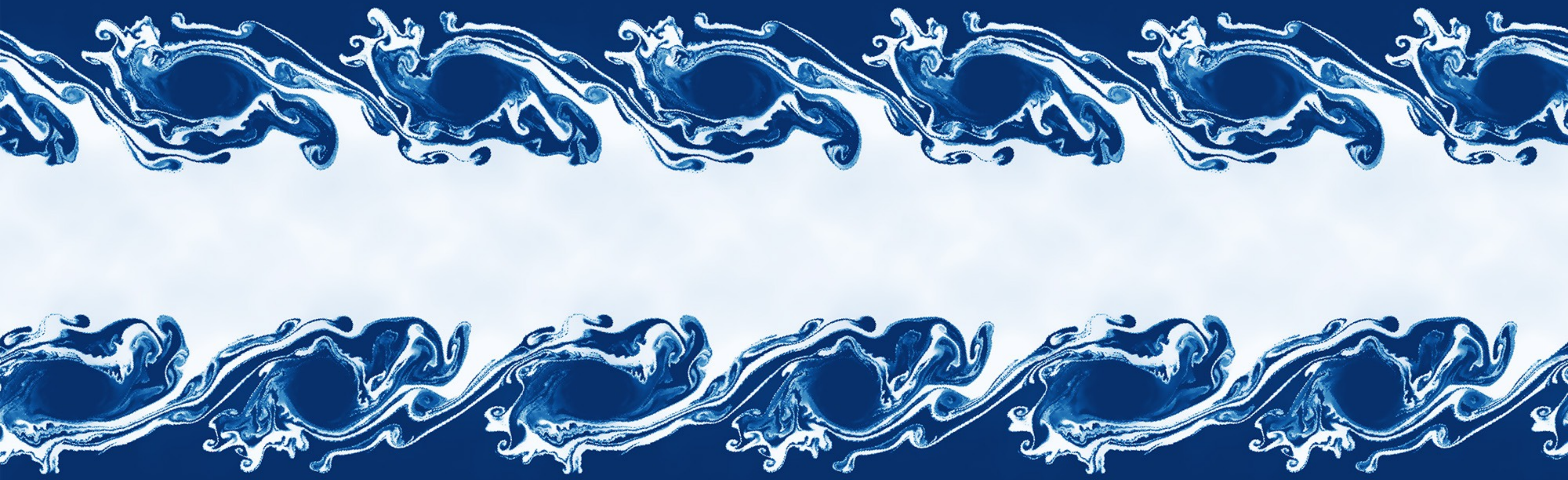
ASTR 670: Interstellar medium and gas dynamics

Prof. Benedikt Diemer



Chapter 8 • Computational hydro III: Finite-volume schemes

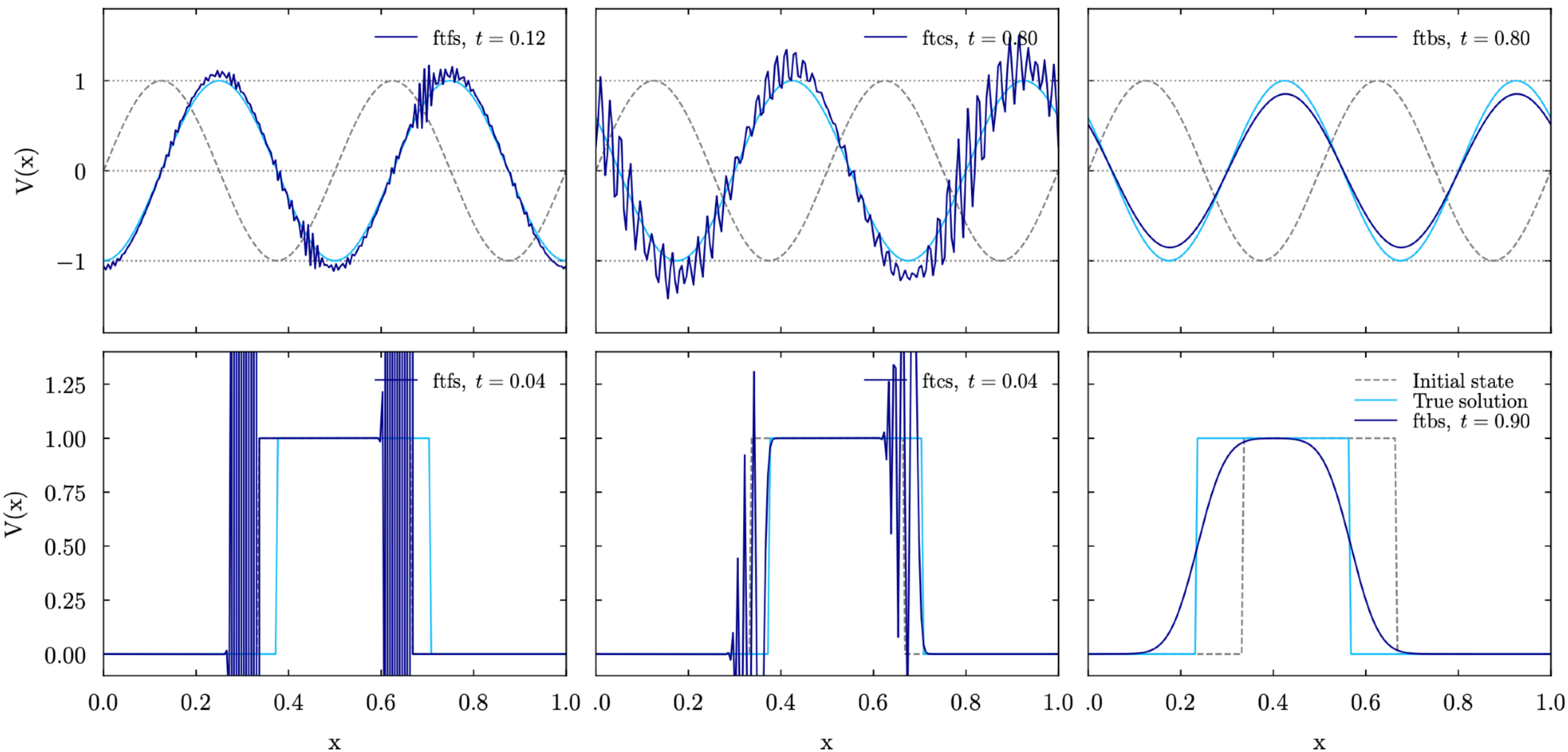
Recap: Finite-difference schemes

Finite-differencing tests

FTFS

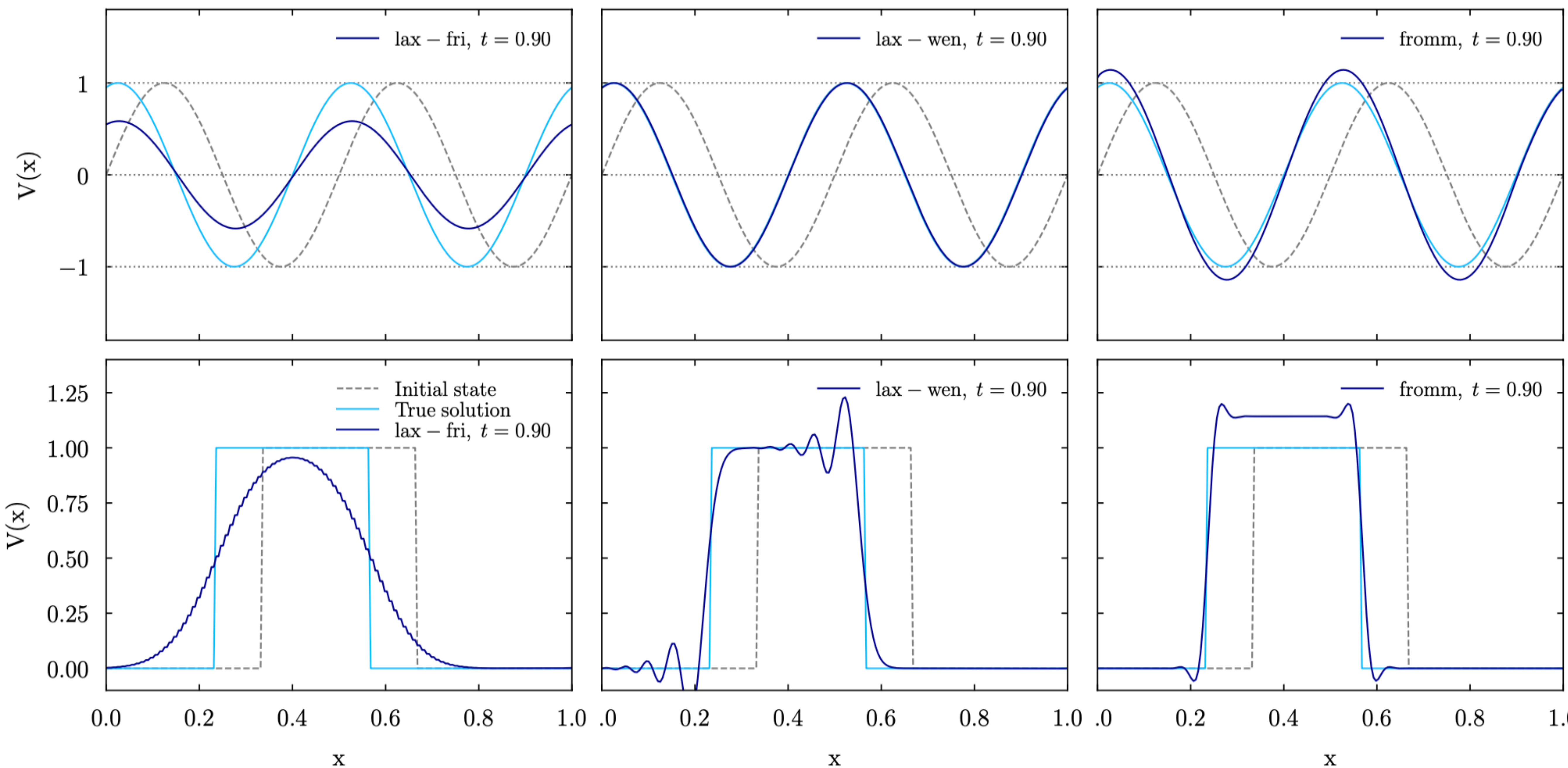
FTCS

FTBS



Finite-differencing tests

Lax-Friedrichs | Lax-Wendroff | Fromm

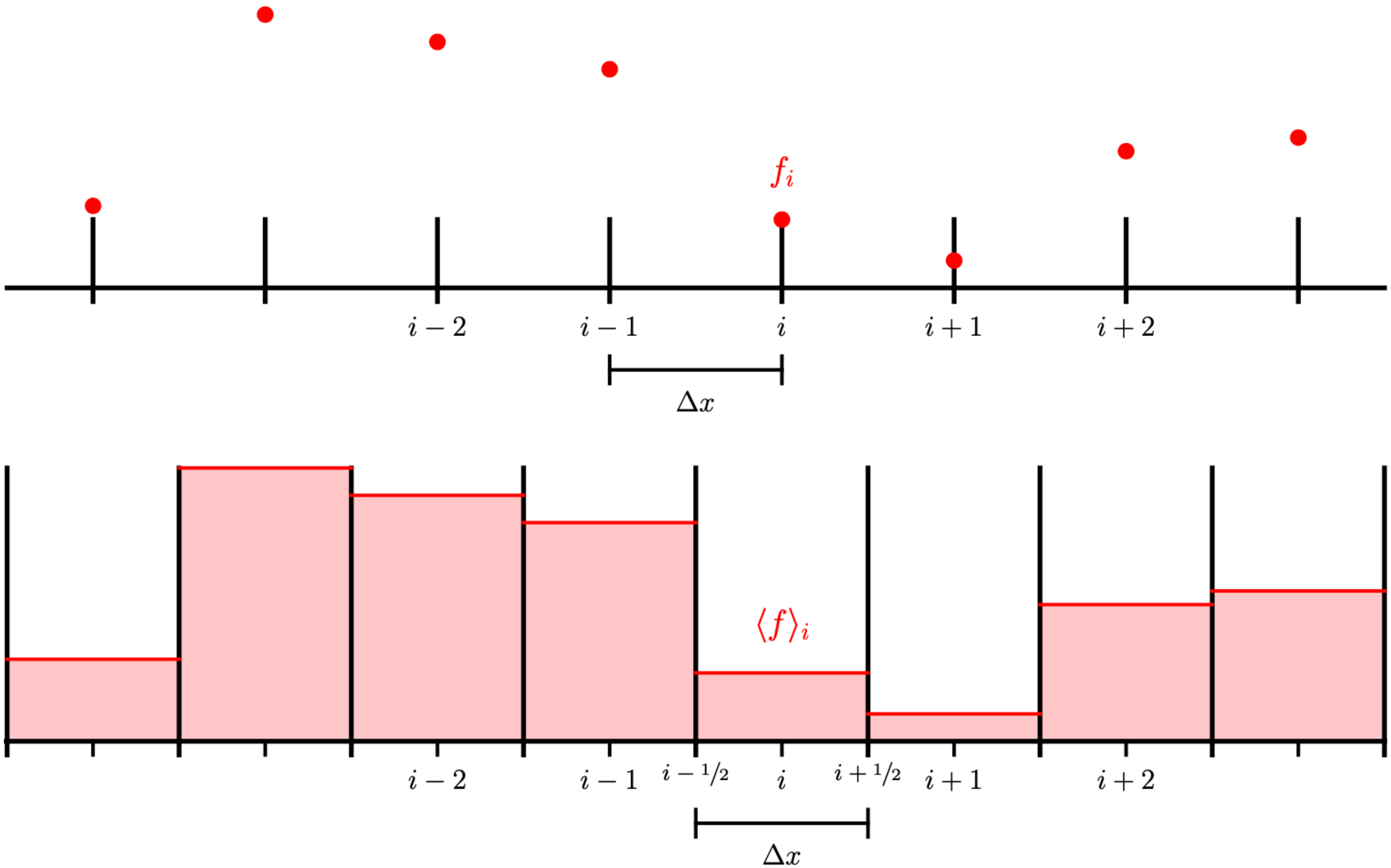


Issues with finite-difference schemes

- Unstable (sometimes)
- Diffusing sharp features (stability vs diffusivity)
- Do not conserve mass/momentum/energy
  - Strictly positive quantities can go negative
- Cannot handle shocks

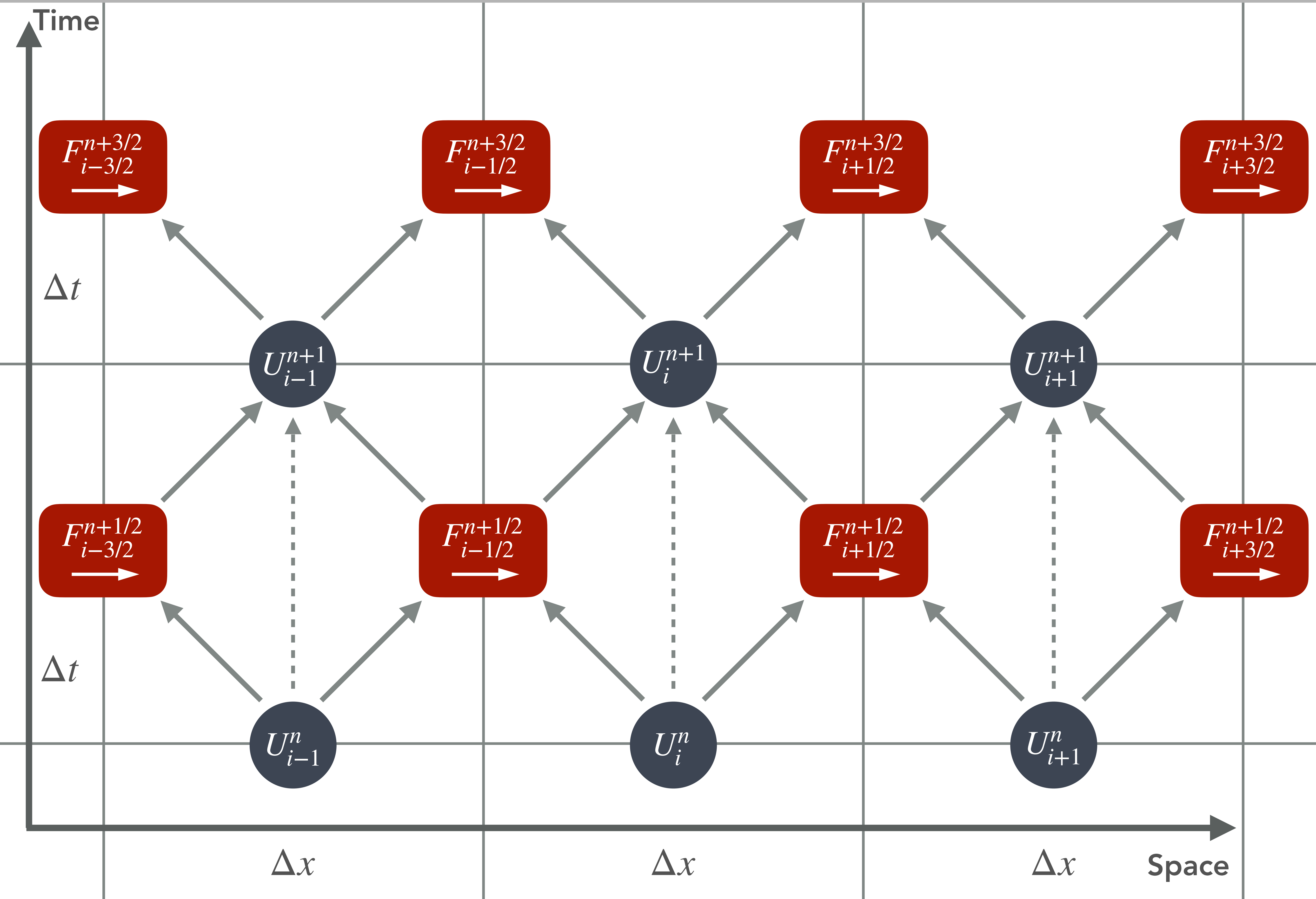
Finite-volume (Godunov) schemes

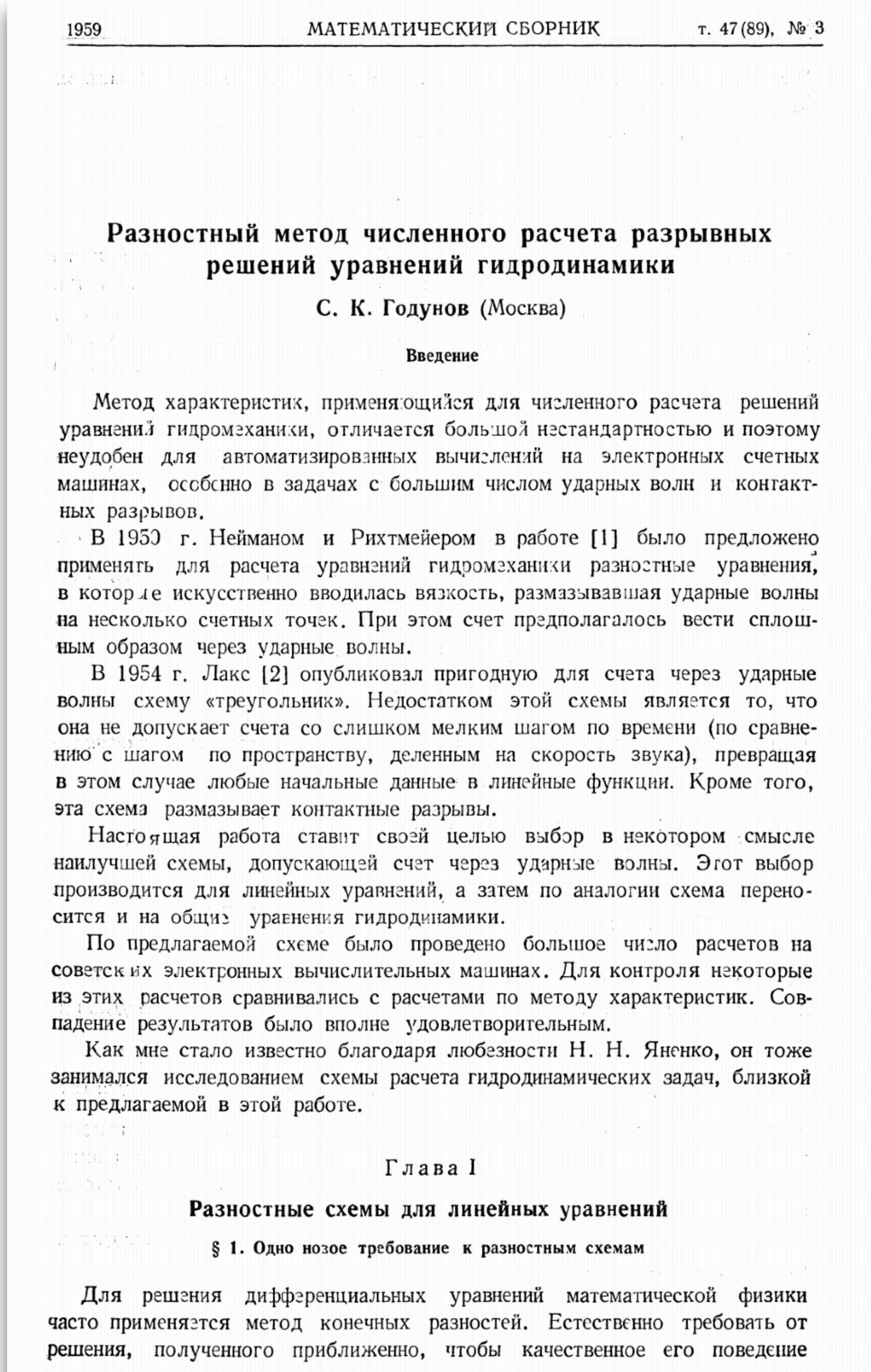
Finite difference vs. finite volume



Zingale 2021

Godunov scheme




1959 МАТЕМАТИЧЕСКИЙ СБОРНИК т. 47 (89), № 3

Разностный метод численного расчета разрывных решений уравнений гидродинамики

С. К. Годунов (Москва)

Введение

Метод характеристик, применяющийся для численного расчета решений уравнений гидромеханики, отличается большой нестандартностью и поэтому неудобен для автоматизированных вычислений на электронных счетных машинах, особенно в задачах с большим числом ударных волн и контактных разрывов.

В 1950 г. Нейманом и Рихтмейером в работе [1] было предложено применять для расчета уравнений гидромеханики разностные уравнения, в которые искусственно вводилась вязкость, размазывавшая ударные волны на несколько счетных точек. При этом счет предполагалось вести сплошным образом через ударные волны.

В 1954 г. Лакс [2] опубликовал пригодную для счета через ударные волны схему «треугольник». Недостатком этой схемы является то, что она не допускает счета со слишком мелким шагом по времени (по сравнению с шагом по пространству, деленным на скорость звука), превращая в этом случае любые начальные данные в линейные функции. Кроме того, эта схема размазывает контактные разрывы.

Настоящая работа ставит своей целью выбор в некотором смысле наилучшей схемы, допускающей счет через ударные волны. Этот выбор производится для линейных уравнений, а затем по аналогии схема переносится и на общие уравнения гидродинамики.

По предлагаемой схеме было проведено большое число расчетов на советских электронных вычислительных машинах. Для контроля некоторые из этих расчетов сравнивались с расчетами по методу характеристик. Совпадение результатов было вполне удовлетворительным.

Как мне стало известно благодаря любезности Н. Н. Яненко, он тоже занимался исследованием схемы расчета гидродинамических задач, близкой к предлагаемой в этой работе.

Глава I

Разностные схемы для линейных уравнений

§ 1. Одно новое требование к разностным схемам

Для решения дифференциальных уравнений математической физики часто применяется метод конечных разностей. Естественно требовать от решения, полученного приближенно, чтобы качественное его поведение


Sergei Godunov

Godunov 1959

$$\frac{\partial u}{\partial t} + B\,\frac{\partial p\,(v,\ E)}{\partial x} = 0,$$

$$\frac{\partial v}{\partial t} - B\,\frac{\partial u}{\partial x} = 0,$$

$$\frac{\partial\left(E + \frac{u^2}{2}\right)}{\partial t} + B\,\frac{\partial pu}{\partial x} = 0.$$

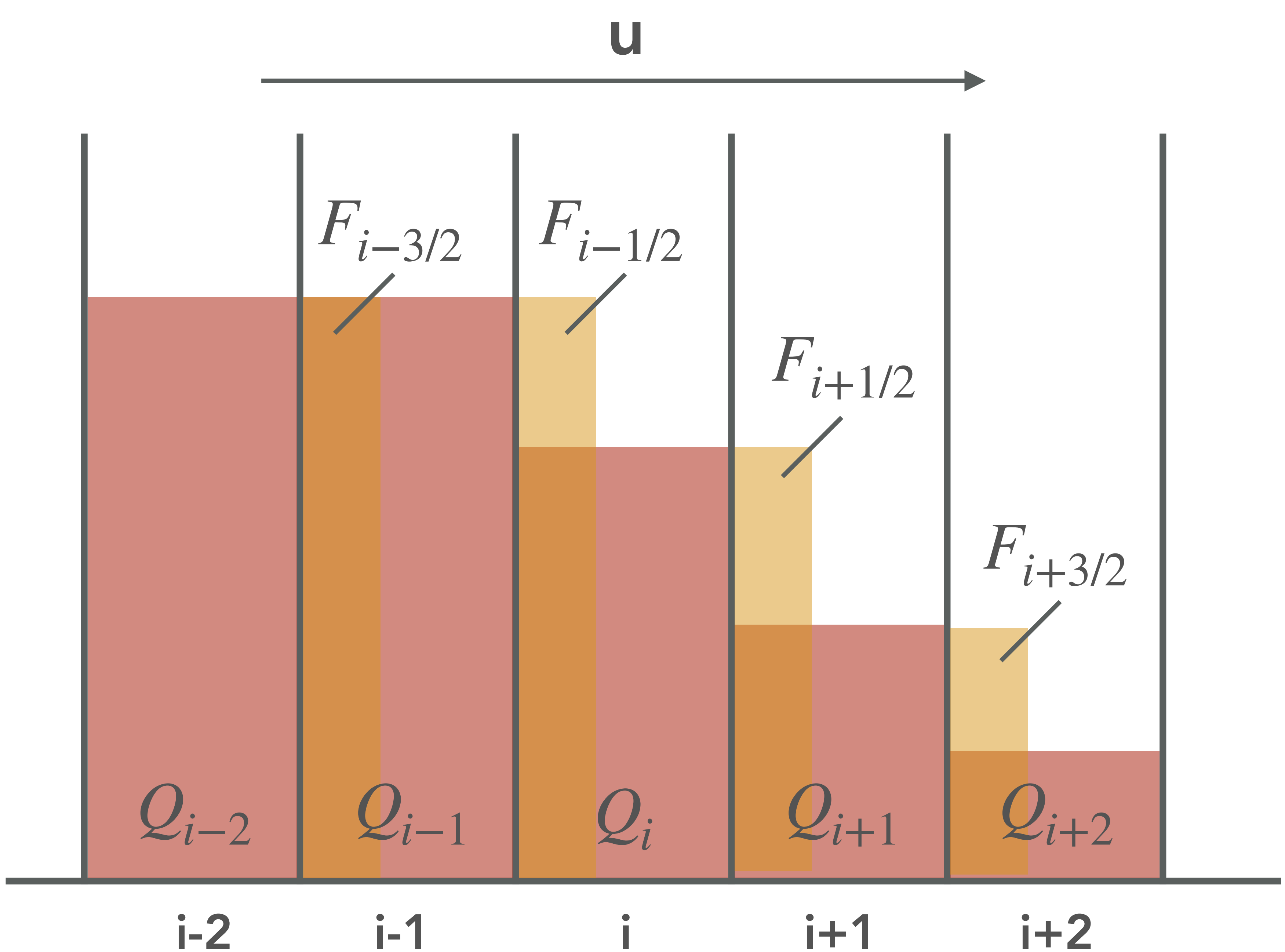
$$u^{m+\frac{1}{2}} = u_{m+\frac{1}{2}} - \frac{\tau B}{h}(P_{m+1} - P_m),$$

$$v^{m+\frac{1}{2}} = v_{m+\frac{1}{2}} + \frac{\tau B}{h}(U_{m+1} - U_m),$$

$$E^{m+\frac{1}{2}} = E_{m+\frac{1}{2}} + \frac{\left(u_{m+\frac{1}{2}}\right)^2}{2} - \frac{\left(u^{m+\frac{1}{2}}\right)^2}{2} - \frac{\tau B}{h}(P_{m+1}U_{m+1} - P_mU_m).$$

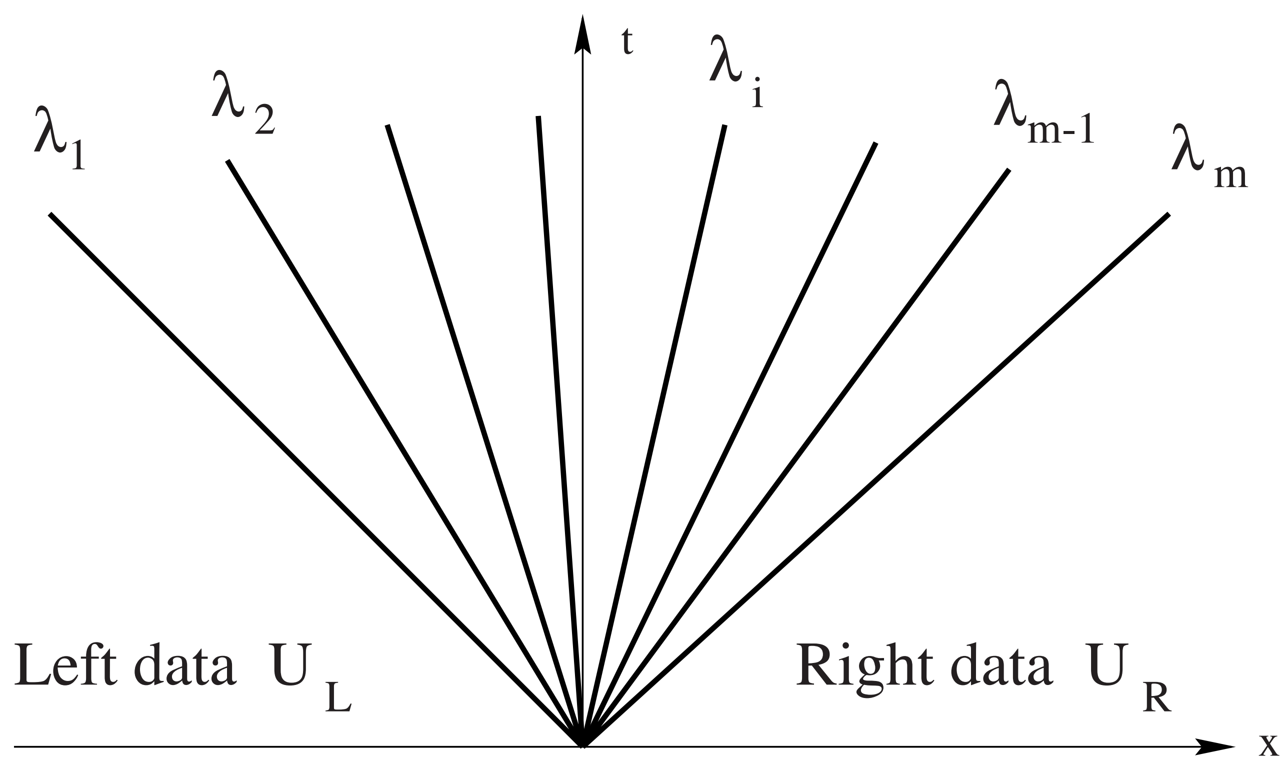
Godunov 1959

Godunov scheme for advection



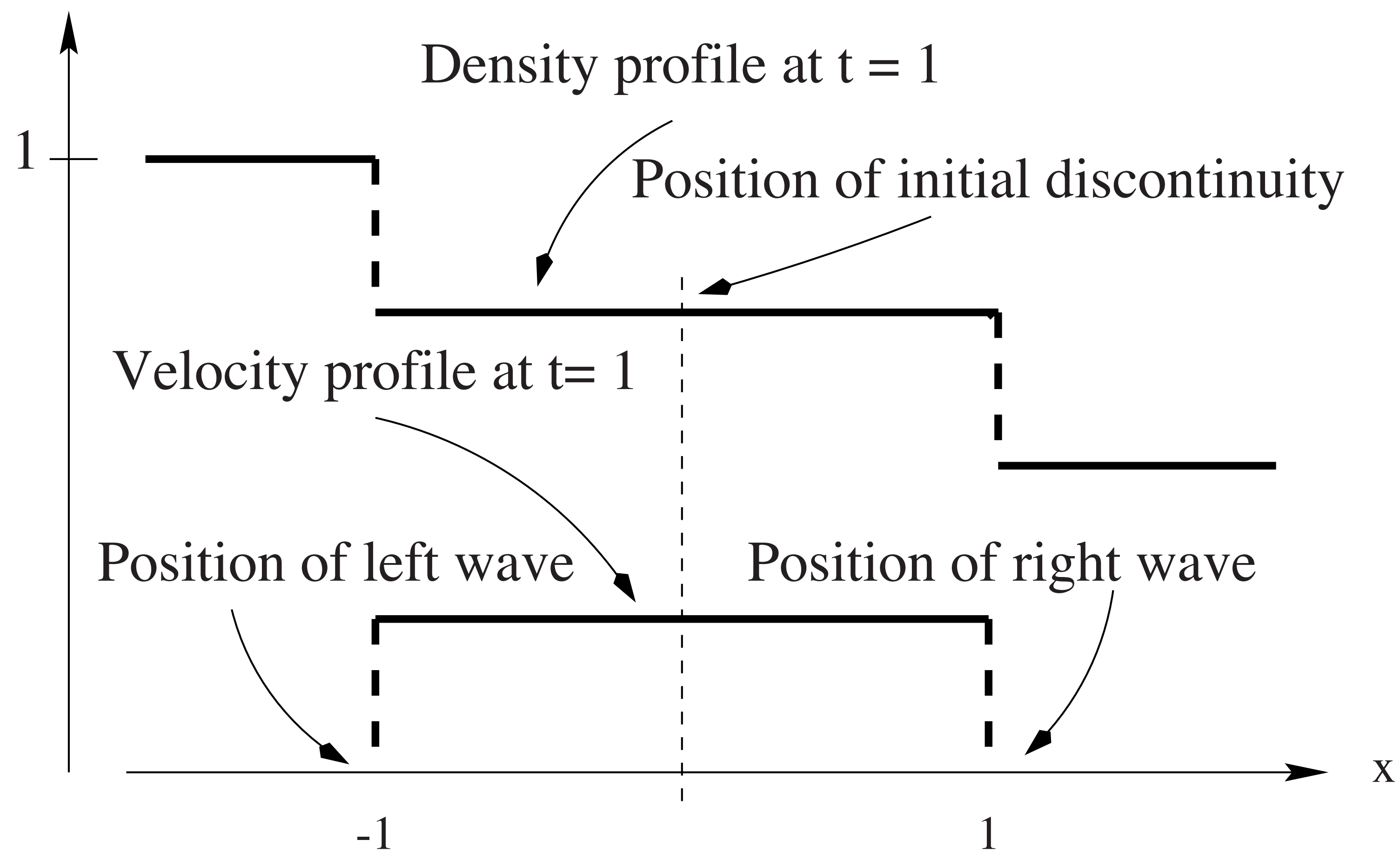
The Riemann problem

The Riemann problem



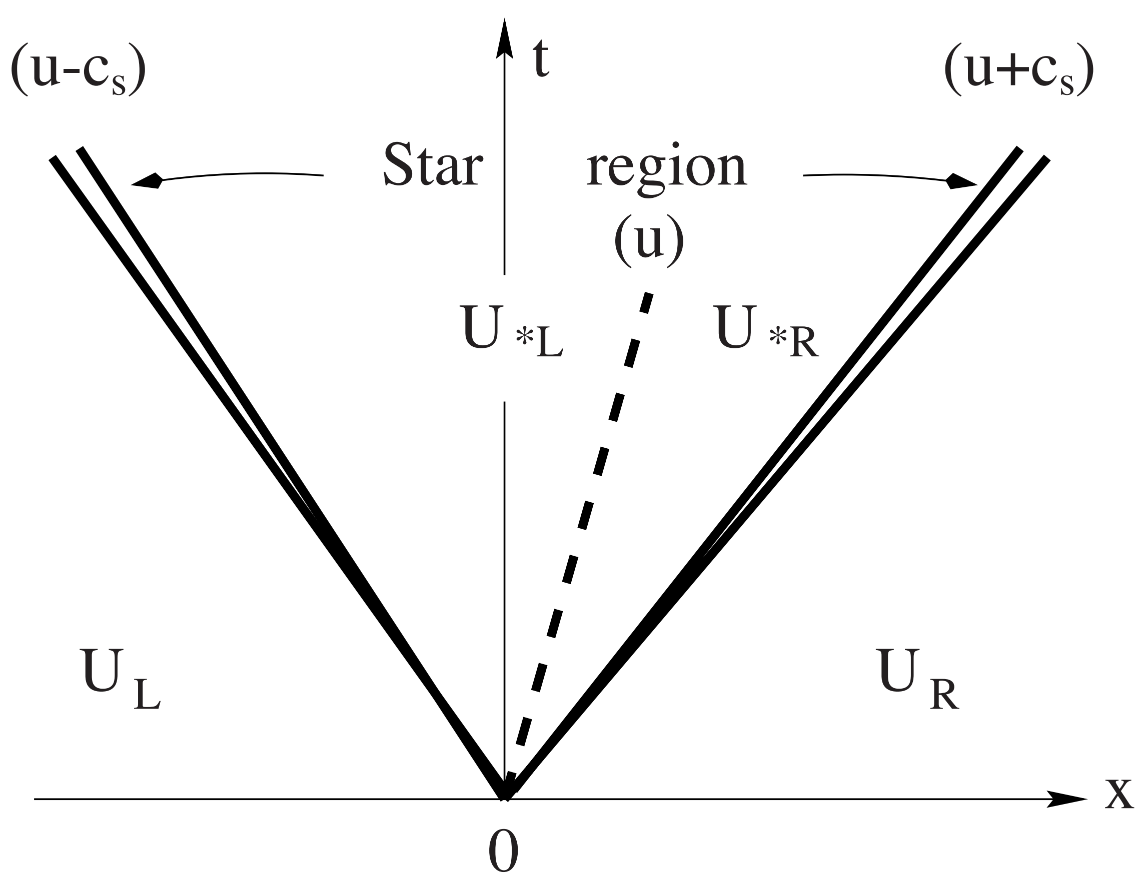
Toro 2009

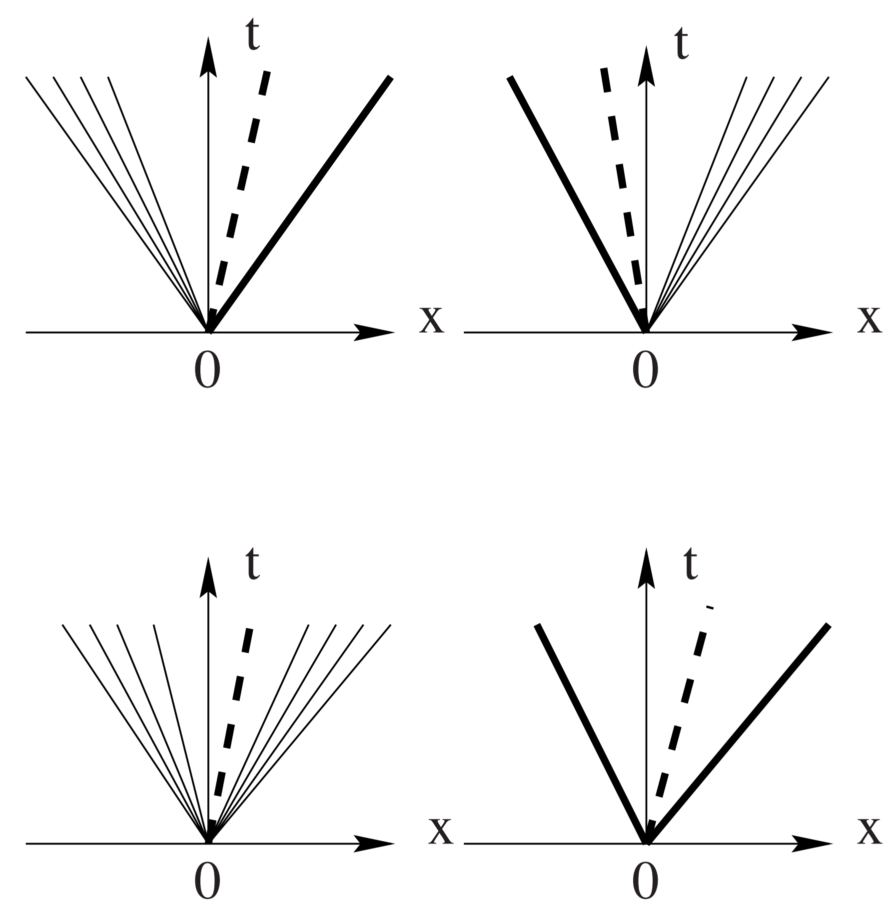
The Riemann problem (linearized Euler)



Toro 2009

The Riemann problem (for the 1D Euler equations)





Toro 2009

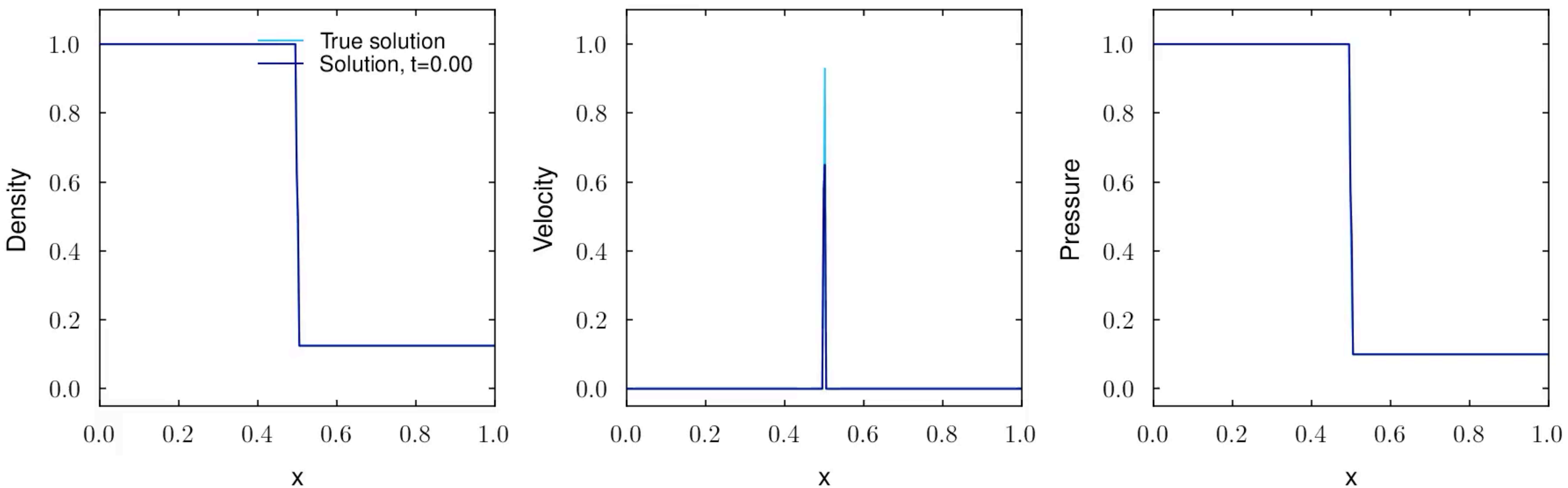
HLL Riemann solver

$$\boldsymbol{\mathcal{F}}_{i\pm1/2}^{\mathrm{HLL}} = \begin{cases} \boldsymbol{\mathcal{F}}_{\mathrm{L}} & \forall \quad S_{\mathrm{L}} \geq 0 \\ \frac{S_{\mathrm{R}}\boldsymbol{\mathcal{F}}_{\mathrm{L}} - S_{\mathrm{L}}\boldsymbol{\mathcal{F}}_{\mathrm{R}} + S_{\mathrm{L}}S_{\mathrm{R}}(\boldsymbol{U}_{\mathrm{R}} - \boldsymbol{U}_{\mathrm{L}})}{S_{\mathrm{R}} - S_{\mathrm{L}}} & \text{otherwise} \\ \boldsymbol{\mathcal{F}}_{\mathrm{R}} & \forall \quad S_{\mathrm{R}} \leq 0 \end{cases} \tag{9.11}$$

We have adopted a common notation of L and R subscripts, which indicate the cells to the left and right of an interface (cells i and $i+1$ for the $i+1/2$ interface, for example). Similarly, $\boldsymbol{\mathcal{F}}_{\mathrm{L}}$ is the cell-centered flux in the left cell, and so on. Moreover, S_{L} and S_{R} are the velocities of the fastest possible waves going left and right, which we approximate as $S_{\mathrm{L}} = u_{\mathrm{L}} - c_{\mathrm{s}}$ and $S_{\mathrm{R}} = u_{\mathrm{R}} + c_{\mathrm{s}}$. If $S_{\mathrm{L}} > 0$, all waves are traveling to the right and we can use the flux corresponding to the left state (as in the advection equation with positive velocity). The opposite case is that $S_{\mathrm{R}} < 0$, meaning that all waves travel to the left. In the majority of cases, we will end up in the intermediate "star region," where use the averaged HLL flux (Equation 9.11). Importantly, we have ignored the

Harten, Lax & van Leer

Sod shocktube



Made with ULULA

The Riemann problem (1D Euler)

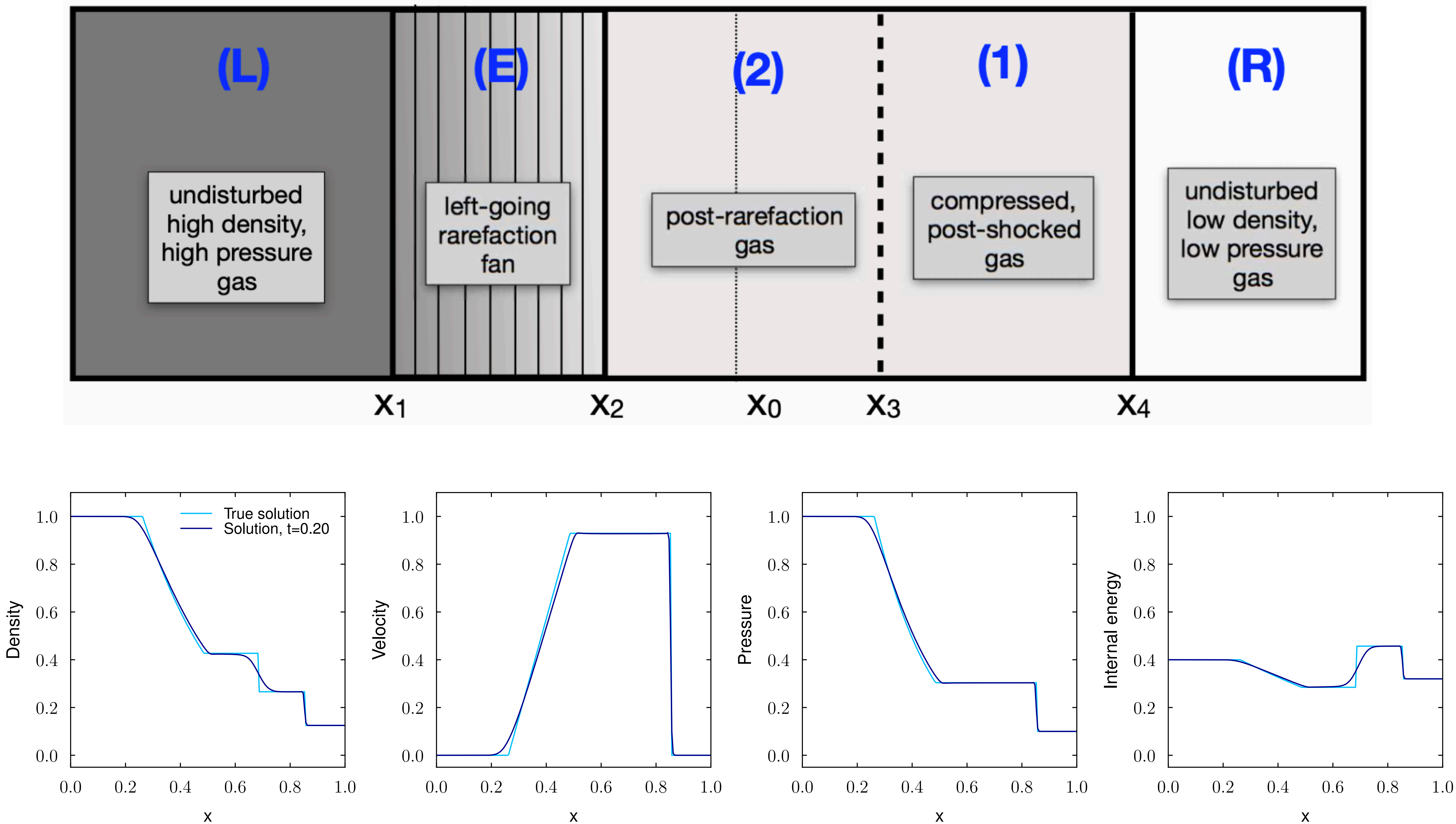
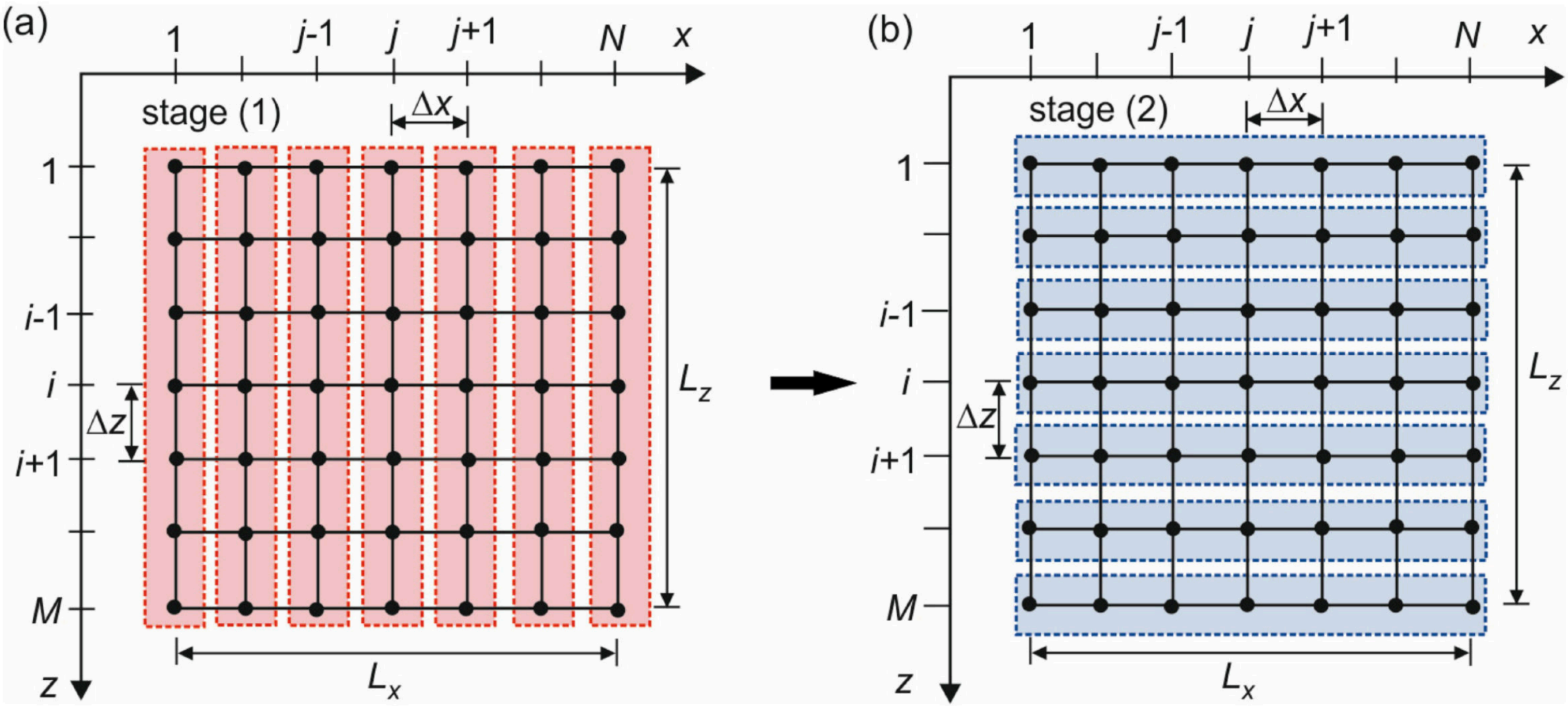


Figure by Frank van den Bosch

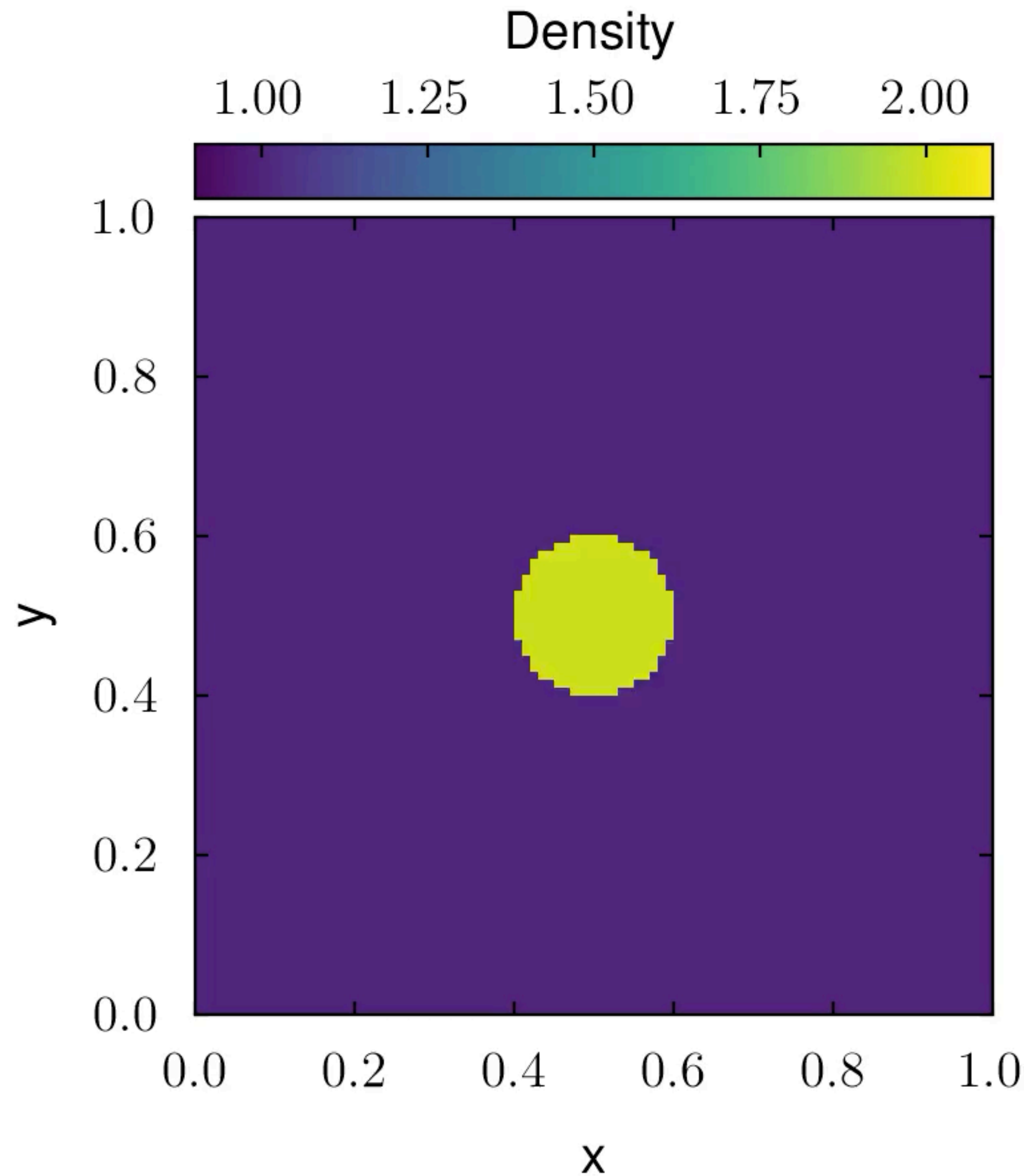
Higher-order Godunov schemes

Dimensional splitting



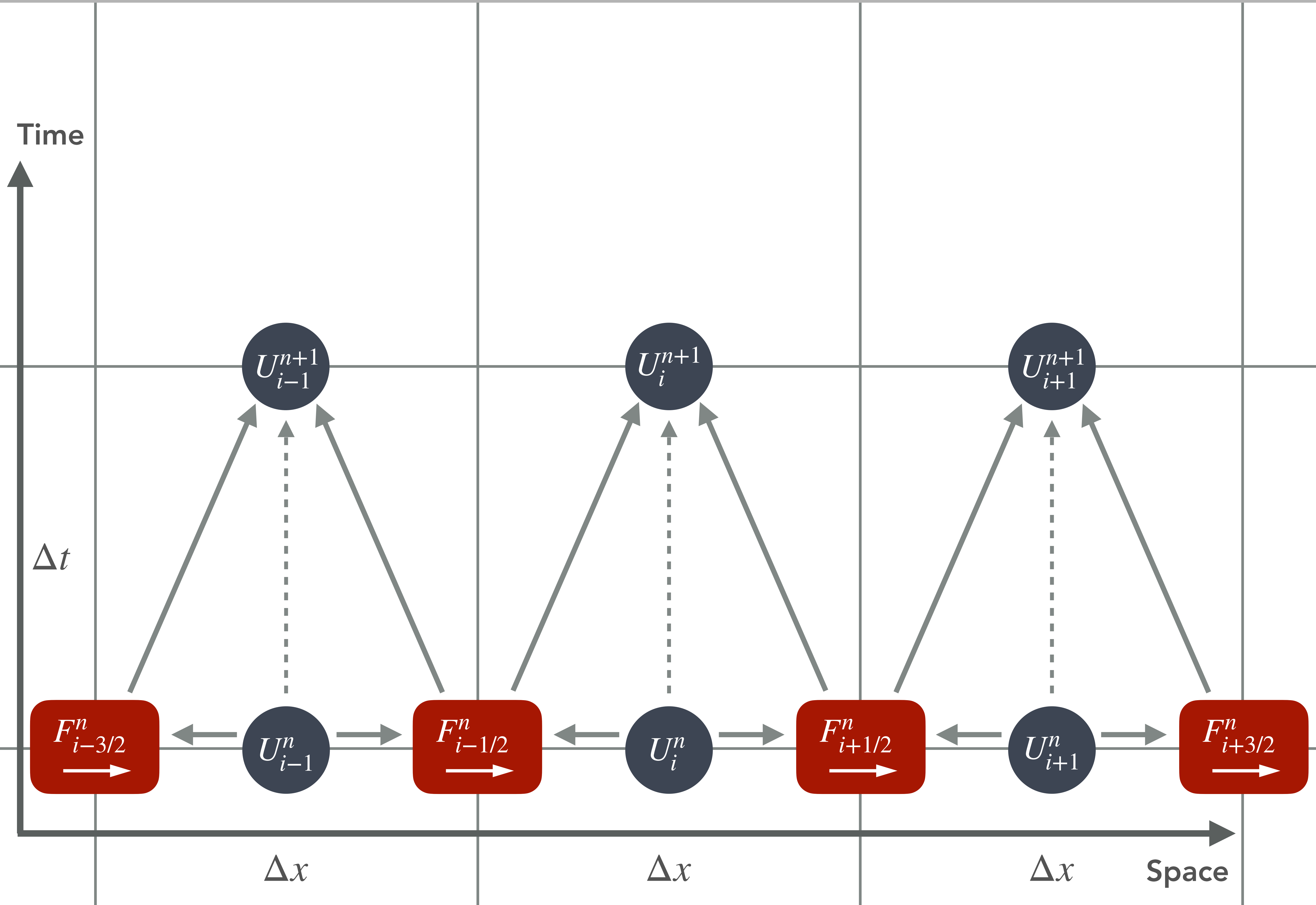
Gasiorowski & Kolerski 2020 • Strang 1968

Advection in 2D

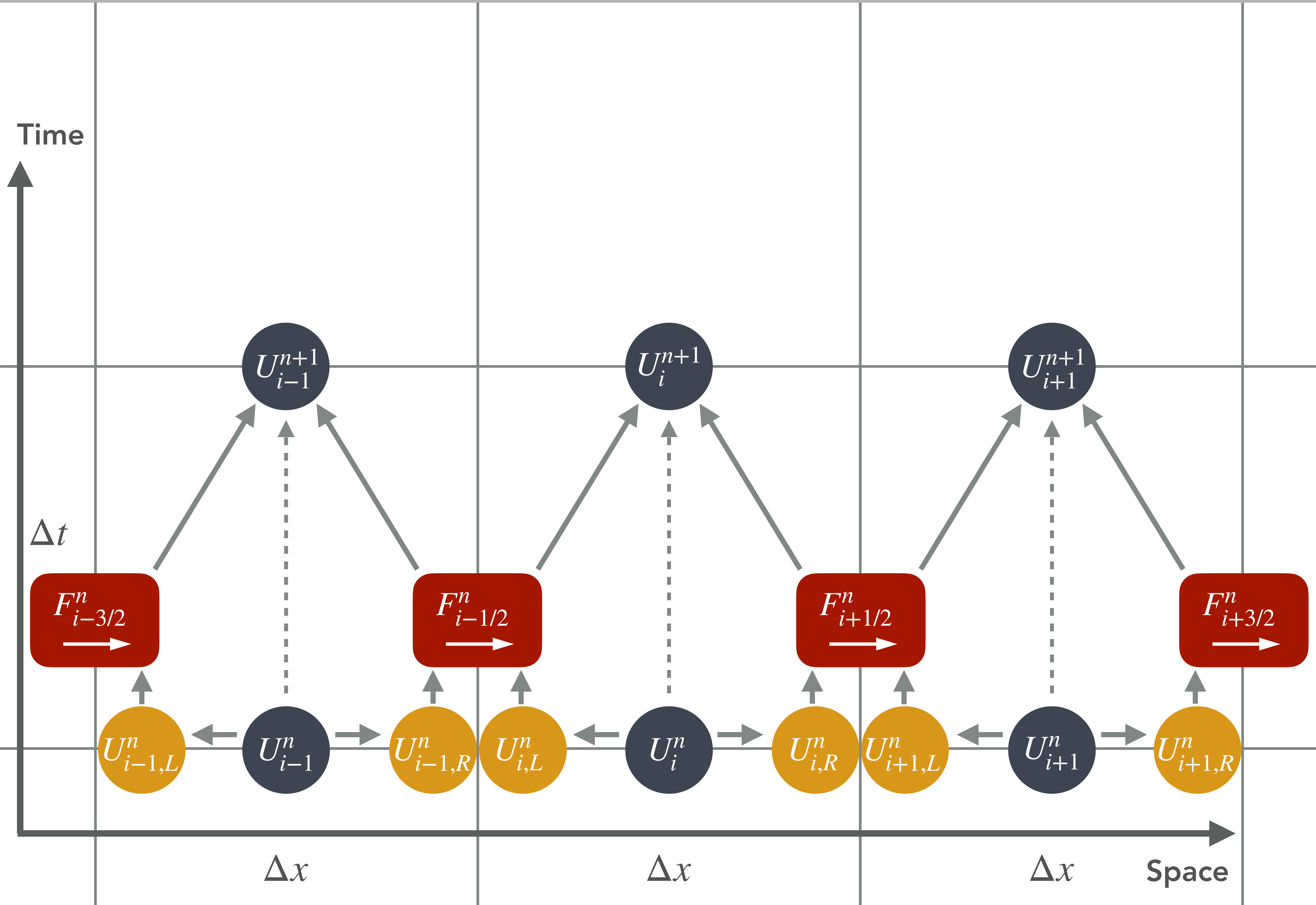


1st-order in space (constant)
1st-order in time
$\alpha_{\mathrm{cfl}} = 0.8$

Godunov scheme (1st order in space and time)



With reconstruction (2nd order in space, 1st in time)

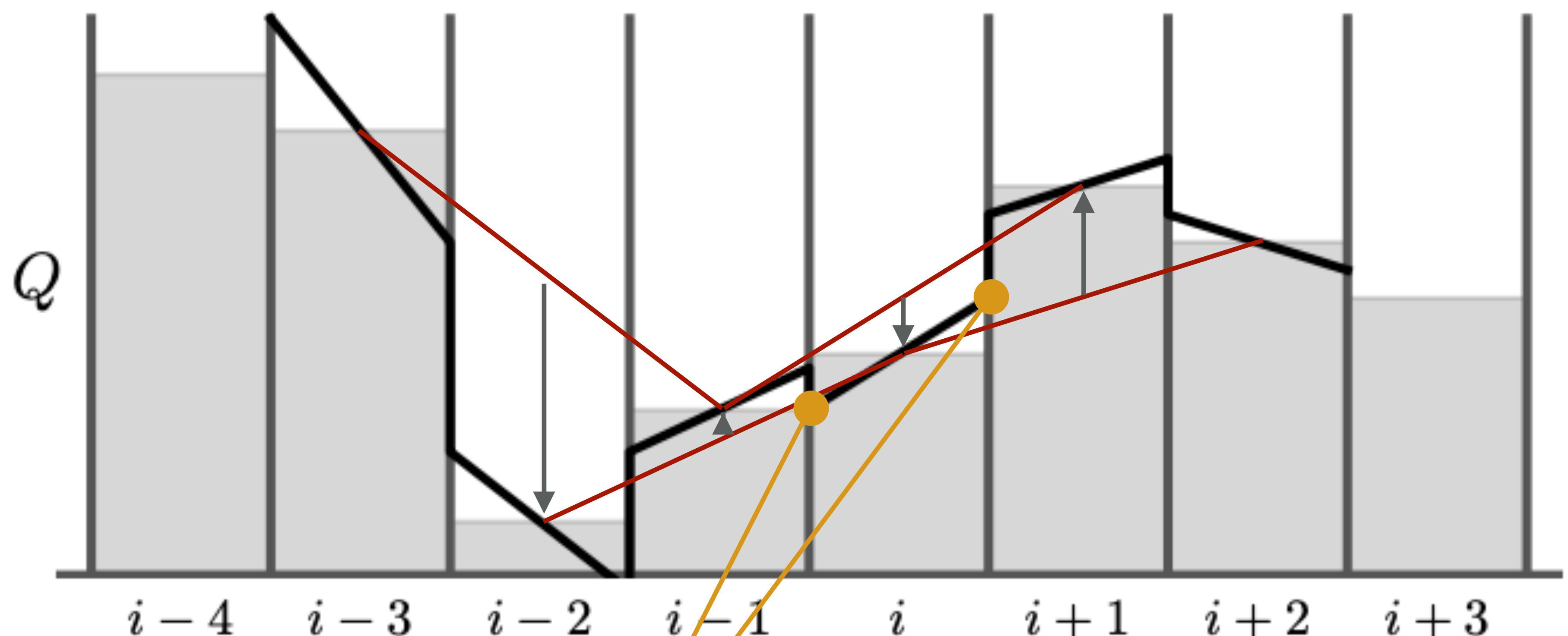


Linear Reconstruction

Define: $\boldsymbol{s}_{\mathrm{L}} \equiv \dfrac{\boldsymbol{V}_i - \boldsymbol{V}_{i-1}}{\Delta x}$ $\quad \boldsymbol{s}_{\mathrm{R}} \equiv \dfrac{\boldsymbol{V}_{i+1} - \boldsymbol{V}_i}{\Delta x}$ $\quad \boldsymbol{s}_{\mathrm{C}} \equiv \dfrac{\boldsymbol{s}_{\mathrm{L}} + \boldsymbol{s}_{\mathrm{R}}}{2} = \dfrac{\boldsymbol{V}_{i+1} - \boldsymbol{V}_{i-1}}{2\Delta x}$

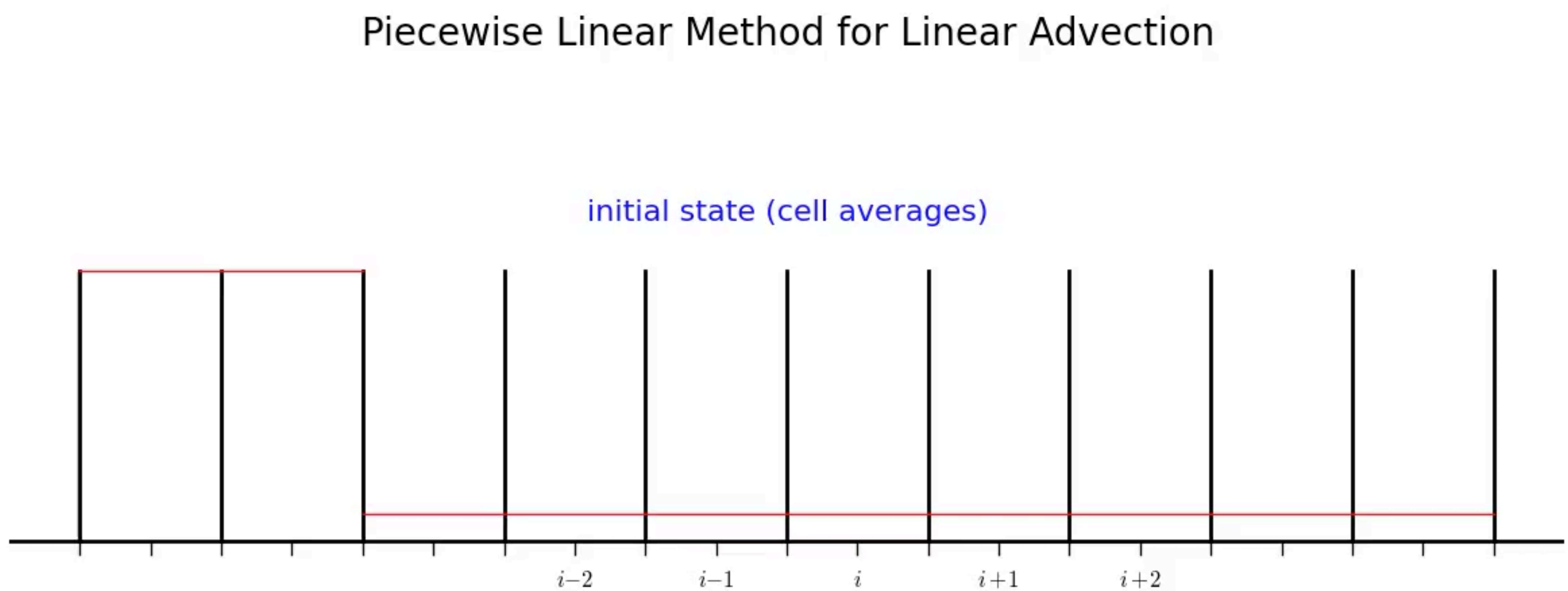
Central derivative: $\boldsymbol{V}(x) = \boldsymbol{V}_i + \boldsymbol{s}_{\mathrm{C}}(x - x_i)$

Example:



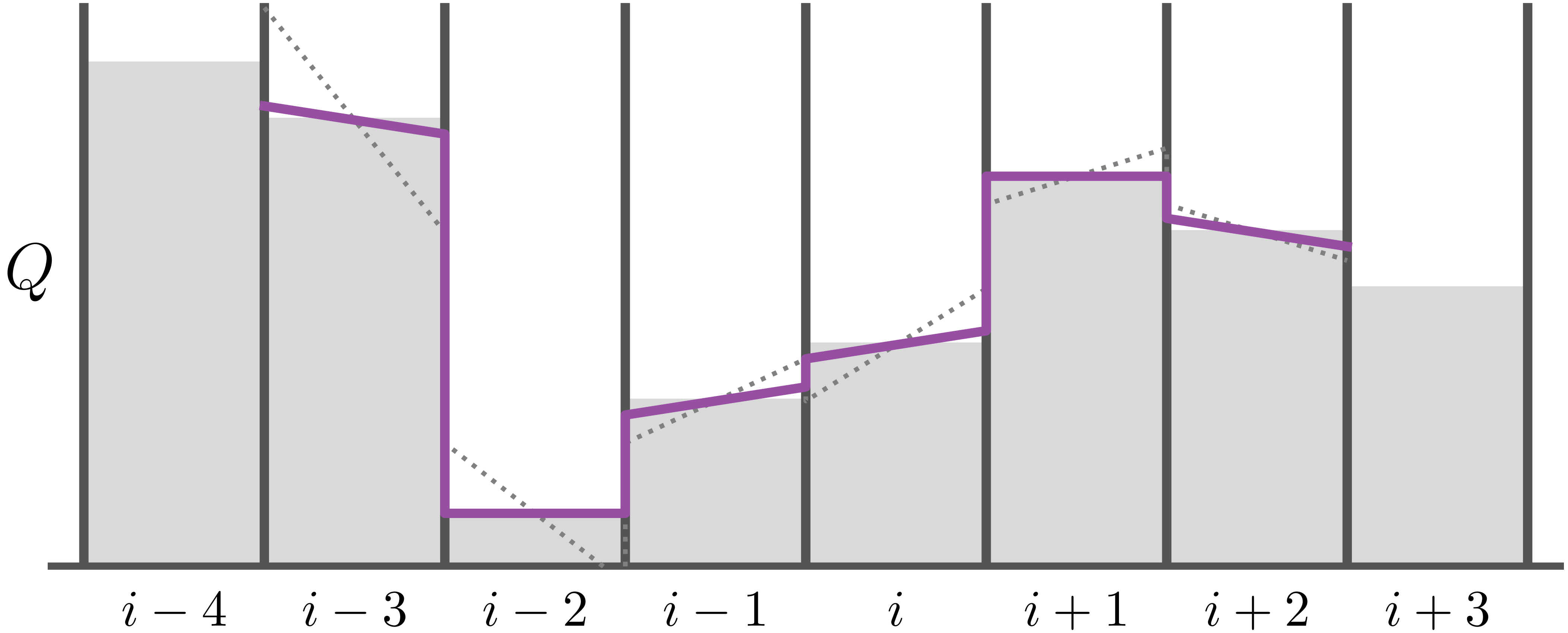
Cell edge states: $\boldsymbol{V}_{i\mp 1/2} = \boldsymbol{V}_i \mp \dfrac{\Delta x}{2}\boldsymbol{s}_{\mathrm{C}}$

Linear Reconstruction



Zingale 2021

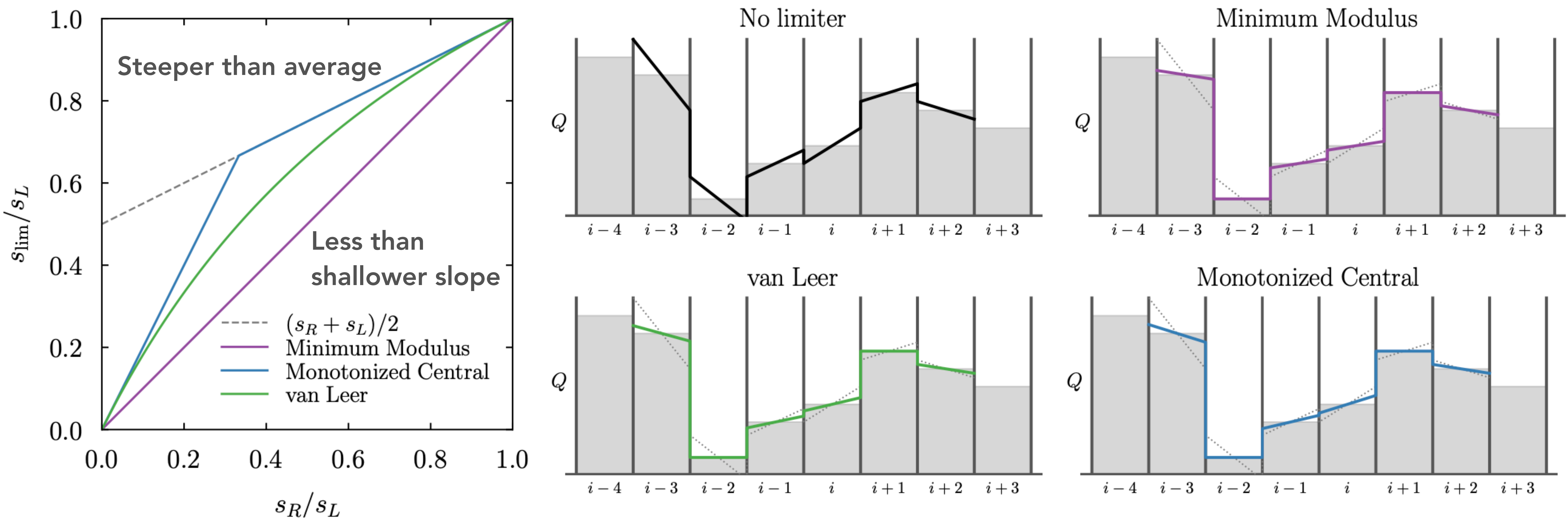
Slope limiters



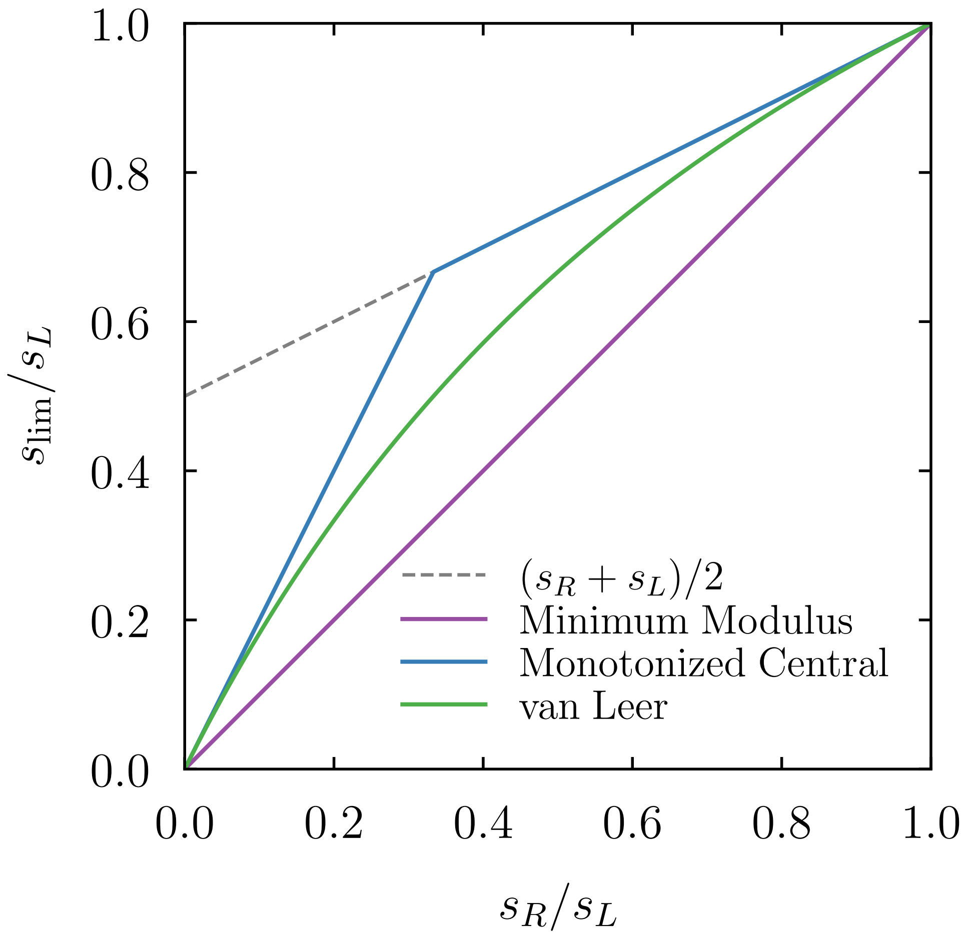
"Minimum Modulus:"

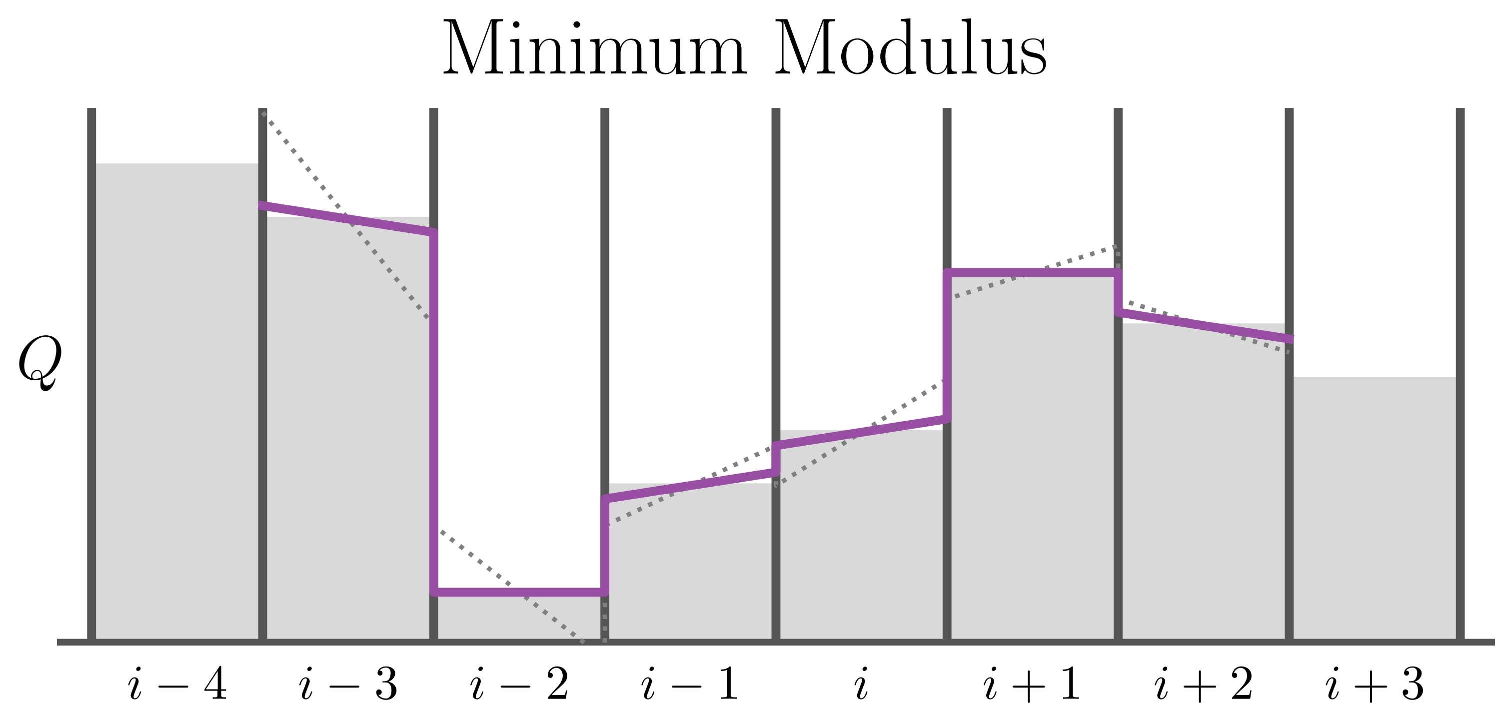
Out of s_L and s_R, take the one with the smaller absolute value, or zero if they have opposite signs.

Linear Reconstruction

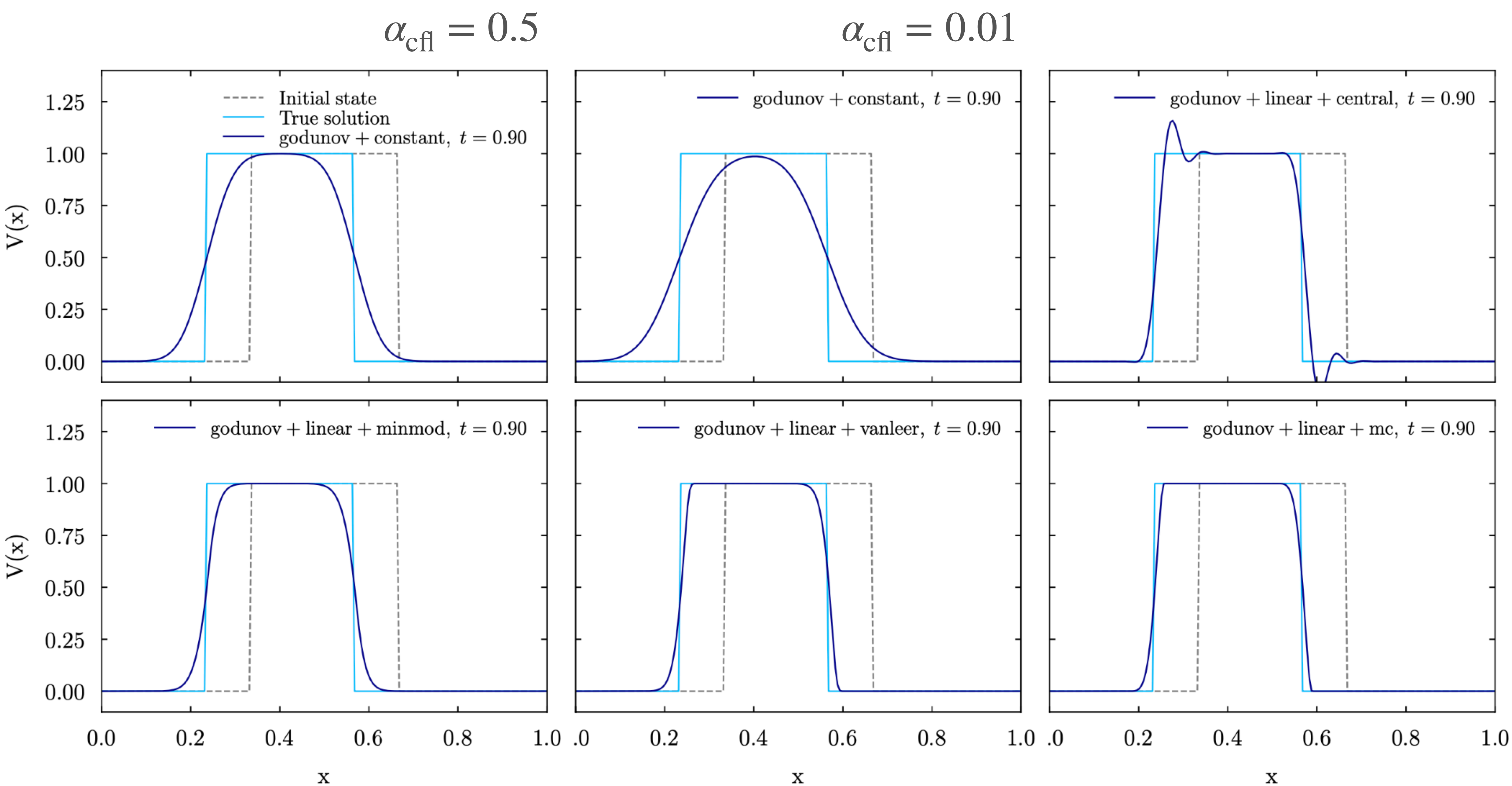


Slope limiters

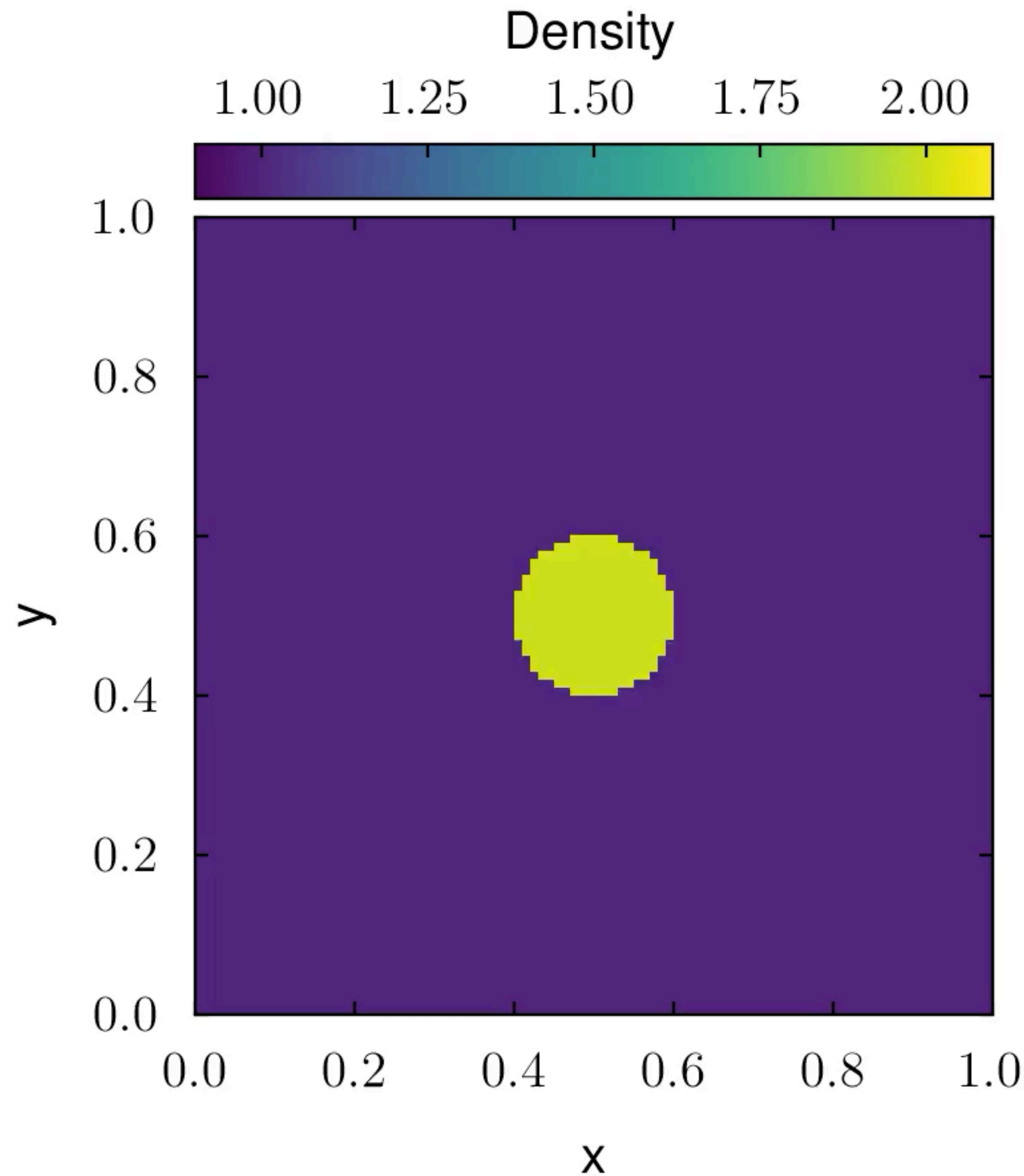




The Riemann problem (1D Euler)

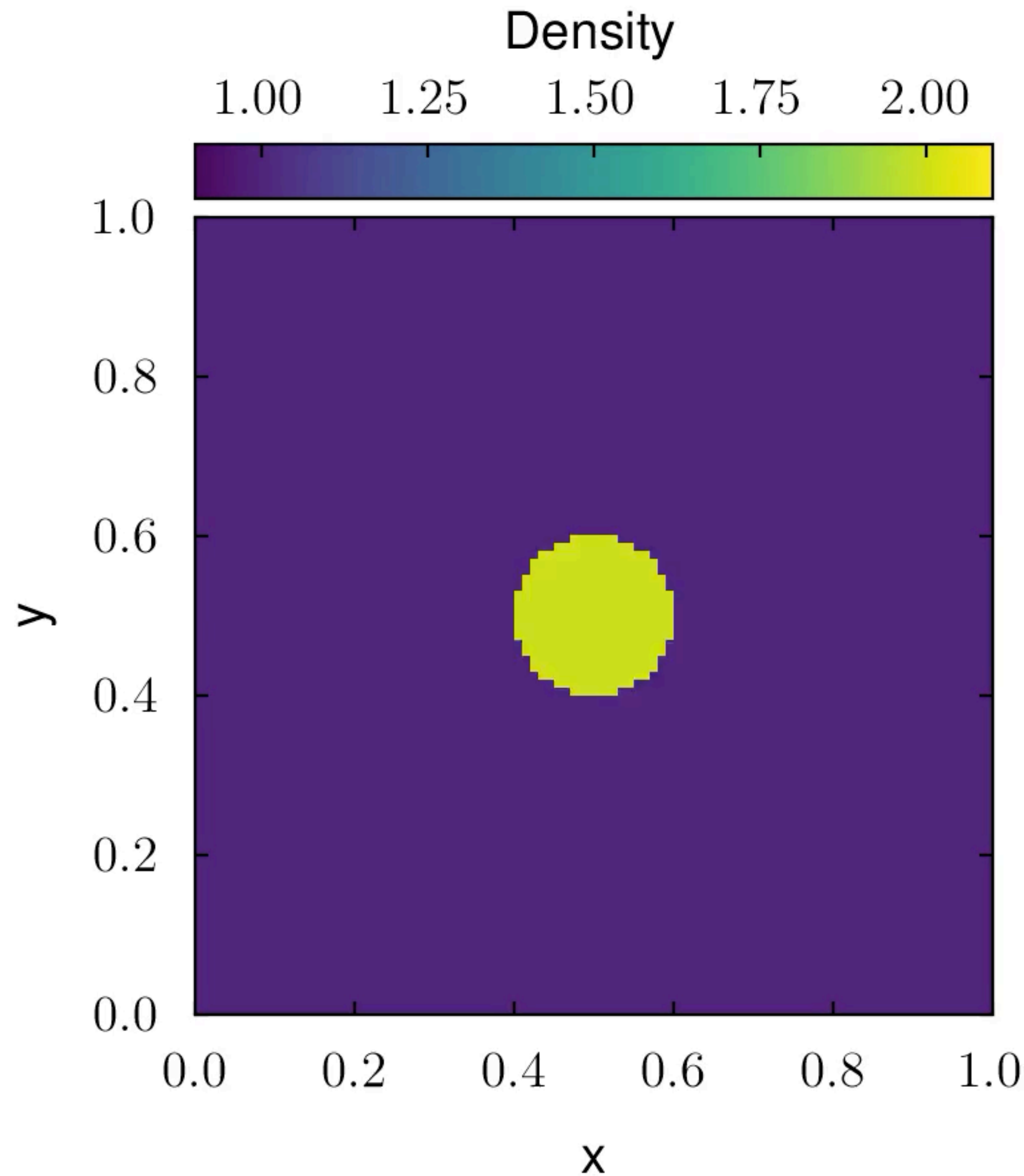


Advection in 2D



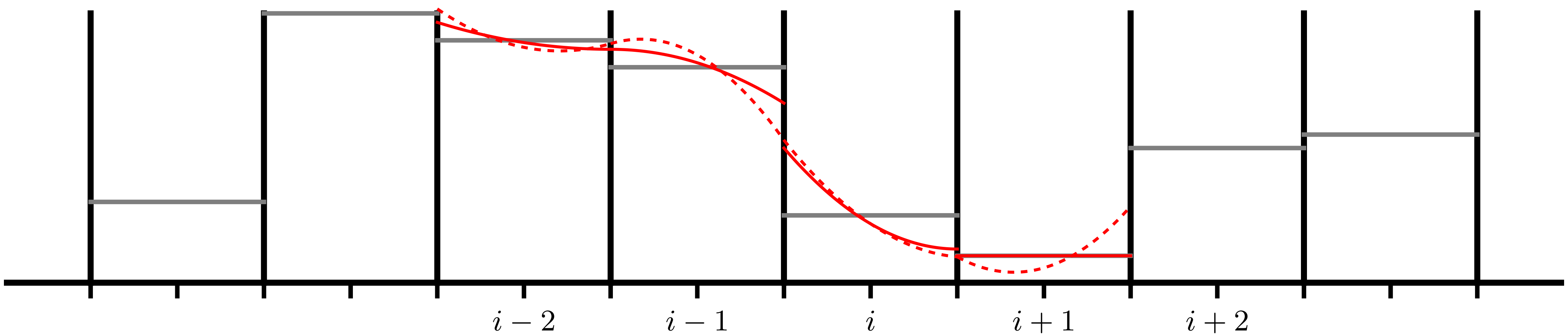
2nd-order in space (linear)
MinMod limiter
1st-order in time
$\alpha_{\mathrm{cfl}} = 0.8$

Advection in 2D



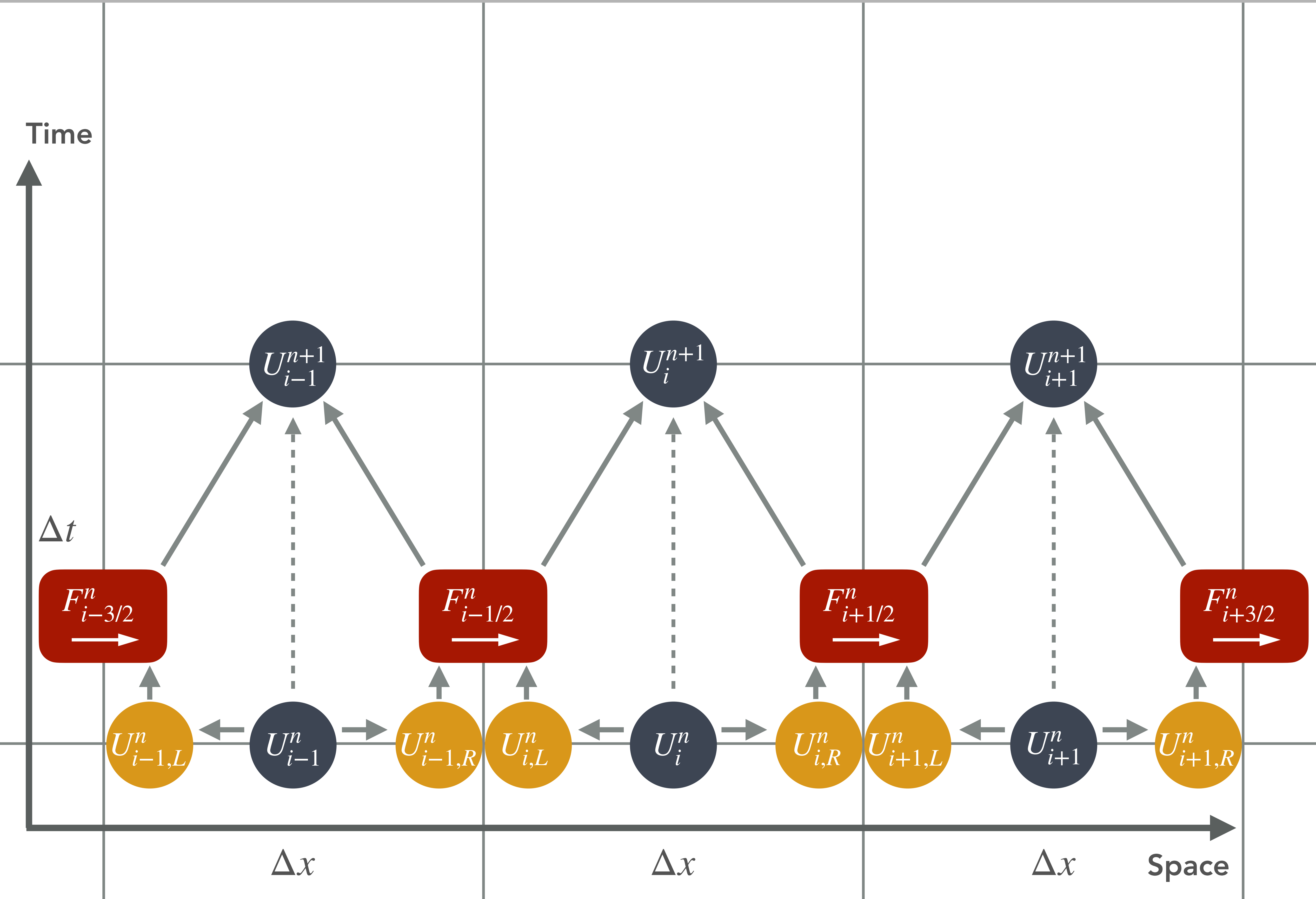
2nd-order in space (linear)
MC limiter
1st-order in time
$\alpha_{\text{cfl}} = 0.8$

Parabolic Reconstruction

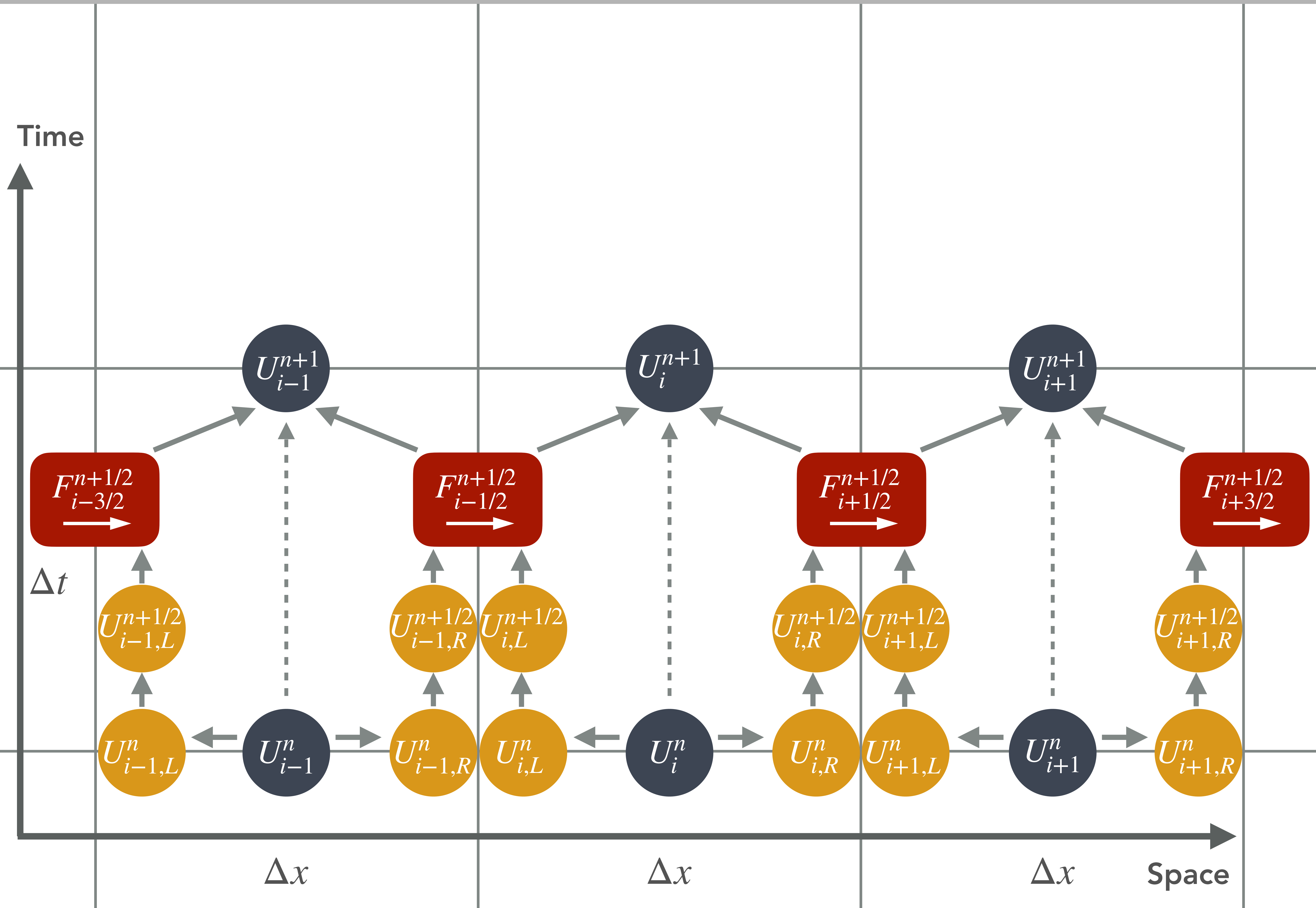


Zingale 2021

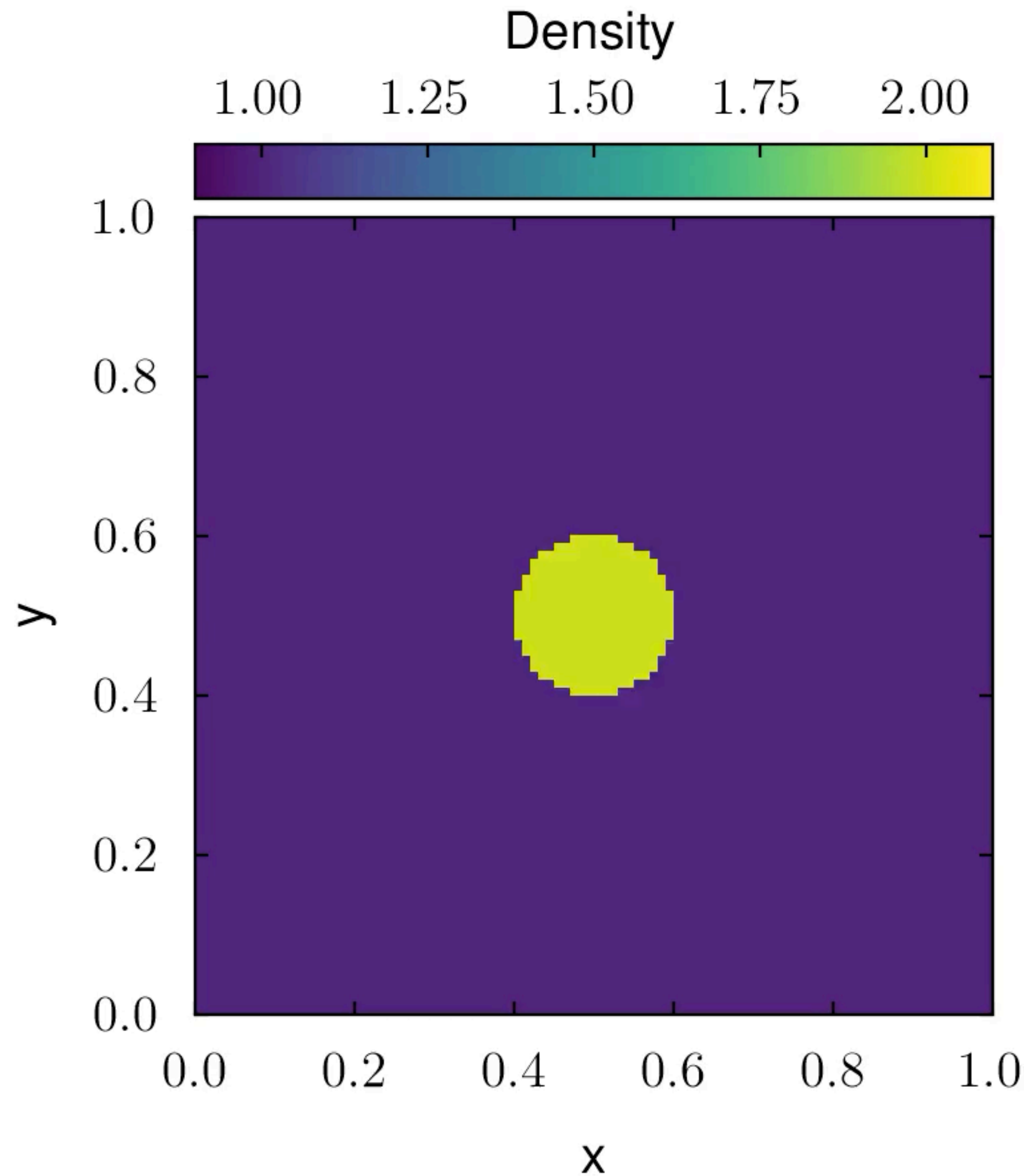
With reconstruction (2nd order in space, 1st in time)



MUSCL-Hancock (2nd order in space and time)



Advection in 2D



2nd-order in space (linear)
MC limiter
2nd-order in time (Hancock)
$\alpha_{\mathrm{cfl}} = 0.8$

ULULA: A 2D python hydro code

ULULA

Quick start

The easiest way to execute Ulula is via the runtime function `run()` (see documentation below). The main input to this function is a `Setup`, a class that contains all data and routines that describe a particular physical problem. The following example runs a simulation of the Kelvin-Helmholtz test:

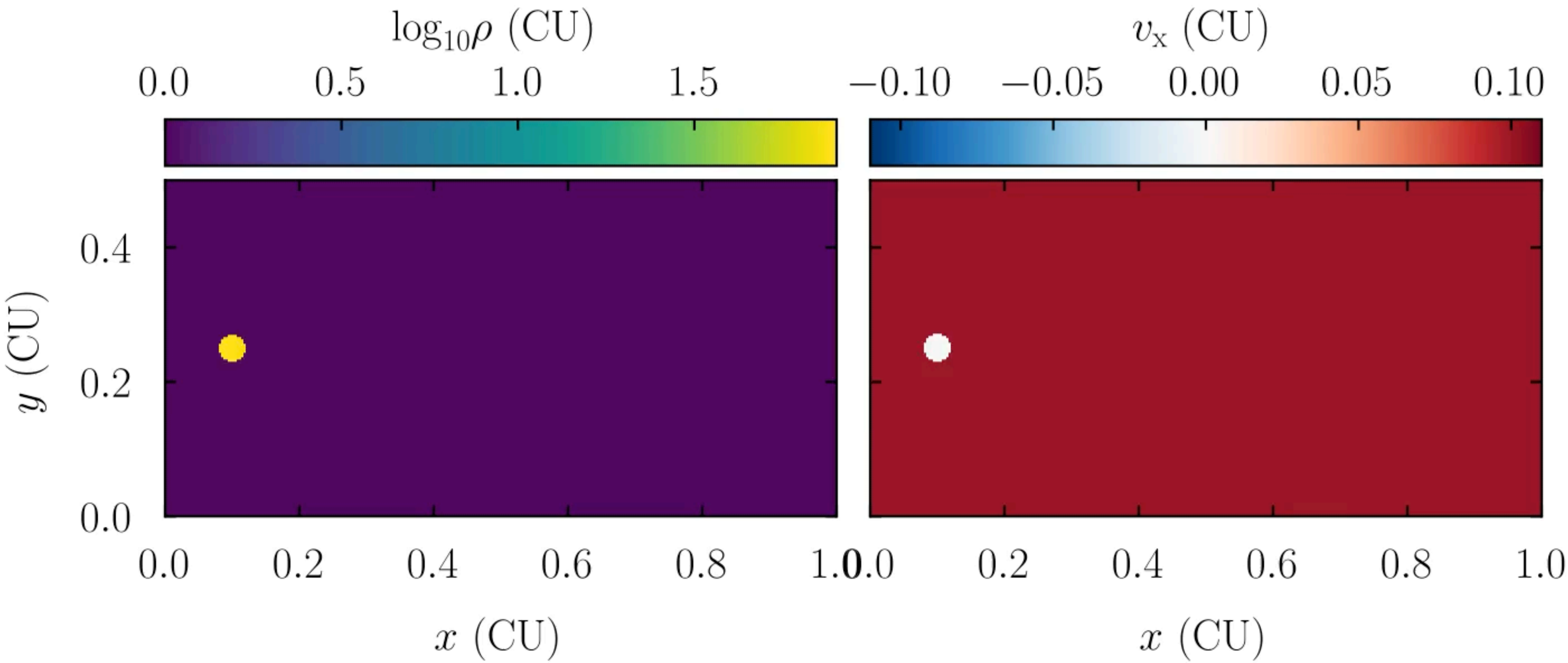
```
import ulula.setups.kelvin_helmholtz as setup_kh
import ulula.run as ulula_run

setup = setup_kh.SetupKelvinHelmholtz()
ulula_run.run(setup, tmax = 4.0, nx = 200)
```

This code will set up a domain with 200x200 cells and run the Kelvin-Helmholtz test until time 4.0 using the default hydro solver. If you want control over the algorithms used, the `HydroScheme` class contains all relevant settings, e.g.:

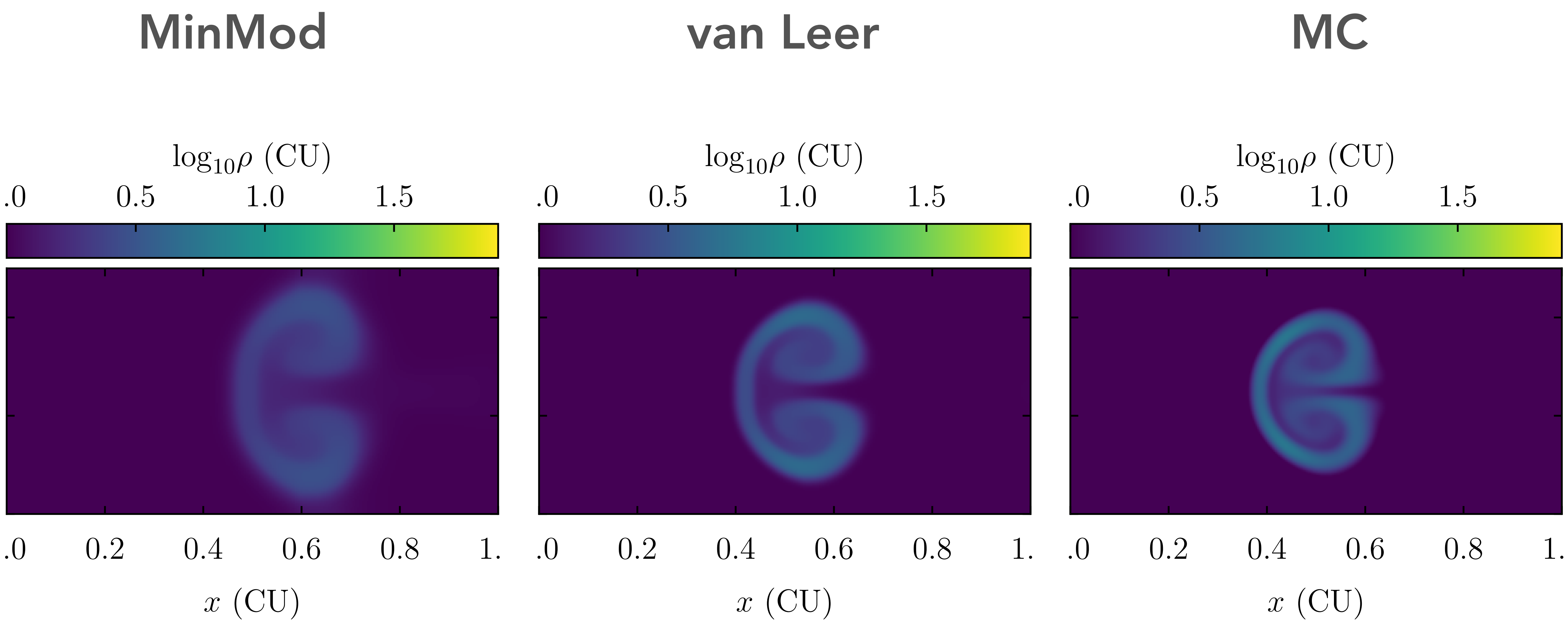
```
import ulula.simulation as ulula_sim

hs = ulula_sim.HydroScheme(reconstruction = 'linear', limiter = 'mc', cfl = 0.9)
ulula_run.run(setup, hydro_scheme = hs, tmax = 4.0, nx = 200)
```

The `run()` function takes a number of additional parameters, many of which govern various possible output products such as interactive and saved figures as well as movies. For example, these code snippets would produce a series of density and pressure plots at time intervals of 0.5 or a movie of the evolution of density:

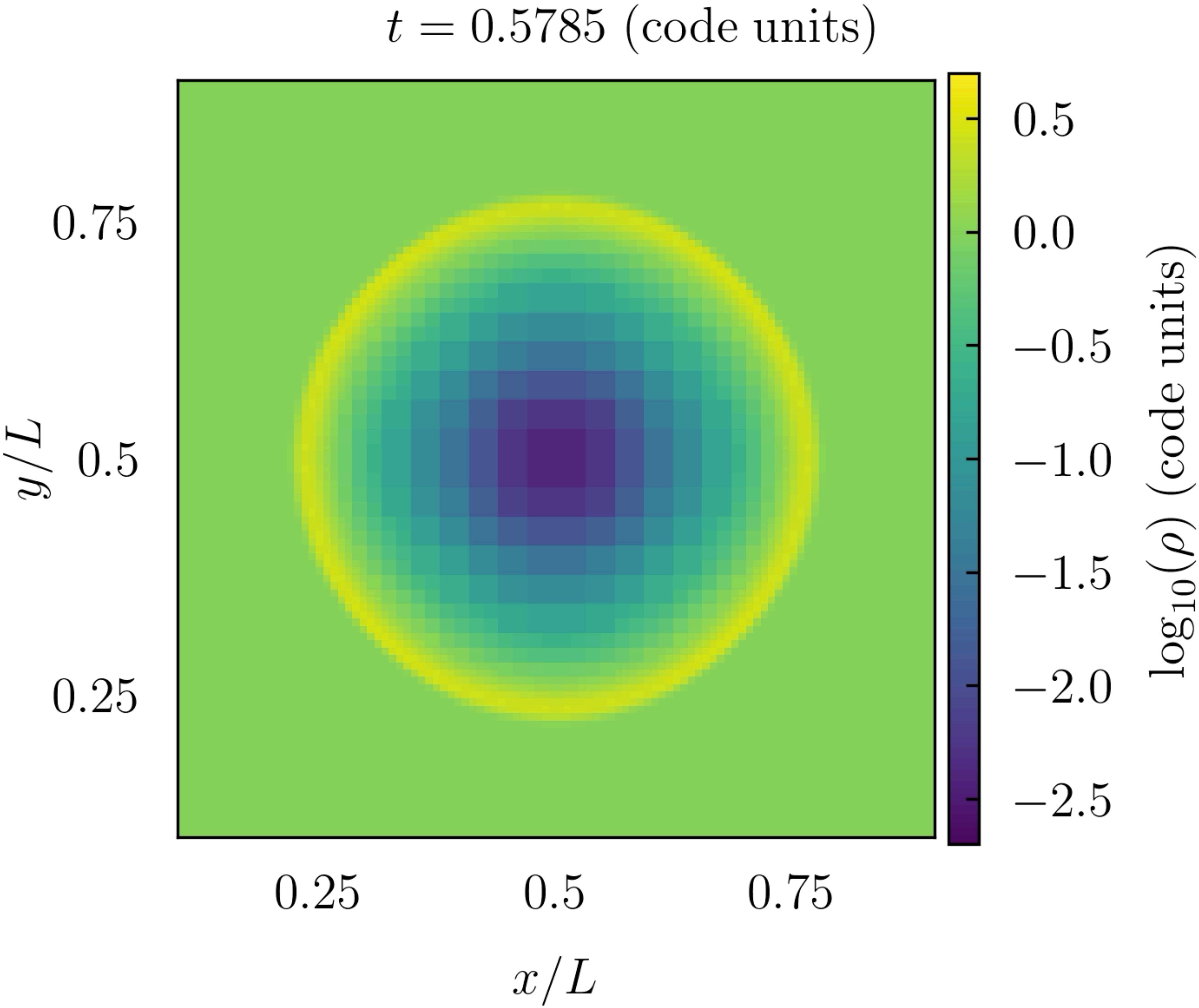
Cloud-crushing setup



Cloud-crushing setup

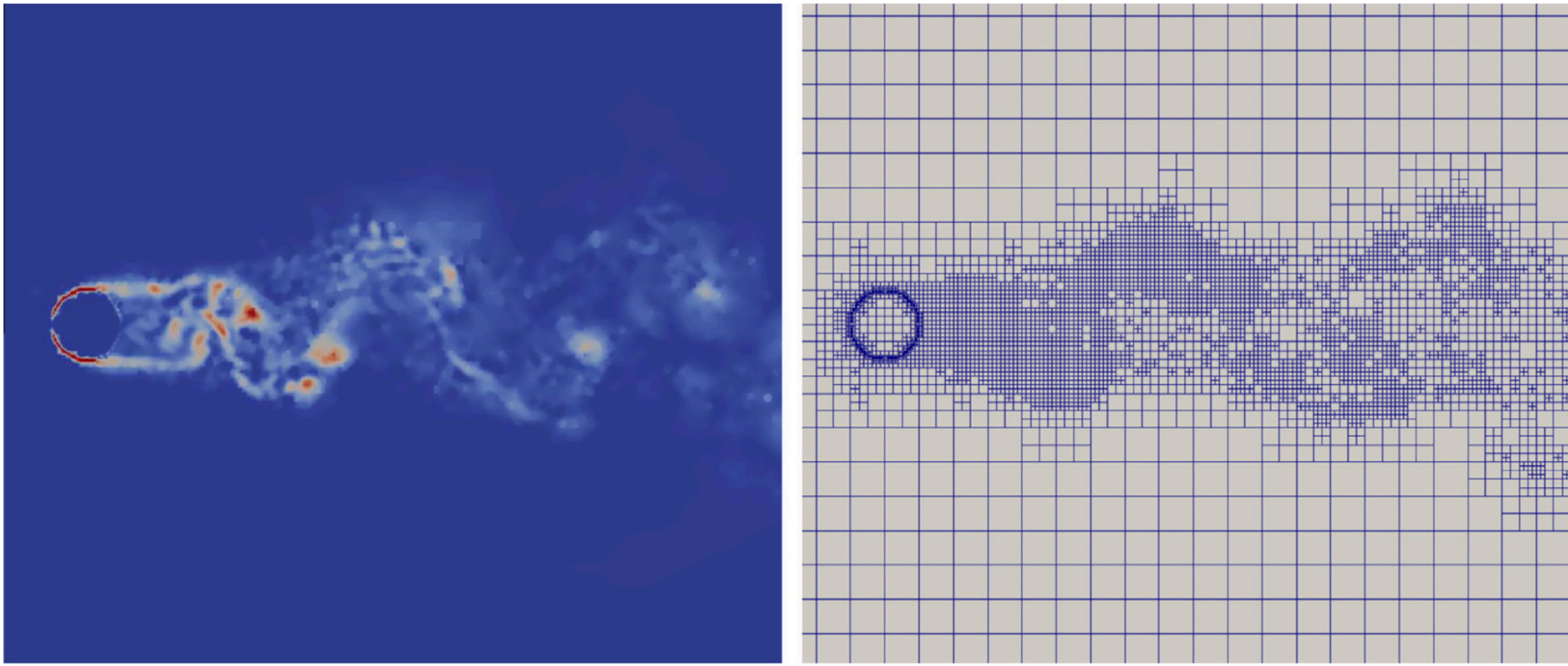


Popular hydro codes in astrophysics



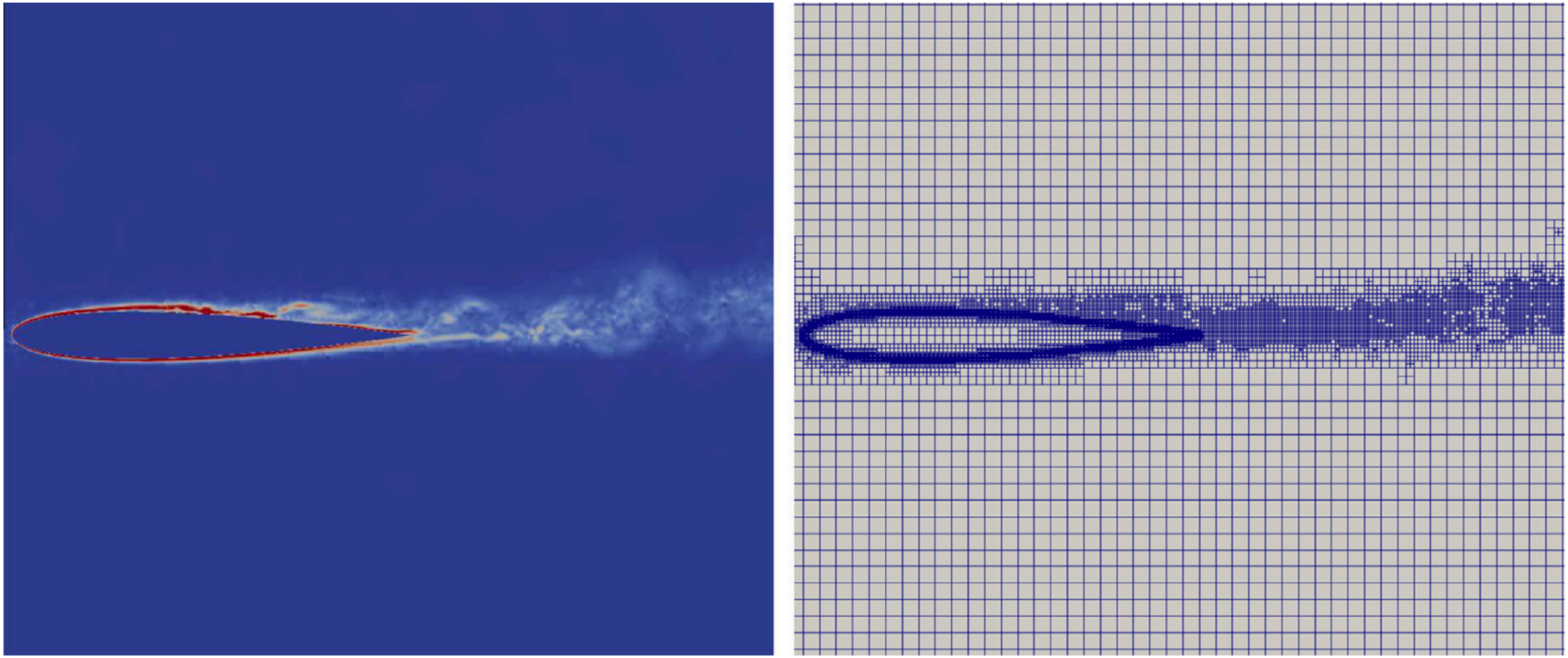
ART (Kravtsov et al. 1997)

Adaptive mesh refinement (AMR)



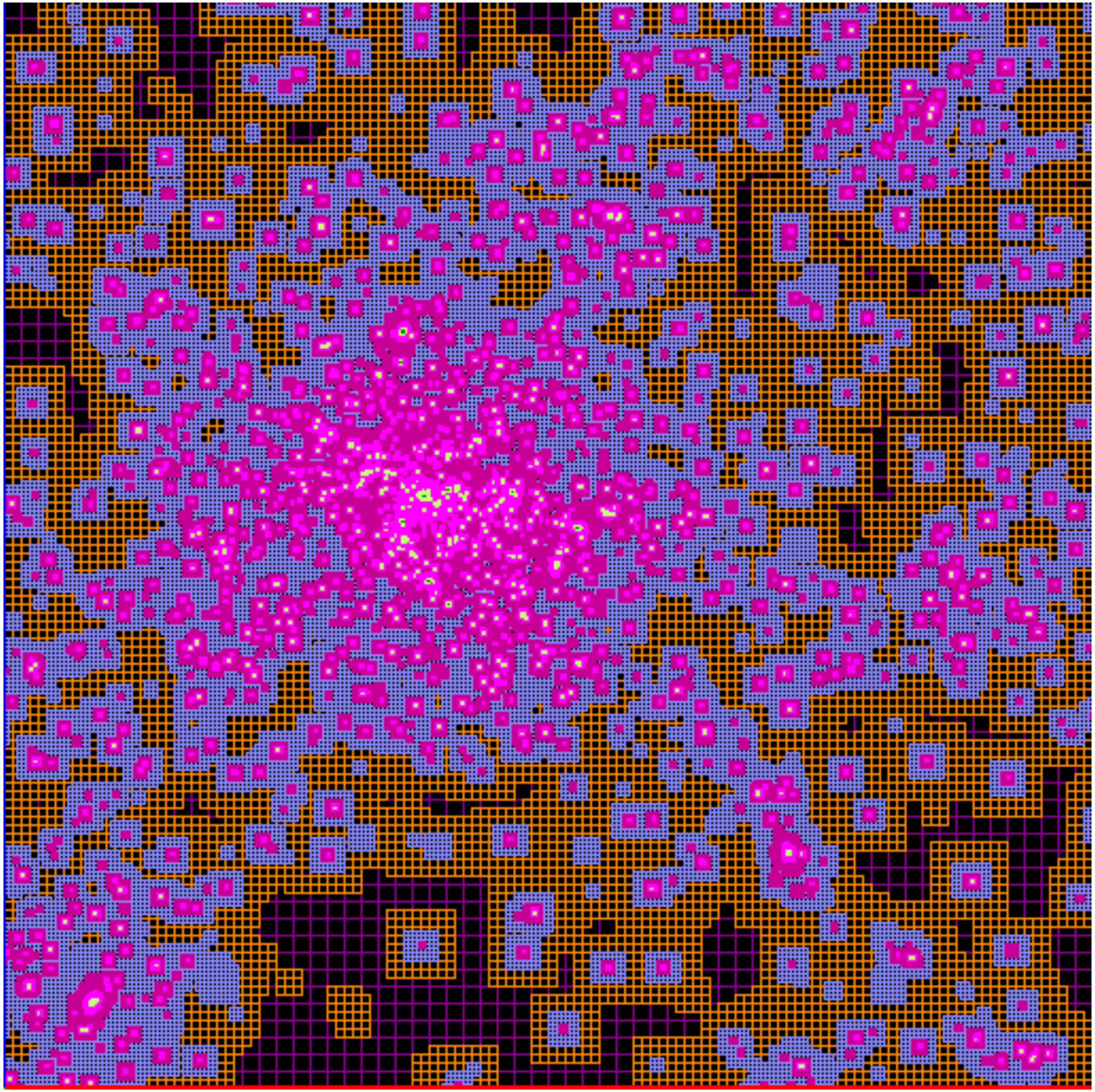
Antepara et al. 2015

Adaptive mesh refinement (AMR)



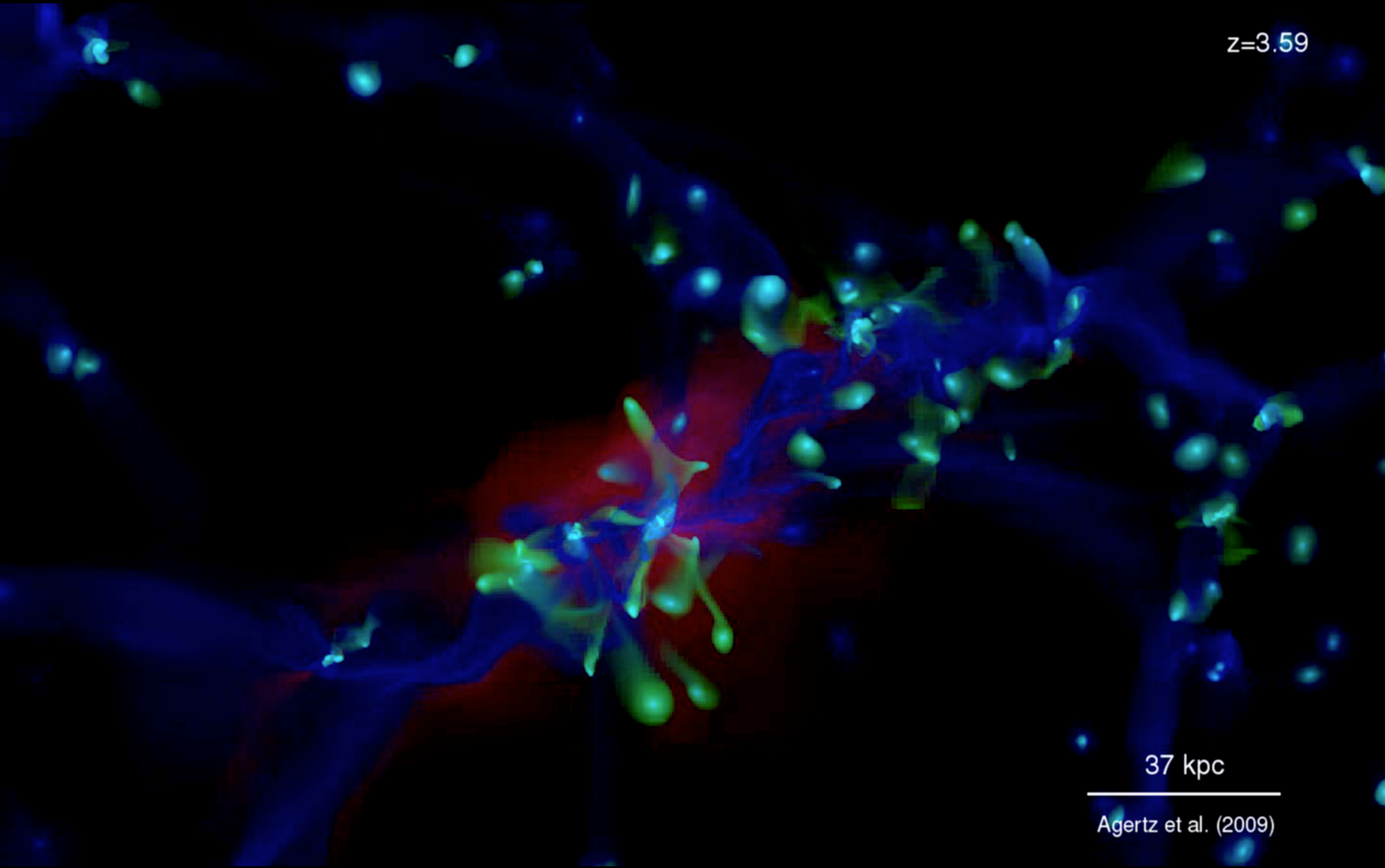
Antepara et al. 2015

Adaptive mesh refinement (AMR)



RAMSES (Teyssier et al. 2006)

RAMSES



RAMSES (Teyssier et al. 2006, Agertz et al. 2009, movie page)

Block-based AMR

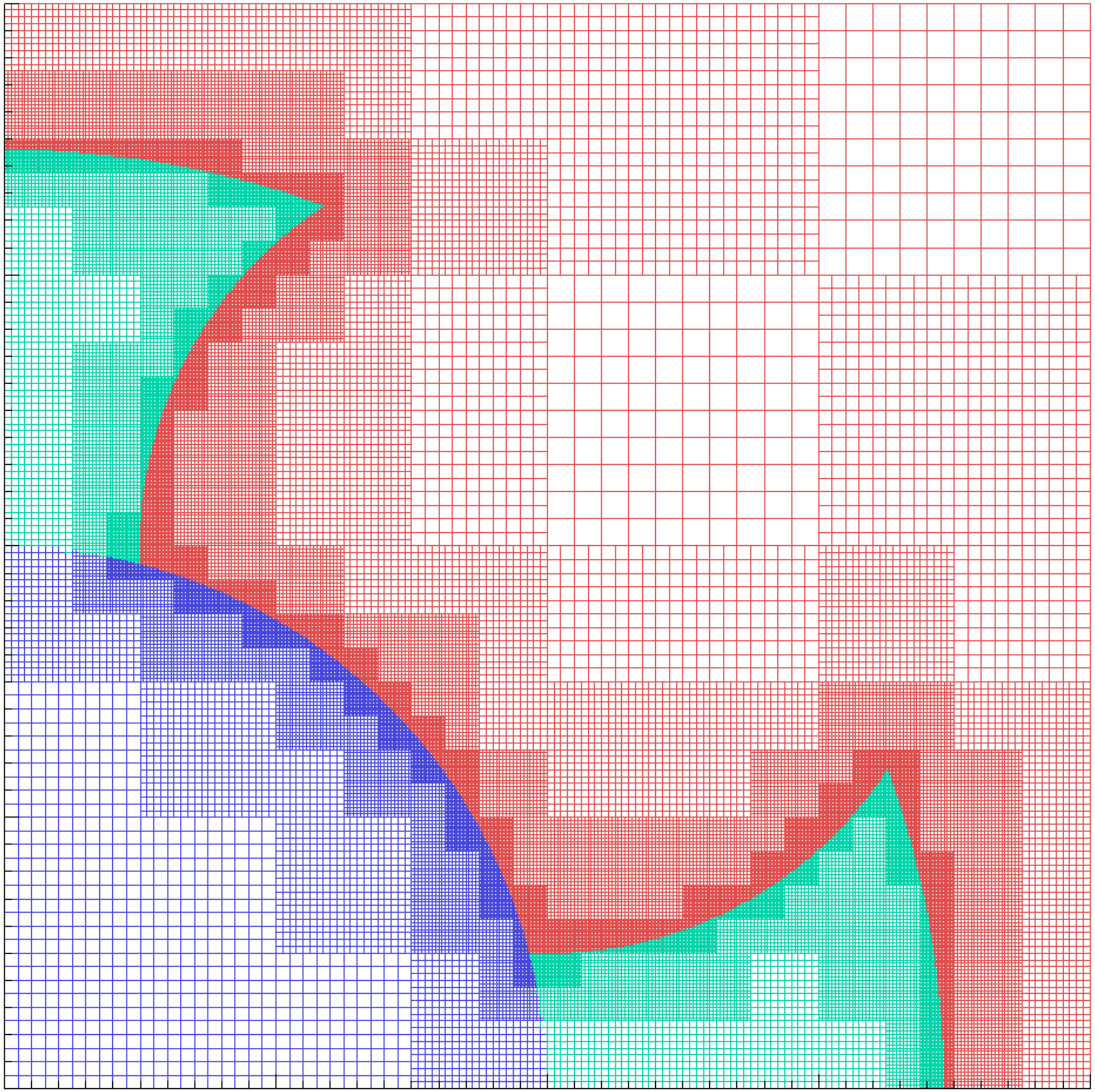
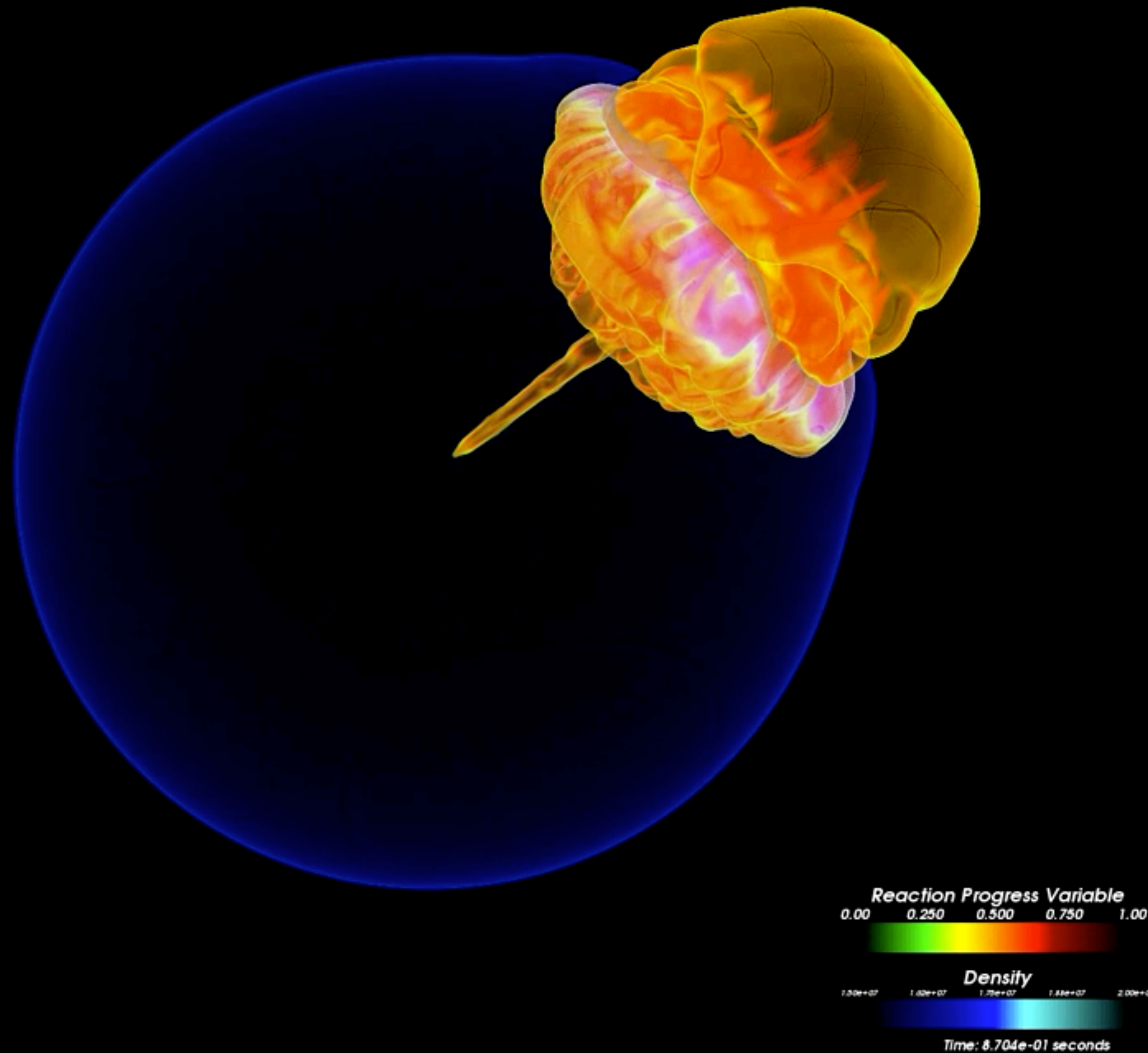


Image: Suhas Jain

FLASH



FLASH (Fryxell et al. 2000, movie page)

Smoothed Particle Hydrodynamics (SPH)

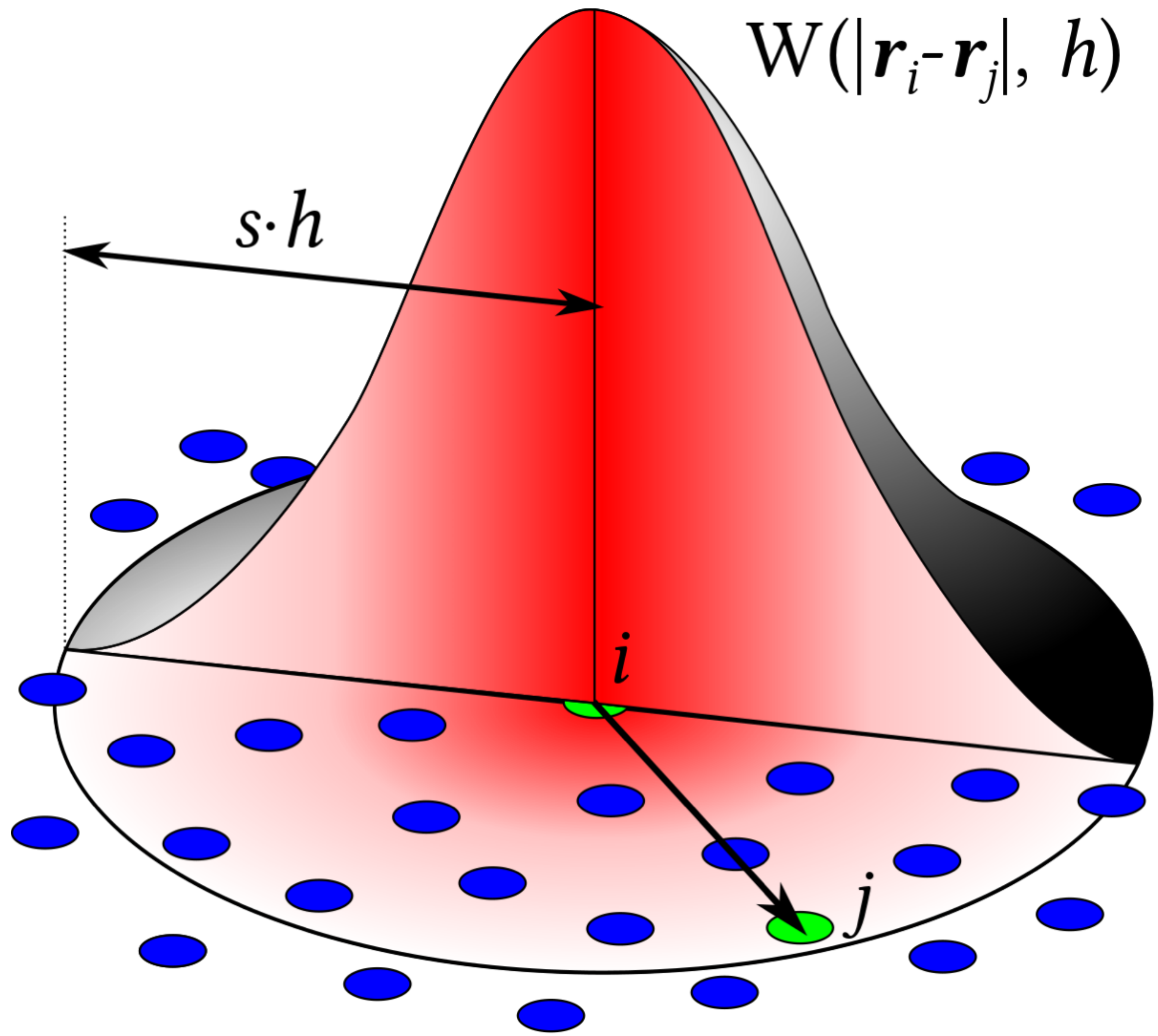
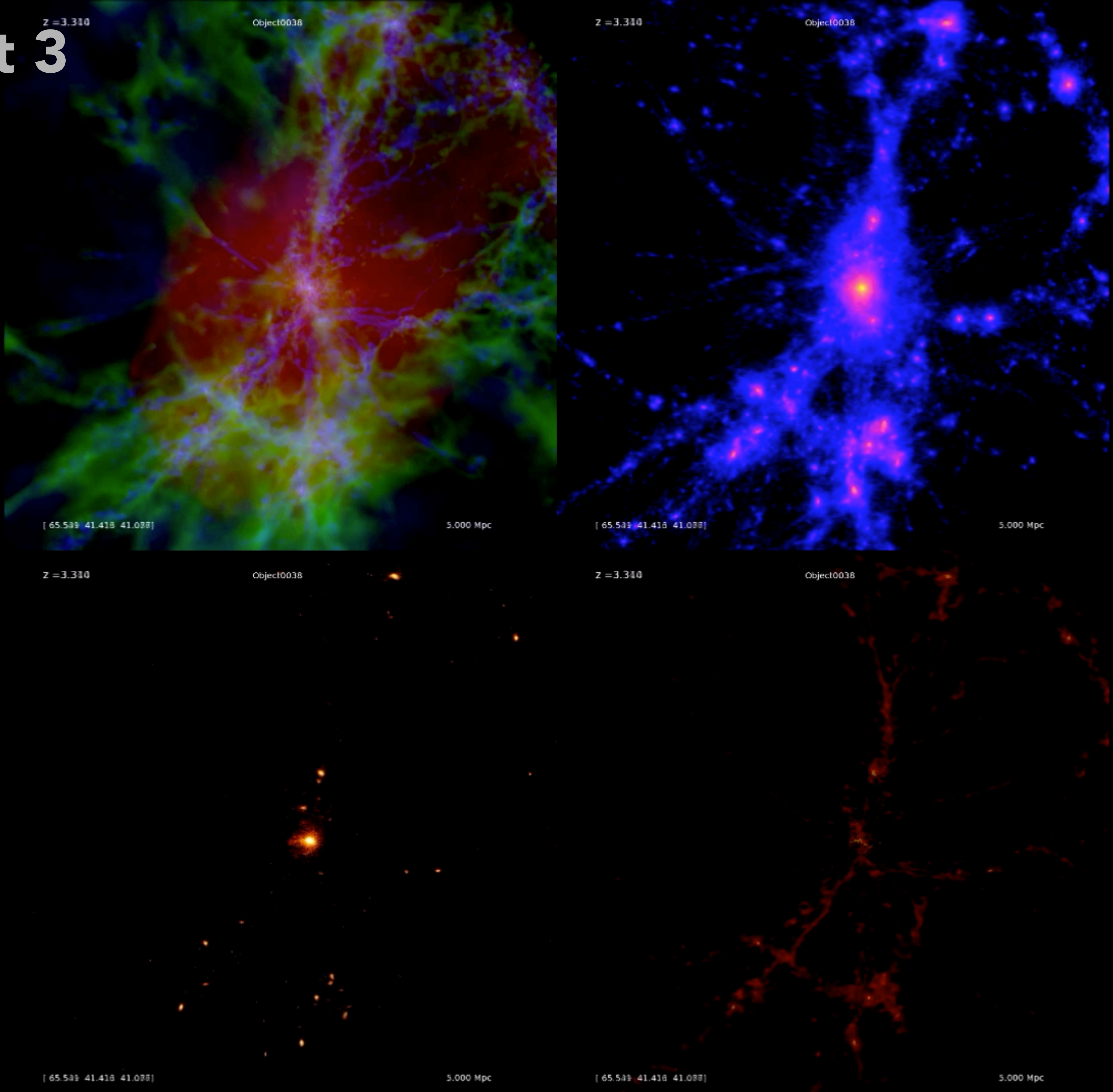


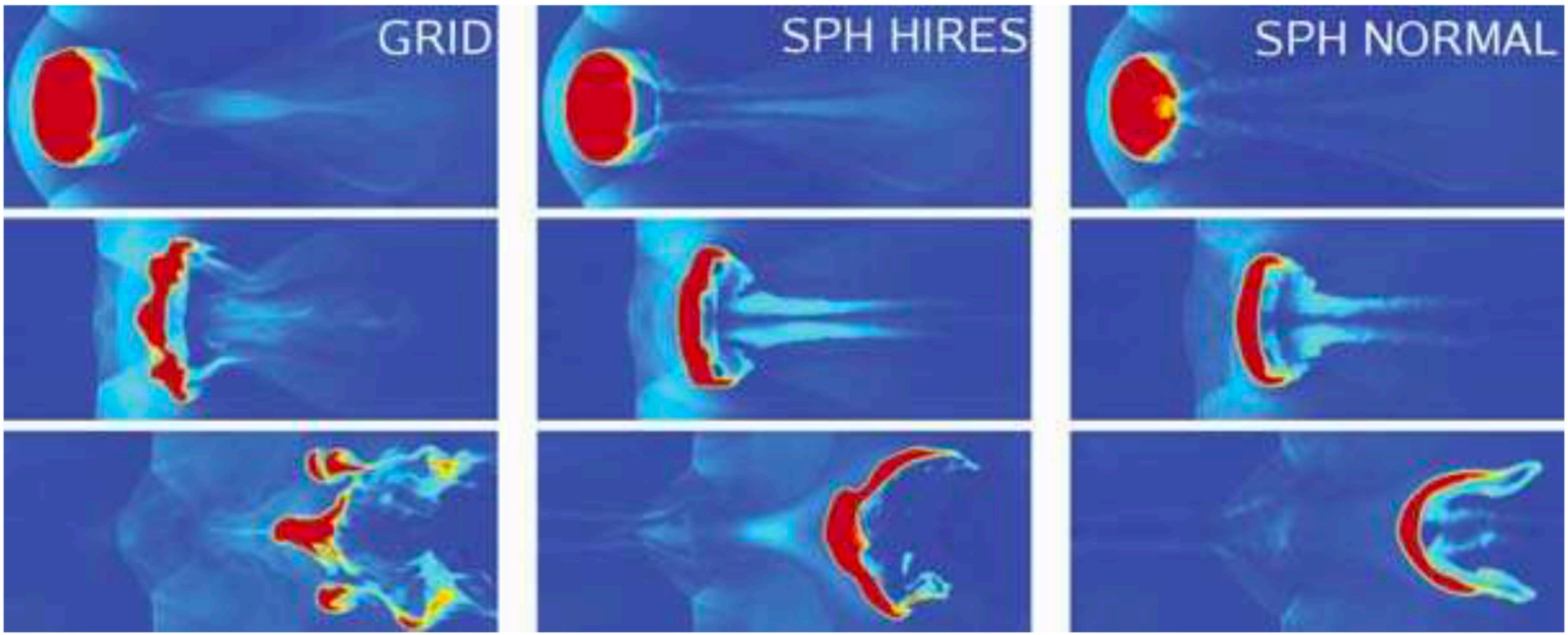
Image: Wikipedia

Gadget 3



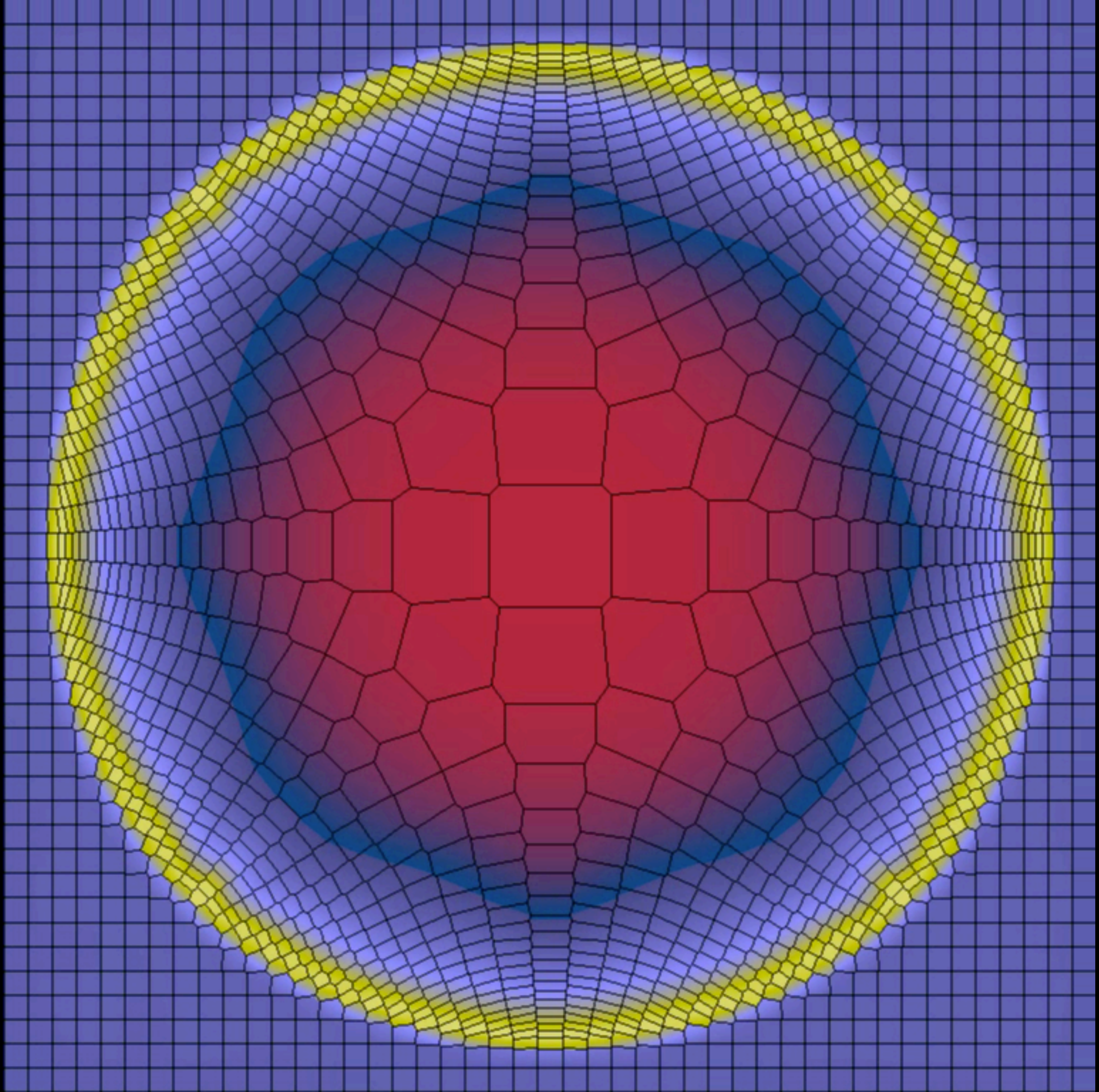
Gadget3 / EAGLE (Springel et al., Schaye et al. 2015, movie page)

Grid vs. SPH



Agertz et al. 2007

Arepo



Arepo (Springel 2010, movie page)

GIZMO

GIZMO (Hopkins 2015, movie page)

Reading

- Recommended: Zingale §7.2, §8.1-4