

Погружение в астрофизические течения – вводный практикум по вычислительной астрофизике

И.А. Кондратьев, С.Г. Моисеенко

ИКИ РАН

III школа ЭЛАГ

Саров – НЦФМ, 25 июня 2025

Содержание практикума

1. Введение и мотивация
2. Описание используемого на занятии кода Piastra2D.
3. Различные модели сплошных сред как законы сохранения.
4. Метод конечного объема.
5. Линейное уравнение переноса как простейшая модель сплошной среды. Программная реализация вычислительных кодов.
6. Разные способы решения уравнения переноса. Устойчивость. Апроксимация. Методы высокого порядка и монотонность.
7. Уравнения газовой динамики. Методы Годуновского типа.
8. Магнитная гидродинамика.
9. Диссипативные процессы – вязкость, электро- и теплопроводность.
10. Методы частиц – smooth particle hydrodynamics (SPH).

Рекомендуемая литература

1. M. Zingale “Introduction to Computational Astrophysical Hydrodynamics”, 2015+ (http://open-astrophysics-bookshelf.github.io/numerical_exercises/CompHydroTutorial.pdf)
2. Д.В. Бисикало, А.Г. Жилкин, А.А. Боярчук «Газодинамика тесных двойных звезд» — М.: Физматлит, 2013
3. А.Г. Куликовский, Н.В. Погорелов, А.Ю. Семенов «Математические вопросы численного решения гиперболических систем уравнений» — М.: Физматлит, 2001
4. E.F. Toro “Riemann solvers and numerical methods for fluid dynamics – Springer, Berlin, 2009
5. D.S. Balsara “Higher-order accurate space-time schemes for computational astrophysics—Part I: finite volume methods”, Living Rev Comput Astrophys 3:2 (2017), <https://doi.org/10.1007/s41115-017-0002-8>
6. G. Toth “The $\nabla \cdot \mathbf{B} = 0$ Constraint in Shock-Capturing Magnetohydrodynamics Codes”, J Comput Phys, 161, 605–652 (2000) <https://doi.org/10.1006/jcph.2000.6519>
7. A. Mignone “High-order conservative reconstruction schemes for finite volume methods in cylindrical and spherical coordinates”, J Comput Phys 270:784–814 (2014) <https://doi.org/10.1016/j.jcp.2014.04.001>
8. J.M. Martí, E. Müller “Grid-based methods in relativistic hydrodynamics and magnetohydrodynamics”, Living Rev Comput Astrophys 1:3 (2015) <https://link.springer.com/article/10.1007/lrcap-2015-3>
9. A. Mignone et al “PLUTO: A Numerical Code for Computational Astrophysics”, Astrophys J Suppl, 170, 228 (2007) <https://iopscience.iop.org/article/10.1086/513316>

Если вы уже занимаетесь моделированием...

- В районе 104-го слайда презентации есть несколько небольших проектов, которые можно сделать на Школе, например:
 1. Моделирование МГД-течений в цилиндрической геометрии
 2. Разработка релятивистского газодинамического кода (СТО)

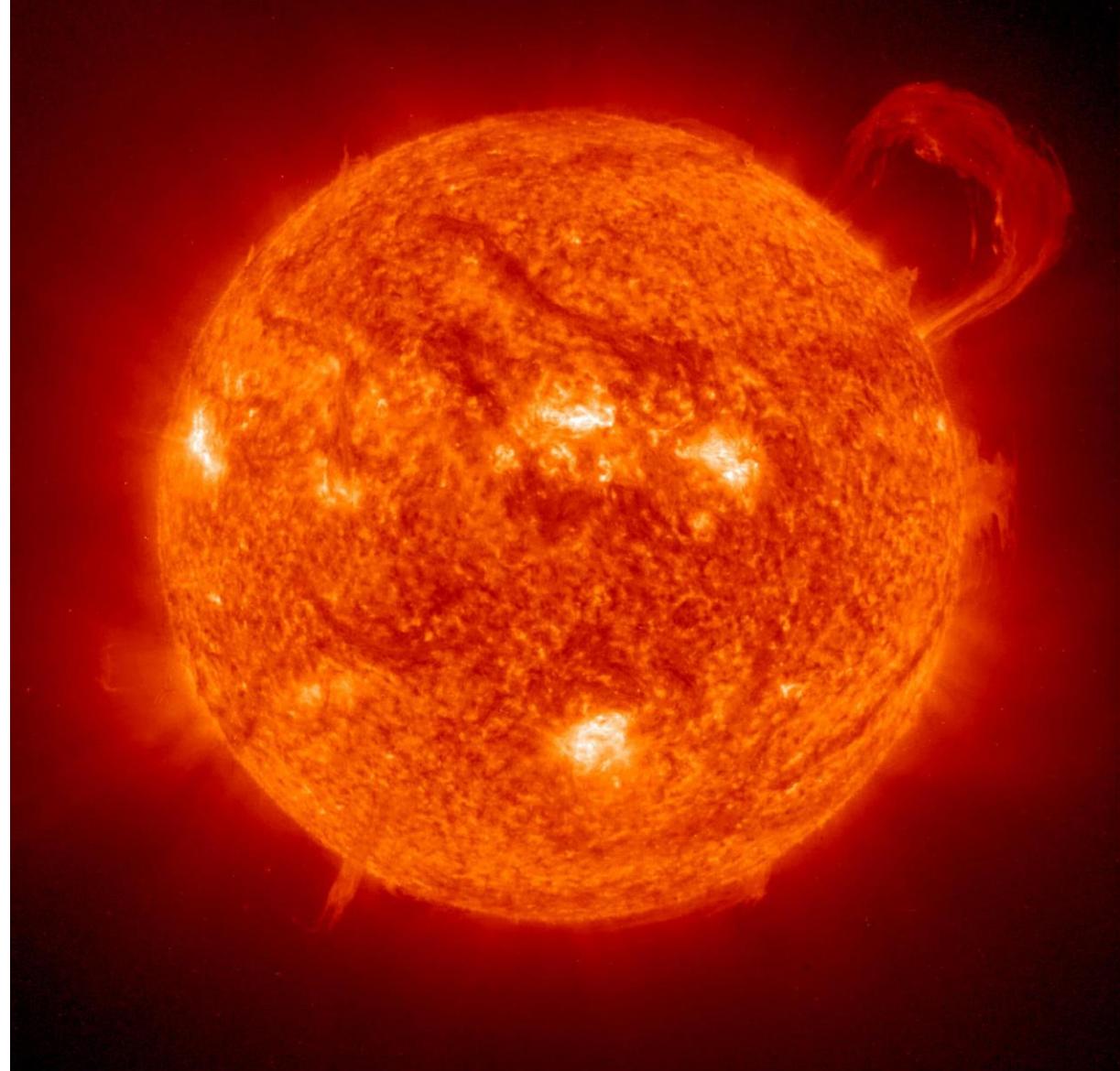
Также есть небольшие задания в ipynb-файлах (они проще и быстрее!)

1. На основе 1D кода (**hydro1D.ipynb**) можно реализовать сферическую газовую динамику с самогравитацией (см. [7]) – очень простая модель коллапса ядер (протозвезды и сверхновые)
2. Вместо линейной адвекции (**advection1D.ipynb**) реализовать решение уравнения Бюргерса $u_t + (0.5u^2)_x = 0$, либо добавить диссиацию (также можно рассмотреть с моделью Бюргерса) $u_t + au_x = \nu u_{xx}$ (см. [2])
3. На основе 2D кода (**SWEsolver2D.ipynb**) с уравнениями мелкой воды рассмотреть распространение цунами или упрощенную модель атмосферных/океанических течений с учетом силы Кориолиса

Введение: зачем это нужно

Давайте построим
математическую
модель звезды
(например, Солнца).

Какие для этого нужны
уравнения?



Необходимые физические процессы

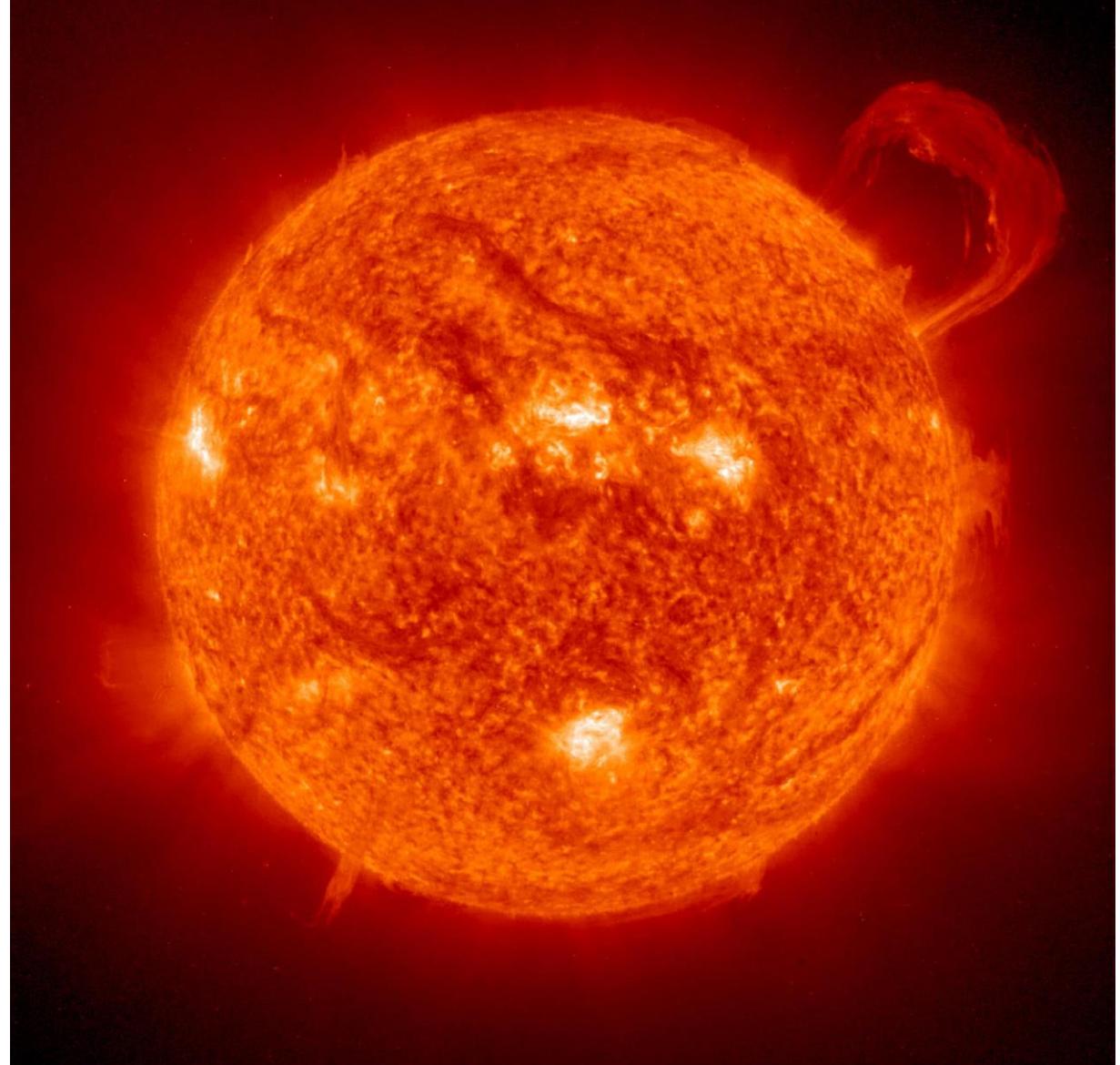
1. (само)гравитация

2. гидродинамика(МГД)

3. ядерные реакции

4. излучение

5. уравнение состояния



Система уравнений для модели звезды

МГД:	$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0,$ $\rho \frac{D\mathbf{v}}{Dt} = -\nabla P + \rho \mathbf{g} + \frac{1}{4\pi} (\nabla \times \mathbf{B}) \times \mathbf{B},$ $\nabla^2 \Phi = 4\pi G \rho.$
Энергия:	$\rho T \frac{Ds}{Dt} = \nabla \cdot (\kappa \nabla T) - \nabla \cdot \mathbf{F}_{\text{rad}} + \epsilon + \frac{ \nabla \times \mathbf{B} ^2}{4\pi \sigma}.$
Излучение:	$\frac{\partial E_\nu}{\partial t} + \nabla \cdot \mathbf{F}_\nu = \kappa_\nu \rho c (B_\nu - E_\nu).$
Реакции:	$\frac{dY_i}{dt} = \sum_j (R_{j \rightarrow i} - R_{i \rightarrow j}), \quad \epsilon = \sum_i \Delta Q_i \frac{dY_i}{dt}.$
Уравнение состояния:	$P = P(\rho, T, Y_i), \quad e = e(\rho, T, Y_i).$

ВСЕ ОЧЕЬ ПЛОХО:

- Разные масштабы для разных явлений (вспышки в короне, эволюция ядра, диффузия фотонов в оболочке, конвекция и т.д.) по времени и пространству
- Нелинейные и многомерные уравнения связаны между собой

Аналитические подходы – почти невозможны даже для описания гидродинамики (обычно возможны только качественные оценки или линеаризованные соотношения для возмущений, также есть очень скромный набор точных решений)

Численные подходы в астрофизике

- В настоящее время промоделировать систему выше целиком для описания всех процессов на Солнце невозможно даже численно (поэтому у нас есть работа ☺)
- По этой причине приходится прибегать к «сценариям»: чтобы проверить некоторую гипотезу о каком-то явлении, делаются (упрощающие) предположения о физике процессов, выбирается численный метод, ставятся начальные условия и проводится численный эксперимент
- Например, мы можем рассматривать 1D эволюцию всей звезды с рядом приближений для многомерных процессов (конвекция, вращение, эволюция магнитного поля и т.д.) – в этом случае мы никак не «ухватим» физику КВМ
- Другой вариант – это численное исследование конкретных частей/процессов в звезде (это же относится и к любым космическим средам) – например, мы можем промоделировать КВМ, но мы не будем обращать внимание на саму звезду
- Примерами могут служить моделирование механизма взрыва сверхновой (МГД + гравитация + нейтрино, $t \sim 1\text{sec}$) и моделирование распространения ударной волны от сверхновой по МЗС (МГД + излучение в остатке, $t \sim 10.000\text{yr}$)

Типичные астрофизические «сценарии»

- | | | |
|--|--|--|
| 1. Коллапс протозвездного облака (<i>dense core</i>), формирование молодых звезд и “ <i>T Tauri</i> ” джетов – МГД + самогравитация + излучение | ЧД/НЗ (<i>AGN, LMXB/HMXB</i>) – (релятивистская) МГД + излучение | взаимодействие с родительской звездой) – ГД/МГД + излучение + химия |
| 2. Взаимодействие солнечного ветра с магнитосферой Земли – МГД + химия | 5. Коллапс звездного ядра и “ <i>central engine</i> ” сверхновой – ГД/МГД + самогравитация + нейтрино | Пульсарные туманности (PWN) – (релятивистская) ГД/МГД |
| 3. Эволюция остатка сверхновой – ГД/МГД + излучение | 6. Слияние нейтронных звезд – ГД/МГД + ОТО + нейтрино | 9. Астрофизические/лабораторные джеты – МГД+ |
| 4. Дисковая акреция на | 7. Протопланетные диски/горячие юпитеры (глобальные модели, | <div style="border: 1px solid black; padding: 10px;"><p>Часто работает гидродинамическое приближение, и задача сводится к использованию гидродинамических для моделирования космических сред</p></div> |

Использование готовых пакетов программ

Очень много МГД-кодов с учетом разной дополнительной физики (вязкость, проводимость, эффект Холла, излучение, пыль....):

1. **PLUTO** (<https://plutocode.ph.unito.it/>)
2. **Athena++** (<https://www.athena-astro.app/>)
3. **FLASH** (<https://flash.rochester.edu/site/flashcode/>)
4. + many-many more!

- «+»: Не нужно писать/отлаживать код; крупные пакеты уже подготовлены для использования на многопроцессорных системах
- «-»: Функционала под нужную именно вам задачу может не быть; в крупных пакетах может быть сложно разобраться; проблема “черного ящика”

Самое главное – всегда нужно понимать, как работает используемый вами метод!

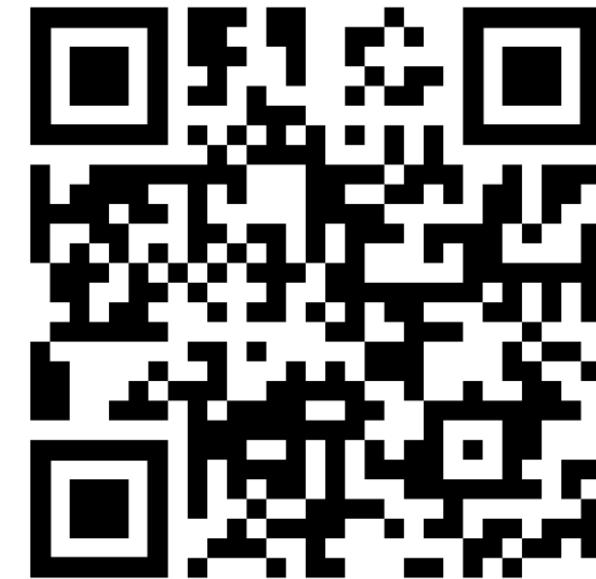
Piastra2D – учебный код для сжимаемых течений в 2D

- Архив можно загрузить по ссылке
<https://github.com/mrkondratyev/Piastra2D>
- Для работы необходимо предварительно установить **Python** с библиотеками **NumPy**, **matplotlib** (обычно эти библиотеки установлены по умолчанию) и **IPython** + **jupyter notebook** (всё можно установить при помощи anaconda или pip, я рекомендую anaconda)
- Запускать модели можно с помощью ipynb-файлов
“*_illustration_jupyter.ipynb”**



Piastra2D – учебный код для сжимаемых течений в 2D

- Код «умеет» решать уравнения линейного переноса, газовой динамики и магнитной гидродинамики в 1D и 2D в декартовой геометрии (см. ниже на данном занятии)
- Имеет все основные детали, которые есть в «настоящих» больших комплексах программ для моделирования плазменных/астрофизических течений
- Для тестирования кода можно запустить файл *****_illustration_jupyter.ipynb** с любой из моделей
- На выходе получится анимация + в конце будет сообщение об успешном завершении



Гидродинамический закон сохранения

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{R}$$

- Изменение сохраняющейся величины внутри ограниченного объема во времени равно потоку этой величины через границы ячеек

$$\frac{\partial \int_V \mathbf{U} dV}{\partial t} + \oint_{\partial V} \mathbf{F} \cdot d\mathbf{S} = \int_V \mathbf{R} dV$$

Большинство кодов решает уравнения сплошной среды в такой форме (Методы конечного объема / Методы типа Годунова)

Различные модели сплошных сред как законы сохранения

- Линейное уравнение переноса (адвекция) – **простейшая линейная модель** сплошной среды – начальный профиль переносится в пространстве с постоянной скоростью

$$\frac{\partial U}{\partial t} + a_x \frac{\partial U}{\partial x} + a_y \frac{\partial U}{\partial y} = 0$$

- Уравнение Бюргерса (Хопфа) – **простейшая нелинейная модель** сплошной среды – возможны аналоги волн разрежения и ударных волн

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} = 0 \quad \rightarrow \quad \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0$$

Уравнения газовой динамики

- Стандартная форма уравнений (Ландау & Лифшиц, том 6):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

- ρ - плотность,
- \mathbf{v} - вектор скорости,
- p - давление,
- ϵ - внутренняя энергия на единицу массы,

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla p$$

$$\frac{\partial \epsilon}{\partial t} + \mathbf{v} \cdot \nabla \epsilon = -\frac{p}{\rho} \nabla \cdot \mathbf{v}$$

**Перепишем их в форме
законов сохранения**

- + начальные и граничные условия
- + уравнение состояния

Различные модели сплошных сред как законы сохранения – газовая динамика

1. Уравнение сохранения массы (непрерывности):

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho v_x)}{\partial x} + \frac{\partial(\rho v_y)}{\partial y} = 0 \quad (1)$$

2. Уравнение сохранения импульса в направлении x :

$$\frac{\partial(\rho v_x)}{\partial t} + \frac{\partial(\rho v_x^2 + p)}{\partial x} + \frac{\partial(\rho v_x v_y)}{\partial y} = 0 \quad (2)$$

3. Уравнение сохранения импульса в направлении y :

$$\frac{\partial(\rho v_y)}{\partial t} + \frac{\partial(\rho v_x v_y)}{\partial x} + \frac{\partial(\rho v_y^2 + p)}{\partial y} = 0 \quad (3)$$

4. Уравнение сохранения энергии:

$$\frac{\partial E}{\partial t} + \frac{\partial((E + p)v_x)}{\partial x} + \frac{\partial((E + p)v_y)}{\partial y} = 0 \quad (4)$$

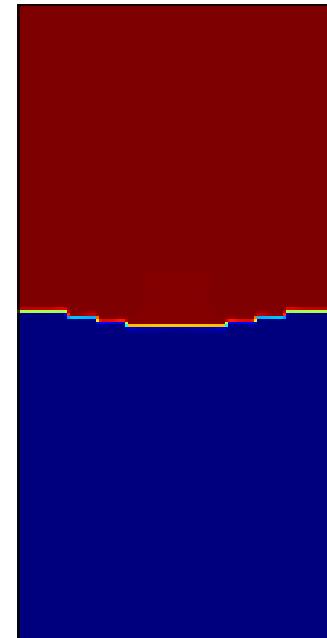
где ρ - плотность, u и v - компоненты скорости в направлениях x и y соответственно, p - давление, E - полная энергия на единицу объёма, связанная с плотностью, скоростью и внутренней энергией следующим образом:

$$E = \rho e + \frac{1}{2} \rho(v_x^2 + v_y^2) \quad (5)$$

где e - внутренняя энергия на единицу массы.

- Газовая динамика – описывает процессы в сжимаемых жидкостях/газах и столкновительной плазме без магнитного поля
- Огромное количество процессов! Взрывы, турбулентность, плавучесть и т.д.

Рис.:
Моделирование
неустойчивости
Рэлея-Тейлора,
Fluid_solver_easy
_run.ipynb
С параметрами
«RTI», **Nx = 200**,
Ny = 100



Уравнения «Мелкой воды»

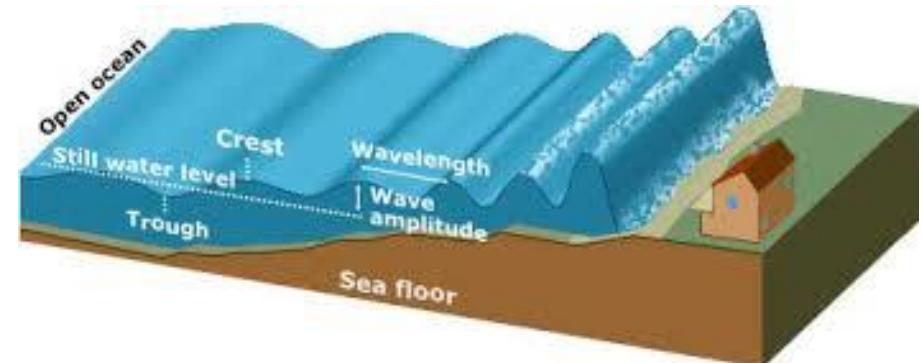
- Модель **несжимаемой** жидкости в приближении, что глубина бассейна вдоль Oz намного меньше, чем его размеры вдоль Ox, Oy
- Хорошая аппроксимация для океана и атмосферы – можно исследовать распространение цунами, волны Россби и пр.
- Уравнения становятся эквивалентны **сжимаемой** газовой динамике с баротропным уравнением состояния => имеем набор законов сохранения

$$\frac{\partial h}{\partial t} + \frac{\partial(hv_x)}{\partial x} + \frac{\partial(hv_y)}{\partial y} = 0$$

$$\frac{\partial(hv_x)}{\partial t} + \frac{\partial}{\partial x} \left(hv_x^2 + \frac{gh^2}{2} \right) + \frac{\partial(hv_x v_y)}{\partial y} = -gh \frac{\partial b}{\partial x} + fhv_y$$

$$\frac{\partial(hv_y)}{\partial t} + \frac{\partial(hv_x v_y)}{\partial x} + \frac{\partial}{\partial y} \left(hv_y^2 + \frac{gh^2}{2} \right) = -gh \frac{\partial b}{\partial y} - fhv_x$$

- $h(x, y, t)$ — глубина жидкости
- $v_x(x, y, t), v_y(x, y, t)$ — компоненты скорости по x и y
- g — ускорение свободного падения
- $b(x, y)$ — рельеф дна ($h + b$ = полная высота поверхности)
- f — параметр Кориолиса ($f = 2\Omega \sin \phi$, где Ω — угловая скорость вращения Земли, ϕ — широта)



В файле **SWEsolver2D.ipynb** запрограммированы уравнения, там же можно посмотреть примеры

Уравнения магнитной гидродинамики

- Стандартная форма уравнений (Ландау & Лифшиц, том 8):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0$$

$$\rho \left(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla) \mathbf{v} \right) = -\nabla p + \mathbf{J} \times \mathbf{B} + \mathbf{f}$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B})$$

$$\frac{\partial s}{\partial t} + \mathbf{v} \cdot \nabla s = 0$$

$$\mathbf{J} = \nabla \times \mathbf{B}$$

- Магнитная гидродинамика (МГД) – всё, что в газовой динамике + магнитное поле
- Аккреционные диски, некоторые виды сверхновых, солнечно-земные связи, лабораторные и астрофизические джеты, термояд, плазменные ускорители и много-много других объектов

**Перепишем их в форме
законов сохранения**

+ нач. и гран. условия
+ уравнение состояния

Различные модели сплошных сред как законы сохранения – магнитная гидродинамика

1. Уравнение сохранения массы:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \quad (1)$$

2. Уравнение сохранения импульса:

$$\frac{\partial(\rho \mathbf{v})}{\partial t} + \nabla \cdot \left[\rho \mathbf{v} \mathbf{v} + \left(p + \frac{\mathbf{B}^2}{8\pi} \right) \mathbf{I} - \frac{\mathbf{B} \mathbf{B}}{4\pi} \right] = 0 \quad (2)$$

3. Уравнение сохранения энергии:

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + p + \frac{\mathbf{B}^2}{8\pi} \right) \mathbf{v} - \frac{(\mathbf{v} \cdot \mathbf{B}) \mathbf{B}}{4\pi} \right] = 0 \quad (3)$$

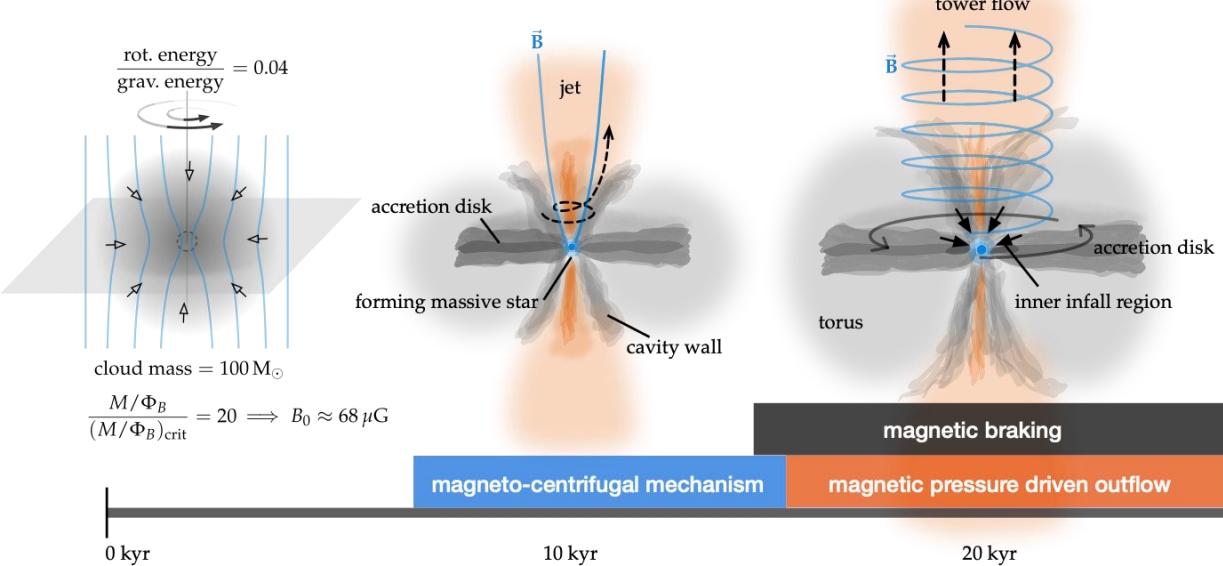
4. Уравнение индукции магнитного поля:

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v}) = 0 \quad (4)$$

где ρ - плотность, \mathbf{v} - вектор скорости, p - давление, \mathbf{B} - магнитное поле, E - полная энергия, включающая внутреннюю энергию ϵ , кинетическую энергию и магнитную энергию:

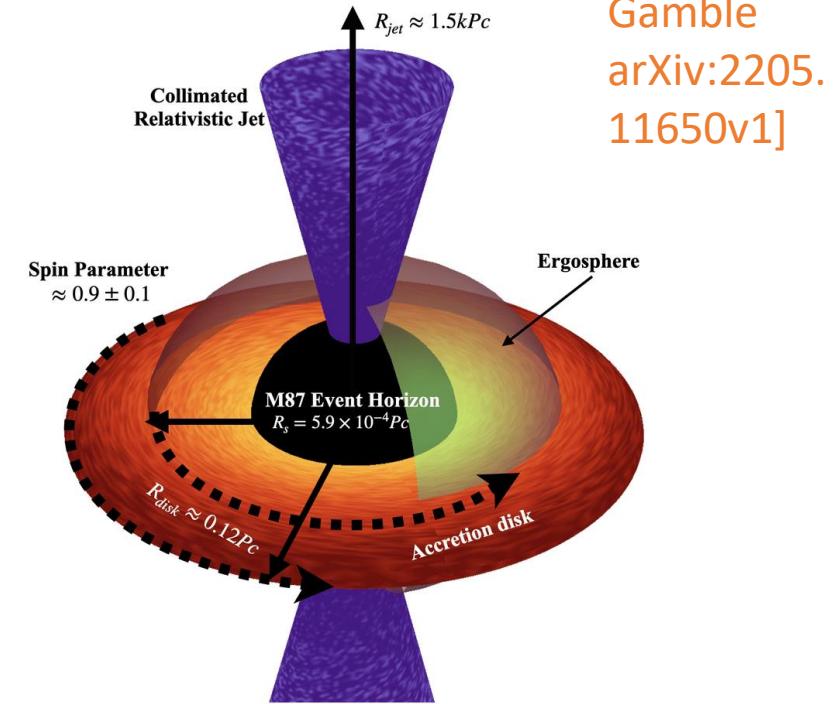
$$E = \rho \epsilon + \frac{1}{2} \rho \mathbf{v}^2 + \frac{\mathbf{B}^2}{8\pi} \quad (5)$$

Рис.: Генерация джетов от протозвезд [REF: <https://www.gandreoliva.org/posters/ringberg23/index.html>]



Другие модели, которые представимы в аналогичной (дивергентной) форме

- Релятивистская газовая динамика (сверхновые, акреция на НЗ и ЧД, АЯГ)
- Релятивистская магнитная гидродинамика (все выше + КГВ + магнитосфера НЗ)
- Уравнения переноса излучения (и нейтрино) в двухмоментной форме (т.н. M1-closure)
- Двухжидкостная гидродинамика плазмы



Все упомянутые системы уравнений являются **гиперболическими** моделями среды (матрица $A = \frac{\partial F}{\partial U}$ имеет вещественные собственные значения при любых входных данных), и к ним могут быть применены схожие подходы

В гиперболических моделях можно рассмотреть **волновые возмущения**, движущиеся с **конечными скоростями**

Jets from
M87 [RS
Gamble
arXiv:2205.
11650v1]

Дивергентная форма записи уравнений и интегральные законы сохранения

- В дифференциальной форме

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \mathbf{F} = \mathbf{R}$$

Изменение во времени объемной плотности сохраняющейся величины равно дивергенции от потока данной величины

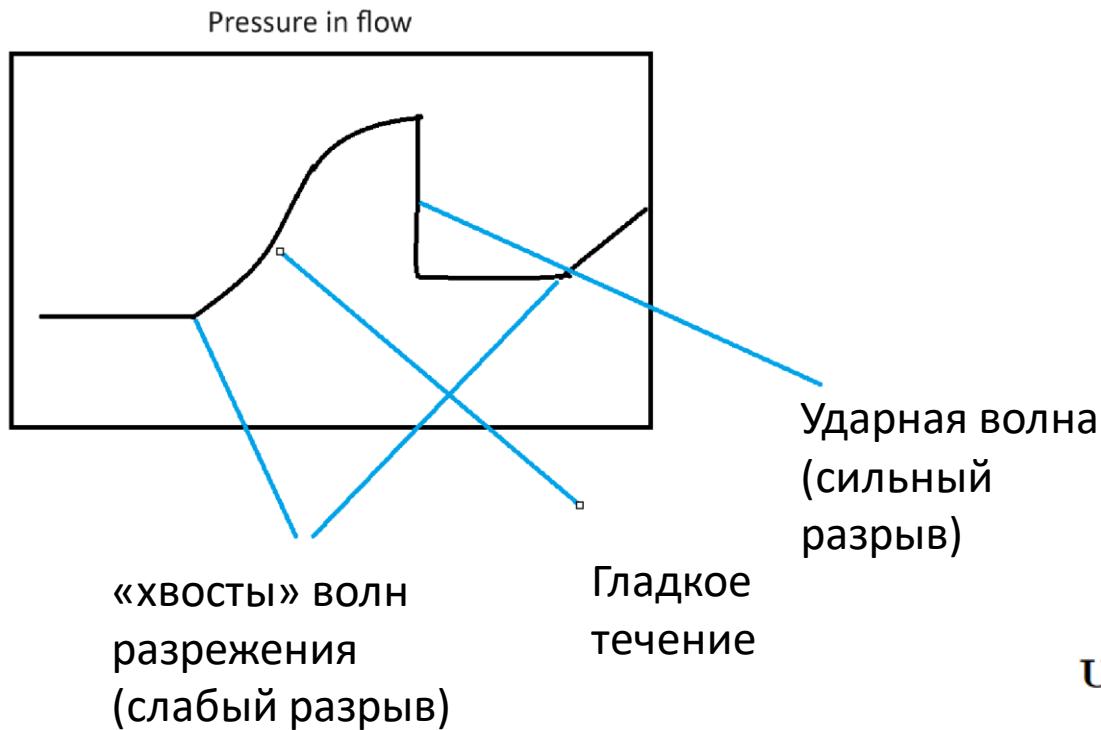
- В интегральной форме (интеграл по объему => изменение самой величины + теорема Остроградского-Гаусса для дивергенции потоков)

$$\frac{\partial \int_V \mathbf{U} dV}{\partial t} + \oint_{\partial V} \mathbf{F} \cdot d\mathbf{S} = \int_V \mathbf{R} dV$$

- «Работает» для любой из рассмотренных систем уравнений

Интегральная форма уравнений лучше дифференциальной?

- Пусть есть течение с разрывами



Дифференциальные
уравнения могут не
существовать,
однако профиль
функции –
интегрируем!
Интегральная
форма позволяет
описать большее
количество решений

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ E \end{pmatrix}, \quad \mathbf{F}_x = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_y \\ (E + p)v_x \end{pmatrix}, \quad \mathbf{F}_y = \begin{pmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + p \\ (E + p)v_y \end{pmatrix}$$

Метод конечного объема (МКО) для уравнений сплошной среды в форме законов сохранения

- Рассмотрим 1D систему законов сохранения в дивергентной форме

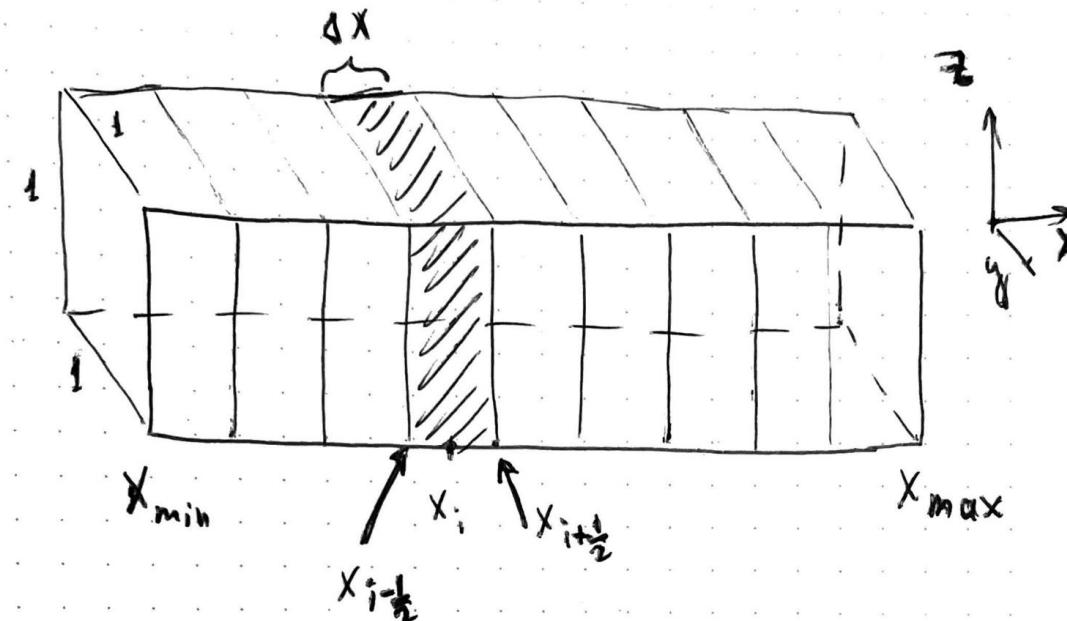
$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = R$$

- Рассмотрим для простоты призму единичного сечения (см. рис. и файл **advection1D.ipynb**) и введем на ней сетку

$$S_{face} = 1 * 1 - \text{площадь грани}$$

$$dV = 1 * 1 * dx - \text{объем ячейки}$$

Поперечные потоки в 1D будут компенсировать друг друга – можем сразу их не учитывать



Метод конечного объема (МКО) для уравнений сплошной среды в форме законов сохранения

- Проинтегрируем закон сохранения по времени (1 шаг) и «объему» ячейки

$$\int_{t^n}^{t^{n+1}} dt \int_{x_{i-1/2}}^{x_{i+1/2}} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} - \mathbf{R} \right) dx = \\ \int_{x_{i-1/2}}^{x_{i+1/2}} [\mathbf{U}^{n+1} - \mathbf{U}^n] dx + \int_{t^n}^{t^{n+1}} [\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}] dt - \int_{t^n}^{t^{n+1}} dt \int_{x_{i-1/2}}^{x_{i+1/2}} dx \mathbf{R} = 0$$

$$\bar{\mathbf{U}}_i^n = \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{U}^n dx$$

$$\bar{\mathbf{R}}_i^{n+1/2} = \frac{1}{\Delta x} \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} dt \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{R} dx$$

$$\bar{\mathbf{F}}_{i+1/2}^{n+1/2} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \mathbf{F}_{i+1/2} dt$$

Введем средние значения для сохраняющихся величин, правой части и потоков

МКО для уравнений сплошной среды в форме законов сохранения

- После подстановки получили следующую аппроксимацию:

$$\frac{\bar{\mathbf{U}}_i^{n+1} - \bar{\mathbf{U}}_i^n}{\Delta t} + \frac{\bar{\mathbf{F}}_{i+1/2}^{n+1/2} - \bar{\mathbf{F}}_{i-1/2}^{n+1/2}}{\Delta x_i} = \bar{\mathbf{R}}_i^{n+1/2}$$

- Умножим на объем ячейки и «проинтегрируем» по всей сетке: $\sum_i f \Delta x_i$:

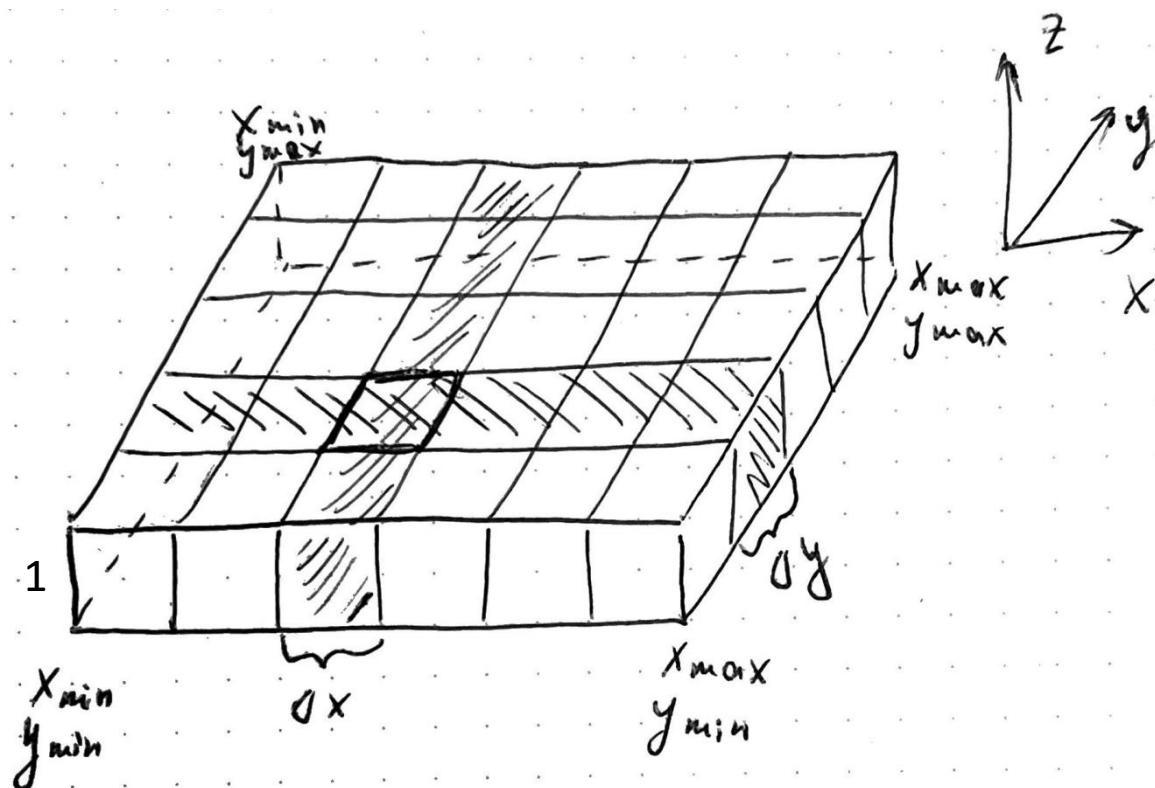
$$\frac{\left(\sum_i \Delta x_i \bar{\mathbf{U}}_i^{n+1} \right) - \left(\sum_i \Delta x_i \bar{\mathbf{U}}_i^n \right)}{\Delta t} + \bar{\mathbf{F}}_R^{n+1/2} - \bar{\mathbf{F}}_L^{n+1/2} - \sum_i \Delta x_i \bar{\mathbf{R}}_i^{n+1/2} = 0$$

Законы сохранения выполняются точно и в дискретном виде! – **консервативность**

МКО для уравнений сплошной среды в форме законов сохранения – многомерный случай

- Рассмотрим теперь двумерный случай (2D)

$$\frac{\partial U}{\partial t} + \frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = R$$



Наша двумерная сетка –
призма с единичной высотой
вдоль оси z

$S_{face,x} = 1 * \Delta y$ – площадь
грани вдоль оси x

$S_{face,y} = 1 * \Delta x$ – площадь
грани вдоль оси y

$\Delta V = 1 * \Delta y * \Delta x$ – объем
ячейки

МКО для уравнений сплошной среды в форме законов сохранения – многомерный случай

Опять запишем интеграл

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \left(\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}_x}{\partial x} + \frac{\partial \mathbf{F}_y}{\partial y} \right) dy dx dt = \int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{R} dy dx dt$$

Используем теорему Гаусса

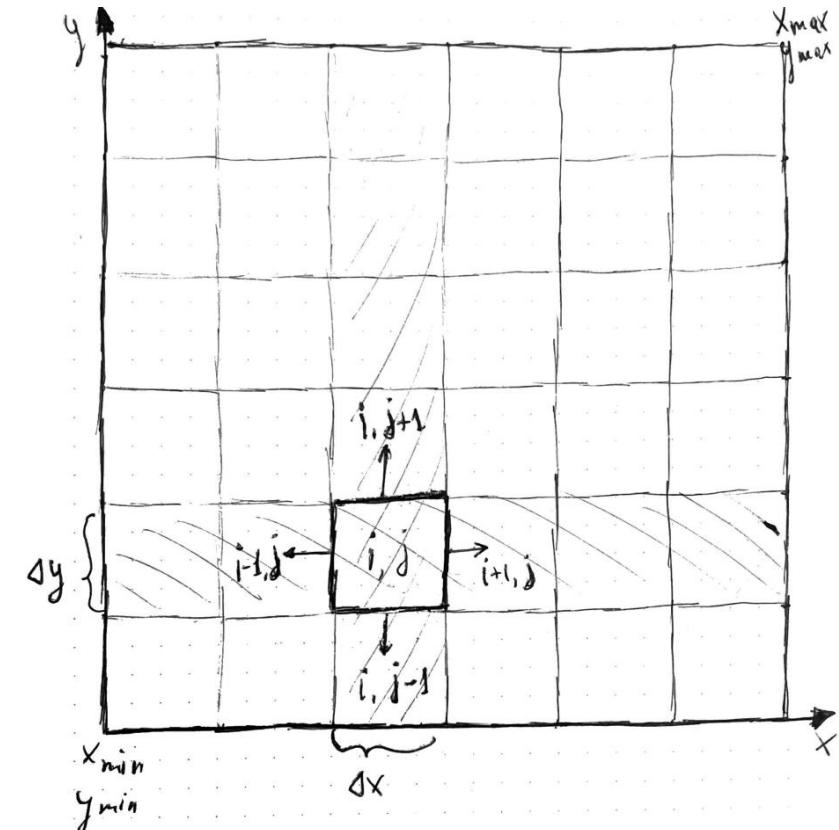
$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \frac{\partial \mathbf{U}}{\partial t} dy dx dt + \int_{t^n}^{t^{n+1}} \oint_{\partial V_{ij}} (\mathbf{F}_x \hat{n}_x + \mathbf{F}_y \hat{n}_y) dS dt =$$

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{R} dy dx dt$$

Интегрируем по прямоугольной ячейке

$$\int_{t^n}^{t^{n+1}} \oint_{\partial V_{ij}} (\mathbf{F}_x \hat{n}_x + \mathbf{F}_y \hat{n}_y) dS dt = \int_{t^n}^{t^{n+1}} \left(\int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}_x(x_{i+1/2}, y, t) dy - \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}_x(x_{i-1/2}, y, t) dy \right) dt$$

$$+ \int_{t^n}^{t^{n+1}} \left(\int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{F}_y(x, y_{j+1/2}, t) dx - \int_{x_{i-1/2}}^{x_{i+1/2}} \mathbf{F}_y(x, y_{j-1/2}, t) dx \right) dt$$



МКО для уравнений сплошной среды в форме законов сохранения – многомерный случай

Интегрируем по времени

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \frac{\partial \mathbf{U}}{\partial t} dy dx dt = \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} (\mathbf{U}(x, y, t^{n+1}) - \mathbf{U}(x, y, t^n)) dy dx$$

Определим средние значения как и в 1D

$$\bar{\mathbf{U}}_{ij}^n = \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{U}(x, y, t^n) dy dx$$

$$\bar{\mathbf{F}}_{x,i+1/2,j}^{n+1/2} = \frac{1}{\Delta y \Delta t} \int_{t^n}^{t^{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{F}_x(x_{i+1/2}, y, t) dy dt$$

$$\int_{t^n}^{t^{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} \mathbf{R} dy dx dt = \Delta x \Delta y \Delta t \bar{\mathbf{R}}_{ij}^{n+1/2}$$

МКО для уравнений сплошной среды в форме законов сохранения

- С учетом средних

$$\int_{t^n}^{t^{n+1}} \oint_{\partial V_{ij}} (\mathbf{F}_x \hat{n}_x + \mathbf{F}_y \hat{n}_y) \, dS \, dt = \Delta t \left(\Delta y \left(\bar{\mathbf{F}}_{x,i+1/2,j}^{n+1/2} - \bar{\mathbf{F}}_{x,i-1/2,j}^{n+1/2} \right) + \Delta x \left(\bar{\mathbf{F}}_{y,i,j+1/2}^{n+1/2} - \bar{\mathbf{F}}_{y,i,j-1/2}^{n+1/2} \right) \right)$$

- В итоге получается следующее выражение для МКО в двумерной геометрии

$$\frac{\bar{\mathbf{U}}_{ij}^{n+1} - \bar{\mathbf{U}}_{ij}^n}{\Delta t} + \frac{\bar{\mathbf{F}}_{x,i+1/2,j}^{n+1/2} - \bar{\mathbf{F}}_{x,i-1/2,j}^{n+1/2}}{\Delta x} + \frac{\bar{\mathbf{F}}_{y,i,j+1/2}^{n+1/2} - \bar{\mathbf{F}}_{y,i,j-1/2}^{n+1/2}}{\Delta y} = \bar{\mathbf{R}}_{ij}^{n+1/2}$$

Трехмерный случай (3D) полностью аналогичен 2D, но является существенно более громоздким вычислительно 29

МКО – общий случай

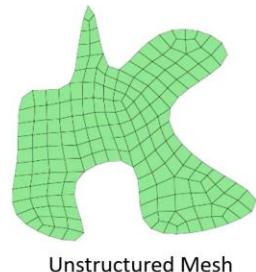
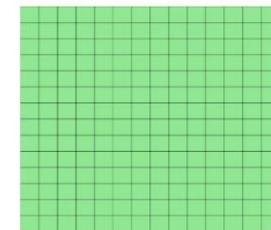
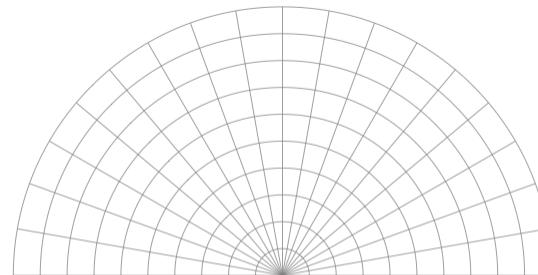
$$\frac{\mathbf{U}_{cell}^{n+1} - \mathbf{U}_{cell}^n}{\Delta t} + \frac{1}{V_{cell}} \sum_{faces \in cell} \mathbf{F}_{face} \cdot \mathbf{n}_{face} S_{face} = \mathbf{R}_{cell}$$

- Работает в произвольной многомерной геометрии
- Все отличия между разными методами/системами лежат в выражении для аппроксимации потока \mathbf{F} и вида нормированной площади/объема
- Декартовые координаты в 3D: $V_{cell} = \Delta x * \Delta y * \Delta z$, $S_{face,x} = \Delta y * \Delta z$, $S_{face,y} = \Delta x * \Delta z$, $S_{face,z} = \Delta x * \Delta y$
- Цилиндрические координаты в 3D: $V_{cell} = \frac{r_{i+1/2}^2 - r_{i-1/2}^2}{2} * \Delta z * \Delta \varphi$,
 $S_{face,r_{i+1/2}} = r_{i+1/2} * \Delta z * \Delta \varphi$, $S_{face,z} = \frac{r_{i+1/2}^2 - r_{i-1/2}^2}{2} * \Delta \varphi$, $S_{face,\varphi} = \Delta r * \Delta z$,

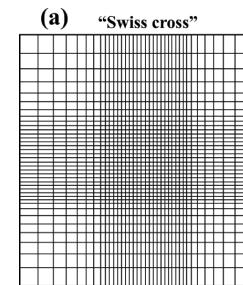
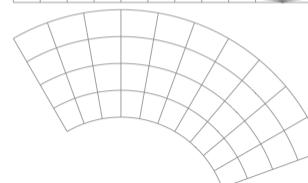
Сетки

- Разные примеры:

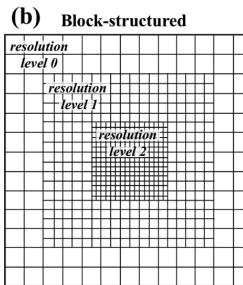
1. Структурированные сетки – можно ввести индексы и многомерные массивы для удобства
2. Неструктурированные сетки – сплошная нумерация ячеек, нужно хранить локальную структуру (площади граней, «соседей» и т.д.), но намного более гибкие области
3. Криволинейные сетки – обычно структурированные – для задач с осевой или сферической симметрией в первом приближении – очень популярны в астрофизике!
4. Сетки с адаптивным разрешением – разные варианты
 1. можно просто сгустить структурированную сетку или рассмотреть вложенную сетку
 2. AMR: Можно использовать блоки или «патчи» с удвоенным разрешением, вложенные в более грубую сетку, для доступа к структуре используется граф/дерево – сложная реализация, но гибкая + оптимальная модель
5. Подвижные сетки – бывают неструктурные (практически настоящий Лагранжев подход) и структурированные (подвижность обычно ограничена вращением, сдвигом, сжатием и т.д.)



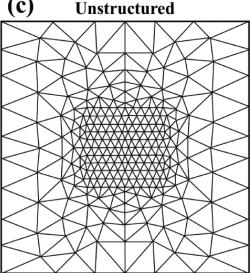
Structured Mesh



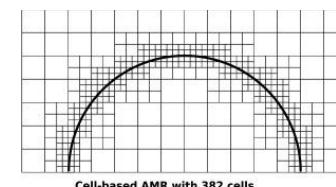
(a) “Swiss cross”



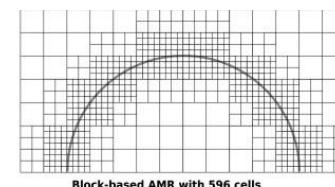
(b) Block-structured



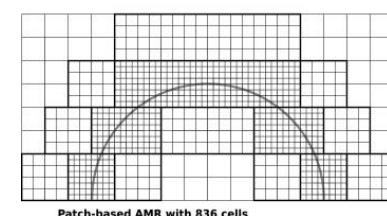
(c) Unstructured



Cell-based AMR with 382 cells

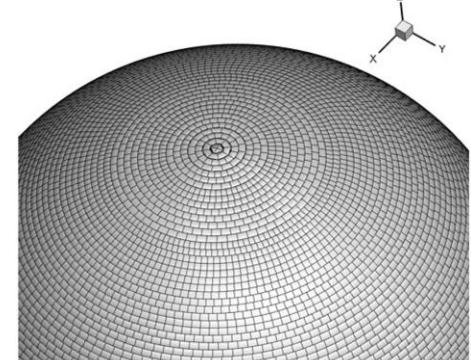


Block-based AMR with 596 cells



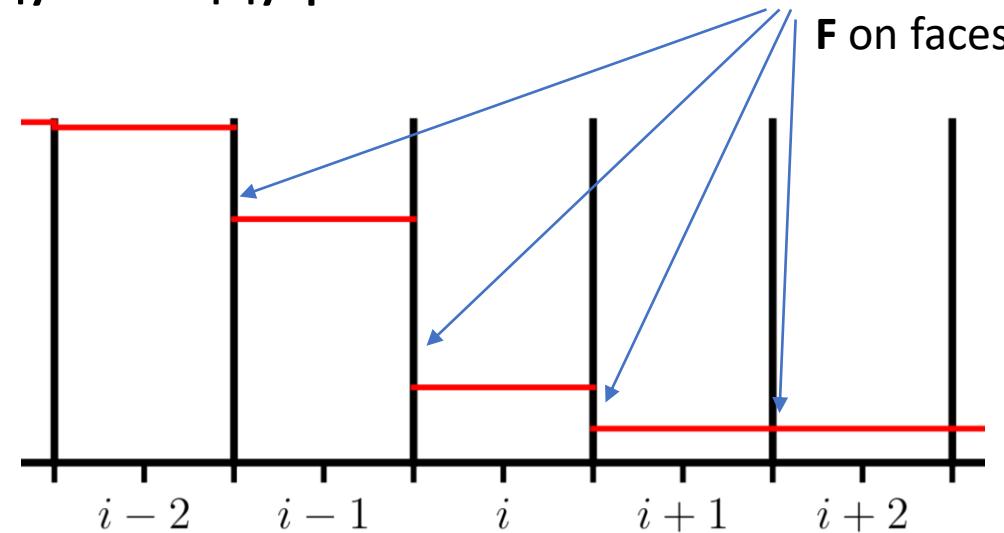
Patch-based AMR with 836 cells

Notes:
Block-based is done using 4 x 4 blocks. For all methods there is never more than 1 level of refinement difference across any face.



МКО для уравнений сплошной среды в форме законов сохранения

- В итоге получаем, что **решение уравнения определено только формой потока** через границу между расчетными ячейками
- Но каким его выбрать?



Рассмотрим пример для уравнения переноса (для него $F = aU$), но сначала обсудим, как вообще устроено его решение, а также как устроены вычислительные коды для моделирования гидродинамических систем

Линейное уравнение переноса как простейшая модель сплошной среды

- Линейное уравнение переноса (адвекции)

$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} = 0, \quad U(x, 0) = U_0(x)$$

- Характеристики (вдоль них решение распространяется без изменений)

$$\frac{dx}{dt} = a, \quad \zeta = x - at, \quad \frac{dU}{d\zeta} = 0$$

- Получили решение в форме бегущей волны

$$U(x, t) = U_0(x - at)$$

Линейное уравнение переноса как простейшая модель сплошной среды

- Метод Фурье-гармоник:

$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} = 0, \quad U(x, t) \sim \exp(ik(x - at))$$

- Решение уравнения при наличии диссипации

$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} = \nu \frac{\partial^2 U}{\partial x^2}, \quad U(x, t) \sim \exp(ik(x - at) - \nu k^2 t)$$

- Решение уравнения при наличии дисперсии

$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} = \mu \frac{\partial^3 U}{\partial x^3}, \quad U(x, t) \sim \exp(ik(x - at) - i\mu k^3 t)$$

Пример. Как устроены гидродинамические коды.

- Файл **advection1D.ipynb** – простая 1D линейная модель адвекции

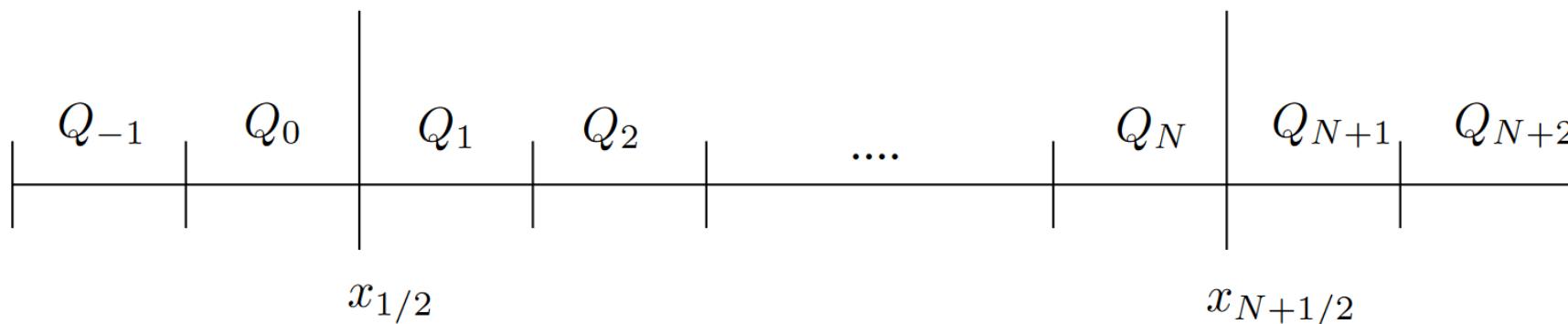
Структура практически любого FV гидродинамического кода:

1. Задание расчетной сетки
2. Задание параметров модели и начальных\границных условий
3. Цикл по времени: (пока $\text{Time} < \text{finalTime}$)
 1. Расчет допустимого шага во времени (для явных схем - из условия CFL)
 2. Учет граничных условий (через заполнение «фиктивных ячеек», см. след. слайд)
 3. Уточнение расчетных величин для расчета в потоках (опционально, порядок точности ++, обсудим через пару минут)
 4. Расчет потоков на границах между ячейками
 5. Дополнительные процедуры (опционально) (учет гравитации, удаление дивергенции магнитного поля, учет вязкости/проводимости, etc.....)
 6. Обновление сохраняющихся (консервативных) переменных методом конечного объема
 7. Восстановление физических (примитивных) переменных из сохраняющихся

(некоторые методы (с высоким порядком по времени) – многостадийные и требуют повторения процедур (3.2)-(3.7) несколько раз!)

Учет граничных условий в МКО – *ghost cells*

- Подход “**ghost cells**” – во всех массивах добавляются элементы в фиктивных ячейках за расчетной областью



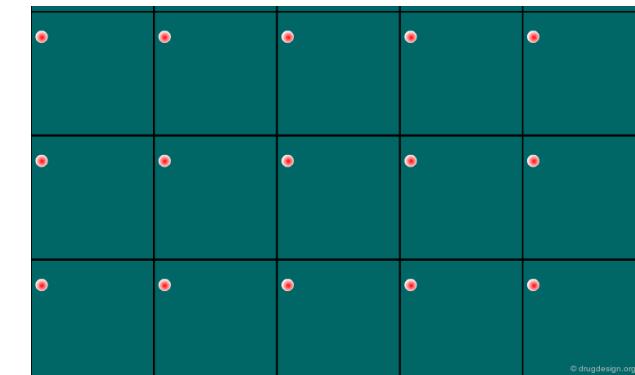
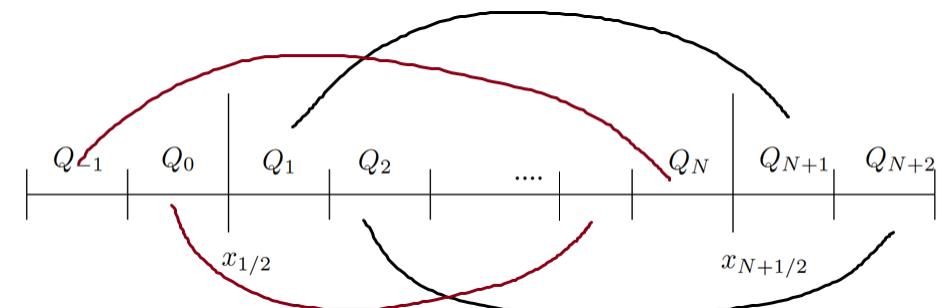
- Заполняя эти элементы величинами из «реальных» ячеек в соответствии с заданными граничными условиями, можно легко моделировать краевые задачи

Границные условия для адвекции – ghost cells

- Зададим **периодические** граничные условия в следующем виде (рис.)
- Такие условия часто используются когда моделируется бесконечная область и/или турбулентные/нерегулярные течения

#задаем граничные условия с использованием #фактивных ячеек (в данном примере они #периодические)

```
for i in range(0,Ngc):
    #внутренняя граница
    U0[i] = U0[Nx+i]      #замыкаем первых на последних
    #внешняя граница
    U0[Nx+Ngc+i] = U0[Ngc+i] #замыкаем последних на первых
```

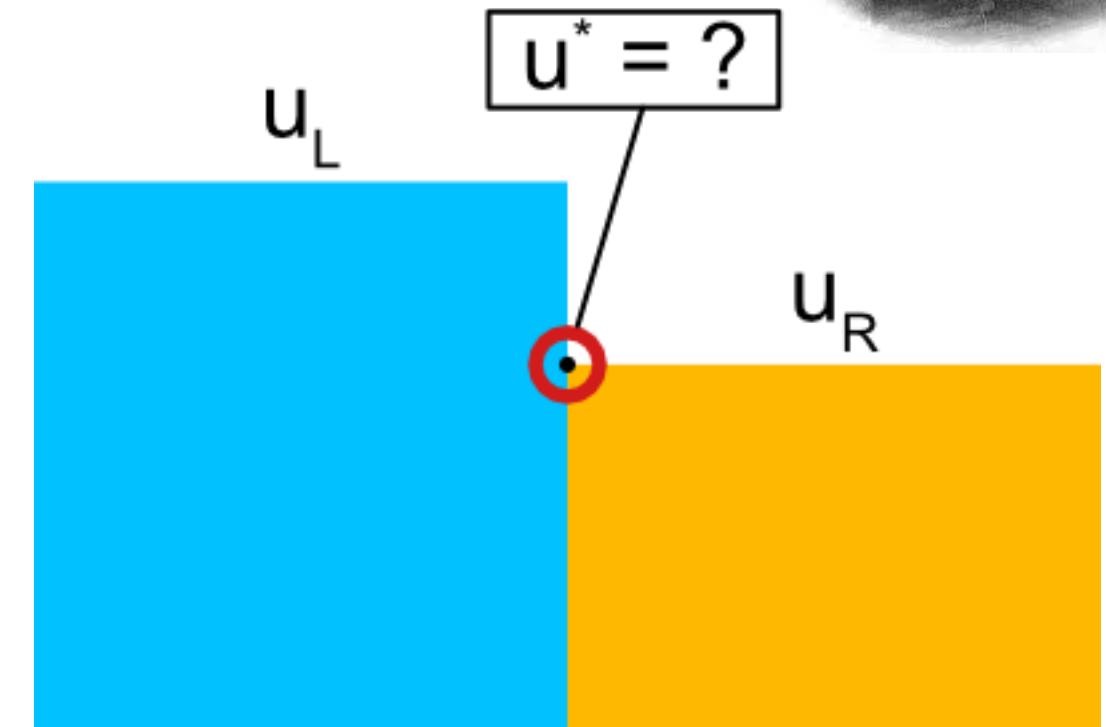


Пробуем разные варианты аппроксимации
для потока адвекции

Расчет потока. Задача Римана.

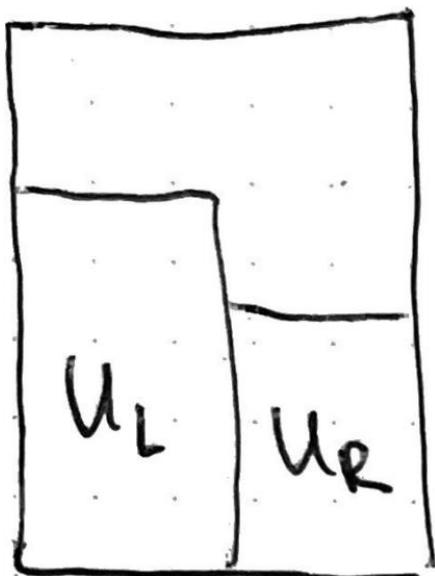
- Поток рассчитывается исходя из физики модели
- Мы можем трактовать решения справа и слева от границы между ячейками как **разрывы** в газе (или в другой системе вроде уравнений переноса излучения), разделенные перегородкой
- Каждый новый шаг по времени мы «разрываем перегородку»
- Из физики моделируемой системы возникает задача Римана
- Она является ключевым элементом для гидродинамического кода

Bernhard Riemann (1860): В начальный момент ($t=0$) есть перегородка между двумя независимыми состояниями среды. Перегородку убирают. Что произойдет со средой при $t > 0$?

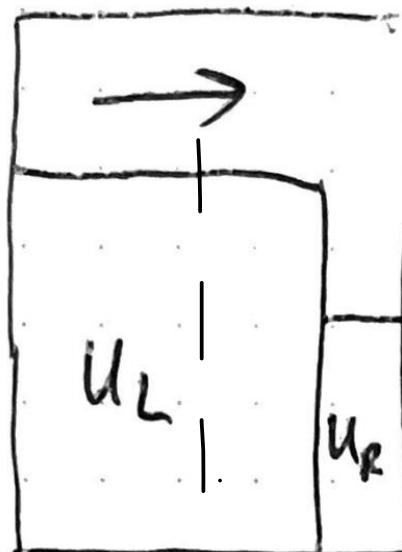


Задача Римана для уравнения адвекции

- Это очень просто!



$$t = 0$$



$$t > 0$$
$$a > 0$$



$$t > 0$$
$$a < 0$$

Численный поток для уравнения адвекции

- Решение задачи Римана приводит к следующему выражению для потока

$$F_{i+1/2} = \frac{aU_{i+1} + aU_i}{2} - |a| \frac{U_{i+1} - U_i}{2} \quad \longleftrightarrow \quad \frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{a}{\Delta x} \begin{cases} u_i^n - u_{i-1}^n, & a \geq 0, \\ u_{i+1}^n - u_i^n, & a < 0. \end{cases}$$

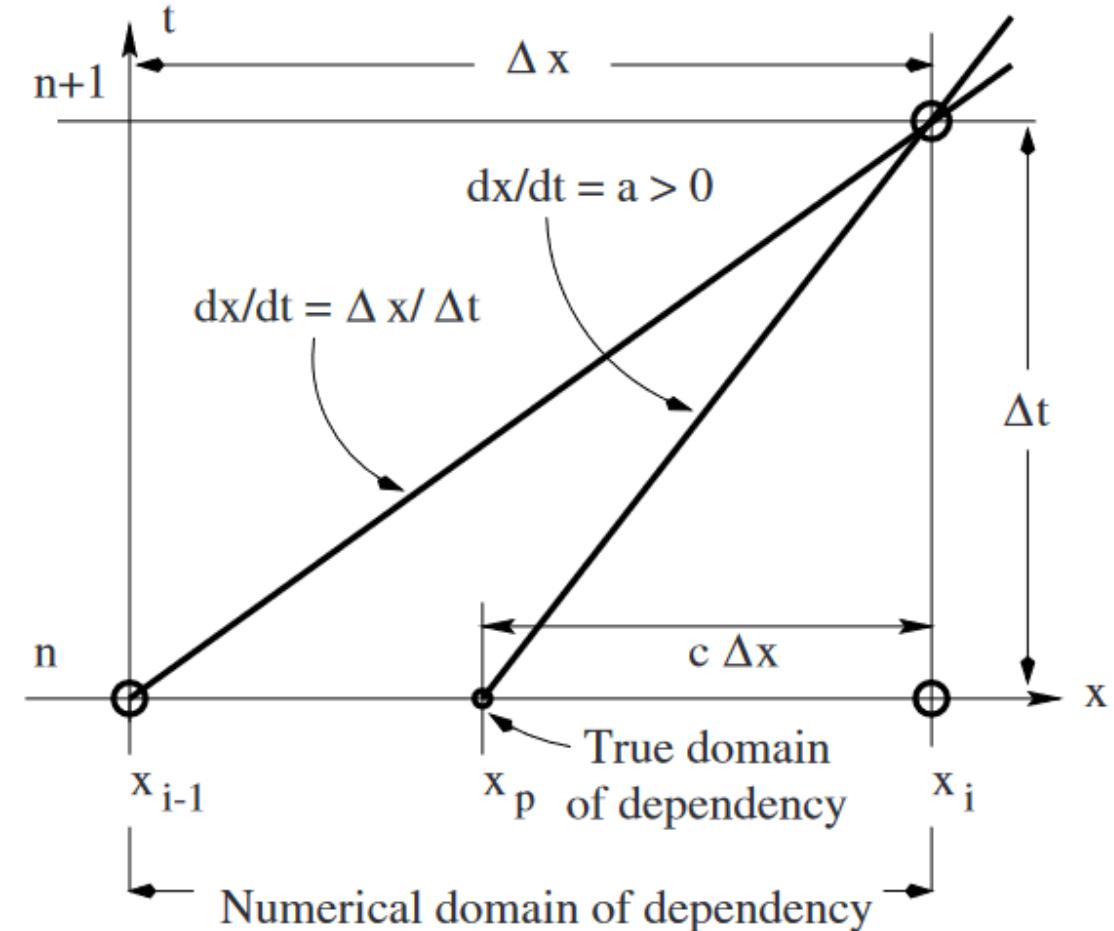
- Метод условно устойчив!** Условие устойчивости метода можно трактовать как независимость задач Римана на каждой грани

$$\text{1D: } \Delta t = CFL * \min_{i,j} \left(\frac{\Delta x}{|a_x|} \right), \quad CFL \leq 1$$

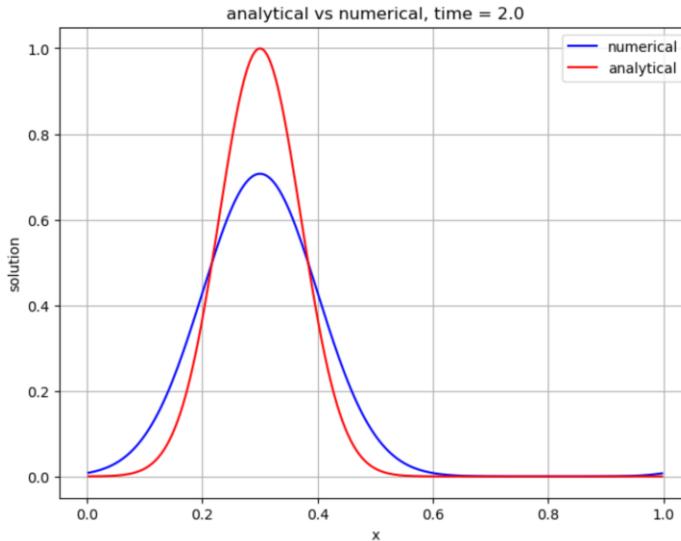
$$\text{2D: } \Delta t = CFL * \min_{i,j} \left(\frac{\Delta x}{|a_x|}, \frac{\Delta y}{|a_y|} \right), \quad CFL \leq \frac{1}{2}$$

Courant-Friedrichs-Lowy (1928) condition – более наглядное объяснение

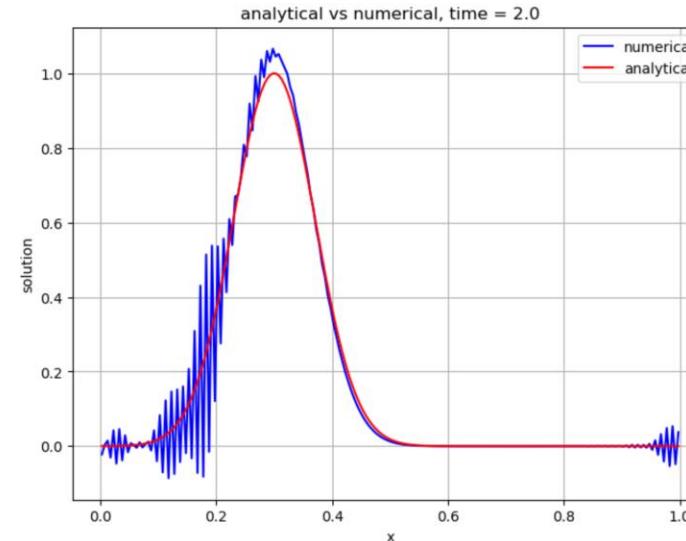
- Условие CFL является неотъемлемой частью **явных** схем для моделирования гидродинамических течений
- При $CFL > 1$ мы выходим физически разрешимую область влияния модели
- Более «обыденное» определение:
За один шаг по времени возмущение не должно проходить более одной ячейки



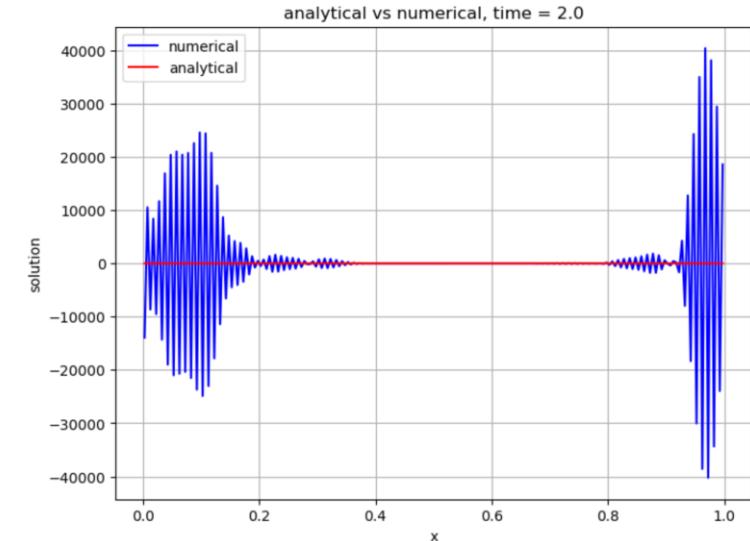
Courant-Friedrichs-Lowy (1928) condition – ещё более наглядное объяснение!



CFL = 0.5



CFL = 1.05



CFL = 1.07

Связь метода конечного объема с методом конечных разностей

- Метод конечного объема (МКО):

$$\frac{\bar{U}_i^{n+1} - \bar{U}_i^n}{\Delta t} + \frac{\bar{F}_{i+1/2}^{n+1/2} - \bar{F}_{i-1/2}^{n+1/2}}{\Delta x_i} = \bar{R}_i^{n+1/2}$$

Пусть $F_{i+1/2}^{n+1/2} = \frac{aU_{i+1}^n + aU_i^n}{2}$ – центральный поток со вторым порядком аппроксимации в пространстве и первым порядком аппроксимации по времени

- Метод конечных разностей (МКР):

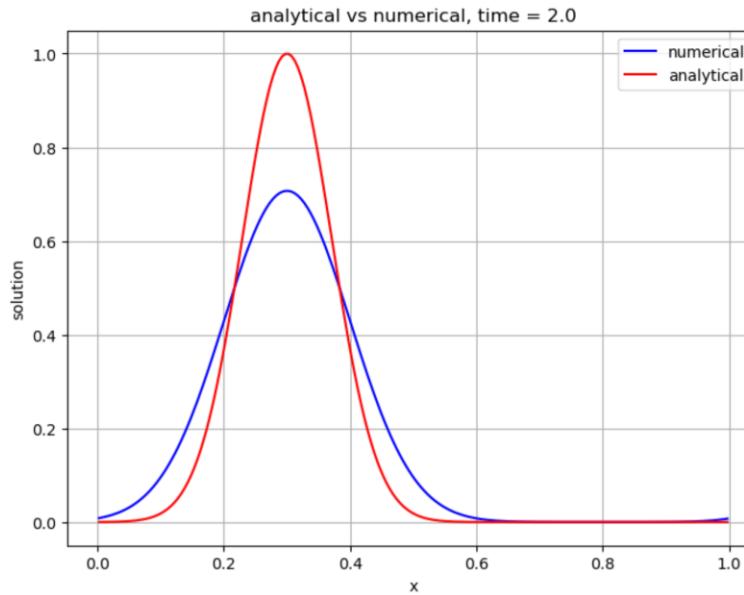
$$\text{Пусть } \frac{U_i^{n+1} - U_i^n}{\Delta t} + a \frac{U_{i+1}^n - U_{i-1}^n}{\Delta x} = 0$$

Тогда при $R_i^{n+1/2} = 0$ получаем, что схемы для МКО и МКР эквивалентные

Трактовка решения немного разная – в МКО решение «размазано» по ячейке, а в МКР оно определено в дискретных точках – узлах сетки

Численная диффузия

- Мы получили решение с первым порядком точности
- В рамках такого подхода решение «размазывается» со временем
- Это можно трактовать как решение «реального» уравнения с учетом члена в форме вязкости $\nu \sim \Delta x$

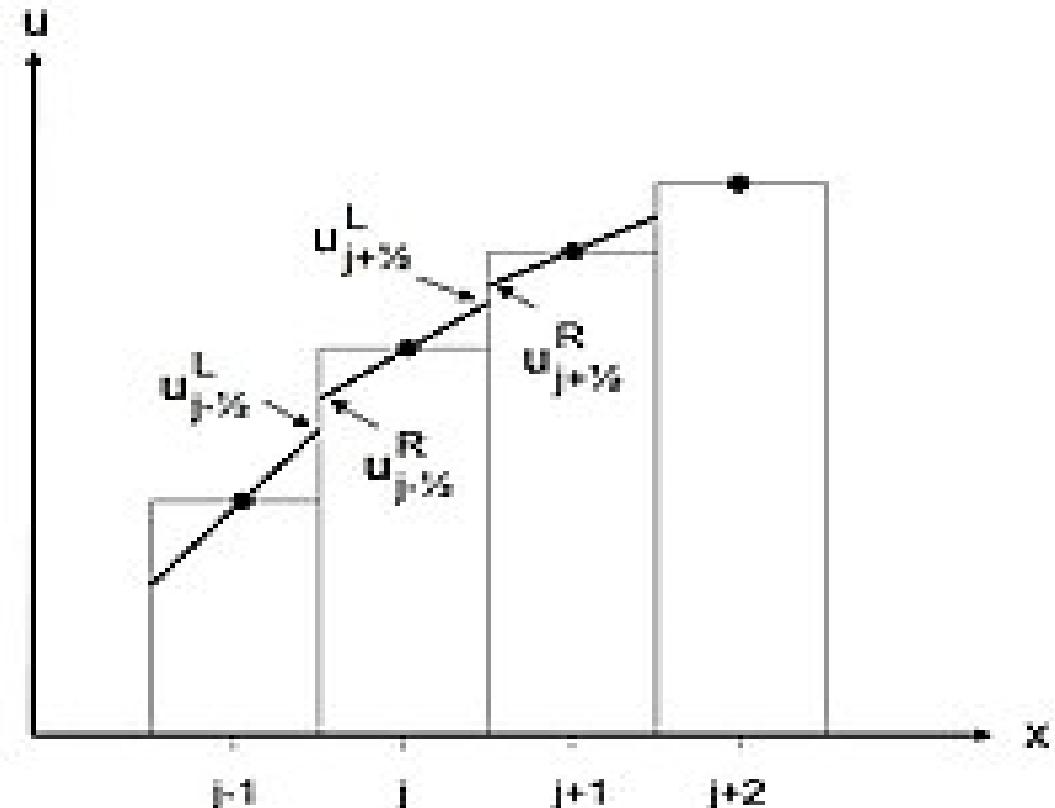


$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} = \nu \frac{\partial^2 U}{\partial x^2}, \quad U(x, t) \sim \exp(i k(x - at) - \nu k^2 t)$$

- Попробуем построить схему с более высоким порядком точности

Схемы второго порядка точности

- Мы использовали кусочно-постоянное решение для вычисления потока F (**PCM – Piecewise constant method**)
- Попробуем взять линейную реконструкцию с учетом данных о соседних ячейках по времени и по пространству (**PLM**)



Piecewise linear extrapolation

Схемы второго порядка точности

- Уточним величины справа и слева от разрыва:

$$U_L = U_i^n + \frac{\Delta x}{2} \left(\frac{\partial U}{\partial x} \right)_i^n + \frac{\Delta t}{2} \left(\frac{\partial U}{\partial t} \right)_i^n; \quad U_R = U_{i+1}^n - \frac{\Delta x}{2} \left(\frac{\partial U}{\partial x} \right)_{i+1}^n + \frac{\Delta t}{2} \left(\frac{\partial U}{\partial t} \right)_{i+1}^n$$

- Используем исходное уравнение для исключения производной по времени:

$$U_L = U_i^n + \frac{\Delta x}{2} \left(\frac{\partial U}{\partial x} \right)_i^n - a \frac{\Delta t}{2} \left(\frac{\partial U}{\partial x} \right)_i^n; \quad U_R = U_{i+1}^n - \frac{\Delta x}{2} \left(\frac{\partial U}{\partial x} \right)_{i+1}^n - a \frac{\Delta t}{2} \left(\frac{\partial U}{\partial x} \right)_{i+1}^n$$

- Получили ($+O(\Delta t^2 + \Delta x^2)$):

$$U_L = U_i^n + \frac{\Delta x - a \Delta t}{2} \left(\frac{\partial U}{\partial x} \right)_i^n; \quad U_R = U_{i+1}^n - \frac{\Delta x + a \Delta t}{2} \left(\frac{\partial U}{\partial x} \right)_{i+1}^n$$

- Пусть $\left(\frac{\partial U}{\partial x} \right)_i^n = \left(\frac{\partial U}{\partial x} \right)_{i+1}^n = \frac{U_{i+1}^n - U_i^n}{\Delta x}$ – получим схему Лакса-Вендроффа (1960)

Схема Лакса-Вендроффа (1960)

- Поток схемы LW равен:

$$F_{i+1/2}^{n+1/2} = \frac{aU_L + aU_R}{2} - |a| \frac{U_R - U_L}{2}$$

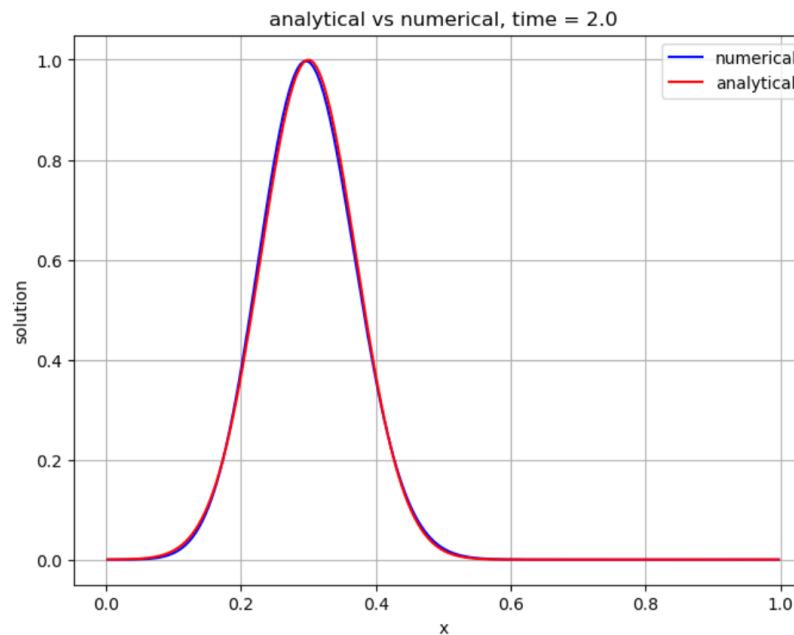
- В исходных обозначениях (см. файл **advection1D.ipynb**):

$$F_{i+1/2}^{n+1/2} = \frac{aU_{i+1}^n + aU_i^n}{2} + a^2 \frac{\Delta t}{2\Delta x} (U_i^n - U_{i+1}^n)$$

- Данная схема имеет второй порядок точности по времени и пространству

Схема Лакса-Вендроффа для гладкого профиля

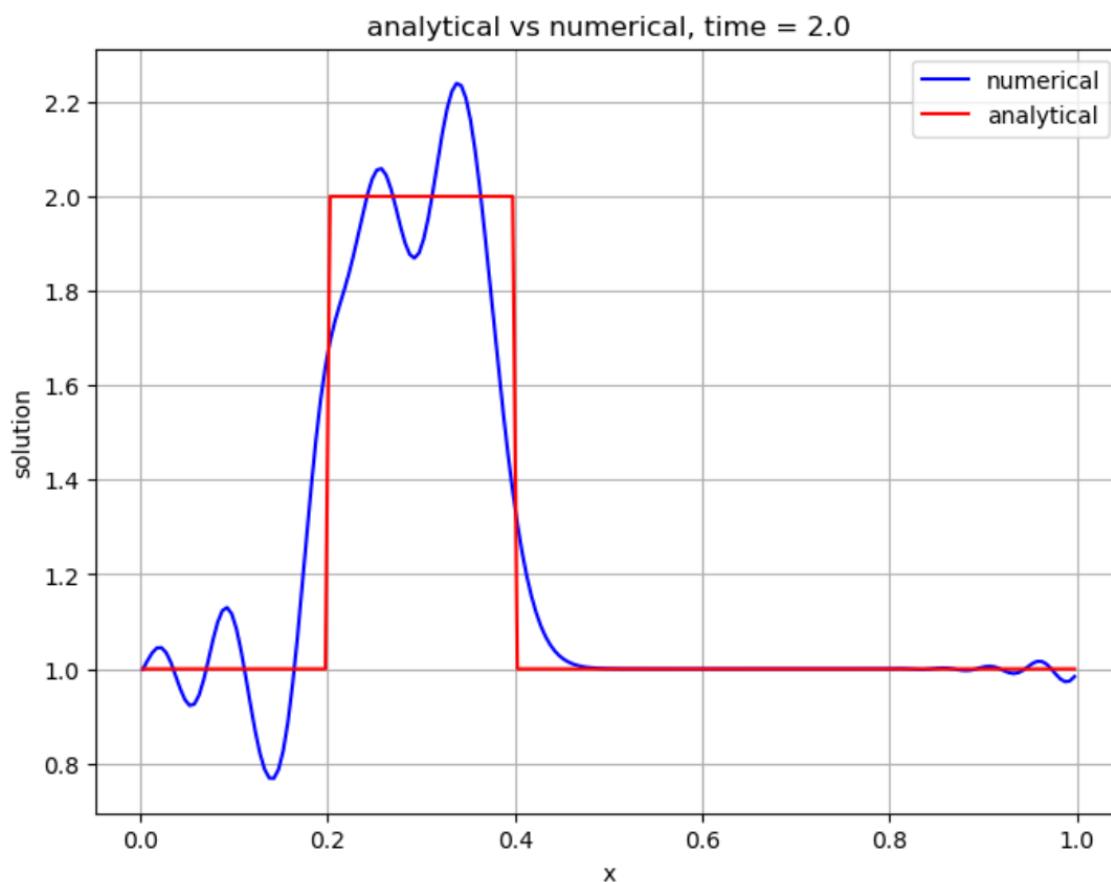
- Можно показать, что такая схема будет иметь меньший уровень численной диссипации (порядок аппроксимации =**2** по времени и пространству вместо =**1** для решения методом задачи Римана)
- Для гладкого профиля решение почти идеально совпадает с аналитическим ('smooth')



- Посмотрим, что происходит с разрывным профилем ('disc')

Схема Лакса-Вендроффа для разрывного профиля

- Получили что-то странное...

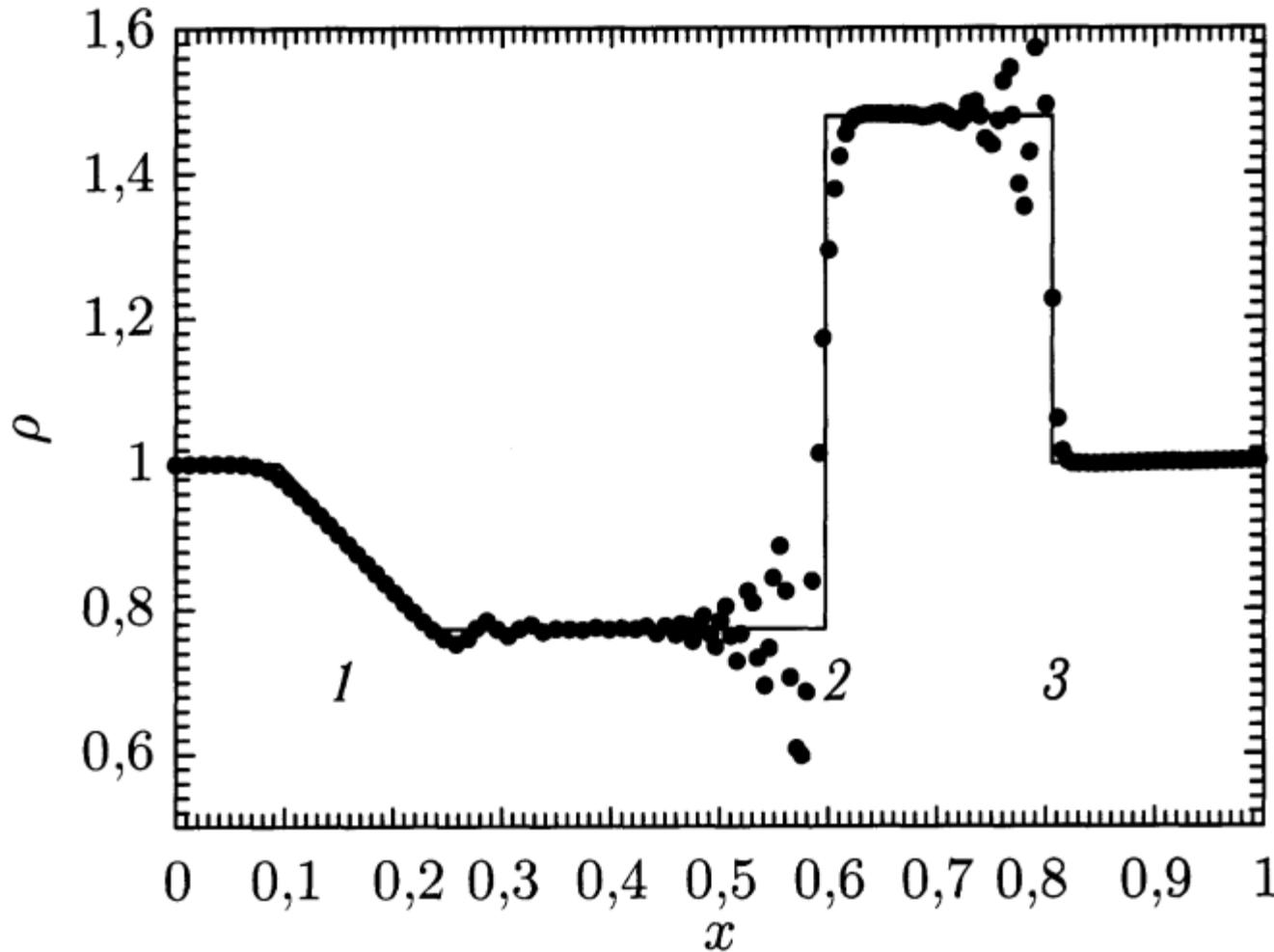


Для коротких волн (разрыв)
имеем дисперсию с
коэффициентом $\mu \sim \Delta x^2$,
т.е. скорость зависит от
длины волны

$$\frac{\partial U}{\partial t} + a \frac{\partial U}{\partial x} = \mu \frac{\partial^3 U}{\partial x^3}, \quad U(x, t) \sim \exp(i k(x - at) - i \mu k^3 t)$$

Получается, что схемы с первым
порядком эффективно решают
уравнения с вязкостью, а со
вторым – с дисперсией
(с ростом порядка дисперсия все
равно останется)

Пример из газовой динамики (т.н. “Gibbs phenomenon”)



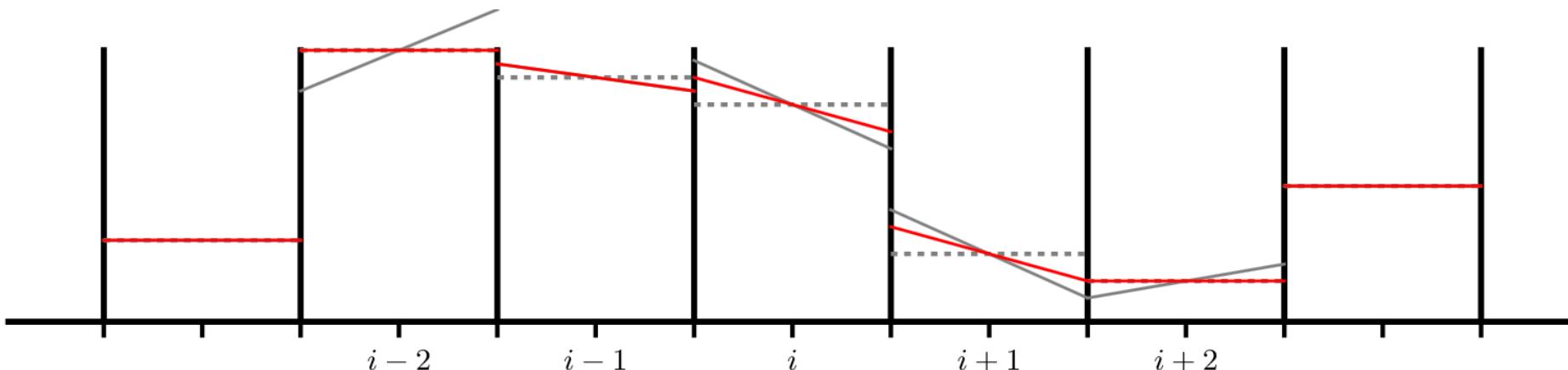
Видим
«overshooting»
для схемы Лакса-
Вендроффа

Д.В. Бисикало, А.Г. Жилкин,
А.А. Боярчук «Газодинамика
тесных двойных звезд» — М.:
Физматлит, 2013

Осцилляции, немонотонность и TVD-принцип

- К уравнению адвекции применим принцип максимума: **решение всегда должно лежать внутри минимального и максимального значений**
- Также у нас перестала сохраняться монотонность решения!
- Численная дисперсия совместно с немонотонностью приводят к возникновению осцилляций вблизи областей с большими градиентами
- В более общем случае, говорят о росте полной вариации решения (A. Harten (1983)):

$$TV(\phi) = \sum_{j=1}^{N-1} |\phi_{j+1} - \phi_j|$$



Свойство TVD – total variation diminishing

- С.К. Годунов (1959) показал, что любая **линейная** схема для уравнения адвекции (справедливо и для других систем) не является монотонной, если ее **порядок точности выше первого**
- Как можно обойти это условие? Сделать схему **нелинейной!**
- Введем «анализатор» гладкости:

$$R = \frac{U_{i+1} - U_i}{U_i - U_{i-1}}$$

- $R \gg 1, R \ll 1 \Rightarrow$ находимся в области сильного градиента (разрыва)
- $R < 0 \Rightarrow$ находимся в точке экстремума
- $R \sim 1 \Rightarrow$ всё ОК

Ограничение наклонов при уточнении переменных (MUSCL-подход, [B. vanLeer 1977, 1982])

- Основная идея: *Попробуем возвращаться к монотонной схеме с первым порядком там, где полная вариация нарастает*
- В точках с сильным градиентом будем ограничивать наклон, а на экстремумах будем отключать второй порядок точности
- $U_L = U_i^n + \frac{\Delta x - a\Delta t}{2} \left(\frac{\partial U}{\partial x} \right)_i^n$; $U_R = U_{i+1}^n - \frac{\Delta x + a\Delta t}{2} \left(\frac{\partial U}{\partial x} \right)_{i+1}^n$ – градиенты надо ограничить!

- $\left(\frac{\partial U}{\partial x} \right)_i^n = \text{limiter} \left(\frac{U_i^n - U_{i-1}^n}{\Delta x}, \frac{U_{i+1}^n - U_i^n}{\Delta x} \right) = \frac{U_i^n - U_{i-1}^n}{\Delta x} * \text{limiter}(1, 1/R)$
- $\left(\frac{\partial U}{\partial x} \right)_{i+1}^n = \text{limiter} \left(\frac{U_{i+1}^n - U_i^n}{\Delta x}, \frac{U_{i+2}^n - U_{i+1}^n}{\Delta x} \right)$

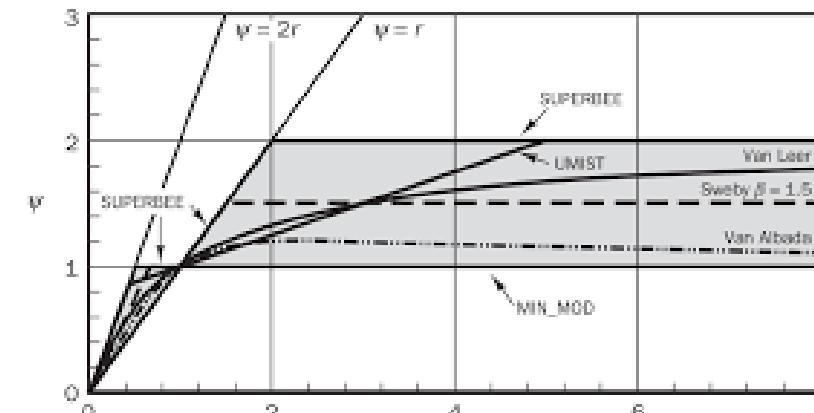
И снова считаем поток по уточненным значениям:

$$F_{i+1/2}^{n+1/2} = \frac{aU_L + aU_R}{2} - |a| \frac{U_R - U_L}{2}$$

Виды ограничителей наклона (limiter'ы)

- Множество разных авторов и вариантов, наиболее популярные (см. также `reconstruction.py`, в нем функции '`PLM_rec`' и '`limiter`'):

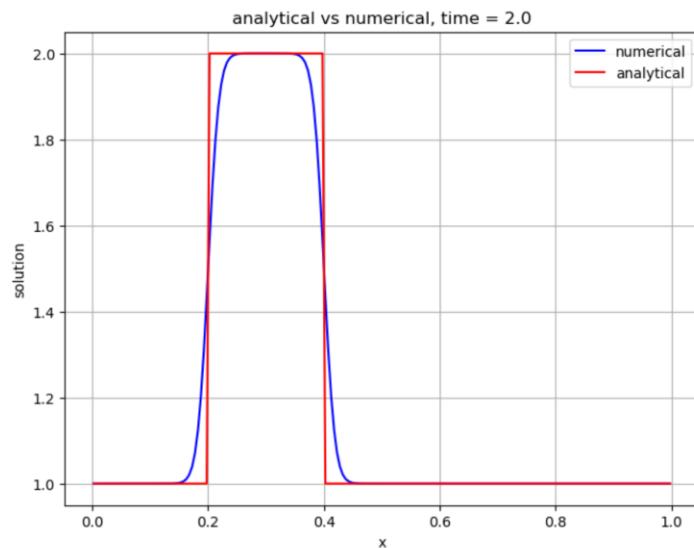
```
def limiter(x, y, limiter_type):  
  
    # Smoothness analyzer  
    r = (y + 1e-14) / (x + 1e-14)  
  
    if limiter_type == 'VL':  
        #vanLeer limiter  
        df = x * (np.abs(r) + r) / (1.0 + np.abs(r))  
  
    elif limiter_type == 'MM':  
        # minmod limiter -- the most diffusive one (but the most stable)  
        df = 0.5 * x * (1.0 + np.sign(r)) * np.minimum(1.0, np.abs(r))
```



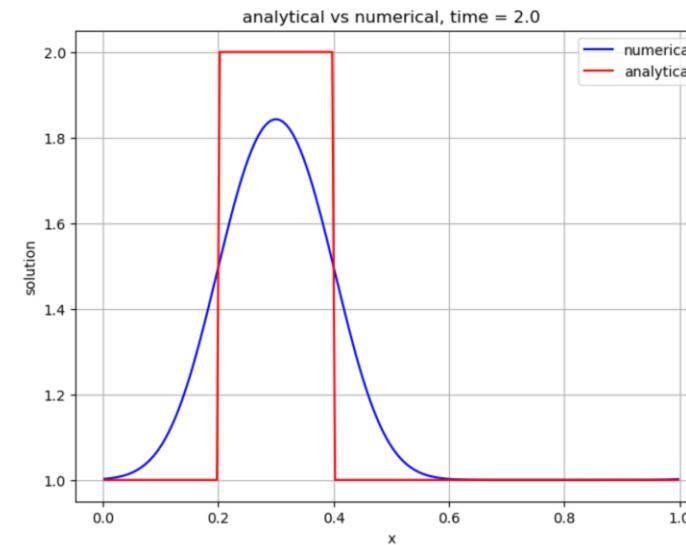
Ref: P. Sweby (1984)

- При использовании готовых пакетов зачастую при вводе параметров симуляции также можно выбрать тип ограничителя

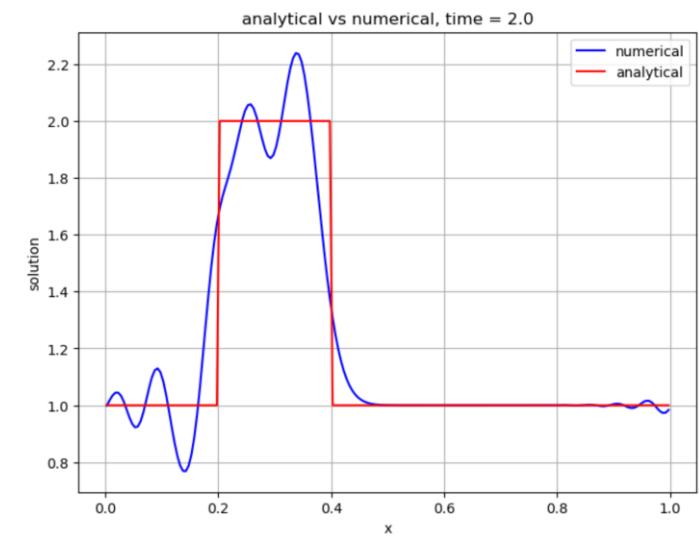
Получили решение, которое «помнит» свою исходную форму



PLM + limiter



PCM (1st order)



PLM + no limiting
(Lax-Wendroff)

Квазимонотонные методы высокого порядка

- В итоге мы получили второй порядок везде кроме областей с большими градиентами/разрывами или экстремумами (**PLM**)
- Для **2D/3D** задач все делается **аналогично** – используются те же формулы, но в другом направлении
- Можно попробовать построить метод с третьим порядком – вместо линейных полиномов возьмем параболы (+ процедура «лимитирования») – такой подход будет называться кусочно-параболическим методом (**PPM**) [**Colella & Woodward 1984**]:

$$u_i(x) = \bar{u}_i + \hat{u}_x x + \hat{u}_{xx} \left(x^2 - \frac{1}{12} \right)$$

Методы Рунге-Кутты для интегрирования по времени (method-of-lines)

- Данный подход позволяет очень просто развивать схемы на высокий порядок по времени путем учета дополнительных стадий [Shu and Osher 1988 – TVD-RK-методы] – удобно для 2D/3D подходов и схем высокого порядка

$$\frac{d\bar{\mathbf{U}}_i}{dt} + \frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{\Delta x} = \bar{\mathbf{R}}_i$$



Получили систему ОДУ во времени после аппроксимации схемы по пространству

- RK2 – второй порядок

$$\mathbf{U}^{(1)} = \mathbf{U}^n + \Delta t L(\mathbf{U}^n)$$

$$\mathbf{U}^{n+1} = \frac{1}{2}\mathbf{U}^n + \frac{1}{2}\mathbf{U}^{(1)} + \frac{1}{2}\Delta t L(\mathbf{U}^{(1)})$$

$L(\mathbf{U})$ – это наш оператор с потоками и уточнением по пространству

- RK3 – третий порядок

$$\mathbf{U}^{(1)} = \mathbf{U}^n + \Delta t L(\mathbf{U}^n)$$

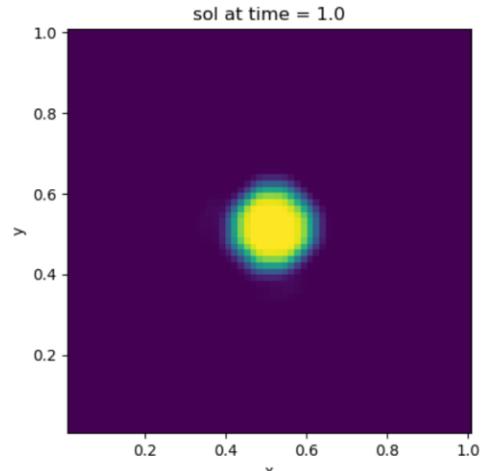
$$\mathbf{U}^{(2)} = \frac{3}{4}\mathbf{U}^n + \frac{1}{4}\mathbf{U}^{(1)} + \frac{1}{4}\Delta t L(\mathbf{U}^{(1)})$$

$$\mathbf{U}^{n+1} = \frac{1}{3}\mathbf{U}^n + \frac{2}{3}\mathbf{U}^{(2)} + \frac{2}{3}\Delta t L(\mathbf{U}^{(2)})$$

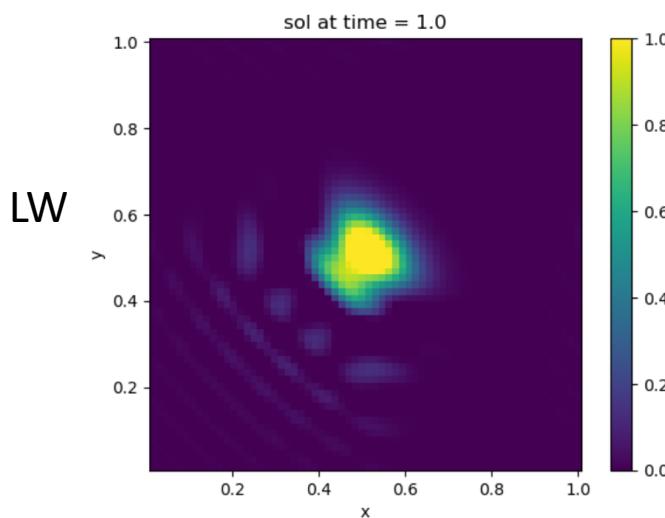
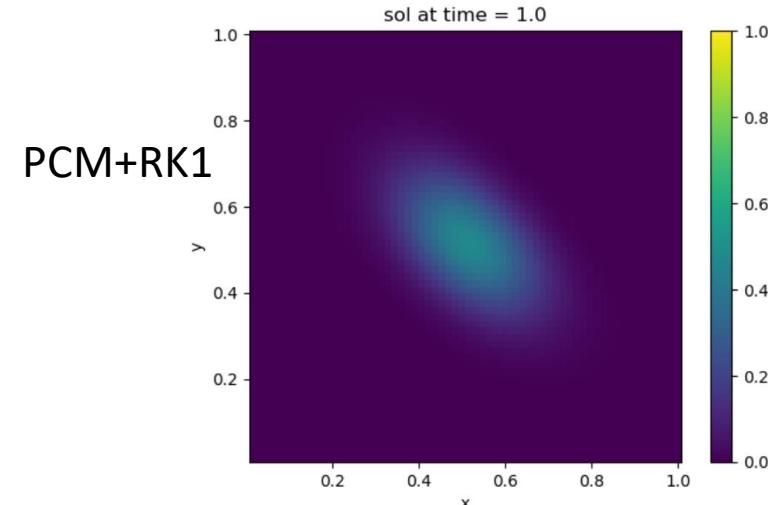
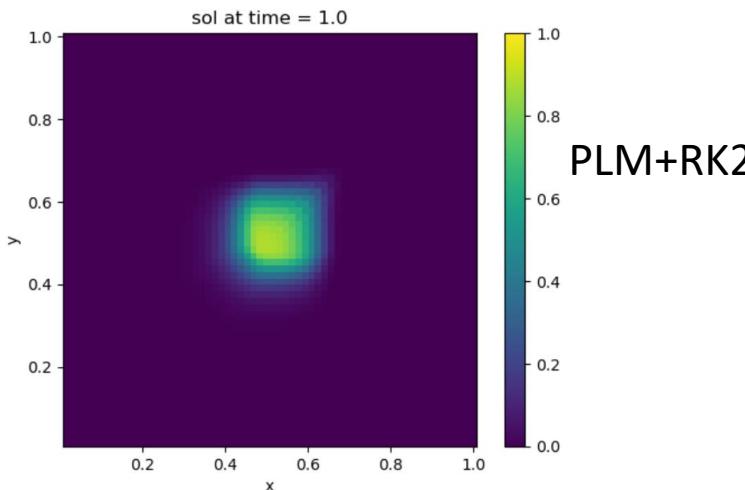
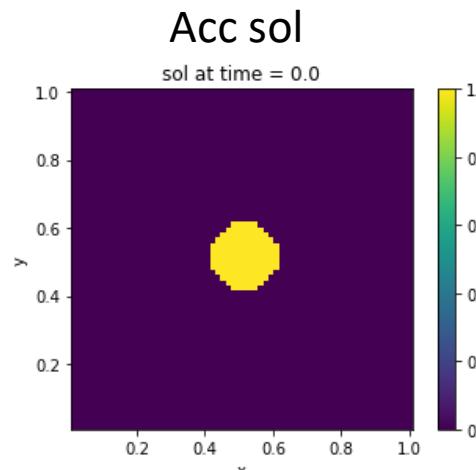
Рассмотрим многомерную адвекцию с разными порядками точности

- Файл `advection_solver_easy_run.ipynb`
- Рассмотрим пример ‘2D’ и методы **PPM + RK3 + adv**, **PLM + RK2 + adv**, **PCM + RK1 + adv**, а также метод **LW**
- PPM почти не изменяет решение, PLM слегка размазывает профиль, а PCM приводит к тому, что профиль полностью размазывается. Как и в 1D, схема LW вносит осцилляции

Многомерная адвекция с разными порядками точности



Сетка 64x64



Теперь мы почти все знаем про адвекцию и можем двигаться дальше!

- ✓ Метод конечного объема и сетка
 - ✓ Задача Римана
 - ✓ Устойчивость
- ✓ Методы высокого порядка

О чем я не говорил:

- Методы SPH – Лагранжев подход (см. [2])
- Схемы Corner Transport Upwind (CTU) – второй порядок по времени без методов Рунге-Кутты, но необходимо учитывать поперечные потоки каждый раз при решении уравнений в любом направлении (см. [1])

Моделирование гидродинамических течений

Газовая динамика

- Уравнения имеют схожую с адвекцией форму

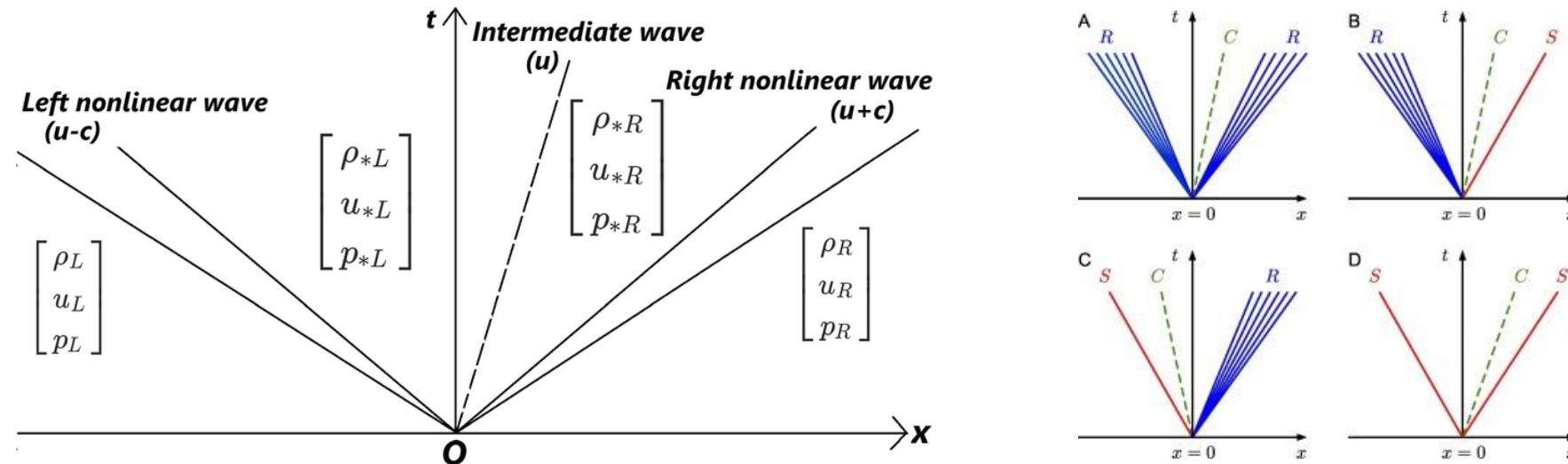
$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ E \end{pmatrix}, \quad \mathbf{F}_x = \begin{pmatrix} \rho v_x \\ \rho v_x^2 + p \\ \rho v_x v_y \\ (E + p)v_x \end{pmatrix}, \quad \mathbf{F}_y = \begin{pmatrix} \rho v_y \\ \rho v_x v_y \\ \rho v_y^2 + p \\ (E + p)v_y \end{pmatrix}$$

- Г.Д. – квазилинейная гиперболическая система дифференциальных уравнений в частных производных
- Три вида характеристик - $v_x \pm c_s$ (звуковые) и v_x (энтропийная)

По аналогии с адвекцией, можно определить задачу Римана вблизи каждой грани (ее также называют *задачей о распаде произвольного разрыва*) – решив эту задачу *точно*, мы получим метод Годунова [С.К. Годунов 1959]

Метод Годунова

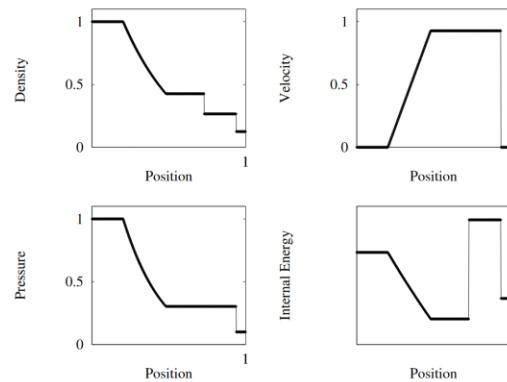
- В задаче о распаде произвольного разрыва могут возникать три вида волн – ударные волны, волны разрежения (простые волны) и контактные разрывы, соответствующие трем характеристикам



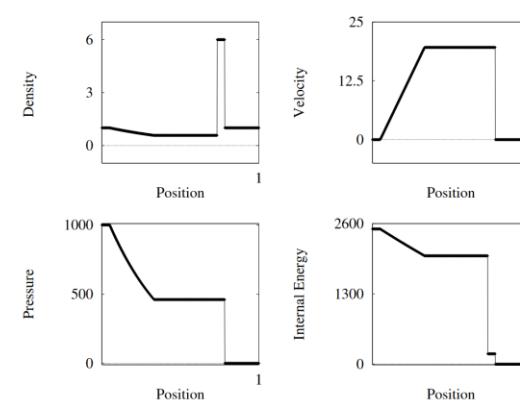
- Условие устойчивости (условие CFL) – задачи Римана не должны зависеть друг от друга – газ в ячейке должен «чувствовать» только локальные волны, которые успевают дойти из соседних ячеек (**Более простая формулировка: самое быстрое возмущение в моделируемой среде не должно распространяться дальше чем на одну ячейку за один шаг по времени**)
- Имеем следующее условие устойчивости: $\Delta t = CFL * \min_{i,j} \left(\frac{\Delta x}{|\nu_x|+c_s}, \frac{\Delta y}{|\nu_y|+c_s} \right)$, $CFL < 1$ for 1D and $CFL < 0.5$ for 2D in PLM methods with RK2 and ~ 0.6 and ~ 0.4 for 2D PPM with RK3

Пример – разные варианты задачи о распаде разрыва в газовой динамике

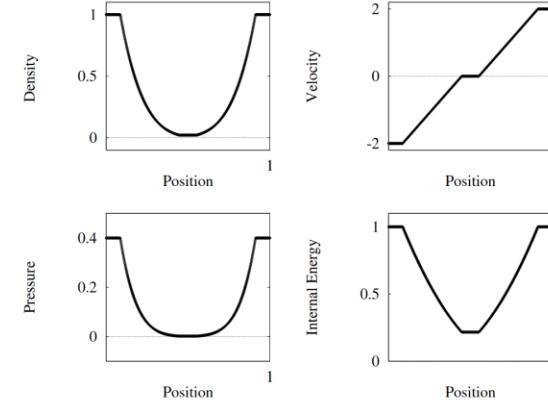
- Файл **fluid_illustration_jupyter.ipynb**



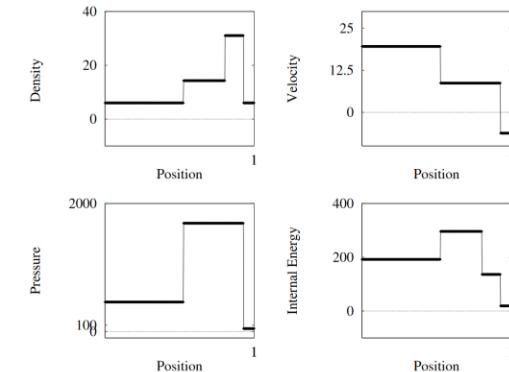
Test problem: Sod1D



Test problem:
strong1D



Две волны разрежения



Две УВ

Кстати, а почему разрыв вообще распадается? ☺

Метод Годунова

- Задача о распаде разрыва решается точно (**[Кочин 1928]**), однако решение данной задачи является очень громоздким и требует решения нелинейного алгебраического уравнения на каждой грани (см. [4], там очень подробно описано решение)
- «+»: позволяет достаточно хорошо разрешать течения с разными разрывами, имеет минимальную численную вязкость
- «-»: вычислительно громоздкий, сложно развить на другие модели (напр., МГД), учесть неидеальное уравнение состояния и пр.

Более того, мы знаем сами состояния газа слева и справа от разрыва лишь приближенно, т.к. мы решаем наши уравнения с конечной точностью! Поэтому решать точно задачу Римана не всегда имеет смысл. Поищем способы попроще!

Методы Годуновского типа

Давайте решим задачу Римана приближенно!

- **Возможны два варианта:**

1. Предположение о более грубой волновой структуре в задаче Римана (напр., возможны только ударные волны) [Harten, Lax and van Leer (1983)] – часто используется физиками и астрофизиками
2. Вместо газодинамической нелинейной задачи решается линеаризованная (акустика), предполагая, что в каждой точке можно локально линеаризовать уравнения [Roe (1981)]

См. книгу [E.F. Toro “Riemann solvers and numerical methods for fluid dynamics – Springer, Berlin, 2009”]

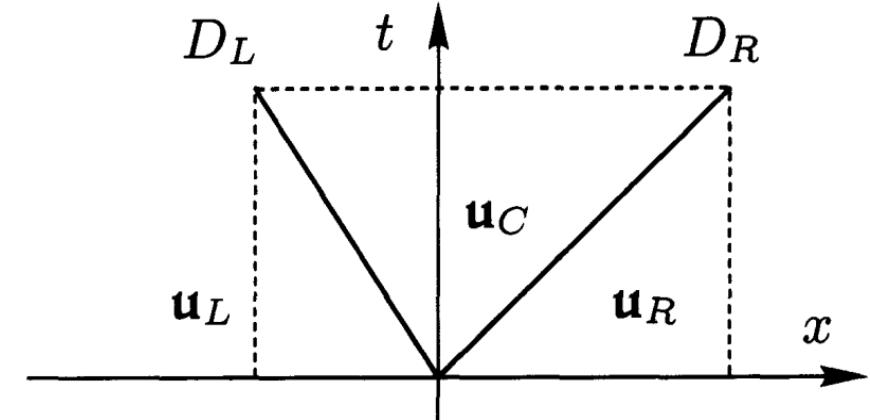
Методы Годуновского типа. Метод HLL (Harten, Lax and van Leer (1983))

- Детали см. в файле **Riemann_fluxes.py**
- Две ударные волны
- Проводится оценка скоростей УВ:

$$D_L = \min(v_{x,L}; v_{x,R}) - \max(c_{s,L}, c_{s,R})$$

$$D_R = \max(v_{x,L}; v_{x,R}) + \max(c_{s,L}, c_{s,R})$$

- Записываются потоки газовых величин на УВ слева и справа от границы
- По характерным скоростям УВ выбирается поток (левый, центральный или правый)



$$\mathbf{F}_* = \begin{cases} \mathbf{F}_L, & D_L \geq 0, \\ \mathbf{F}_C, & D_L \leq 0 \leq D_R, \\ \mathbf{F}_R, & D_R \leq 0. \end{cases}$$

Методы Годуновского типа. Метод HLL (Harten, Lax and van Leer (1983))

- Условия Гюгонио для ударных волн

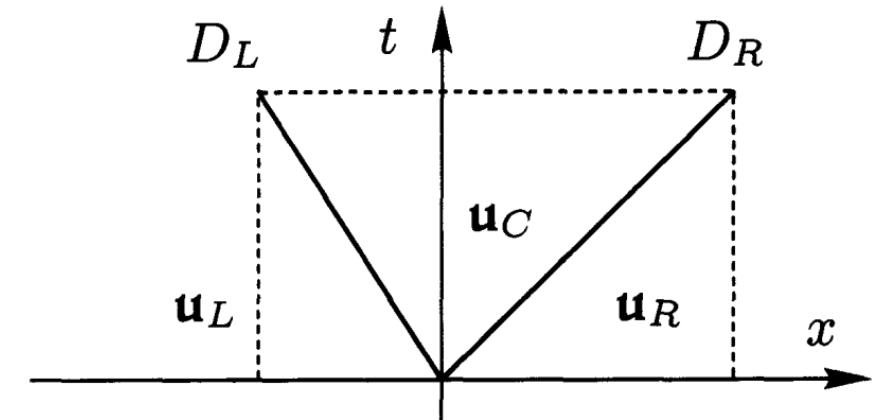
$$D_L (\mathbf{u}_C - \mathbf{u}_L) = \mathbf{F}_C - \mathbf{F}_L,$$

$$D_R (\mathbf{u}_R - \mathbf{u}_C) = \mathbf{F}_R - \mathbf{F}_C.$$

- Получаем решение для состояния газа потока внутри области между УВ

$$\mathbf{u}_C = \frac{D_R \mathbf{u}_R - D_L \mathbf{u}_L - \mathbf{F}_R + \mathbf{F}_L}{D_R - D_L},$$

$$\mathbf{F}_C = \frac{D_R \mathbf{F}_L - D_L \mathbf{F}_R + D_L D_R (\mathbf{u}_R - \mathbf{u}_L)}{D_R - D_L}.$$



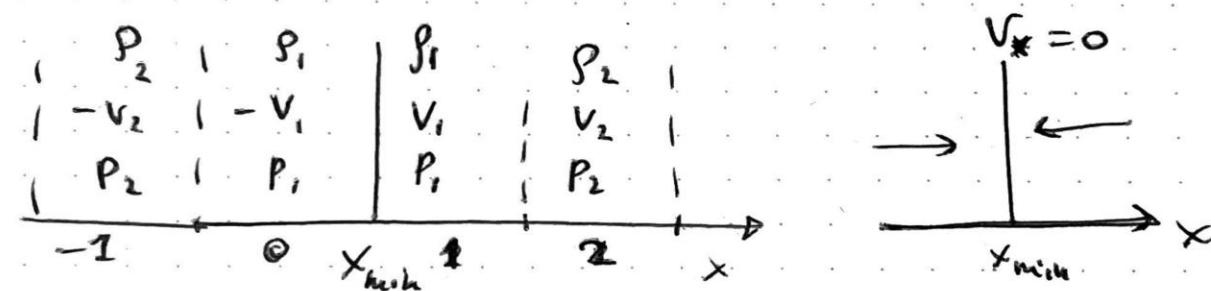
Развитие метода – HLLC, учитывающий контактные разрывы [Toro, Spruce and Pierce (1994)], см. файл **Riemann_fluxes.py**

Такой метод намного лучше моделирует турбулентные течения

Границные условия – ghost cells

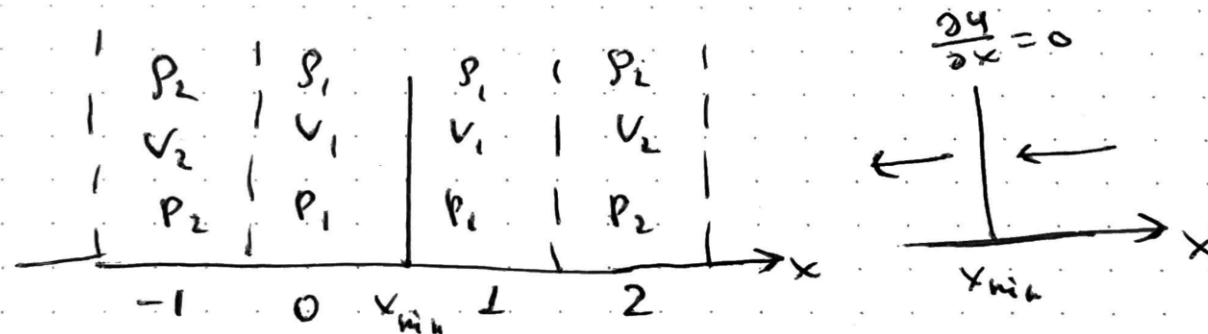
- Периодические – все как для адвекции, только для всех величин газа
- Условие «Стенка»

$$\rho_0 = \rho_1, v_{x,0} = -v_{x,1}, v_{y,0} = v_{y,1}, p_0 = p_1$$



- Условие «Свободная граница» (нулевой градиент)

$$\rho_0 = \rho_1, v_{x,0} = v_{x,1}, v_{y,0} = v_{y,1}, p_0 = p_1$$

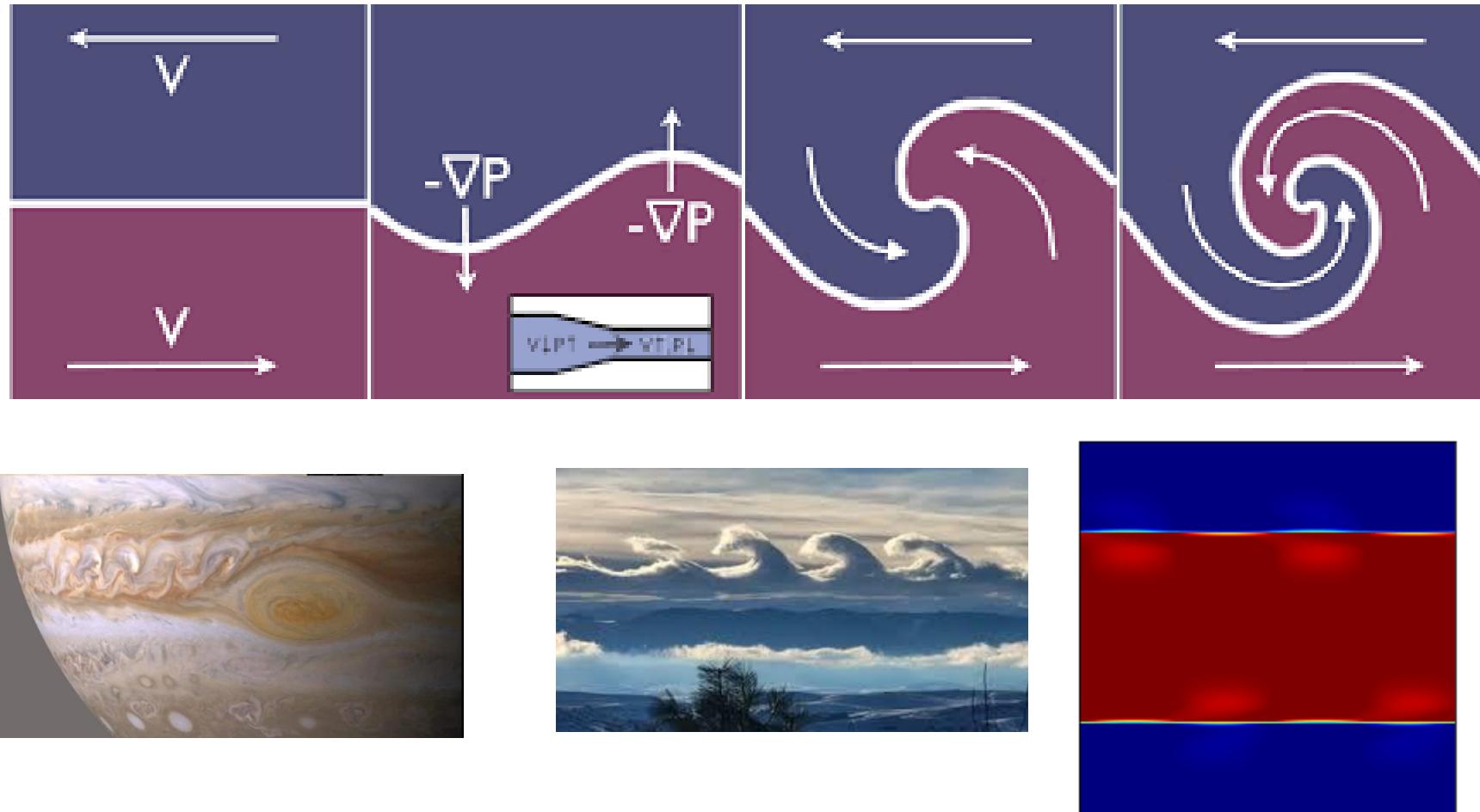


Как быть с высоким порядком и 2D?

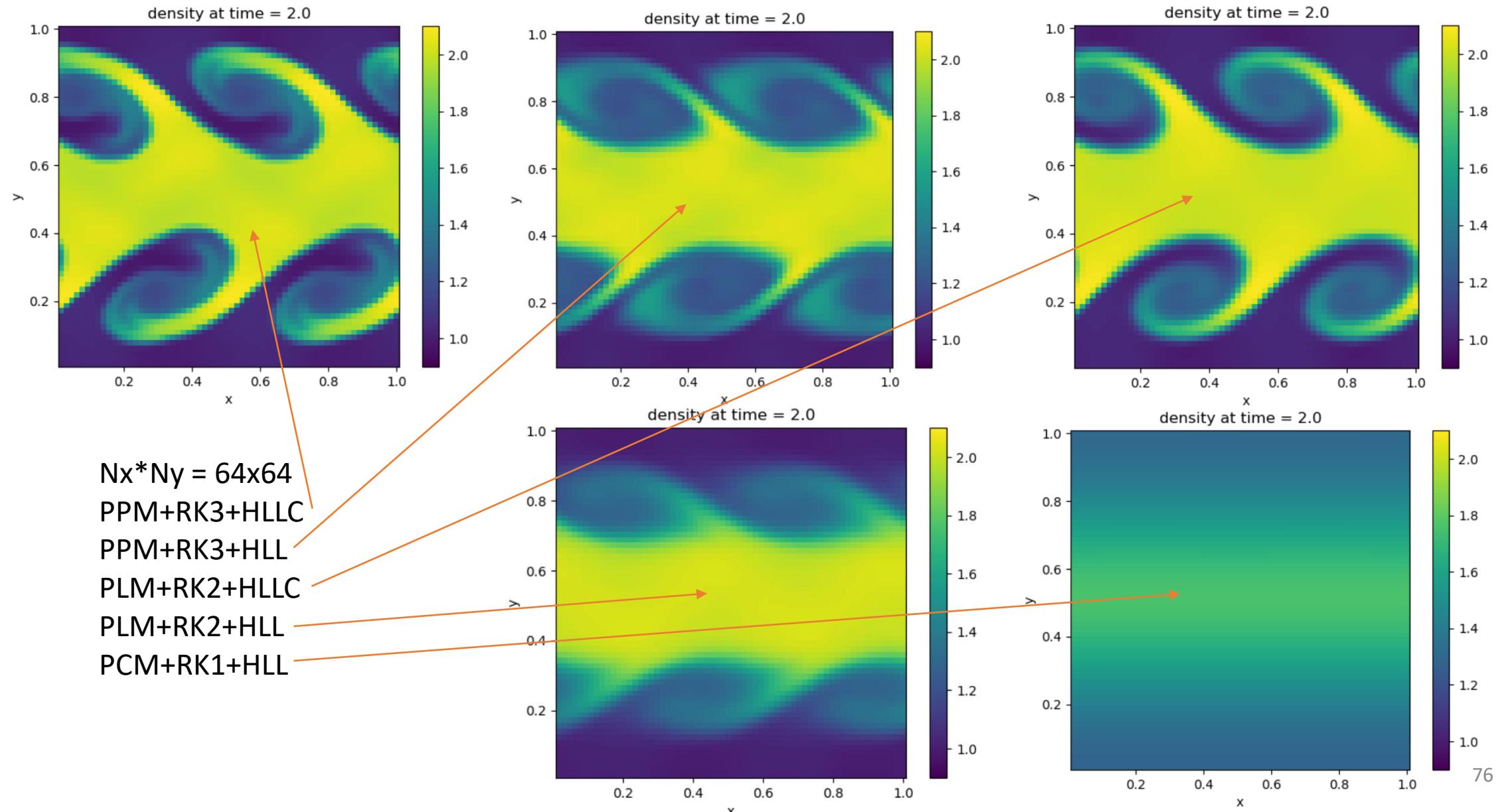
- Точно также! Однако, можно уточнять физические величины (плотность, скорость, давление), либо же сохраняющиеся (массу, импульс, энергию), а также «характеристические» переменные – инварианты Римана
- Выбирается метод аппроксимации по времени, по пространству, а также метод для нахождения потока (см. `Fluid_solver_easy_run.ipynb`)
- Потоки считаются вдоль x и вдоль y независимо друг от друга, для повышения порядка по времени используем методы `RK_X`
- В качестве сравнения методов рассмотрим генерацию неустойчивости Кельвина-Гельмгольца в `fluid_illustration_jupyter.ipynb`
- Детали - см. файлы `one_step_***.py`

Пример: неустойчивость Кельвина-Гельмгольца

- Неустойчивость сдвигового течения в жидкости
- Встречается в огромном количестве космических и земных объектов
- Для запуска можно задать “КНІ” + методы



Неустойчивость Кельвина-Гельмгольца



Моделирование магнитогидродинамических течений

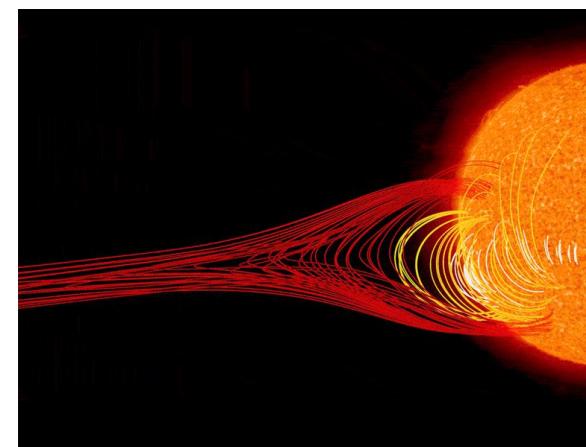
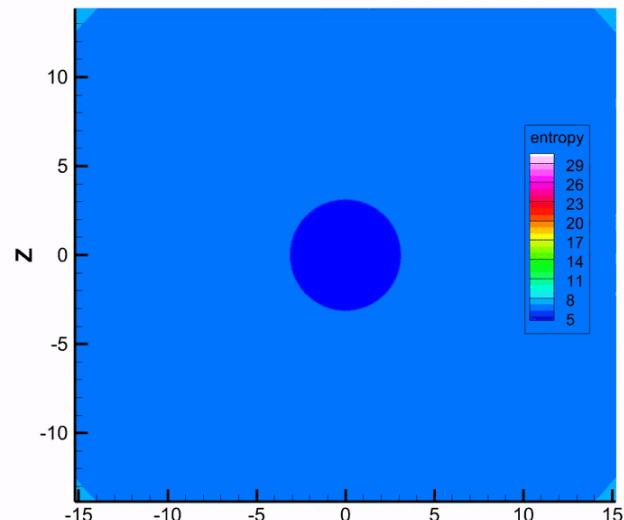
МГД-описание процессов в космосе

Космос

- Магнитосфера звезд и планет
- Дисковая акреция (от протозвезд до АЯГ)
- Коллимация джетов
- Магниторотационный механизм взрыва сверхновой
- Корональные выбросы массы
- Пульсарные туманности
- T Tauri, объекты Хербига-Аро
- Звездный магнетизм и активность

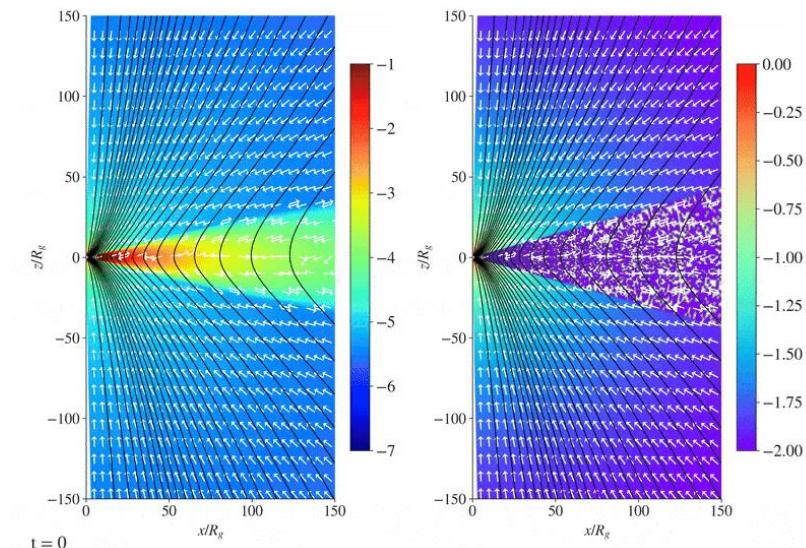
I A Kondratyev et al, Phys Rev D (2024)

Пример моделирования взрыва
магниторотационной сверхновой



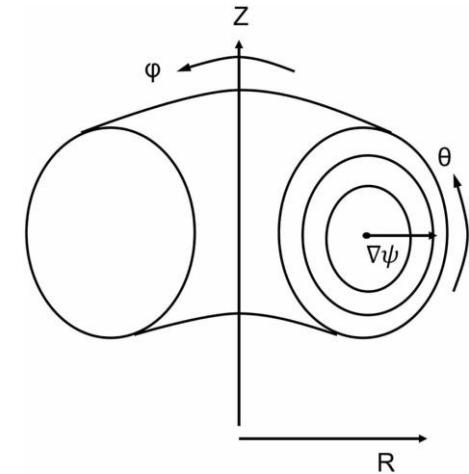
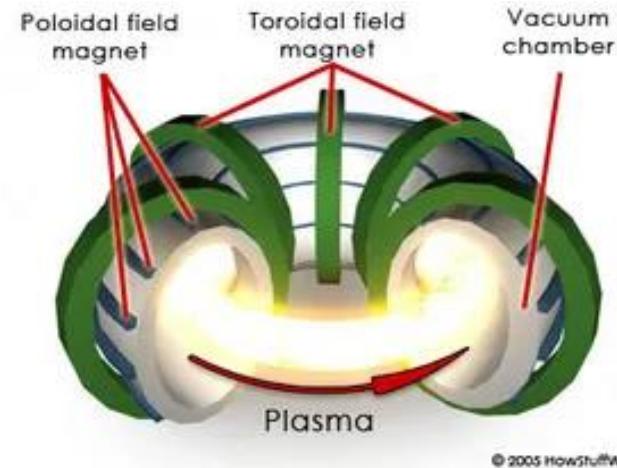
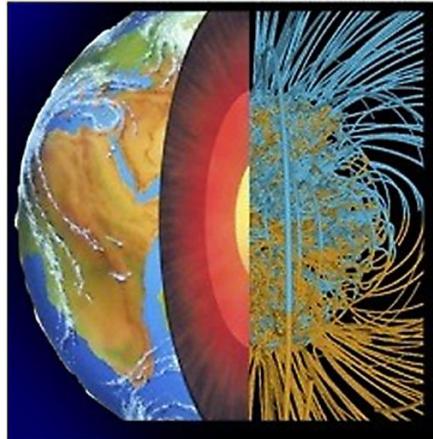
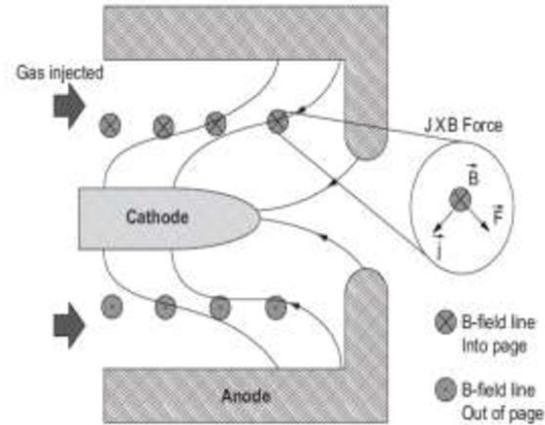
[Vourellis, Fendt, Qian, Noble, ApJ 882, 2V \(2019\)](#),

[Qian, Fendt, Vourellis, ApJ 859, 28Q \(2018\)](#),
[Qian, Fendt, Noble, Bugli \(2017\)](#). – пример расчетов акреции и джетов вблизи ЧД



Корональный выброс
массы Солнцем

МГД-описание процессов на Земле



Земля

- Геодинамо, солнечно-земные связи, aurora borealis и т.д.
- МГД-генераторы, плазменные ускорители
- Магнитное удержание плазмы (термояд)

$$\nabla \cdot \mathbf{B} = 0, \quad \nabla \times \mathbf{B} = \frac{4\pi}{c} \mathbf{j}, \quad \nabla p + \frac{\mathbf{j} \times \mathbf{B}}{c} = 0.$$

2D axisymmetry – Grad-Shafranov equation

$$\frac{1}{r} \Delta^* \Psi = \frac{\partial}{\partial r} \left(\frac{1}{r} \frac{\partial \Psi}{\partial r} \right) + \frac{\partial}{\partial z} \left(\frac{1}{r} \frac{\partial \Psi}{\partial z} \right) = -j_\varphi = - \left(r \frac{dp}{d\Psi} + \frac{1}{r} f \frac{df}{d\Psi} \right)$$

Магнитная гидродинамика

- Вид в форме законов сохранения ($\mathbf{E} = -\mathbf{v} \times \mathbf{B}$)

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ \rho v_z \\ \mathcal{E} \\ B_x \\ B_y \\ B_z \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho v_x & \rho v_x \\ \rho v_x^2 + P + B^2/8\pi - B_x^2/4\pi & \rho v_x v_y - B_x B_y/4\pi \\ \rho v_x v_y - B_x B_y/4\pi & \rho v_x v_z - B_x B_z/4\pi \\ \rho v_x v_z - B_x B_z/4\pi & (\mathcal{E} + P + \mathbf{B}^2/8\pi)v_x - B_x(\mathbf{v} \cdot \mathbf{B})/4\pi \\ 0 & -E_z \\ -E_z & E_y \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} \rho v_y & \rho v_y \\ \rho v_x v_y - B_x B_y/4\pi & \rho v_y^2 + P + B^2/8\pi - B_y^2/4\pi \\ \rho v_y^2 + P + B^2/8\pi - B_y^2/4\pi & \rho v_y v_z - B_y B_z/4\pi \\ \rho v_y v_z - B_y B_z/4\pi & (\mathcal{E} + P + \mathbf{B}^2/8\pi)v_y - B_y(\mathbf{v} \cdot \mathbf{B})/4\pi \\ E_z & 0 \\ 0 & -E_x \end{pmatrix} + \frac{\partial}{\partial z} \begin{pmatrix} \rho v_z & \rho v_z \\ \rho v_x v_z - B_x B_z/4\pi & \rho v_z^2 + P + B^2/8\pi - B_z^2/4\pi \\ \rho v_z^2 + P + B^2/8\pi - B_z^2/4\pi & (\mathcal{E} + P + \mathbf{B}^2/8\pi)v_z - B_z(\mathbf{v} \cdot \mathbf{B})/4\pi \\ -E_y & -E_y \\ E_x & 0 \end{pmatrix} = 0$$

- Имеем уже 7 волн – две быстрых и две медленных магнитозвуковых волны, две альфеновских, а также энтропийная волна

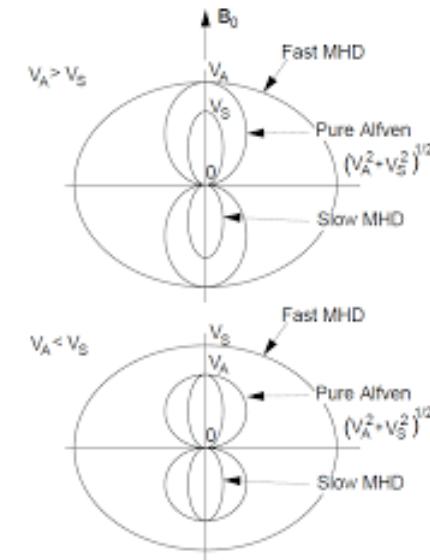
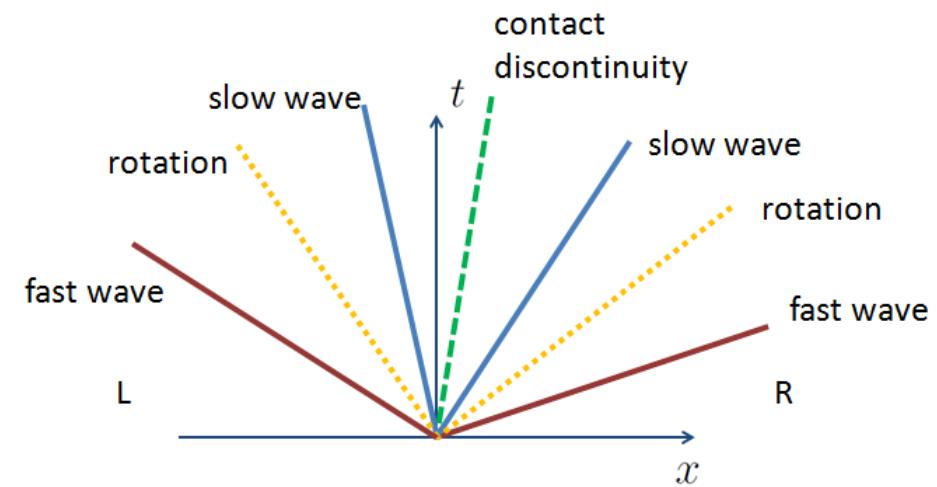
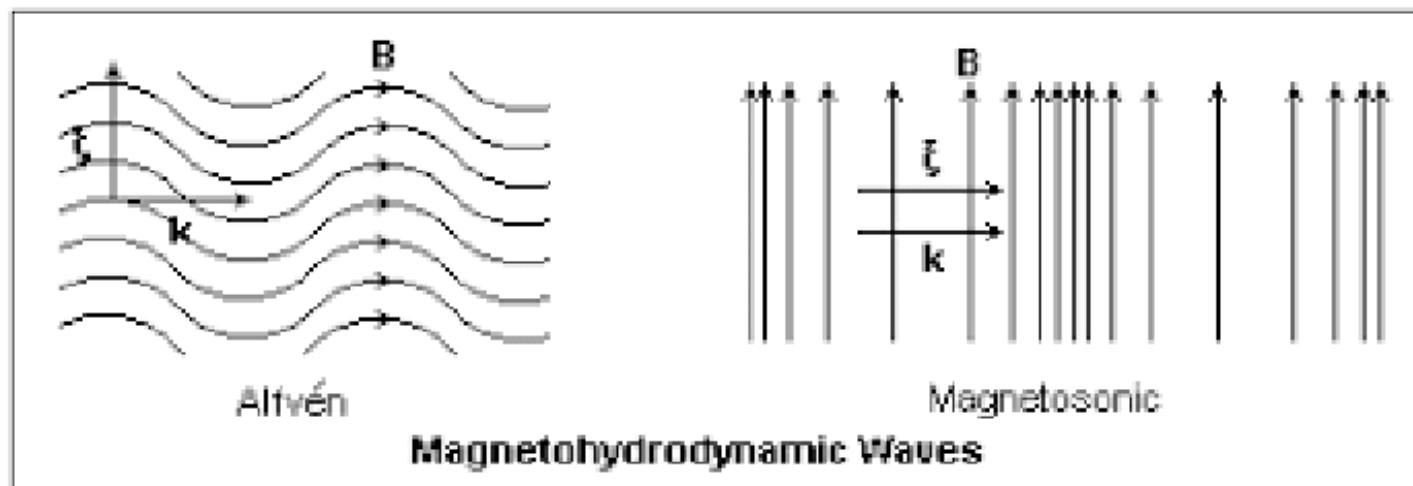
Магнитная гидродинамика – собственные значения и волновые решения

$$\lambda_B = 0, \quad \lambda_E = v_x, \quad \lambda_{\pm A} = v_x \pm a_x,$$

$$\lambda_{\pm S} = v_x \pm u_S, \quad \lambda_{\pm F} = v_x \pm u_F,$$

$$\mathbf{a} = \frac{\mathbf{B}}{\sqrt{\rho}}$$

$$u_{F,S} = \sqrt{\frac{c_s^2 + \mathbf{a}^2}{2}} \pm \frac{1}{2} \sqrt{(c_s^2 + \mathbf{a}^2)^2 - 4a_x^2 c_s^2}.$$



Метод конечного объема (теорема О.-Г. внутри ячейки, 1D)

$$\begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ E \\ B_x \\ B_y \end{pmatrix}_j^{n+1} = \begin{pmatrix} \rho \\ \rho v_x \\ \rho v_y \\ E \\ B_x \\ B_y \end{pmatrix}_j^n - \mathbf{dt} \frac{S_{j+1/2} F_{j+1/2}^{n+1/2} - S_{j-1/2} F_{j-1/2}^{n+1/2}}{V_j}$$

Для получения физически корректного решения необходимо построить монотонную и устойчивую схему, как и раньше, конкретизируя выражение для численных потоков F .

Методы Годуновского типа для МГД

Но точное решение задачи о распаде произвольного разрыва в МГД может иметь...

400+ вариантов – DEAD END!



Однако, мы снова можем использовать приближенные методы + всё, что мы обсуждали про методы высокого порядка, также применимо в МГД

Пример: Схема Русанова (Local Lax-Friedrichs) –
две одинаковых ударных волны)

$$F_{j+1/2} = \frac{F_L + F_R}{2} - S_r \frac{U_R - U_L}{2}$$

- $S_r = \max(\max|\lambda|_L, \max|\lambda|_R)$, т.е. берется локально максимальная скорость распространения волн в системе для левого и правого состояний среды – эквивалентно тому, что у нас распространяются две одинаковых сильных быстрых МГД УВ
- $\Delta t \leq \min_i \left(\frac{\Delta x_i}{\max|\lambda|} \right)$ – условие устойчивости
- Очень простая и универсальная схема для решения широкого класса уравнений

Пример: Метод HLL для уравнений МГД

- Детали см. в файле **Riemann_fluxes.py**

- Две быстрые МГД ударные волны

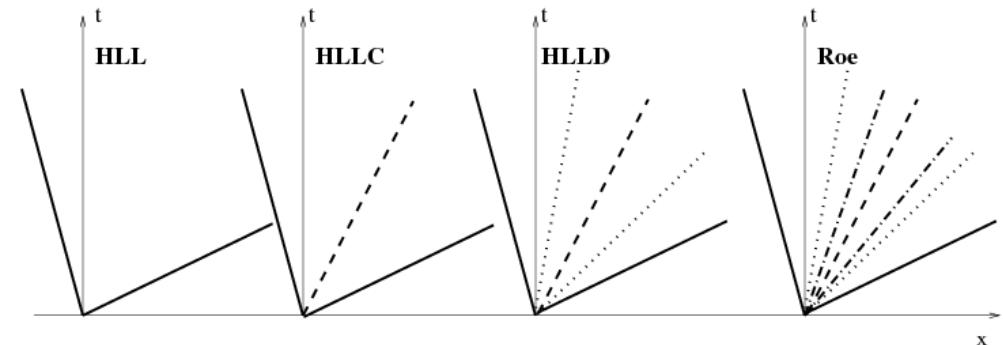
- Проводится оценка скоростей УВ:

$$D_L = \min(v_{x,L}; v_{x,R}) - \max(u_{F,L}, u_{F,R})$$

$$D_R = \max(v_{x,L}; v_{x,R}) + \max(u_{F,L}, u_{F,R})$$

- Записываются потоки МГД-величин для УВ слева и справа от границы

- По характерным скоростям УВ выбирается поток (левый, центральный или правый)

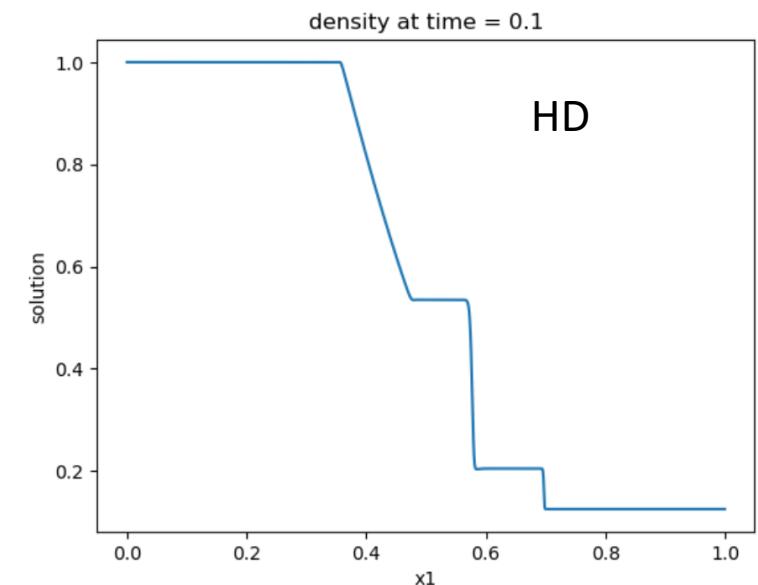
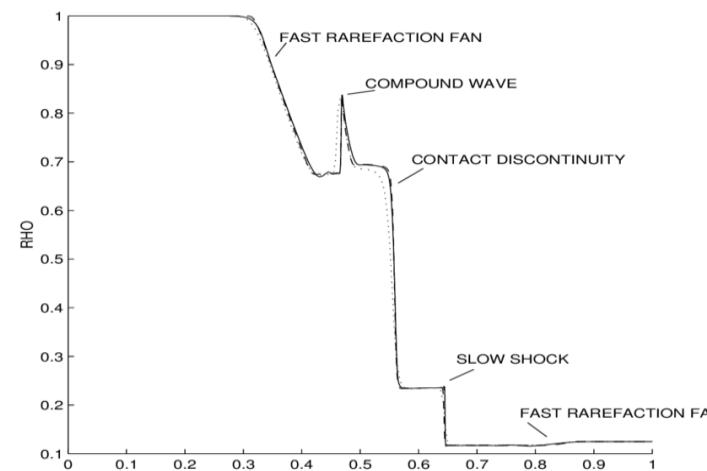
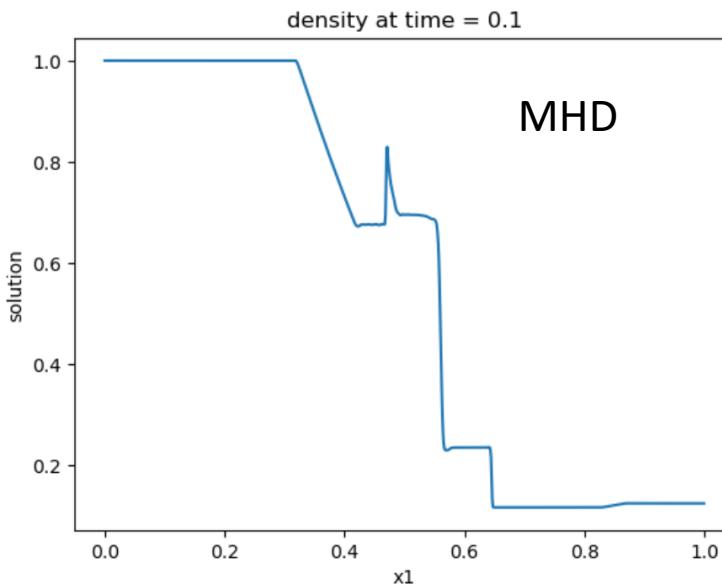


$$\mathbf{F}_* = \begin{cases} \mathbf{F}_L, & D_L \geq 0, \\ \mathbf{F}_C, & D_L \leq 0 \leq D_R, \\ \mathbf{F}_R, & D_R \leq 0. \end{cases}$$

$$\mathbf{F}_C = \frac{D_R \mathbf{F}_L - D_L \mathbf{F}_R + D_L D_R (\mathbf{u}_R - \mathbf{u}_L)}{D_R - D_L}$$

Сравнение с газовой динамикой для разных задач с разрывами

- Рассм. файл `mhd_illustration_jupyter.ipynb`
- Запустим варианты BW1D и BW1D_0



1D МГД уравнения не представляют сложности по сравнению с газовой динамикой; по-настоящему интересными МГД процессы становятся только при многомерном подходе

Проблема возникновения ненулевой дивергенции в многомерных МГД-моделях

- 2D и 3D МГД-расчеты при помощи метода конечного объема могут приводить к численной неустойчивости
- Это происходит ввиду накопления магнитного заряда в точках сетки
- **Дискретный** аналог закона Фарадея не совпадает с дискретным уравнением на магнитное поле из метода конечного объема

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{v} \times \mathbf{B}) \quad \neq \quad \frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{vB} - \mathbf{Bv}) = 0$$

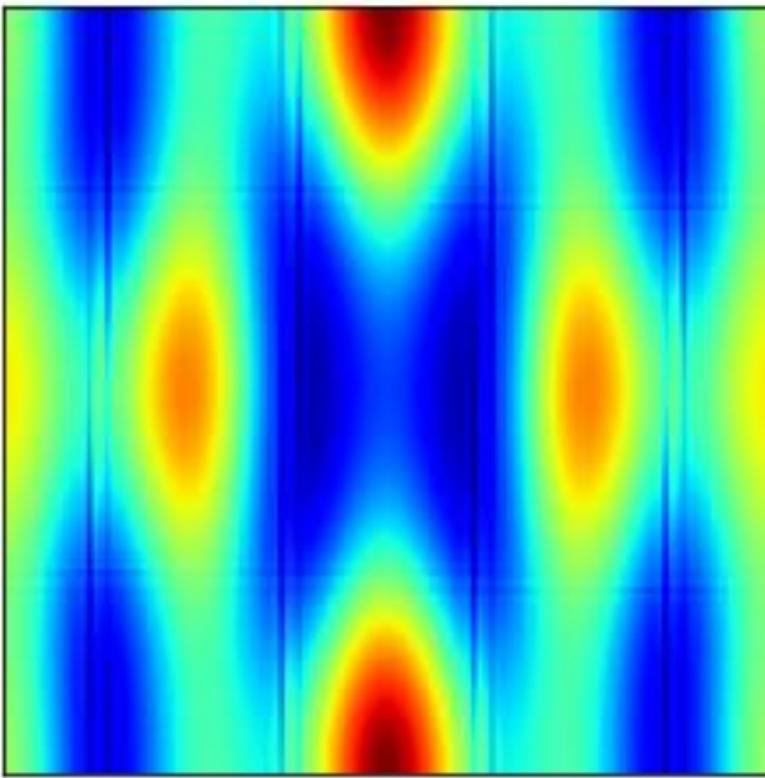
Связывает площадь и контур (магнитный поток через площадку сохраняется)

$$\nabla \cdot \mathbf{B} = 0$$

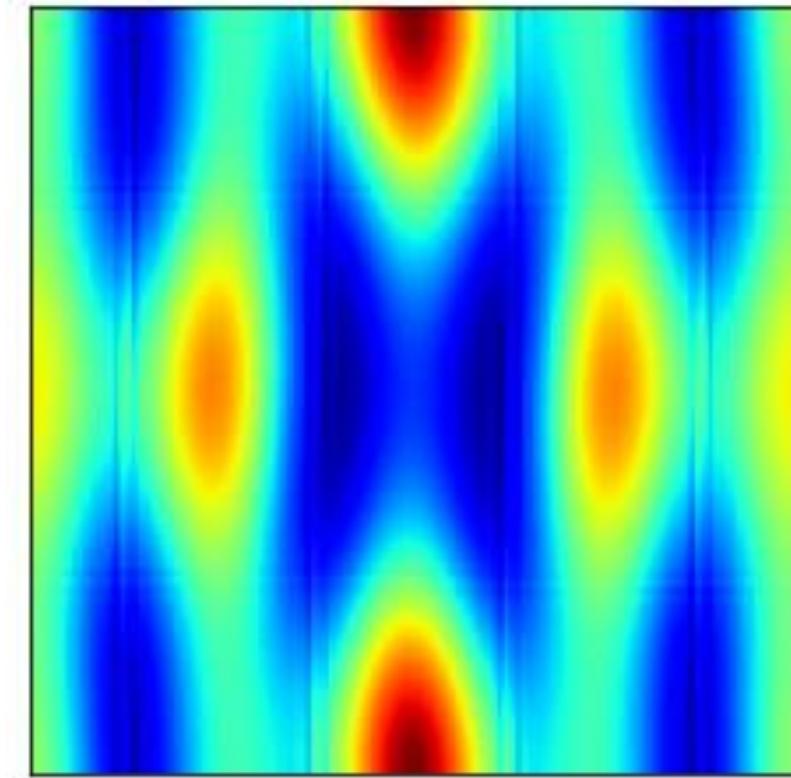
Связывает объем и площадь
(магнитный поток не сохраняется)

$$\nabla \cdot \mathbf{B} \neq 0$$

Пример – переход к МГД-турбулентности (Течение Орзага-Танга)



Есть контроль
дивергенции
(см. ниже)



Ничего не
делаем
(МКО)

Попробуйте убрать очистку
дивергенции в файле
`one_step_mhd.py` и посмотреть, что
получится с другими тестами 😊

Проблема возникновения ненулевой дивергенции в многомерных МГД-моделях

Что происходит? Пусть $\nabla \cdot \mathbf{B} = q_m \neq 0$ – есть монополи

- Тогда по аналогии с силой, действующей на заряд вдоль электрического поля, должна возникать продольная полю «магнитодвижущая» сила

$$F \sim q_m * \mathbf{B} = \mathbf{B} \nabla \cdot \mathbf{B}$$

- Мы никак не учтываем в МКО эту силу + она является не физичной! Однако, поле эволюционирует так, что она должна быть

Методы решения

1. Ничего не делать, надеясь на лучшее (т.к. ошибка в дивергенции порядка ошибок аппроксимации)
2. Каждый шаг по времени удалять из распределения магнитного поля паразитную часть с дивергенцией (проекция на пространство бездивергентных функций)
3. Использовать метод, в котором магнитный заряд бы не возникал и на дискретном уровне
4. Модифицировать МГД-систему, вводя дополнительное уравнение на эволюцию дивергенции/магнитного заряда (чтобы $\text{div} \mathbf{B}$ затухала, «вымывалась» потоком из расчетной области и т.п.)

Варианты 2-4 активно используются

Очистка дивергенции

- Пусть получено поле \mathbf{B}^* после МКО, его можно разложить на вихревую и потенциальную части:

$$\mathbf{B}^* = \nabla \times \mathbf{A} + \nabla \phi,$$

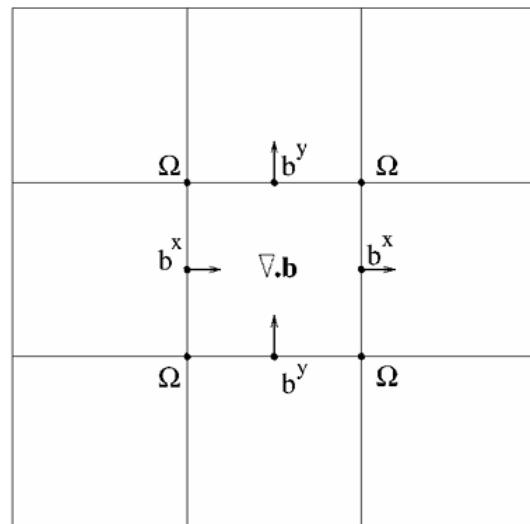
- Возьмем от него дивергенцию: $\nabla^2 \phi = \nabla \cdot \mathbf{B}^*$
- Получили уравнение Пуассона для поправки ϕ . Решив его, имеем:

$$\mathbf{B}^{n+1} = \mathbf{B}^* - \nabla \phi.$$

«+» работает почти всегда
«-» уравнение Пуассона
бывает дорого решать

Метод Constrained transport (CT) – МГД-моделирование с $\operatorname{div} \mathbf{B} = 0$

- Дискретный закон Фарадея для магнитного поля может быть записан на грани ячеек ($\Omega = E_z$) – «переселим» поле на грань



$$b_{j+1/2,k}^{x,n+1} = b_{j+1/2,k}^{x,n} - \Delta t \frac{\Omega_{j+1/2,k+1/2} - \Omega_{j+1/2,k-1/2}}{\Delta y}$$

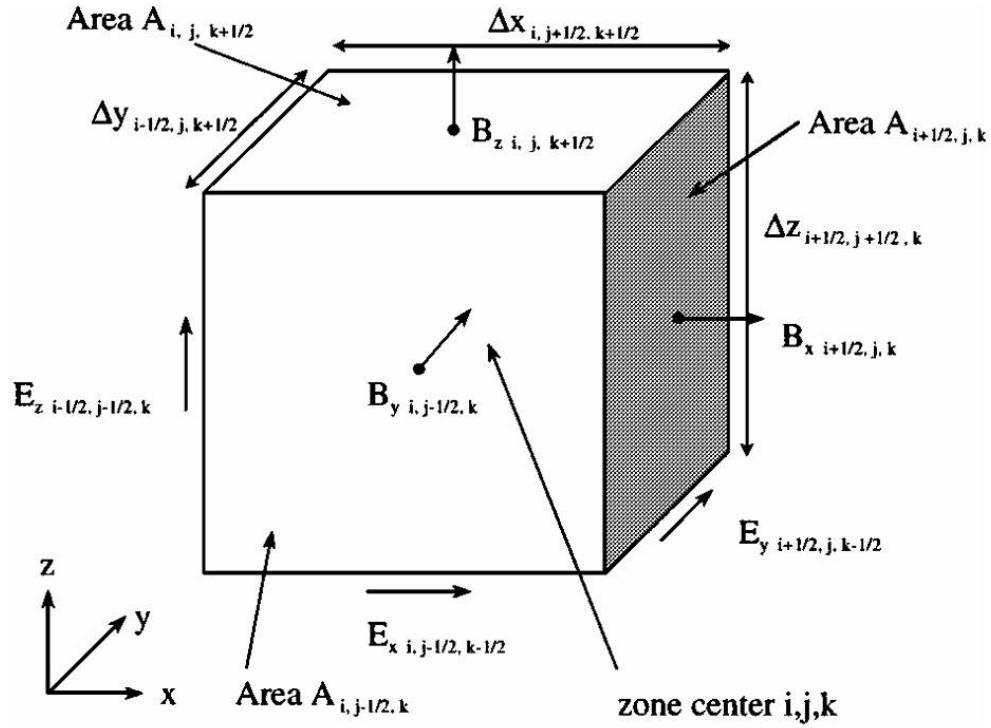
$$b_{j,k+1/2}^{y,n+1} = b_{j,k+1/2}^{y,n} + \Delta t \frac{\Omega_{j+1/2,k+1/2} - \Omega_{j-1/2,k+1/2}}{\Delta x}.$$

Можно показать, что в этом случае будет сохраняться дивергенция поля в следующем виде:

Но как нам найти электрическое поле в углах?

$$(\nabla \cdot \mathbf{b})_{j,k} = \frac{b_{j+1/2,k}^x - b_{j-1/2,k}^x}{\Delta x} + \frac{b_{j,k+1/2}^y - b_{j,k-1/2}^y}{\Delta y}$$

CT – 3D



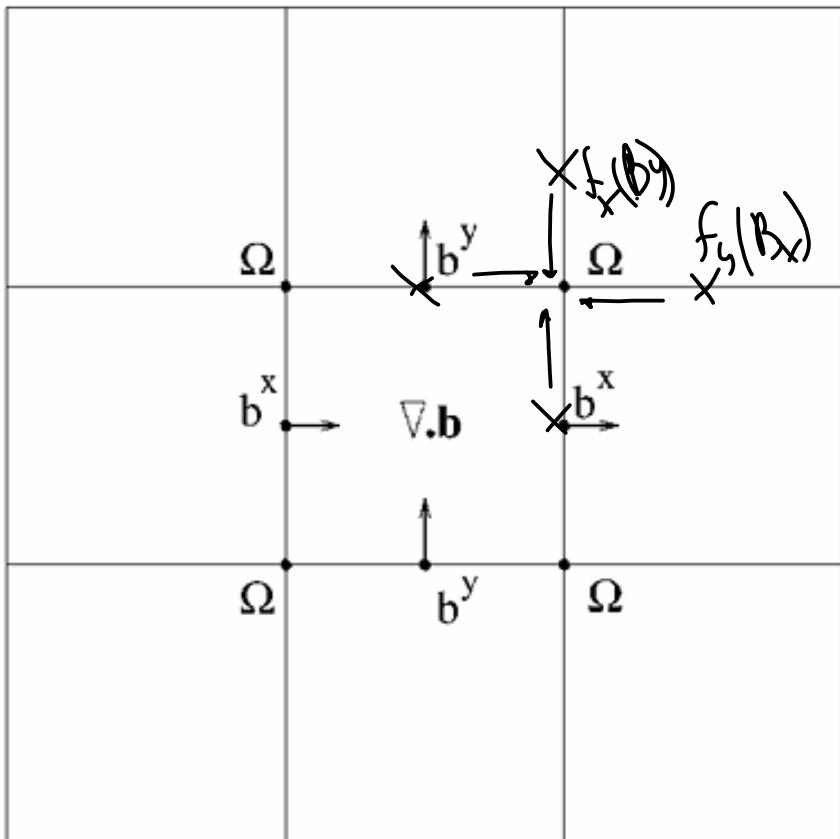
$$B_{x, i+1/2, j, k}^{n+1} = B_{x, i+1/2, j, k}^n - \frac{\Delta t}{A_{i+1/2, j, k}} \begin{pmatrix} \Delta z_{i+1/2, j+1/2, k} E_{z, i+1/2, j+1/2, k}^{n+1/2} \\ -\Delta z_{i+1/2, j-1/2, k} E_{z, i+1/2, j-1/2, k}^{n+1/2} \\ +\Delta y_{i+1/2, j, k-1/2} E_{y, i+1/2, j, k-1/2}^{n+1/2} \\ -\Delta y_{i+1/2, j, k+1/2} E_{y, i+1/2, j, k+1/2}^{n+1/2} \end{pmatrix}$$

$$B_{y, i, j-1/2, k}^{n+1} = B_{y, i, j-1/2, k}^n - \frac{\Delta t}{A_{i, j-1/2, k}} \begin{pmatrix} \Delta x_{i, j-1/2, k+1/2} E_{x, i, j-1/2, k+1/2}^{n+1/2} \\ -\Delta x_{i, j-1/2, k-1/2} E_{x, i, j-1/2, k-1/2}^{n+1/2} \\ +\Delta z_{i-1/2, j-1/2, k} E_{z, i-1/2, j-1/2, k}^{n+1/2} \\ -\Delta z_{i+1/2, j-1/2, k} E_{z, i+1/2, j-1/2, k}^{n+1/2} \end{pmatrix}$$

$$B_{z, i, j, k+1/2}^{n+1} = B_{z, i, j, k+1/2}^n - \frac{\Delta t}{A_{i, j, k+1/2}} \begin{pmatrix} \Delta x_{i, j-1/2, k+1/2} E_{x, i, j-1/2, k+1/2}^{n+1/2} \\ -\Delta x_{i, j+1/2, k+1/2} E_{x, i, j+1/2, k+1/2}^{n+1/2} \\ +\Delta y_{i+1/2, j, k+1/2} E_{y, i+1/2, j, k+1/2}^{n+1/2} \\ -\Delta y_{i-1/2, j, k+1/2} E_{y, i-1/2, j, k+1/2}^{n+1/2} \end{pmatrix}.$$

Метод Constrained transport

- Будем интерполировать электрическое поле в углы наших ячеек, зная, что потоки магнитного из МКО поля равны $E_z = \Omega$ на гранях



$$F_y(B_x) = v_y B_x - v_x B_y = E_z = \Omega$$

$$F_x(B_y) = -v_y B_x + v_x B_y = -E_z = -\Omega$$

$$\Omega_{j+1/2,k+1/2} = \frac{1}{4} (-f_{j+1/2,k}^{x,*} - f_{j+1/2,k+1}^{x,*} + f_{j,k+1/2}^{y,*} + f_{j+1,k+1/2}^{y,*})$$

В конце шага усредним поле в ячейку из граней

**Получили поле без
дивергенции!**
«+» – метод
максимально
физичный
«-» – требует
наличия
разнесенных сеток

$$B_{j,k}^{x,n+1} = \frac{b_{j+1/2,k}^{x,n+1} + b_{j-1/2,k}^{x,n+1}}{2}$$

$$B_{j,k}^{y,n+1} = \frac{b_{j,k+1/2}^{y,n+1} + b_{j,k-1/2}^{y,n+1}}{2}$$

Модификация МГД-систем – 8-wave

- Два наиболее употребляемых варианта – 8-вольновой метод (Powell (1999)) и метод обобщенного множителя Лагранжа (Dedner et al (2002))
- Восьмиволновой метод: рассмотрим МГД-систему, в которой **есть** дивергенция магнитного поля. Учтем эту силу во всех уравнениях.

Детали в файле
`one_step_mhd.py`

$$\frac{\partial(\rho\mathbf{v})}{\partial t} + \nabla \cdot \left[\rho\mathbf{v}\mathbf{v} + \left(p + \frac{\mathbf{B}^2}{8\pi} \right) \mathbf{I} - \frac{\mathbf{B}\mathbf{B}}{4\pi} \right] = -\mathbf{B}(\nabla \cdot \mathbf{B})$$

$$\frac{\partial E}{\partial t} + \nabla \cdot \left[\left(E + p + \frac{\mathbf{B}^2}{8\pi} \right) \mathbf{v} - \frac{(\mathbf{v} \cdot \mathbf{B})\mathbf{B}}{4\pi} \right] = -\mathbf{B} \cdot \mathbf{v}(\nabla \cdot \mathbf{B})$$

$$\frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{B} - \mathbf{B}\mathbf{v}) = -\mathbf{v}(\nabla \cdot \mathbf{B})$$

Дивергенция поля переносится
вместе с потоком (8-ая волна)

$$\frac{\partial(\nabla \cdot \mathbf{B})}{\partial t} + \nabla \cdot ((\nabla \cdot \mathbf{B})\mathbf{v}) = 0$$

«+» – очень
простой
«-» – портится
консервативность

Модификация МГД-систем - GLM

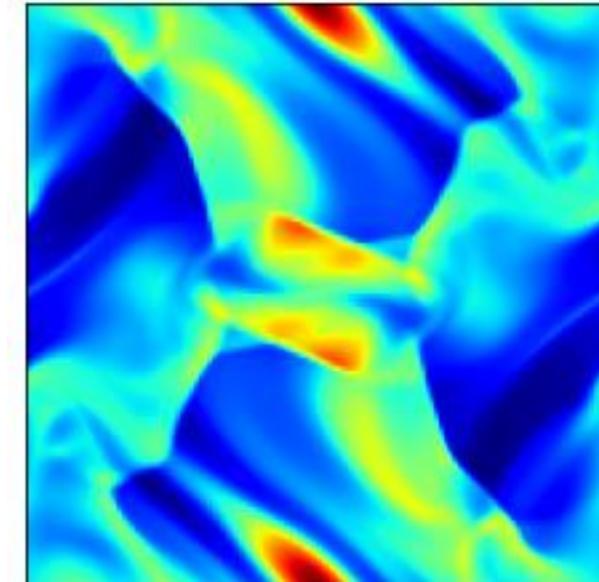
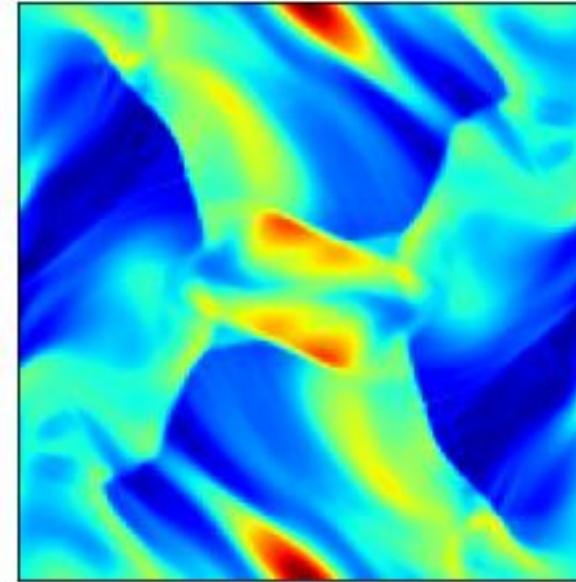
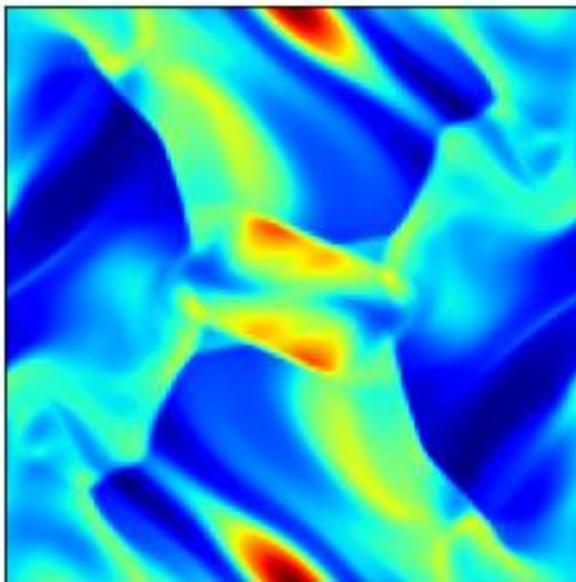
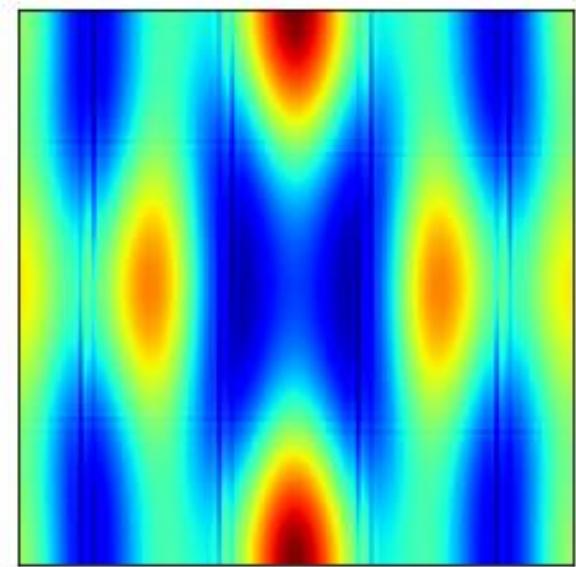
- В рамках подхода **GLM** решается дополнительное уравнение на множитель Лагранжа, который связан с переносом нормальной компоненты поля
- Дивергенция магнитного поля подчиняется телеграфному уравнению (перенос со скоростью c_h + затухание на времени $\sim \tau$):

$$\partial_t \mathbf{B} + \nabla \cdot (\mathbf{u} \mathbf{B}^T - \mathbf{B} \mathbf{u}^T) + \nabla \psi = 0, \quad \mathcal{D}(\psi) + \nabla \cdot \mathbf{B} = 0, \quad \xrightarrow{\hspace{1cm}} \frac{\partial \mathbf{B}}{\partial t} + \nabla \cdot (\mathbf{v} \mathbf{B} - \mathbf{B} \mathbf{v}) + \nabla \psi = 0,$$

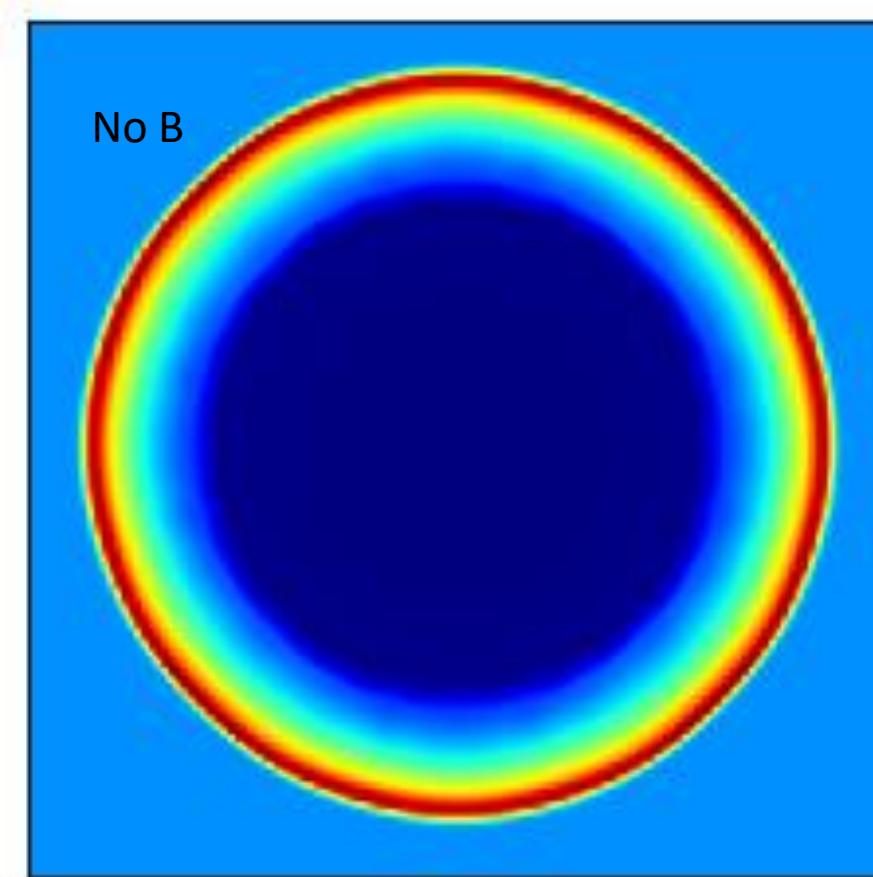
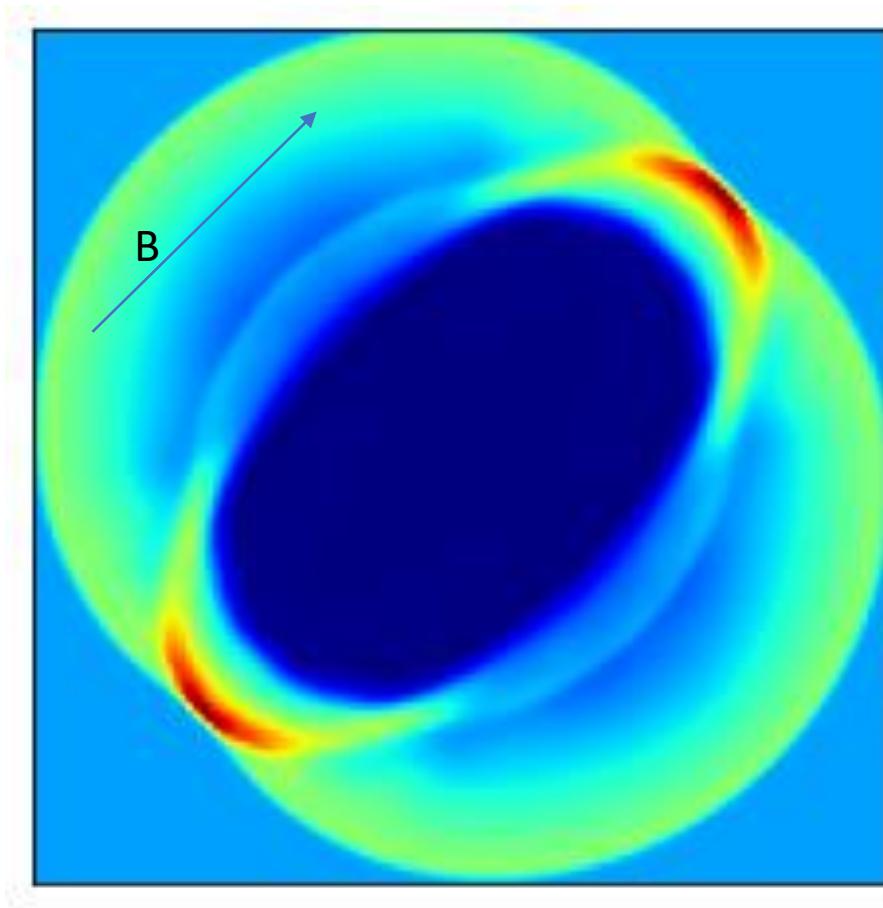
c_h - выбирается из условия
устойчивости; $\frac{1}{\tau} \sim 0.1 \frac{c_h}{\Delta x}$

$$\frac{\partial \psi}{\partial t} + c_h^2 \nabla \cdot \mathbf{B} = \frac{\psi}{\tau}.$$

Течение Орзага-Танга – переход к МГД-турбулентности как самый популярный МГД-тест



2D Взрыв в замагниченной среде vs в не замагнитенном газе



Пример: expl2d

Заключение/take home

- Большое количество моделей плазмы/газа/жидкости в астрофизических и геофизических условиях можно исследовать численно, применяя относительно простые и универсальные подходы, которые используют понятия конечного объема и обмена потоками сохраняющихся величин между ячейками
- Написать код для моделирования разных плазменных течений можно, используя «кирпичики» в виде функций потока, уточнения величин и т.д. (Это проще, чем кажется! 😊)
- Также можно использовать готовые пакеты, если это возможно, но необходимо понимать, как они работают!

Спасибо за внимание!

Проекты по численным методам
(если захочется больше практики)

Кондратьев И.А. – ИКИ РАН (mrkondratyev95@gmail.com)

Brief description

- **1-ая задача** – это моделирование МГД-течений в цилиндрической геометрии, **2-ая задача** – учет самогравитации газа в сферической геометрии, **3-ий проект** – разработка релятивистского газодинамического кода (СТО), а **4-ый** – это простой код для моделирования атмосферных/океанических течений на основе уравнений «мелкой воды»
- Для всех задач можно использовать готовую основу кода (хотя можно также написать свой), которую можно скачать по ссылке <https://github.com/mrkondratyev/Piastra2D>

Перед выполнением проекта рекомендуется сохранить копию кода!

Задачи попроще (на часок)

Для advection1D.ipynb

- Реализовать уравнение Бюргерса вместо линейной модели:

$$u_t + \left(\frac{u^2}{2} \right)_x = 0$$

- Добавить явную диссиpацию (также можно рассмотреть моделью Бюргерса + помнить про условие устойчивости $\Delta t = \min\left(\Delta t^{adv}, \frac{\Delta x^2}{2\nu}\right)$):

$$u_t + au_x = \nu u_{xx}$$

Для SWEsolver2D.ipynb

- Поэкспериментировать с формой дна и рассмотреть разные варианты задачи о распространении цунами в 1D
- Добавить силу Кориолиса и посмотреть на формирование волн Россби в 2D
- Детали в файле

1. МГД в цилиндрических координатах

- Что нужно сделать:

1. Найти источниковые члены S для уравнений МГД (вы можете выбрать (R,Z) или (R,ϕ) геометрию)
2. Создать новый файл сетки для цилиндрической геометрии, в котором необходимо записать выражения для площадей и объемов (`grid_setup`)
3. Учесть источниковые члены в правой части уравнений МГД внутри процедуры (`fluid_flux_calc`)
4. В качестве теста можно рассмотреть задачу о взрыве в замагниченной среде или распространение джета*
5. Для очистки дивергенции магнитного поля можно использовать 8-wave метод, либо написать схему СТ*

Уравнение сохранения массы

$$\frac{\partial \rho}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r \rho v_r) + \frac{1}{r} \frac{\partial}{\partial \phi} (\rho v_\phi) + \frac{\partial}{\partial z} (\rho v_z) = 0$$

Уравнения сохранения импульса

$$\frac{\partial}{\partial t} (\rho v_r) + \frac{1}{r} \frac{\partial}{\partial r} r (\rho v_r^2 + P_{tot} - B_r^2) + \frac{1}{r} \frac{\partial}{\partial \phi} (\rho v_r v_\phi - B_r B_\phi) + \frac{\partial}{\partial z} (\rho v_r v_z - B_r B_z) = S_R^v$$

$$\frac{\partial}{\partial t} (\rho v_\phi) + \frac{1}{r} \frac{\partial}{\partial r} r (\rho v_r v_\phi - B_r B_\phi) + \frac{1}{r} \frac{\partial}{\partial \phi} (\rho v_\phi^2 + P_{tot} - B_\phi^2) + \frac{\partial}{\partial z} (\rho v_\phi v_z - B_\phi B_z) = S_\phi^v$$

$$\frac{\partial}{\partial t} (\rho v_z) + \frac{1}{r} \frac{\partial}{\partial r} r (\rho v_r v_z - B_r B_z) + \frac{1}{r} \frac{\partial}{\partial \phi} (\rho v_\phi v_z - B_\phi B_z) + \frac{\partial}{\partial z} (\rho v_z^2 + P_{tot} - B_z^2) = S_z^v$$

Уравнение сохранения энергии

$$\frac{\partial E}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} r (v_r (E + P_{tot}) - B_r (\mathbf{B} \cdot \mathbf{v})) + \frac{1}{r} \frac{\partial}{\partial \phi} (v_\phi (E + P_{tot}) - B_\phi (\mathbf{B} \cdot \mathbf{v})) + \frac{\partial}{\partial z} (v_z (E + P_{tot}) - B_z (\mathbf{B} \cdot \mathbf{v})) = 0$$

Где полная энергия определяется как:

$$E = \frac{P}{\gamma - 1} + \frac{1}{2} \rho \mathbf{v}^2 + \frac{1}{2} \mathbf{B}^2$$

Уравнения индукции магнитного поля

$$\frac{\partial B_r}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (v_r B_r - B_r v_r) + \frac{1}{r} \frac{\partial}{\partial \phi} (v_\phi B_r - B_\phi v_r) + \frac{\partial}{\partial z} (v_z B_r - B_z v_r) = S_r^B$$

$$\frac{\partial B_\phi}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (r (v_r B_\phi - B_r v_\phi)) + \frac{1}{r} \frac{\partial}{\partial \phi} (v_\phi B_\phi - B_\phi v_\phi) + \frac{\partial}{\partial z} (v_z B_\phi - B_z v_\phi) = S_\phi^B$$

$$\frac{\partial B_z}{\partial t} + \frac{1}{r} \frac{\partial}{\partial r} (v_r B_z - B_r v_z) + \frac{1}{r} \frac{\partial}{\partial \phi} (v_\phi B_z - B_\phi v_z) + \frac{\partial}{\partial z} (v_z B_z - B_z v_z) = S_z^B$$

Цилиндрические координаты в 3D: $V = \frac{r_{i+1/2}^2 - r_{i-1/2}^2}{2} \Delta z \Delta \phi$,

$$S_{f,r_{i+1/2}} = r_{i+1/2} \Delta z \Delta \phi, S_{f,z} = \frac{r_{i+1/2}^2 - r_{i-1/2}^2}{2} \Delta \phi, S_{f,\phi} = \Delta r \Delta z$$

1. МГД в цилиндрических координатах: инструкция

1. Сетка

1. В файле `grid_setup.py` находится класс `Grid`
2. Сейчас там есть только декартовая сетка, которая задается функцией `uniCartGrid(self, x1ini, x1fin, x2ini,x2fin)`
3. Необходимо построить в этом классе функцию с заданием цилиндрической сетки по аналогии с `uniCartGrid` (например, `x1 == R, x2 == Z` или `x1 == R, x2 == Phi`)

2. Источниковые члены

1. В правых частях цилиндрических МГД-уравнений появляются источники, возникающие из-за перехода в криволинейную систему координат (например, члены вида $S_v1 = (P/R + ...)$), которые следуют из-за возникновения центробежной силы, а также ввиду аппроксимации градиента скалярной функции P при помощи дивергенции тензора $P \cdot \hat{I}$) – необходимо их вычислить
2. Далее в файле `one_step_mhd.py` нужно изменить (или создать новую) функцию с расчетом потоков `flux_calc_mhd` – нужно дополнить ее источником сразу после обновления приращений консервативных величин $\text{ResV1} = ((\text{Fmom1+}) * (\text{fS1+}) - (\text{Fmom1-}) * (\text{fS1-})) / \text{cVol}$ (будет что-то вроде $\text{ResV1} = \text{ResV1} - S_v1$)

3. Границы

1. Вблизи оси (граница 0) нужно добавить также граничное условие на ней так, чтобы v_ϕ и v_R равнялись нулю на оси. Это можно сделать, добавив в файле `boundaries.py` внутри функции `boundCond_mhd(grid, mhd)` еще один тип границы (например, задать `mhd.boundMark[0] == 103`, далее нужно поменять знаки перед скоростями так, как если бы это была твердая стенка)

4. Начальные условия и сравнение с «эталоном»

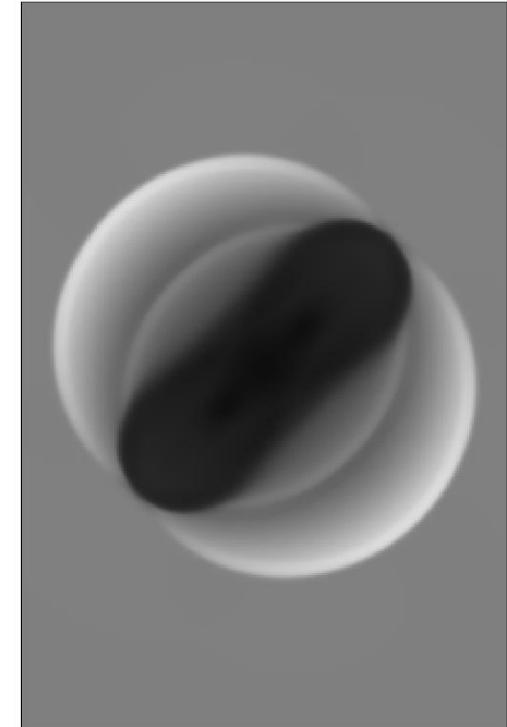
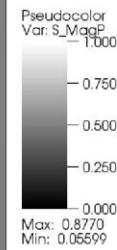
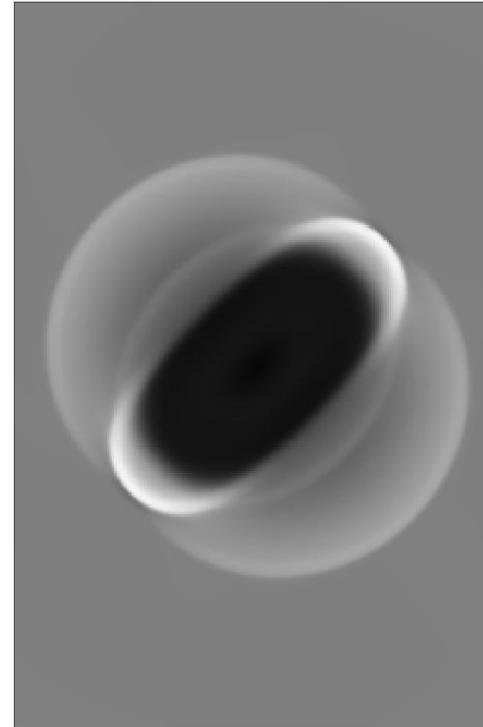
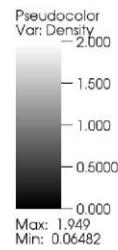
1. Наконец, в файле начальных условий `init_cond_mhd.py` надо задать начальные условия для вашей модели, там же маркируются границы (см. далее в презентации). Сетку для решателя можно построить в `main`-файле или при задании начальных условий
2. Для моделирования распространения джета* нужно поменять граничные условия на «втекание» вещества с одной из границ

5. Constrained transport* (опционально)

1. Нужно записать закон Фарадея в цилиндрической геометрии и переписать этот кусок кода в процедуре `flux_calc_MHD_CT` файла `one_step_mhd_CT.py`
2. Необходимо продемонстрировать, что код работает корректно, а дивергенция магнитного поля равна нулю

Начальные условия и «точное» решение для взрыва в замагниченной среде (1)

- <https://astro.uni-bonn.de/~jmackey/jmac/node12.html>
- 3D blast wave setup:
- $r < 0.1 \Rightarrow P = 10$, $r \geq 0.1 \Rightarrow P = 0.1$
- Dens = 1, Bz = 1 – everywhere



Для задачи о распространении джета см. [Торопина и др. Астрономический журнал (2020)]

2. Сферическая газовая динамика с самогравитацией

- Что нужно сделать:

1. Создать новый файл сетки для сферической геометрии, в котором необходимо записать выражения для площадей и объемов (вдоль направления x_2 можно удалить ненужные части кода) (**grid setup**)
2. Учесть источниковые члены в правой части уравнений газовой динамики внутри процедуры **fluid_flux_calc**
3. Запрограммировать решение для ускорения в гравитационном поле
4. В качестве теста можно рассмотреть задачу о коллапсе пылевого облака, для нее есть точное решение (чтобы найти $r(t)$, необходимо решить алгебраическое уравнение)

$$\left(\frac{8\pi G}{3}\rho_0\right)^{1/2} t = \left(\frac{r(t)}{r_0}\right)^{1/2} \left(1 - \frac{r(t)}{r_0}\right)^{1/2} + \sin^{-1} \left(1 - \frac{r(t)}{r_0}\right)^{1/2}$$

Уравнение сохранения массы

$$\frac{\partial \rho}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 \rho v_r) = 0$$

Уравнение сохранения импульса

$$\frac{\partial}{\partial t} (\rho v_r) + \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 (\rho v_r^2 + P)) = -\rho \frac{\partial \Phi}{\partial r}$$

Уравнение сохранения энергии

$$\frac{\partial E}{\partial t} + \frac{1}{r^2} \frac{\partial}{\partial r} (r^2 v_r (E + P)) = -\rho v_r \frac{\partial \Phi}{\partial r}$$

Где полная энергия:

$$E = \frac{P}{\gamma - 1} + \frac{1}{2} \rho v_r^2$$

Уравнение Пуассона для гравитационного потенциала

$$\frac{1}{r^2} \frac{\partial}{\partial r} \left(r^2 \frac{\partial \Phi}{\partial r}\right) = 4\pi G \rho$$

Решение для гравитационного ускорения

$$\Psi'_{i+\frac{1}{2}} = \frac{4\pi G}{r_{i+\frac{1}{2}}^2} \sum_{l=1}^i \rho_l \Delta \mathcal{V}_l .$$

$$\Delta \mathcal{V}_{r,i} = \frac{r_{i+\frac{1}{2}}^3 - r_{i-\frac{1}{2}}^3}{3}$$

<https://arxiv.org/pdf/1404.0537>

2. Сферическая газовая динамика с самогравитацией: инструкция

1. Сетка

1. В файле `grid_setup.py` находится класс `Grid`
2. Сейчас там есть только декартовая сетка, которая задается функцией `uniCartGrid(self, x1ini, x1fin, x2ini,x2fin)`
3. Необходимо построить в этом классе функцию с заданием сферической сетки по аналогии с `uniCartGrid` (например, `x1 == r, x2 == theta`). Можно не описывать члены с углом `theta`, заменив их нулями)
4. Сферический радиус в центре ячейки теперь будет равен $cx1(i) = 2(fx1(i+1)^3 - fx1(i)^3)/3(fx1(i+1)^2 - fx1(i)^2)$

2. Источниковые члены

1. В правых частях сферического уравнения на радиальный импульс возникнет источник $S_v1 = 2P/r$, возникающий из-за аппроксимации градиента скалярной функции P при помощи дивергенции тензора $P \cdot \hat{I}$. В файле `one_step_fluid.py` нужно изменить (или создать новую) функцию с расчетом потоков `flux_calc_fluid` – нужно дополнить ее источником сразу после обновления приращений консервативных величин $ResV1 = ((Fmom1+)*(fS1+) - (Fmom1-)*(fS1-)) / cVol$ (будет что-то вроде $ResV1 = ResV1 - S_v1$), потоки вдоль второй координаты можно удалить – мы их не используем для этого проекта

3. Сила тяжести

1. Необходимо написать простую функцию, которая по заданному распределению плотности рассчитывает гравитационное ускорение (в коде для есть массивы $F1$ и $F2$ для этого – их нужно рассчитать)
2. Эту функцию нужно вызывать в начале каждого шага по времени (внутри функции `oneStep_fluid_RK` файла `one_step_fluid.py`)

4. Начальные условия и сравнение с аналитикой

1. Наконец, в файле начальных условий `init_cond_fluid.py` надо задать начальные условия для вашей модели – рассматривается однородный коллапс облака с единичной плотностью. Область должна быть больше `r_max == 1`, давление во всей области и плотность за облаком нужно задать совсем небольшим – такими, чтобы они не влияли на коллапс. Сетку для решателя можно построить в `main`-файле или при задании начальных условий
2. Далее нужно провести расчет до `t < t_ff` и сравнить полученные распределения плотности с аналитическим решением (см. предыдущий слайд)

3. Релятивистская газовая динамика (1D или 2D в декартовой геометрии)*

- Что нужно сделать:

1. Выписать уравнения релятивистской газовой динамики в форме для метода конечного объема (см. [8] и [Ландау и Лифшиц VI])
2. Условие CFL и выражение для потока сохраняющихся величин можно взять с использованием скорости света ($c = 1$)
3. Необходимо также расписать процедуры **prim2cons** и **cons2prim**, которые задают отношения между примитивными (физическими) и консервативными переменными
4. **Cons2prim** – самая сложная часть, поскольку ввиду наличия Лоренц-фактора отношение между переменными становится нелинейным и требует использования метода Ньютона для решения
5. Для проверки правильности решим задачу о распаде разрыва и сравним с точным решением (также можно взять решение для взрывной волны Бленфорда-МакКи*)

$$\frac{\partial U}{\partial t} + \frac{\partial F^x(U)}{\partial x} + \frac{\partial F^y(U)}{\partial y} = 0, \quad (1)$$

where $U = (D, m_x, m_y, E)$ is the unknown vector of conservative variables, whereas F^x and F^y are, respectively, the fluxes along the x and y directions:

$$F^x(U) = \begin{pmatrix} Dv_x \\ m_x v_x + p \\ m_y v_x \\ m_x \end{pmatrix}, \quad F^y(U) = \begin{pmatrix} Dv_y \\ m_x v_y \\ m_y v_y + p \\ m_y \end{pmatrix}. \quad (2)$$

The relation between conserved variables U and physical quantities $V = (\rho, v_x, v_y, p)$ is

$$D = \gamma \rho, \quad m = Dh\gamma v, \quad E = Dh\gamma - p, \quad (3)$$
$$h = 1 + \frac{\gamma}{\gamma - 1} \frac{P}{\rho}, \quad \gamma = (1 - \mathbf{v} \cdot \mathbf{v})^{-\frac{1}{2}}$$

Equation (3) gives U in terms of the primitive state vector V . The inverse relation involves the solution of a nonlinear equation for the pressure p :

$$E + p = D\gamma + \frac{\Gamma}{\Gamma - 1} p\gamma^2 \quad (5)$$

where $\gamma = [1 - |\mathbf{v}|^2/(E + p)^2]^{-\frac{1}{2}}$. Equation (5) can be solved by any standard root finding algorithm.

3. Релятивистская газовая динамика (1D или 2D в декартовой геометрии): инструкция

1. Функция потоков консервативных величин

1. В файле `Riemann_fluxes.py` находятся разные аппроксимации решения задачи Римана. Нужно создать на основе `Riemann_flux_nr_fluid(rhol, rhor, vxl, vxr, vyl, vyx, vzl, vzl, pl, pr, eos, flux_type, dim)` функцию для релятивистских уравнений с таким же вводом и выводом
2. В ней нужно описать решение задачи Римана методом Лакса-Фридрихса, заменяя c везде, где потребуется, скоростью волн на скорость света $c = 1$ (мы используем максимально упрощенную структуру решения в этом случае)

2. Условие Куранта

1. В файле `CFL_condition.py` нужно создать функцию по аналогии с `CFLcondition_fluid(grid, fluid, eos, CFL)`, вместо скоростей волн в них также должны стоять единицы, т.к. мы используем упрощение, основанное на наличии максимальной скорости распространения волн в системе

3. Переход между физическими и консервативными переменными

1. Файл `prim_cons_fluid_MHD.py`
2. Функция `prim2cons_idealHydro(dens, vel1, vel2, vel3, pres, eos)` переводит физические переменные к консервативным. Ее нужно переписать под релятивистские уравнения (лучше просто сделать копию и изменить ее) – это достаточно просто
3. Функция `cons2prim_idealHydro(mass, mom1, mom2, mom3, etot, eos)` переводит консервативные переменные в физические, она сложнее, чем в обычной газовой динамике, т.к. все величины связаны через Лоренц-фактор. Для нее нужно в каждой ячейке решить уравнение на давление, и только потом, зная его, можно рассчитать плотность и скорости газа. Для этого необходимо отдельно написать функцию с решением уравнения на давление методом Ньютона

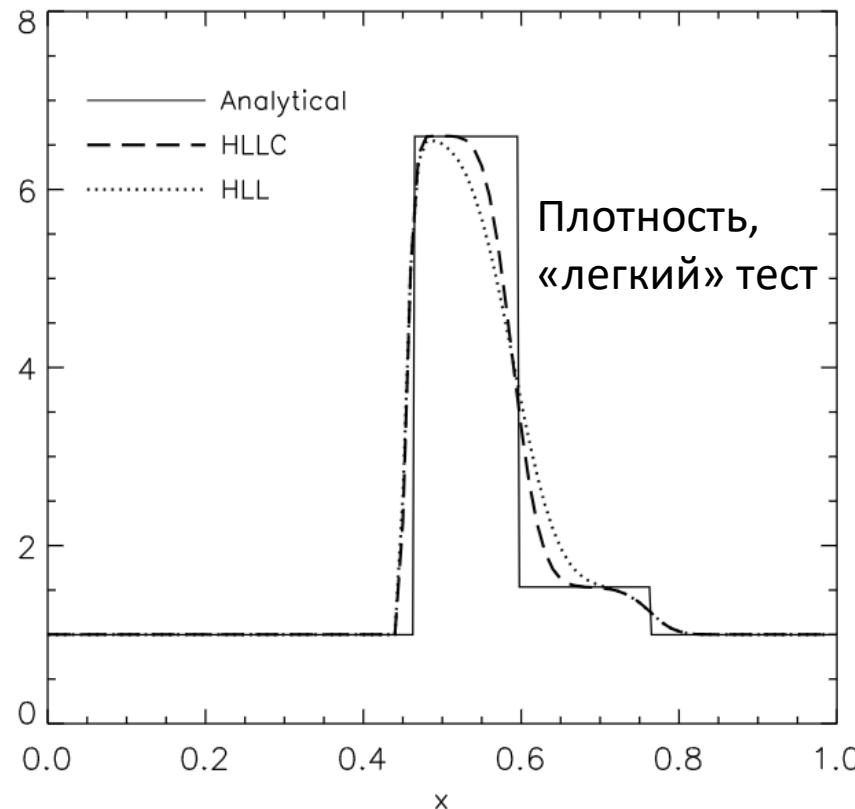
4. Итоговый файл с решателем

1. Нужно собрать все в файл решателя по аналогии с тем, как это сделано в `flux_calc_fluid` и `oneStep_fluid_RK` файла `one_step_fluid.py` для обычной гидродинамики
2. Наконец, нужно организовать вызов всех ключевых элементов расчета в `main`-файле

5. Начальные условия и сравнение с аналитикой

1. Наконец, в файле начальных условий `init_cond_fluid.py` надо задать начальные условия для вашей модели – рассматриваются различные виды релятивистской задачи о распаде разрыва (см. далее в презентации)

Релятивистская гидродинамика – тестовые задачи



The first test consists in a Riemann problem with initial data

$$(\rho, v_x, p) = \begin{cases} (1, 0.9, 1) & \text{for } x < 0.5, \\ (1, 0, 10) & \text{for } x > 0.5. \end{cases} \quad (32)$$

Integration is carried with $CFL = 0.8$ until $t = 0.4$ and an ideal equation of state with $\Gamma = 4/3$ is used. The breakup of the discontinuity results in the formation of two shock waves separated by a contact discontinuity.

Ref.: arXiv:1104.3562v3 – там же можно найти 2D задачи

Hard Relativistic Shock Tube	$x < .5$	$x > .5$
	ρ	1
v_x	0	0
$N = 100$.9	.9
$\Gamma = 5/3$	1000	.01
$t = 0.6$		

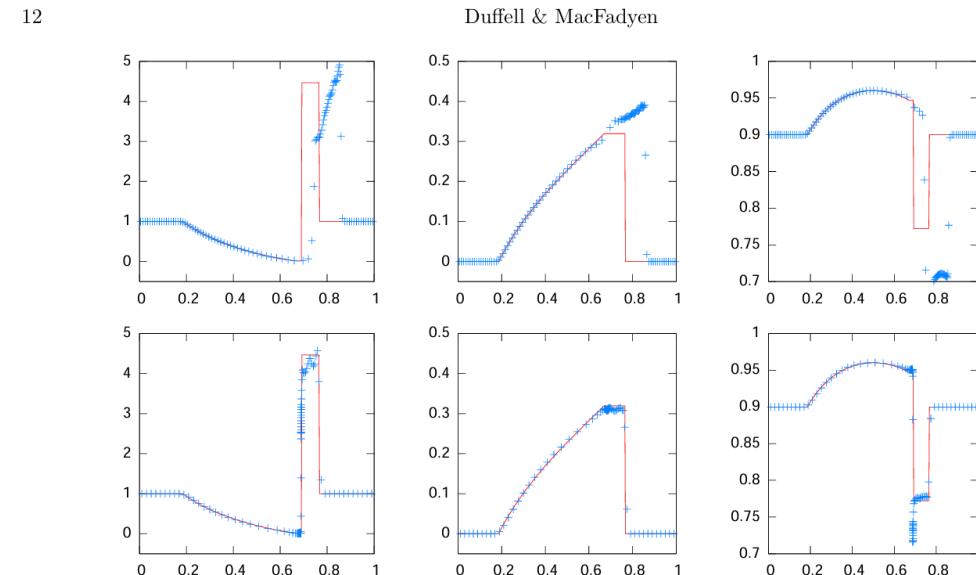
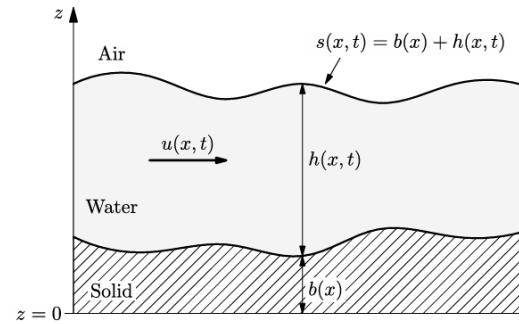


FIG. 11.— Hard relativistic shock tube at $t = 0.6$ (solid line is the exact solution). Both tests were run with a moving mesh using 100 zones, but the lower test concentrated 50 of the zones within a small region near the initial discontinuity, momentarily giving an effective resolution of roughly 10,000.

4. 2D модель уравнений мелкой воды (SWE)

- Что нужно сделать:

1. Написать решатель для двумерных уравнений мелкой воды с использованием HLL подхода
2. Характерные скорости волн $\frac{\omega}{k} = v \pm \sqrt{gh}$
3. Для валидации кода можно решить задачу о прорыве плотины и т.н. «модель ванны» (обсудим на занятии)
4. SWE очень важны для практических приложений. С переменной глубиной дна можно моделировать распространение цунами в океане*, дополнительно можно учесть силу Кориолиса* и исследовать волны Россби в атмосфере



$$\begin{aligned} u_x + v_y + w_z &= 0, \\ u_t + uu_x + vu_y + wu_z &= -\frac{1}{\rho} p_x, \\ v_t + uv_x + vv_y + wv_z &= -\frac{1}{\rho} p_y, \\ w_t + uw_x + vw_y + ww_z &= -\frac{1}{\rho} p_z - g. \end{aligned}$$



$$\partial_t \mathbf{Q} + \partial_x \mathbf{F}(\mathbf{Q}) + \partial_y \mathbf{G}(\mathbf{Q}) = \mathbf{S}(\mathbf{Q}),$$

$$\left. \begin{aligned} \mathbf{Q} &= \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, & \mathbf{F}(\mathbf{Q}) &= \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \\ huv \end{bmatrix}, \\ \mathbf{G}(\mathbf{Q}) &= \begin{bmatrix} hv \\ hvu \\ hv^2 + \frac{1}{2}gh^2 \end{bmatrix}, & \mathbf{S}(\mathbf{Q}) &= \begin{bmatrix} s_1 \\ s_2 \\ s_3 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh\partial_x b \\ -gh\partial_y b \end{bmatrix}. \end{aligned} \right\}$$

4. 2D модель уравнений мелкой воды (SWE): ИНСТРУКЦИЯ

1. Файл с переменными среды

1. Здесь у нас всего три независимых переменных вместо пяти для гидродинамики. Можно либо создать отдельный класс на основе файла `fluid_state.py`, либо использовать класс из гидродинамики, помня, что теперь у нас поменялся физический смысл наших переменных (`dens` – высота, `vel1, vel2` – усредненные по высоте скорости жидкости)
2. Дополнительно надо добавить величину (просто константу), описывающую гравитационное ускорение `g`

2. Функция потоков консервативных величин

1. В файле `Riemann_fluxes.py` находятся разные аппроксимации решения задачи Римана. Нужно создать на основе функции `Riemann_flux_nr_fluid(rhol, rhor, vxl, vxr, vyl, vyr, vzl, vzs, pl, pr, eos, flux_type, dim)` функцию для уравнений мелкой воды. Здесь можно предусмотреть, что переменных всего три, либо же аккуратно занулить все ненужные массивы, чтобы использовать идентичный ввод/вывод

3. Условие Куранта

1. В файле `CFL_condition.py` нужно создать функцию по аналогии с `CFLcondition_fluid(grid, fluid, eos, CFL)`, вместо скорости звука в них должна стоять величина \sqrt{gh}

4. Итоговый файл с решателем

1. Нужно собрать всё в файл решателя по аналогии с тем, как это сделано в `flux_calc_fluid` и `oneStep_fluid_RK` файла `one_step_fluid.py` для обычной гидродинамики
2. В некоторых файлах нужно будет убрать лишнее, т.к. модель стала проще, чем газовая динамика (например, в `prim2cons_idealHydro` и `cons2prim_idealHydro` файла `prim_cons_fluid_MHD.py` нужно удалить часть процедур с вычислением давления)
3. Наконец, нужно организовать вызов всех ключевых элементов расчета в `main`-файле

5. Начальные условия и сравнение с аналитикой

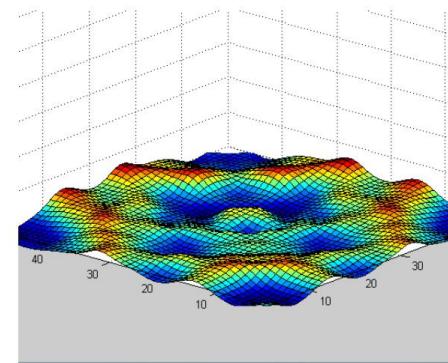
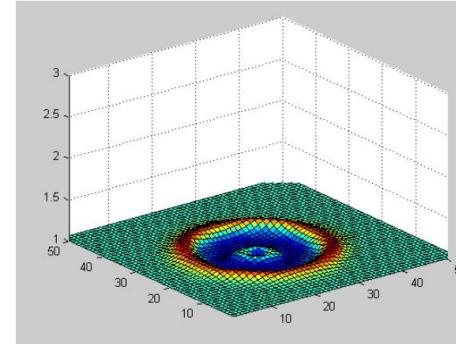
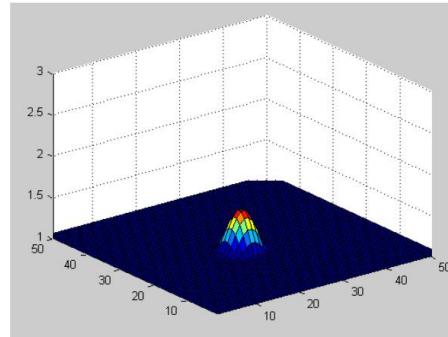
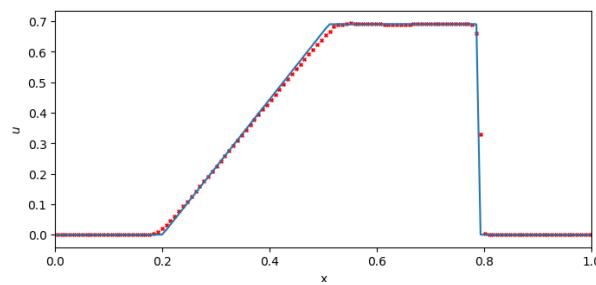
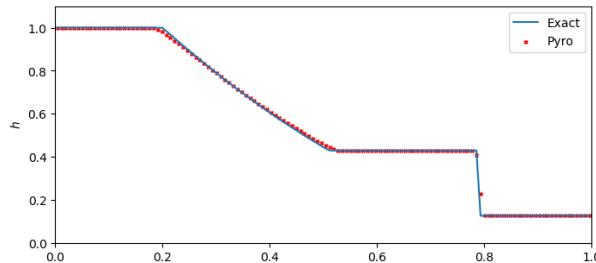
1. Наконец, в файле начальных условий `init_cond_***.py` надо задать начальные условия для вашей модели – сначала рассматривается аналог теста Сода – прорыв плотины (см. далее в презентации)
2. Также можно промоделировать течение воды в ванной – возникнет турбулентная картина, которую вы наверняка могли наблюдать в жизни (см. далее в презентации)

SWE – Dam break problem (прорыв плотины, слева) и bathtub simulation (течение воды в «ванне», справа) (4)

dam

parameters:

name	default
dam.direction	x
dam.h_left	1.0
dam.h_right	0.125
dam.u_left	0.0
dam.u_right	0.0



$$0 < x, y < 50$$

$$v_1 = v_2 = 0$$

$$g = 9.8$$

$$h = h_0 + h_1 \cdot \max(0, \frac{\sin(r)}{r})$$

$$h_0 = 1, h_1 = 1, r = \sqrt{(x - x_0)^2 + (y - y_0)^2}$$

$$x_0 = 25, y_0 = 25$$

Ref.: Pyro code by Michael Zingale
https://python-hydro.github.io/pyro2/swe_basics.html

Ref:https://homes.civil.aau.dk/jen/VisualizationExercise3_old/ShallowWaterEquations.pdf