

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ  
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)



КУРСОВОЙ ПРОЕКТ ПО КУРСУ  
«МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ»

---

**Численное моделирование двумерной  
нестационарной задачи газовой динамики  
с использованием MPI и CUDA**

---

*Студент:*  
Колесников Е. В.

*Преподаватель:*  
Тишкин В. Ф.

27 апреля 2016 г.

# Содержание

<b>1 Физическая постановка задачи</b>	<b>2</b>
1.1 Исследуемый объект . . . . .	2
<b>2 Математическая модель</b>	<b>3</b>
2.1 Двумерные уравнения газовой динамики . . . . .	3
<b>3 Вычислительный алгоритм</b>	<b>4</b>
3.1 Идея алгоритма . . . . .	4
3.2 Метод Лакса-Фридригса . . . . .	5
3.3 Задача распада произвольного разрыва . . . . .	6
3.4 Окончательный алгоритм решения . . . . .	6
<b>4 Программа на ЭВМ</b>	<b>8</b>
4.1 Архитектура программы . . . . .	8
4.2 Результаты работы . . . . .	10
<b>5 Вывод</b>	<b>11</b>
<b>Список литературы</b>	<b>12</b>

# 1 Физическая постановка задачи

## 1.1 Исследуемый объект

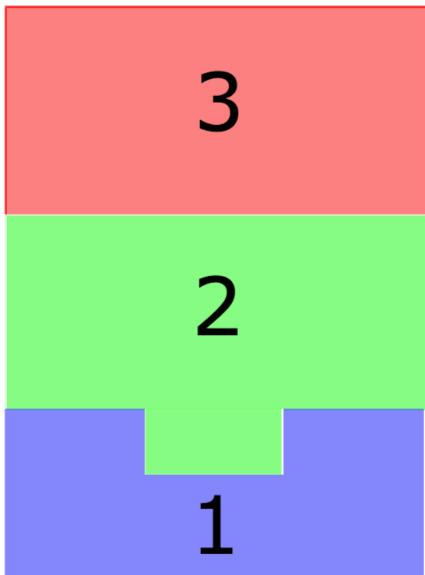


Рис. 1: Начальные области

Задача состоит в том, чтобы исследовать поведение жидкости в конечном объеме с течением времени. По условию задачи исходный объем разделен на три области (Рис. 1):

- Параметры первой области:

$$\{\rho = 1.0, v_x = 0.0, v_y = 0.0, e = 1.0\}.$$

- Параметры второй области:

$$\{\rho = 5.0, v_x = 0.0, v_y = 0.0, e = 0.2\}.$$

- Параметры третьей области определяются из условий Гюгонио с параметром маха  $M = 2.0$  (и учетом того, что плотность третьей области больше плотности второй области из-за увеличения энтропии), т.к. в системе присутствует ударная волна, расположенная на стыке третьей и второй областей и идущая вниз вдоль оси  $Y$ .

Условия Гюгонио:

$$\begin{cases} \rho(D - v) = 5D \\ \rho(Dv - \frac{2}{3}e - v^2) = \rho \frac{2}{15} \\ \rho(D(e + \frac{v^2}{2}) - v(\frac{5}{3}e + \frac{v^2}{2})) = D \\ D = M\sqrt{\gamma(\gamma - 1)e^{(2)}} = 2\sqrt{\frac{2}{9}} \end{cases} \quad (1.1.1)$$

Выражая из условий Гюгонио параметры третьей области получаем следующие значения:

$$\{\rho = 20.0, v_x = 0.0, v_y = 0.707107, e = 0.05\}.$$

В качестве граничных условий используется условие “жесткой стенки” или “условие отражения”.

Условие отражения заключается в том, что вокруг исследуемой области добавляется один ряд ячеек со специальными параметрами (Рис. 2):

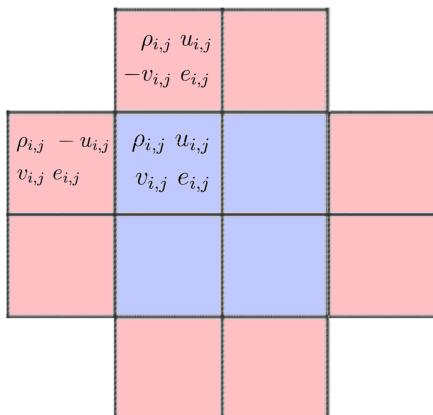


Рис. 2: Условие отражения

аналогично если граничная ячейка соседствует с “обычной” вдоль оси  $Y$ , то все параметры граничной ячейки эквивалентны параметрам “обычной” ячейки, за исключением скорости по оси  $X$ , которая в свою очередь получается домножением исходной скорости на  $-1$ ,

- аналогично если граничная ячейка соседствует с “обычной” вдоль оси  $Y$ , то все параметры граничной ячейки эквивалентны параметрам “обычной” ячейки, за исключением скорости по оси  $Y$ , которая в свою очередь получается домножением исходной скорости на  $-1$ .

## 2 Математическая модель

### 2.1 Двумерные уравнения газовой динамики

Двумерная система уравнений газовой динамики, описывающая изменение параметров жидкости или газа с течением времени в двумерном пространстве, представляется в виде:

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial \rho u}{\partial x} + \frac{\partial \rho v}{\partial y} = 0, \\ \frac{\partial \rho u}{\partial t} + \frac{\partial(p + \rho u^2)}{\partial x} + \frac{\partial \rho uv}{\partial y} = 0, \\ \frac{\partial \rho v}{\partial t} + \frac{\partial \rho uv}{\partial x} + \frac{\partial(p + \rho v^2)}{\partial y} = 0, \\ \frac{\partial \rho \left(e + \frac{u^2 + v^2}{2}\right)}{\partial t} + \frac{\partial \rho u \left(e + \frac{p}{\rho} + \frac{u^2 + v^2}{2}\right)}{\partial x} + \frac{\partial \rho v \left(e + \frac{p}{\rho} + \frac{u^2 + v^2}{2}\right)}{\partial y} = 0, \\ p = (\gamma - 1)\rho e \Leftrightarrow pV = \nu RT, \end{cases} \quad (2.1.1)$$

где параметры  $\{\rho, p, u, v, e, \gamma\}$  определяются следующим образом:

- $\rho$  – плотность элементарного объема;
- $p$  – давление элементарного объема;
- $u$  – скорость по оси  $X$  элементарного объема;
- $v$  – скорость по оси  $Y$  элементарного объема;
- $e$  – внутренняя энергия элементарного объема;
- $\gamma$  – показатель адиабаты (в нашем случае  $\gamma = \frac{5}{3}$ ).

Система уравнений газовой динамики в двумерном пространстве состоит из пяти уравнений: первые четыре – законы сохранения, а последнее – уравнение состояния:

- (1) – закон сохранения массы,
- (2-3) – законы сохранения импульса по разным направлениям,
- (4) – закон сохранения энергии,
- (5) – уравнение состояния (связь термодинамических параметров системы), которым для идеальных газов является уравнение Клапейрона-Менделеева. Мы привыкли записывать уравнение Клапейрона-Менделеева в форме  $pV = \nu RT$ , но в газовой динамике удобнее записывать его в форме  $p = (\gamma - 1)\rho e$ .

### 3 Вычислительный алгоритм

#### 3.1 Идея алгоритма

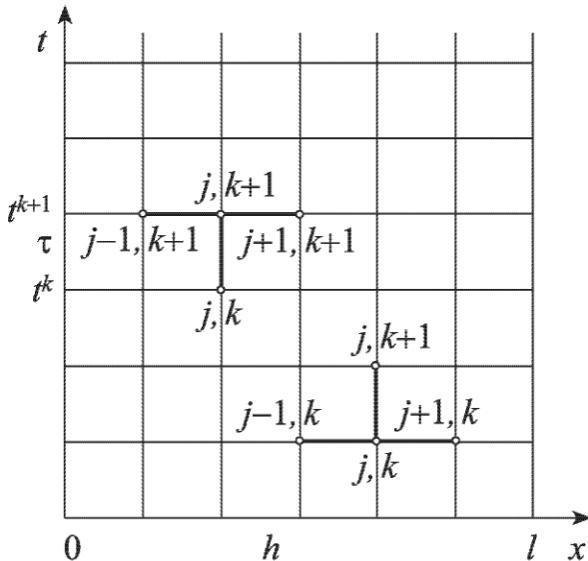
Для численного решения системы (2.1.1) используется метод контрольного объема (интегро-интерполяционный метод).

Если метод конечных разностей решает задачу в дифференциальной форме, которая была приведена на слайде, то метод конечных объемов решает задачу в интегральной форме. Следовательно необходимо переписать систему уравнений в интегральной форме:

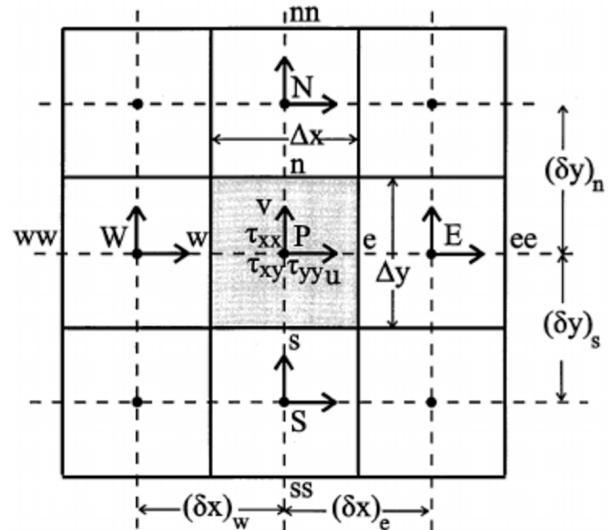
$$\left\{ \begin{array}{l} \iint \rho dx dy + \rho u dy dt + \rho v dx dt = 0 \\ \iint \rho u dx dy + (p + \rho u^2) dy dt + \rho u v dx dt = 0 \\ \iint \rho v dx dy + \rho u v dy dt + (p + \rho v^2) dx dt = 0 \\ \iint \rho \left( e + \frac{u^2 + v^2}{2} \right) dx dy + \rho u \left( e + \frac{p}{\rho} + \frac{u^2 + v^2}{2} \right) dy dt + \\ \quad + \rho v \left( e + \frac{p}{\rho} + \frac{u^2 + v^2}{2} \right) dx dt = 0 \\ p = (\gamma - 1)\rho e \end{array} \right.$$

Перевод системы из дифференциальной формы в интегральную производится с помощью теоремы Остроградского-Гаусса, теоремы Грина и теоремы Стокса из курса дифференциальной геометрии.

Для того, чтобы численно решить такую систему необходимо ввести дискретную сетку в пространстве, однако не таким образом, как это делается при решении методами конечных разностей, когда узлы клетки находятся в углах ячеек, а таким, чтобы узлы клеток были в центре ячеек, как видно на (Рис. 3). Все величины  $\{\rho, p, u, v, e\}$  внутри



(a) Метод конечных разностей



(b) Метод конечных объемов

Рис. 3: Построение сеток

одной ячейки считаются постоянными и им присваиваются номера этой ячейки.

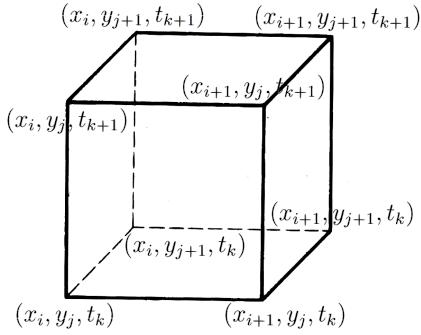


Рис. 4: Поверхность интегрирования

В интегральной записи интеграл по замкнутой поверхности означает интегрирование по поверхности трехмерного куба, где двумя основаниями куба являются элементарные ячейки с одними и теми же пространственными индексами, но в разные моменты времени, как видно на (Рис. 4).

Для того, чтобы решить интегральную систему, раскроем каждое интегральное уравнение отдельно и упростим.

Рассмотрим первое уравнение системы. Если раскрыть подробно этот двойной интеграл по замкнутому контуру, то получится уравнение такого вида:

$$\begin{aligned} \rho_{i,j}^{k+1} = \rho_{i,j}^k - \frac{1}{h_x h_y} & \left( \int_{t_k}^{t_{k+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} (\rho u)_{x=x_{i+\frac{1}{2}}^k} dy dt - \int_{t_k}^{t_{k+1}} \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} (\rho u)_{x=x_{i-\frac{1}{2}}^k} dy dt + \right. \\ & + \int_{t_k}^{t_{k+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\rho v)_{y=y_{j+\frac{1}{2}}^k} dx dt - \left. \int_{t_k}^{t_{k+1}} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (\rho v)_{y=y_{j-\frac{1}{2}}^k} dx dt \right) \end{aligned}$$

В нашей задаче предполагается, что величины  $\{\rho u, \rho v\}$  на границах ячейки постоянны, хотя по факту это не так. Для этих постоянных величин мы примем обозначения  $\{RU, RV\}$  (“большие” величины). В новых обозначениях уравнение перепишется в виде:

$$\rho_{i,j}^{k+1} = \rho_{i,j}^k - \frac{\tau}{h_x} \left( (RU)_{i+\frac{1}{2}, j} - (RU)_{i-\frac{1}{2}, j} \right) - \frac{\tau}{h_y} \left( (RV)_{i, j+\frac{1}{2}} - (RV)_{i, j-\frac{1}{2}} \right)$$

Аналогичным образом можно переписать все интегральные уравнения. Получающаяся система хоть и абсолютно правильная, но очень громоздкая и некрасивая. Ее можно переписать в очень удобном векторном виде. Если посмотреть на исходную систему, то можно заметить, что она имеет определенный вид:

$$\frac{\partial \cdot}{\partial t} + \frac{\partial \cdot}{\partial x} + \frac{\partial \cdot}{\partial y} = 0.$$

### 3.2 Метод Лакса-Фридригса

Для того, чтобы определить метод Лакса-Фридригса, необходимо сперва задать систему в векторном виде:

$$\frac{\partial f}{\partial t} + \frac{\partial F(f)}{\partial x} + \frac{\partial G(f)}{\partial y} = 0.$$

Уравнения такого вида называют уравнениями переноса.

Это возможно сделать, произведя замену переменных:

$$f = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho \left( e + \frac{u^2 + v^2}{2} \right) \end{pmatrix} \quad F = \begin{pmatrix} \rho u \\ (p + \rho u^2) \\ \rho u v \\ \rho u \left( e + \frac{p}{\rho} + \frac{u^2 + v^2}{2} \right) \end{pmatrix} \quad G = \begin{pmatrix} \rho v \\ \rho u v \\ (p + \rho v^2) \\ \rho v \left( e + \frac{p}{\rho} + \frac{u^2 + v^2}{2} \right) \end{pmatrix}$$

Разностная формула уравнения переноса записывается в виде:

$$\frac{f_{ij}^{k+1} - f_{ij}^k}{\tau} + \frac{F_{i+\frac{1}{2},j}^k - F_{i-\frac{1}{2},j}^k}{h_x} + \frac{G_{i,j+\frac{1}{2}}^k - G_{i,j-\frac{1}{2}}^k}{h_y} = 0$$

Т.о. мы получили систему уравнений с единственной проблемой: мы не знаем значений на границах ячеек, эти значения в скалярной форме были обозначены “большими” буквами. Для нахождения этих значений необходимо решить так называемую задачу распада произвольного разрыва.

### 3.3 Задача распада произвольного разрыва

Два элементарных объема, разделенных бесконечно тонкой пластинкой, содержат газ с разными термодинамическими характеристиками, например с разными плотностями. Исходная функция распределения плотности разрывна. Необходимо найти распределение плотности газа на стыке двух объемов после того, как пластинку уберут. Визуально представить задачу можно с помощью (Рис. 5).

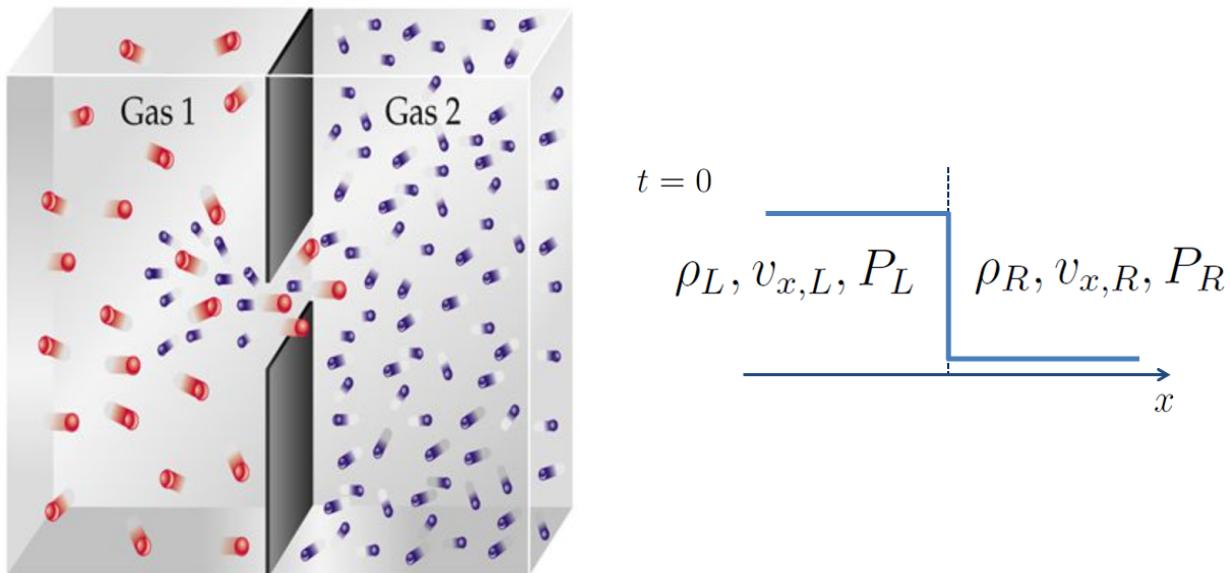


Рис. 5: Задача распада произвольного разрыва

### 3.4 Окончательный алгоритм решения

Для удобства записи введем обозначения:

$$f_{ij}^k := f_{ij} \text{ в момент времени } k$$

$$F_{ij}^k := F(f_{ij}^k)$$

$$G_{ij}^k := G(f_{ij}^k)$$

$\tau$  – временной шаг

$h_x, h_y$  – шаги по пространственным переменным

В таких обозначениях окончательный вид алгоритма Лакса-Фридригса можно запи-

сать в виде:

$$\left\{ \begin{array}{l} \frac{f_{ij}^{k+1} - f_{ij}^k}{\tau} + \frac{F_{i+\frac{1}{2},j}^k - F_{i-\frac{1}{2},j}^k}{h_x} + \frac{G_{i,j+\frac{1}{2}}^k - G_{i,j-\frac{1}{2}}^k}{h_y} = 0 \\ F_{i+\frac{1}{2},j}^k = \frac{1}{2}(F_{i+1,j}^k + F_{i,j}^k) - \frac{1}{2}D_{i+1,j}^k(f_{i+1,j}^k - f_{i,j}^k) \\ F_{i-\frac{1}{2},j}^k = \frac{1}{2}(F_{i,j}^k + F_{i-1,j}^k) - \frac{1}{2}D_{i-1,j}^k(f_{i,j}^k - f_{i-1,j}^k) \\ G_{i,j+\frac{1}{2}}^k = \frac{1}{2}(G_{i,j+1}^k + G_{i,j}^k) - \frac{1}{2}D_{i,j+1}^k(f_{i,j+1}^k - f_{i,j}^k) \\ G_{i,j-\frac{1}{2}}^k = \frac{1}{2}(G_{i,j}^k + G_{i,j-1}^k) - \frac{1}{2}D_{i,j-1}^k(f_{i,j}^k - f_{i,j-1}^k) \\ D_{i+1,j}^k = \max(|u_{i,j}^k| + c_{i,j}^k, |u_{i+1,j}^k| + c_{i+1,j}^k) \\ D_{i-1,j}^k = \max(|u_{i,j}^k| + c_{i,j}^k, |u_{i-1,j}^k| + c_{i-1,j}^k) \\ D_{i,j+1}^k = \max(|v_{i,j}^k| + c_{i,j}^k, |v_{i,j+1}^k| + c_{i,j+1}^k) \\ D_{i,j-1}^k = \max(|v_{i,j}^k| + c_{i,j}^k, |v_{i,j-1}^k| + c_{i,j-1}^k) \\ c_{i,j}^k = \sqrt{\gamma(\gamma-1)e_{i,j}^k} \end{array} \right. \quad (3.4.1)$$

После того, как выразили  $f_{ij}^{k+1}$ , используем уравнение состояния для того, чтобы выразить каждый параметр отдельно:

$$f = \begin{pmatrix} \rho_{i,j}^{k+1} \\ \rho_{i,j}^{k+1}u_{i,j}^{k+1} \\ \rho_{i,j}^{k+1}v_{i,j}^{k+1} \\ \rho_{i,j}^{k+1}\left(e_{i,j}^{k+1} + \frac{(u_{i,j}^{k+1})^2 + (v_{i,j}^{k+1})^2}{2}\right) \end{pmatrix} \Rightarrow \begin{pmatrix} \rho_{i,j}^{k+1} \\ u_{i,j}^{k+1} \\ v_{i,j}^{k+1} \\ e_{i,j}^{k+1} \end{pmatrix}$$

Метод Лакса-Фридригса является явной схемой аппроксимации решения исходной системы: в момент времени  $n$  параметры известны, из уравнений определяются параметры в момент времени  $n+1$ . Данная схема, являясь явной, не является абсолютно устойчивой, поэтому, измельчая шаг по пространству, необходимо уменьшать шаг по времени. Схема является консервативной, т.к. сконструирована на основе интегро-интерполяционного метода.

## 4 Программа на ЭВМ

### 4.1 Архитектура программы

Программа написана с использованием технологий MPI и CUDA. Идея использования MPI, заложенная в программу, заключается в том, чтобы разбить исследуемую область вдоль оси  $X$  на равные блоки (подпространства), над которыми в параллельном режиме будут “работать” различные MPI узлы. Идею MPI архитектуры можно видеть на (Рис. 6), на котором показан более общий случай исследования 3D пространства. Параметрами

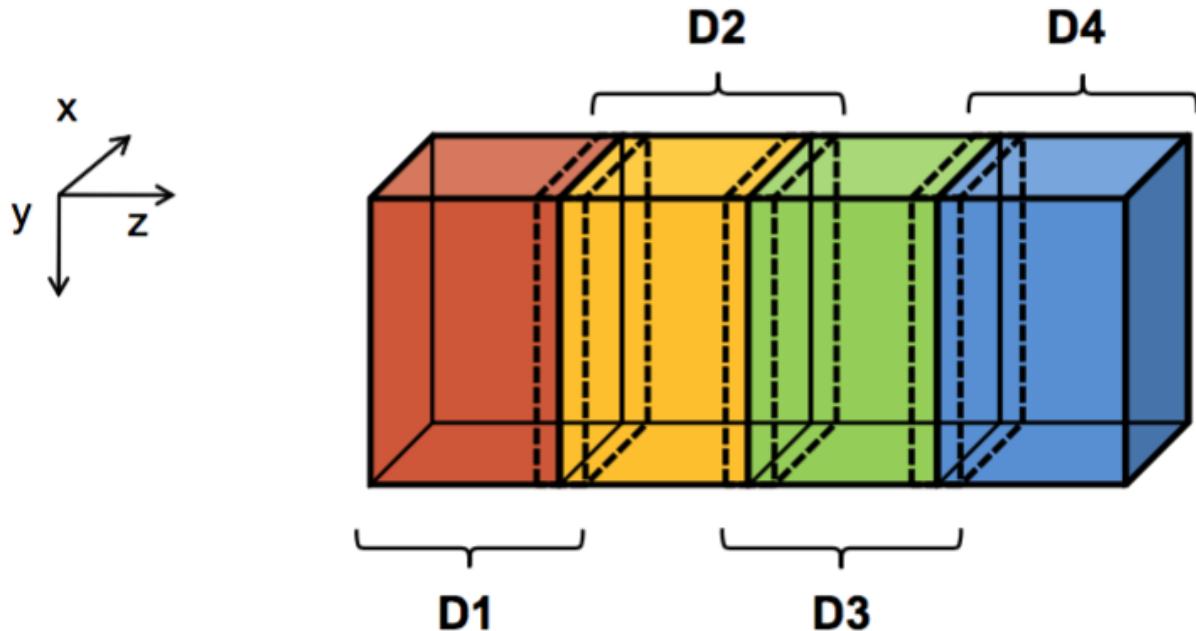


Рис. 6: MPI архитектура

$\{D1, D2, D3, D4\}$  обозначаются подпространства, обрабатывающиеся параллельно различными MPI узлами.

Помимо вычислительных узлов, необходим также серверный узел, основной задачей которого является операции ввода-вывода и, в конкретной реализации, создании изображений для исследования результатов работы программы.

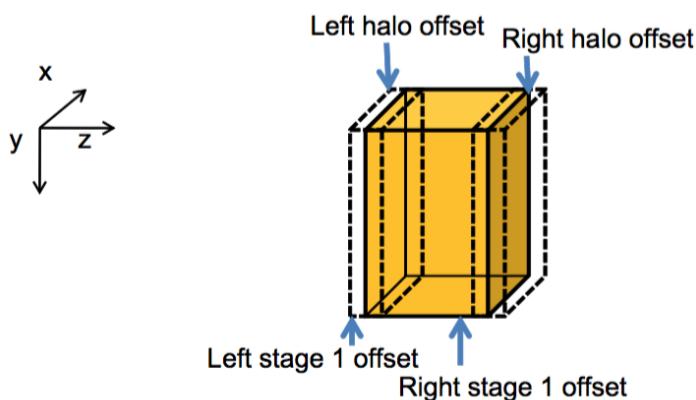


Рис. 7: Границные и halo ячейки

каждый узел будет обмениваться с каждым соседним узлом одним рядом ячеек для того, чтобы у каждого узла всегда была корректная информация о параметрах пространства вокруг. На (Рис. 7) можно видеть, что граничные ячейки показаны в виде пунктирной

Помимо ячеек самого узла необходимо также хранить один дополнительный ряд ячеек вокруг, для того, чтобы грамотно обрабатывать граничные ячейки. И если ячейки дополнительного ряда по оси  $Y$  являются фиктивными ячейками, то по оси  $X$  – ячейки соседних узлов, которые вычисляются независимо от рассматриваемого узла. Поэтому при создании архитектуры программы необходимо учитывать тот факт, что

области внутри исследуемого (желтого) подпространства, в то время как halo область, относящаяся к подпространству соседнего узла, показана в виде пунктирной области вне исследуемого (желтого) подпространства.

На (Рис. 6) можно видеть, что  $i$ -ый узел использует один ряд ячеек  $(i - 1)$ -ого узла и один ряд ячеек  $(i + 1)$ -ого узла. Фиктивные ячейки обновляются на каждой итерации, точно так же как ячейки соседних узлов. Каждый MPI блок в свою очередь мы будем делять на CUDA блоки, но, в отличии от MPI блоков, CUDA блоки будут покрывать исследуемое подпространство двумерно, как видно на (Рис. 8).

Точно также, как и MPI блокам, каждому CUDA блоку необходимо хранить ряд дополнительных ячеек вокруг для корректного вычисления. Для ускорения работы каждый CUDA блок с дополнительным рядом ячеек можно загружать в разделяемую память, обрабатывать и только потом выгружать обратно в глобальную память. Ускорение достигается благодаря тому, что каждая CUDA нить обращается к четырем своим соседям при расчетах.

Вычисления на каждой итерации можно разбить на два этапа для увеличения скорости работы программы, как показано на (Рис. 9). Первым делом необходимо произвести

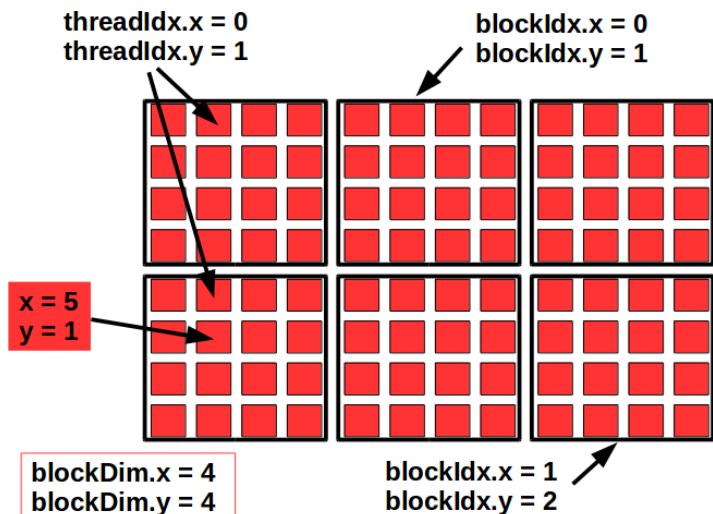


Рис. 8: Двумерное разбиение подпространства

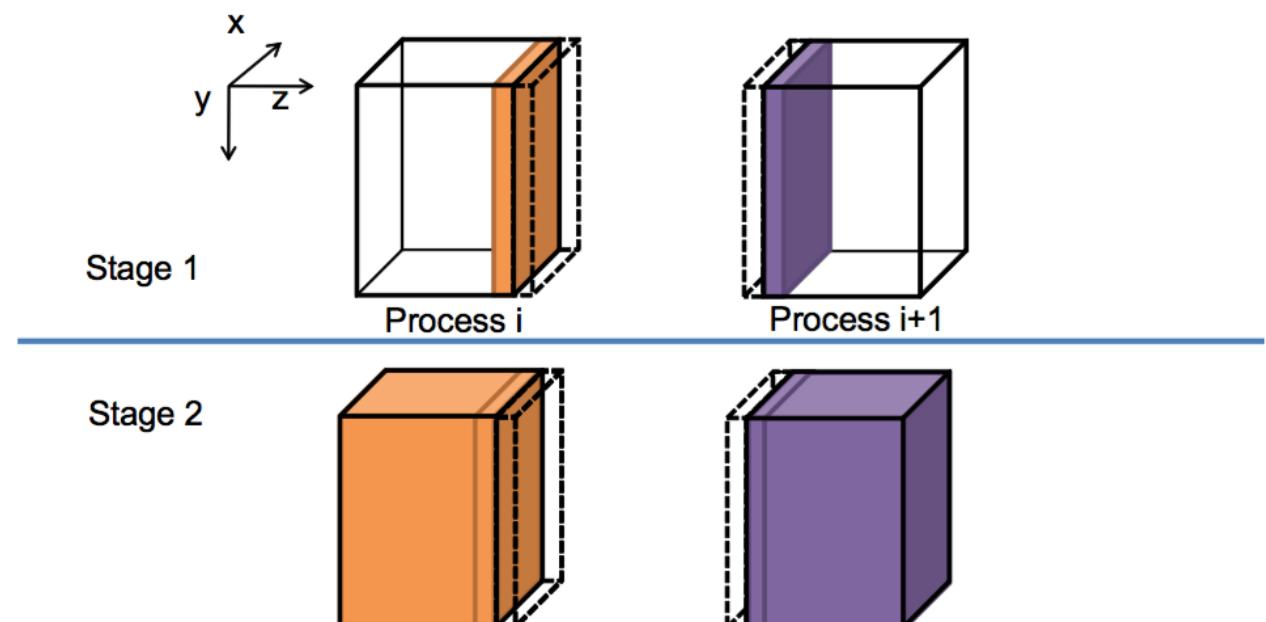


Рис. 9: Этапы вычислений

вычисление граничных ячеек. Делается этого для того, чтобы на следующем этапе вычислений параллельно производить вычисление внутренних ячеек подпространства и обмен граничных и halo ячеек между соседними MPI узлами.

## 4.2 Результаты работы

Результатом работы программы является четыре видеофайла, каждый из которых показывает изменяющееся во времени поле либо плотности, либо скорости по оси  $X$ , либо скорости по оси  $Y$ , либо энергии.

Несколько скриншотов работы программы можно видеть на (Рис. 10) и (Рис. 11).

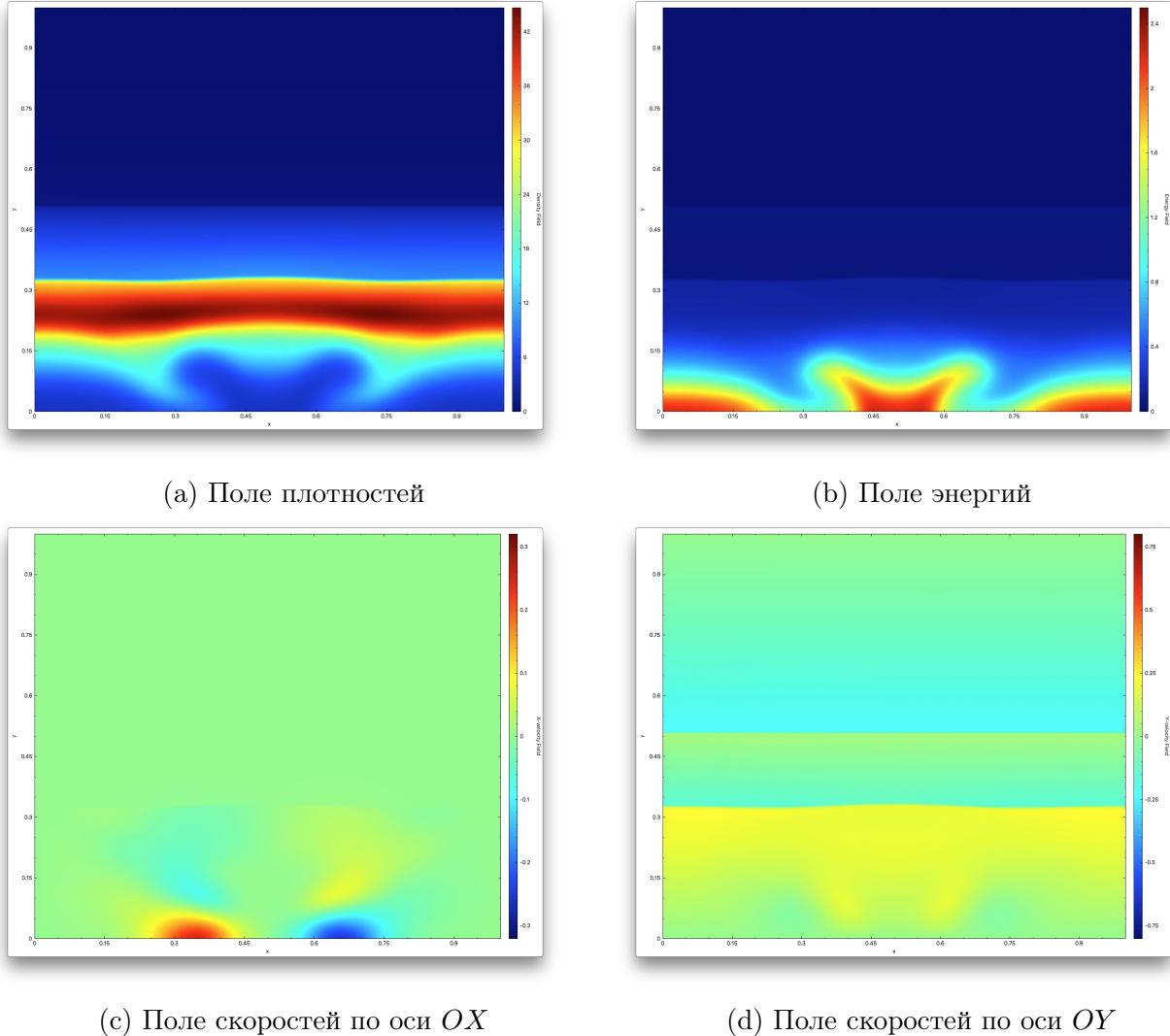


Рис. 10: Скриншот работы программы в момент времени  $t_1$

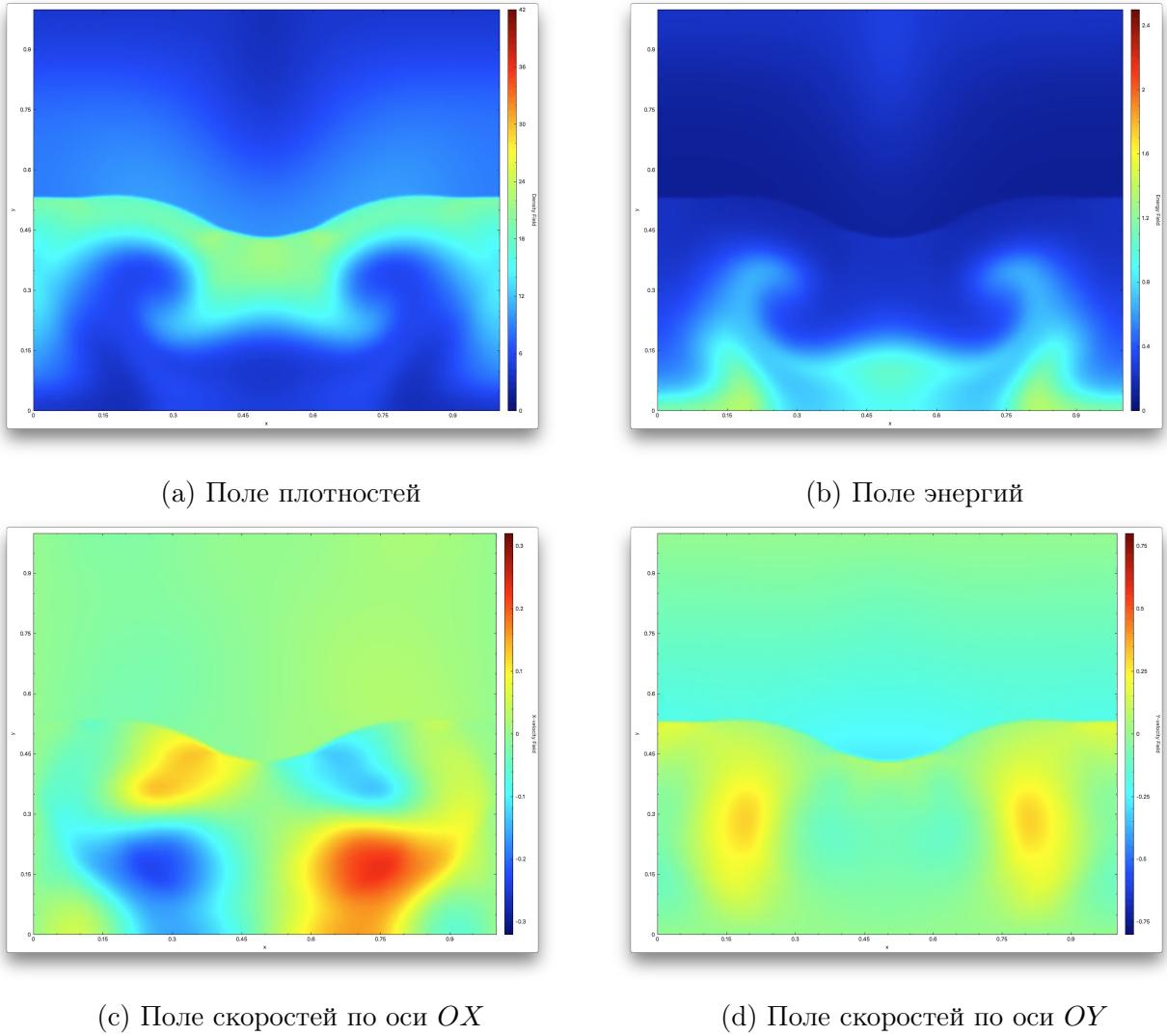


Рис. 11: Скриншот работы программы в момент времени  $t_2$

## 5 Вывод

При написании данной работы была создана программа с использованием таких технологий, как: MPI, CUDA, Qt Framework. Возможно сам алгоритм решения системы дифференциальных уравнений и простой, но основная сложность в данной работе заключалась в том, чтобы безошибочно запрограммировать взаимодействие между соседними MPI узлами, а также вычисление CUDA блоков с использованием разделяемой памяти.

В программах такого объема уже заметна сложность отладки, в связи с чем пришлось реализовать качественную систему логирования, без которой реализация данной программы была бы, попросту, невозможной.

Исходный код программы можно найти по ссылке:

<https://github.com/Eugene-Kolesnikov/Hydrodynamics>

## Список литературы

- [1] David B. Kirk and Wen-mei W. Hwu. 2013. Programming Massively Parallel Processors (Second ed.). Elsevier Inc.
- [2] С. К. Годунов. 1976. Численное решение многомерных задач газовой динамики. Издательство “Наука”.
- [3] С. К. Годунов, А. В. Забродин, Г. П. Прокопов. 1961. Расностная схема для двумерных нестационарных задач газовой динамики и расчет обтекания с отошедшей ударной волной. Ж. вычисл. матем. и матем. физ., 1961, том 1, номер 6, 1020– 1050.
- [4] Li Jiequan, Sheng Wancheng, Zheng Tong, Zheng Yuxi. 2008. Two-dimensional Riemann problems: from scalar conservation laws to compressible Euler equations. Acta Mathematica Scientia.
- [5] Т. Г. Елизарова. 2007. Квазигазодинамические уравнения и методы расчета вязких течений. Издательство “Научный Мир”.