_dev,
        rx_m_dev, ry_m_dev, px_m_dev, py_m_dev,
        ntri_dev, tri_neigh1_dev, cm_dev,
        r_dev, u_dev, v_dev, p_dev, ntri_new_dev);
    calcFluxes<<<blocks, threads>>>(r_dev, p_dev, u_dev, v_dev,
        rtmp_dev, utmp_dev, vtmp_dev, etmp_dev,
        ux_m_dev, uy_m_dev, vx_m_dev, vy_m_dev,
        rx_m_dev, ry_m_dev, px_m_dev, py_m_dev,
        x1_dev, tri_neigh1_dev, tri_v1_dev,
        cm_dev, ntri_dev, ntri_new_dev, nx_new_dev);
    gasDinStep<<<blocks, threads>>>(rn_dev, pn_dev, un_dev, vn_dev,
        ro_dev, ru_dev, rv_dev, en_dev,
        rtmp_dev, utmp_dev,
        vtmp_dev, etmp_dev, s_dev, ntri_dev);
    t = t+tau;
    step++;
}
```

Процедура `boundcond<<<blocks, threads>>>()` определяет граничные условия.

Процедура $\text{calcU}(\langle\langle\text{blocks}, \text{threads}\rangle\rangle)()$ вычисляет консервативные переменные в соответствии с формулой (3).

В процедуре $\text{ENO_mass}(\langle\langle\text{blocks}, \text{threads}\rangle\rangle)()$ реализован алгоритм реконструкции газодинамических параметров на границе ячеек.

В процедуре $\text{calcFluxes}(\langle\langle\text{blocks}, \text{threads}\rangle\rangle)()$ рассчитываем дискретные потоки по схеме Лакса-Фридрихса в соответствии с формулой (8).

Процедура $\text{gasDinStep}(\langle\langle\text{blocks}, \text{threads}\rangle\rangle)()$ рассчитывает значения консервативных, а затем примитивных переменных на каждом шаге по времени.

4. Скопировать полученные данные обратно в CPU. Пример: газодинамические параметры (r,p,u,v) копируем на CPU

```
cudaMemcpy(r, rn_dev, (ntri_new_m[0])*sizeof(float), cudaMemcpyDeviceToHost);  
cudaMemcpy(p, pn_dev, (ntri_new_m[0])*sizeof(float), cudaMemcpyDeviceToHost);  
cudaMemcpy(u, un_dev, (ntri_new_m[0])*sizeof(float), cudaMemcpyDeviceToHost);  
cudaMemcpy(v, vn_dev, (ntri_new_m[0])*sizeof(float), cudaMemcpyDeviceToHost);
```

5. Освободить память GPU.

```
cudaFree(r_dev);  
cudaFree(p_dev);  
cudaFree(u_dev);  
cudaFree(v_dev);
```

где единственным параметром функции является указатель освобождаемой памяти.

3. Вычислительный эксперимент. Большой интерес с практической точки зрения представляет задача взаимодействия ударной волны с зоной турбулентного перемешивания. Такие виды задач являются хорошими тестовыми примерами для проверки различных алгоритмов, методов и программ. На практике такие задачи позволяют изучить ход образования вихрей при контактном взаимодействии нескольких газов с разными плотностями и уравнениями состояний [6].

Выполнено численное исследование динамики развития газодинамической неустойчивости в контактной границе элегаз-воздух для условий эксперимента на ударной трубе [7]. Параметры ударной трубы следующие: длина равна 0,4 м, а ширина равна 0,04 м. Труба разделена на три области:

1. область длиной 0,08 м, заполнена элегазом с высоким давлением ($\text{SF}_6\text{-HI}$);
2. область длиной 0,02 м, заполнена элегазом с низким давлением ($\text{SF}_6\text{-LO}$);
3. область длиной 0,03 м, заполнена воздухом (AIR).

В качестве расчетных данных была принята следующая физическая постановка задачи: начальная температура 291 К в камере низкого давления давление составляет 10^{-4}

ГПа = 1 бар. Непосредственно за ударной волной в элегазе давление составляет 2,152 бар, плотность – $1,209 \cdot 10^{-2}$ г/см³, скорость ударной волны – 195,2 м/с, скорость течения за ударной волной – 97,76 м/с. Начальные плотности элегаза и воздуха в камере низкого давления равны $6,037 \cdot 10^{-3}$ и $1,198 \cdot 10^{-3}$ г/см³ соответственно. Физические свойства элегаза и воздуха следующие: оба вещества являются невязкими, нетеплопроводными и идеальными газами с показателями адиабаты $\gamma = 1,094$ (SF_6) и $\gamma = 1.4$ (воздух), отношение молекулярных масс (SF_6 /воздух) принято равным 5,04.

Необходимо смоделировать развитие неустойчивости при многократном прохождении ударной волны через контактный разрыв.

На рисунке 1 приведен график плотности в начальный момент времени. На рисунках 2 и 3 представлены изменения плотности при первом и повторном прохождении ударной волны через контактную границу.

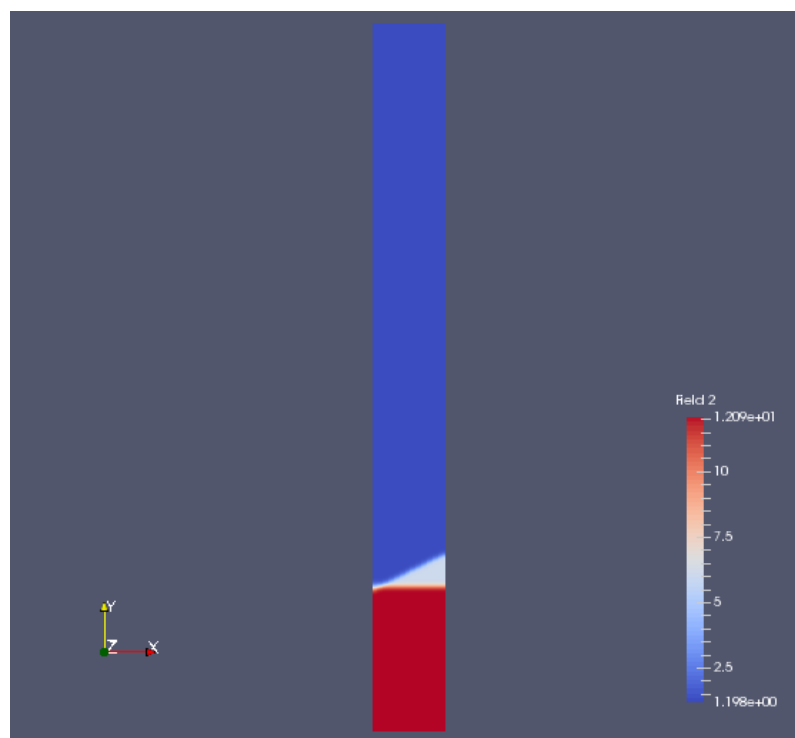


Рис. 1. Изменение плотности в начальный момент времени.

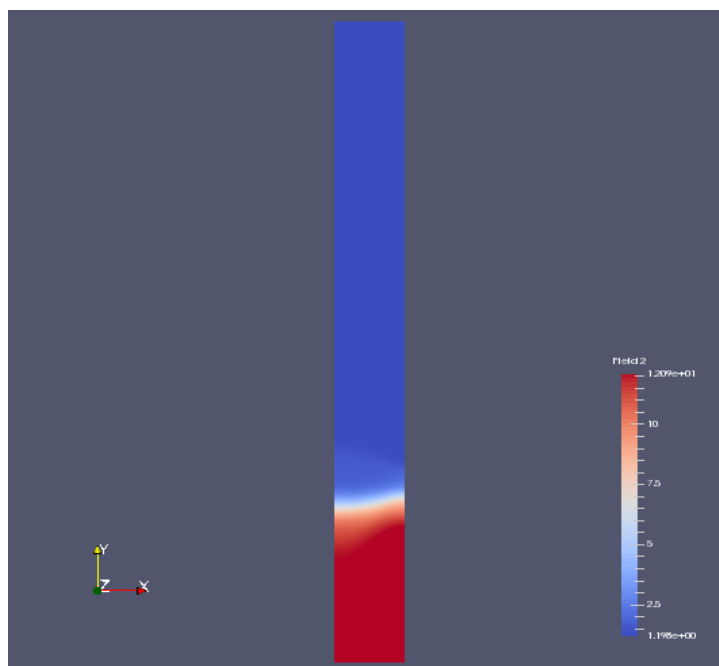


Рис. 2. Изменение плотности при первом прохождении ударной волны через контактную границу.

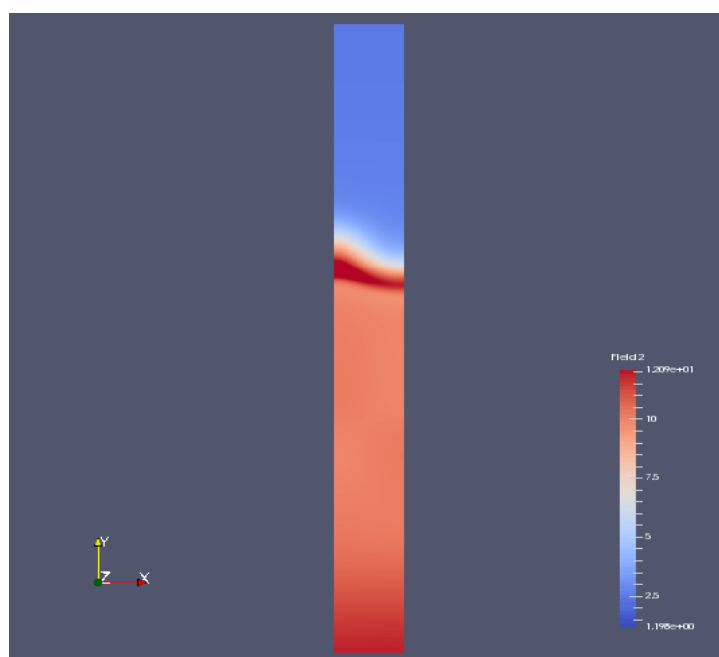


Рис. 3. Изменение плотности при втором прохождении ударной волны через контактную границу.

Заключение. Разностная схема и программа расчета были протестированы на задаче с известными экспериментальными данными [7]. Результаты численных расчетов показали соответствие теоретическим предположениям и экспериментальным данным. Таким образом, проведенное тестирование предложенной схемы и программы моделирования

гидродинамических течений показывает, что данную схему можно использовать при решении задач нестационарной газовой динамики.

ЛИТЕРАТУРА

1. NVIDIA CUDA. Programming Guide [Электронный ресурс]. – Режим доступа: <http://developer.nvidia.com/cuda-toolkit-40>.
2. Kurganov A., Chi-Tien L. On the reduction of numerical dissipation in central-upwind schemes // Commun. Comput. Phys. February. – 2007. – Vol. 2, No. 1. – pp. 141–163.
3. Жалнин Р. В. О построении параллельного вычислительного алгоритма высокого порядка точности для гиперболических систем уравнений // Журнал Средневолжского математического общества. – 2007. – Т. 9, № 1. – С. 145–153.
4. Жалнин Р. В., Змитренко Н. В., Ладонкина М. Е., Тишкин В. Ф. Численное моделирование развития неустойчивости Рихтмайера-Мешкова с использованием схем высокого порядка точности // Математическое моделирование. – 2007. – Т. 19, № 10. – С. 61–66.
5. Ладонкина М. Е. Численное моделирование турбулентного перемешивания с использованием высокопроизводительных систем: дисс. ... канд. физ.-мат. наук. – М., 2005. – 157 с.
6. Lebo I. G., Nikishin V. V., Rozanov V. B., Tishkin V. F. On the development of instability near the contact boundary between two equal-density gases in passage of a shock wave // Bulletin of the Lebedev Physics Institute (Kratkie Soobsheniya po Fizike). – New York: Allerton Press, Inc, 1997.
7. Poggi F., Thorembe M.-H., Rodriguez G. Velocity measurements in turbulent gaseous mixtures induced by Richtmyer-Meshkov instability // Physics of Fluids. – 1998. – Vol. 10, No. 11. – P. 2698–2700.