Finite volume schemes

An introduction

Bert Vandenbroucke, bv7@st-andrews.ac.uk

Some problems can be solved analytically…

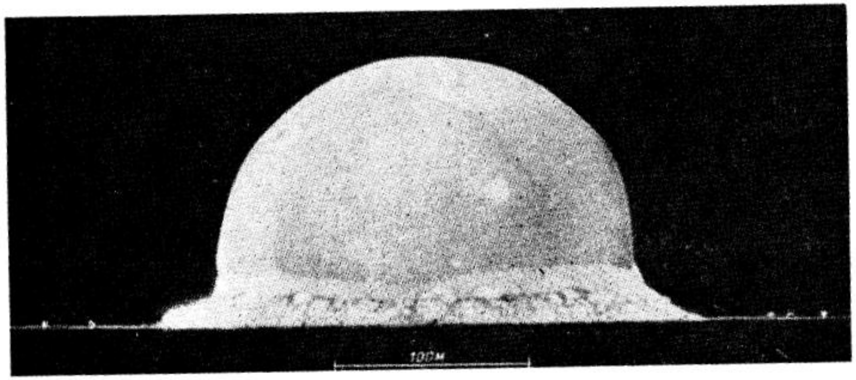


FIGURE 68. The fire ball at $t = 15 \times 10^{-3}$ s.

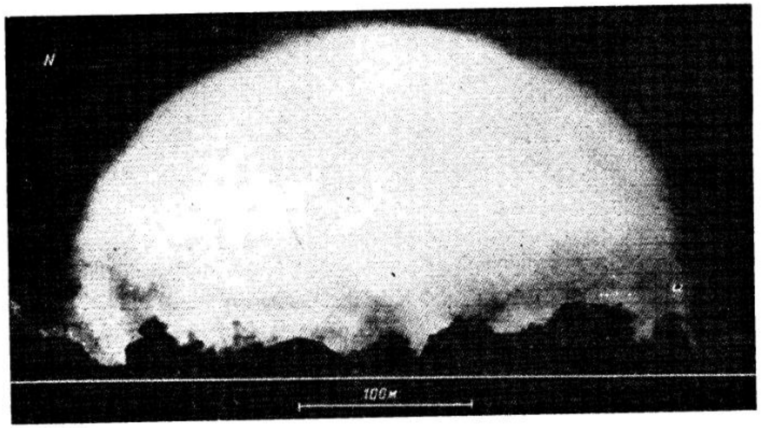
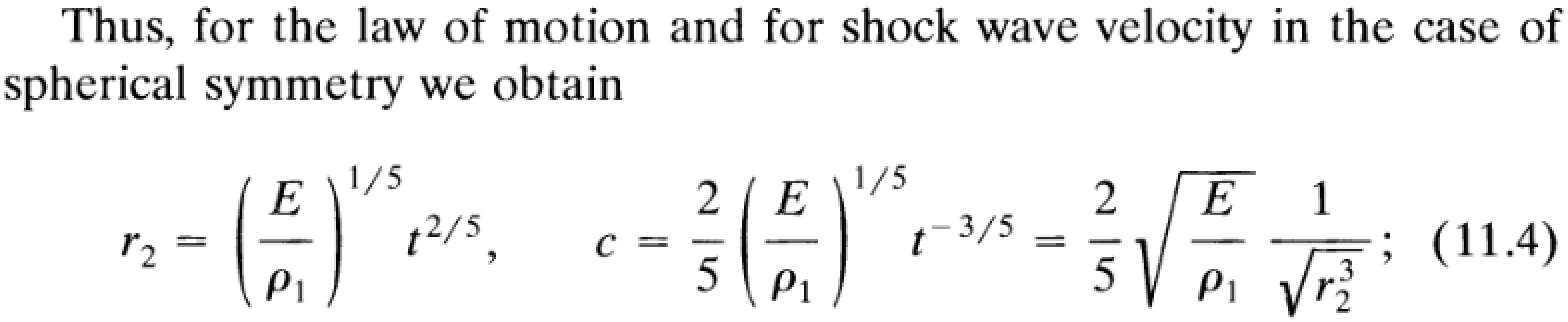


FIGURE 69. Photograph of the explosion at $t = 127 \times 10^{-3}$ s.

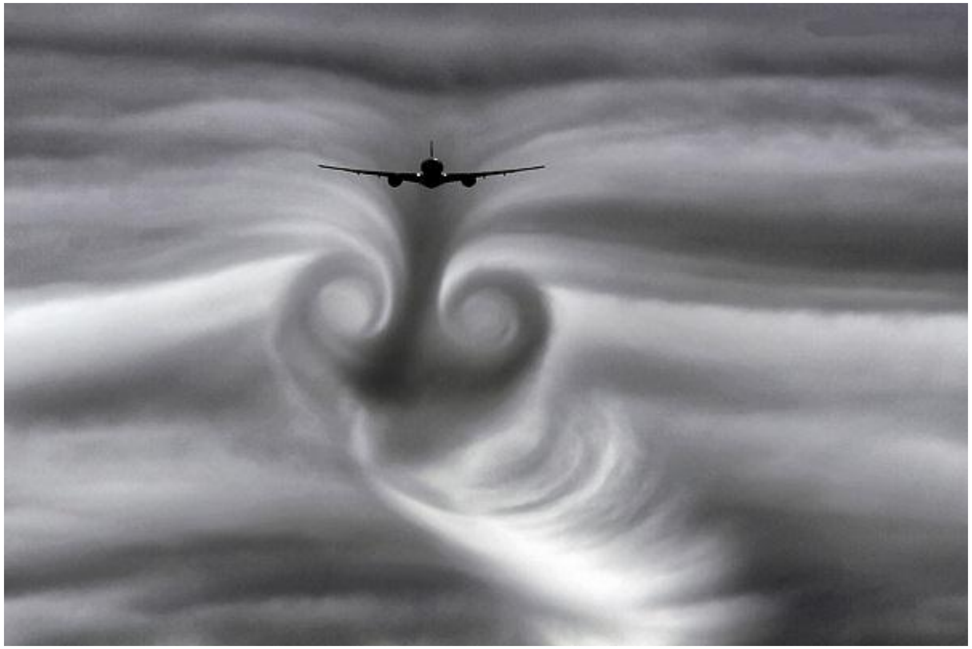


Thus, for the law of motion and for shock wave velocity in the case of spherical symmetry we obtain

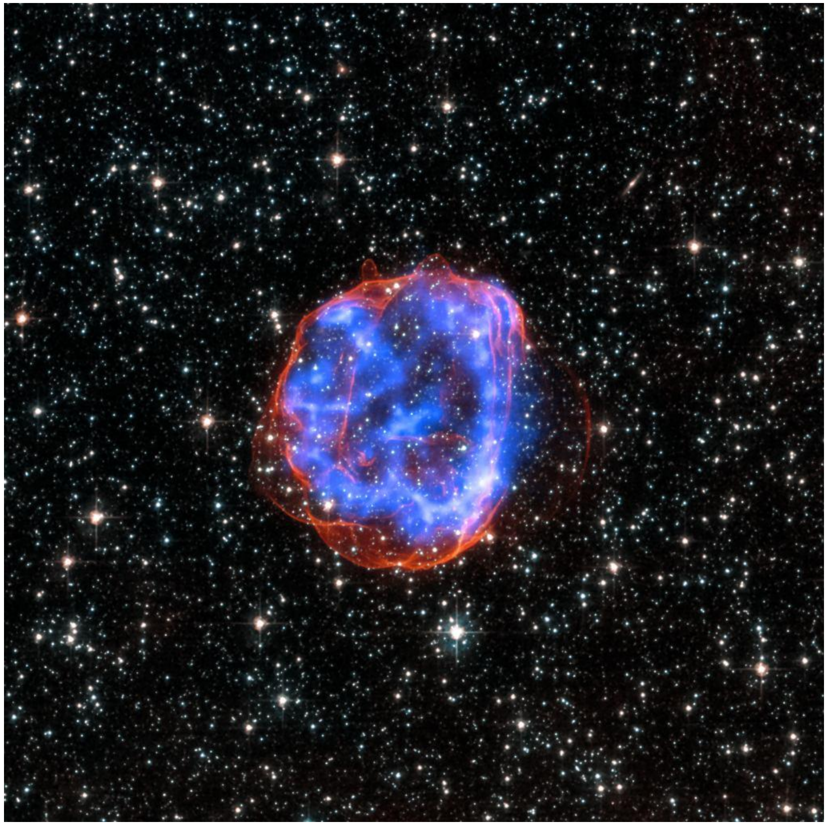
$$r_2 = \left(\frac{E}{\rho_1}\right)^{1/5} t^{2/5}, \qquad c = \frac{2}{5}\left(\frac{E}{\rho_1}\right)^{1/5} t^{-3/5} = \frac{2}{5}\sqrt{\frac{E}{\rho_1}}\,\frac{1}{\sqrt{r_2^3}}; \quad (11.4)$$

Sedov, L. I., Similarity and dimensional methods in mechanics, 10th edition (CRC Press, 1993), pp. 261-290

… most cannot.

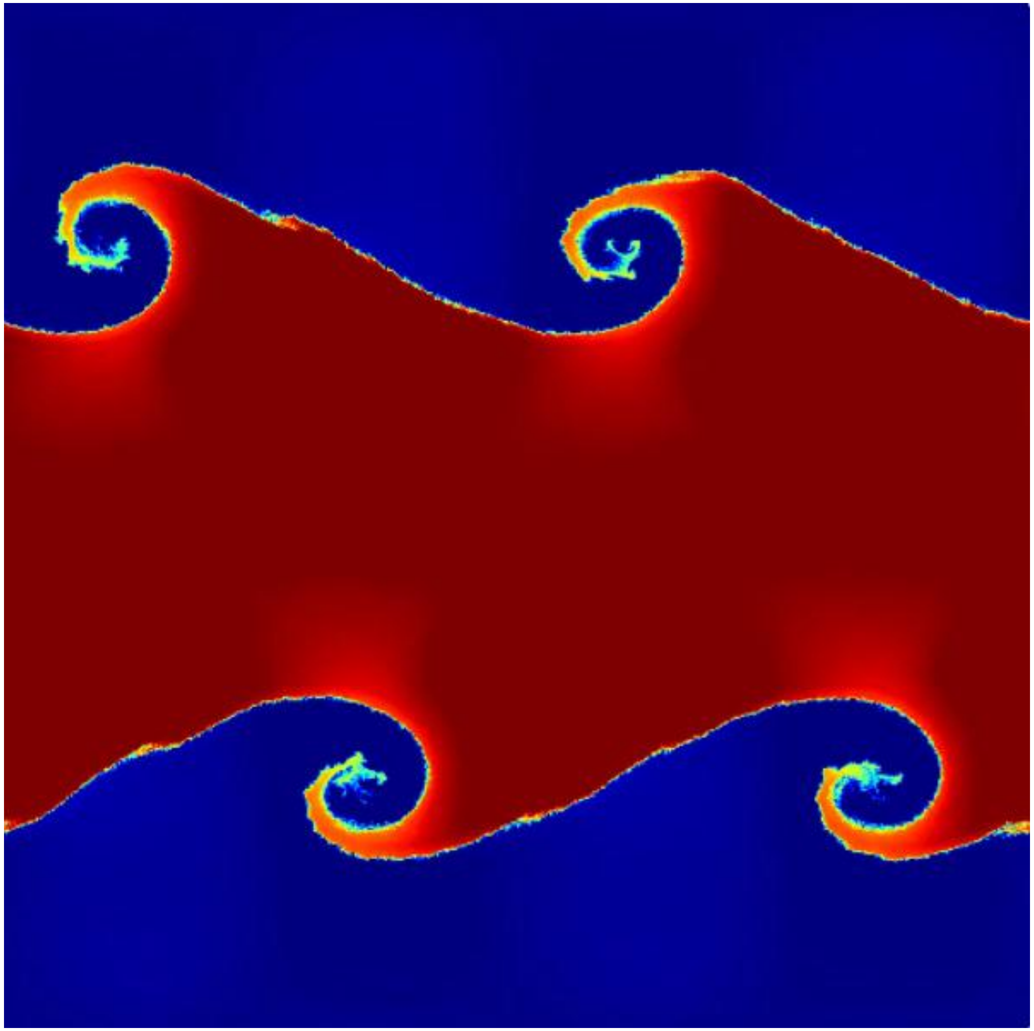


https://www.pinterest.com/flywithcptjoe/pretty-flying-pictures/



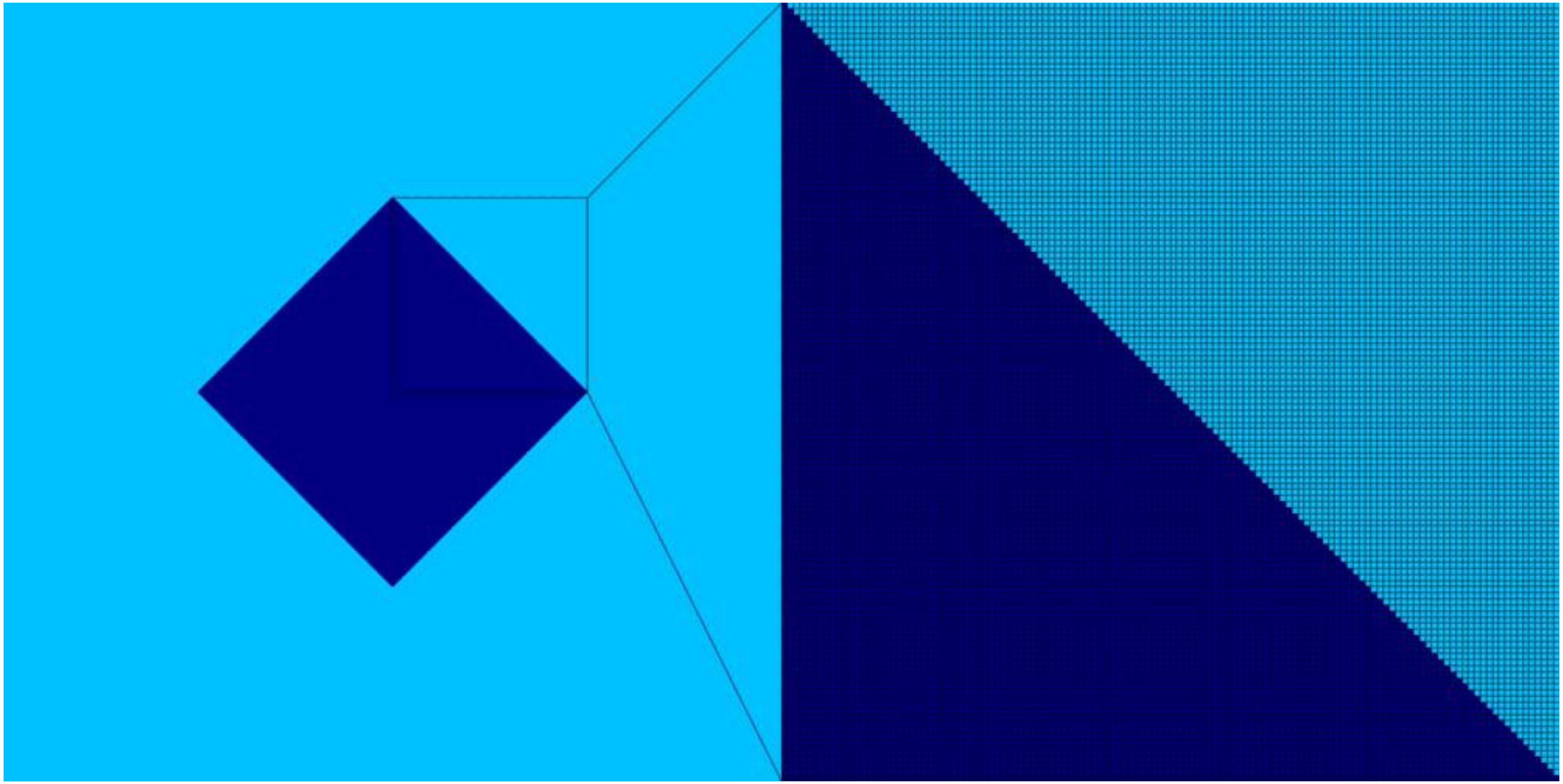
X-ray: NASA/CXC/Rutgers/J.Hughes; Optical: NASA/STScI

We need numerical simulations.



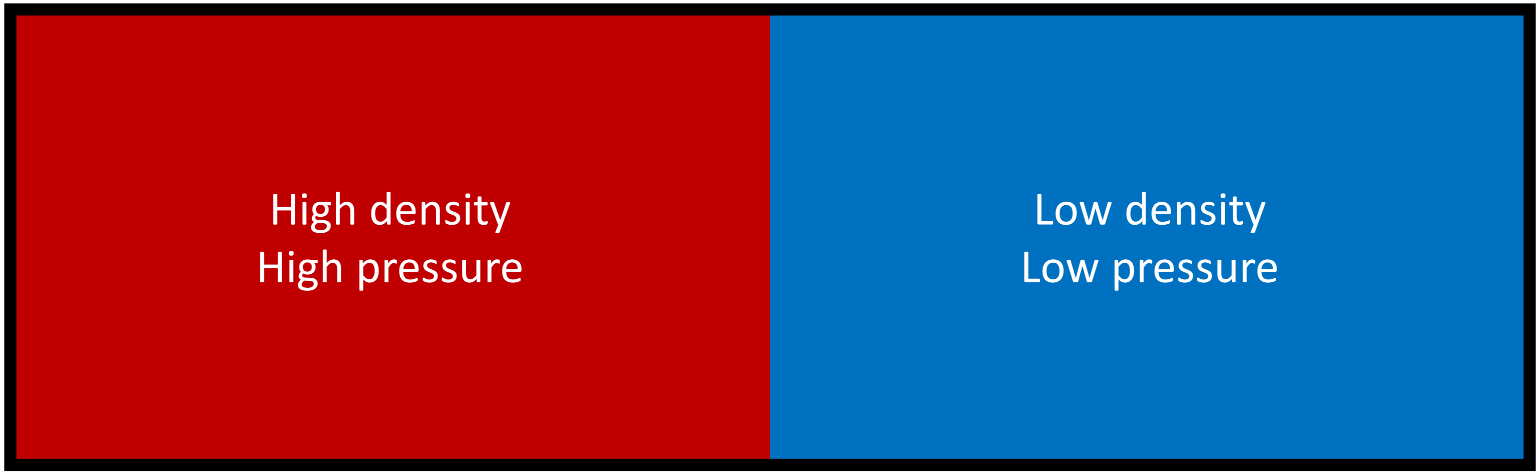
https://gitlab.cosma.dur.ac.uk/swift/swiftsim

Because they are awesome.



hadowfax

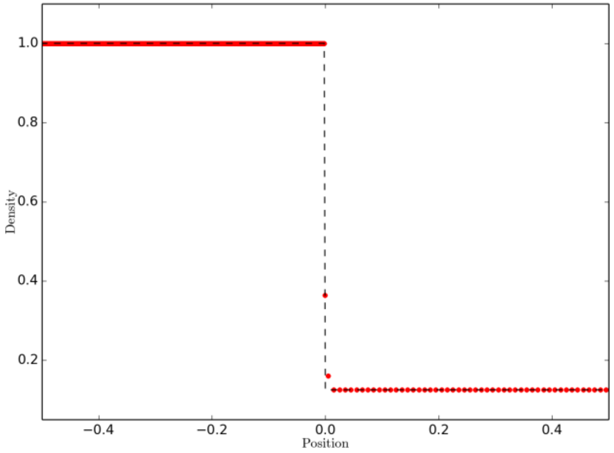
The 1D Sod shock tube



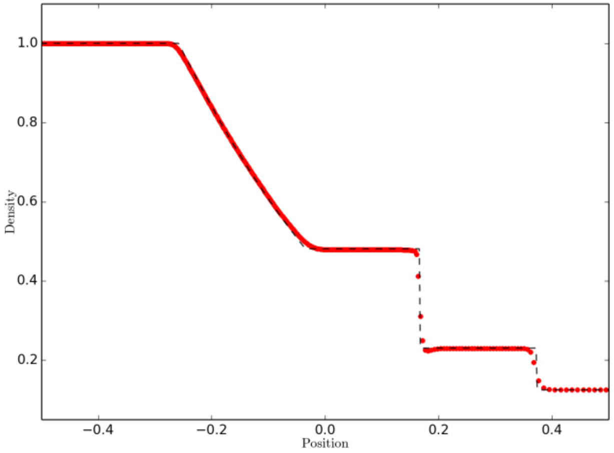
High density
High pressure

Low density
Low pressure

t = 0.0

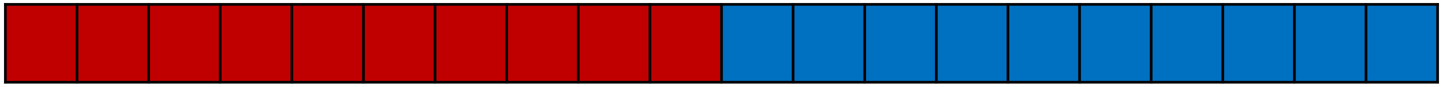


t = 0.2



Finite volume schemes: discretization in space

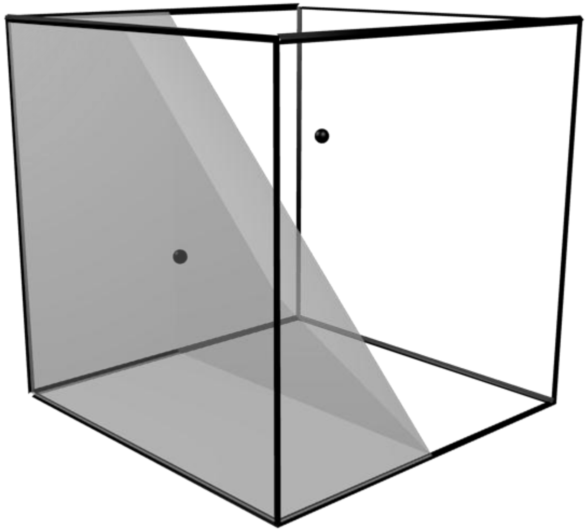
The 1D “volume” is subdivided into cells. Each cell holds a single value for the density, pressure, and fluid velocity.



We need to solve the equations of hydrodynamics for this discrete set of cells.

We will derive equations for a general, 3D, unstructured cell.

A 1D regular cell is just a special case of this more general cell shape.



We start from the Euler equations:

Continuity equation

$$\boxed{\frac{\partial \rho}{\partial t} + \underline{\nabla}.\left(\rho \underline{u}\right) = 0}$$

Momentum equation

$$\boxed{\rho \frac{\partial \underline{u}}{\partial t} + \rho\left(\underline{u}.\underline{\nabla}\right)\underline{u} = -\underline{\nabla} p + \rho \underline{g}}$$

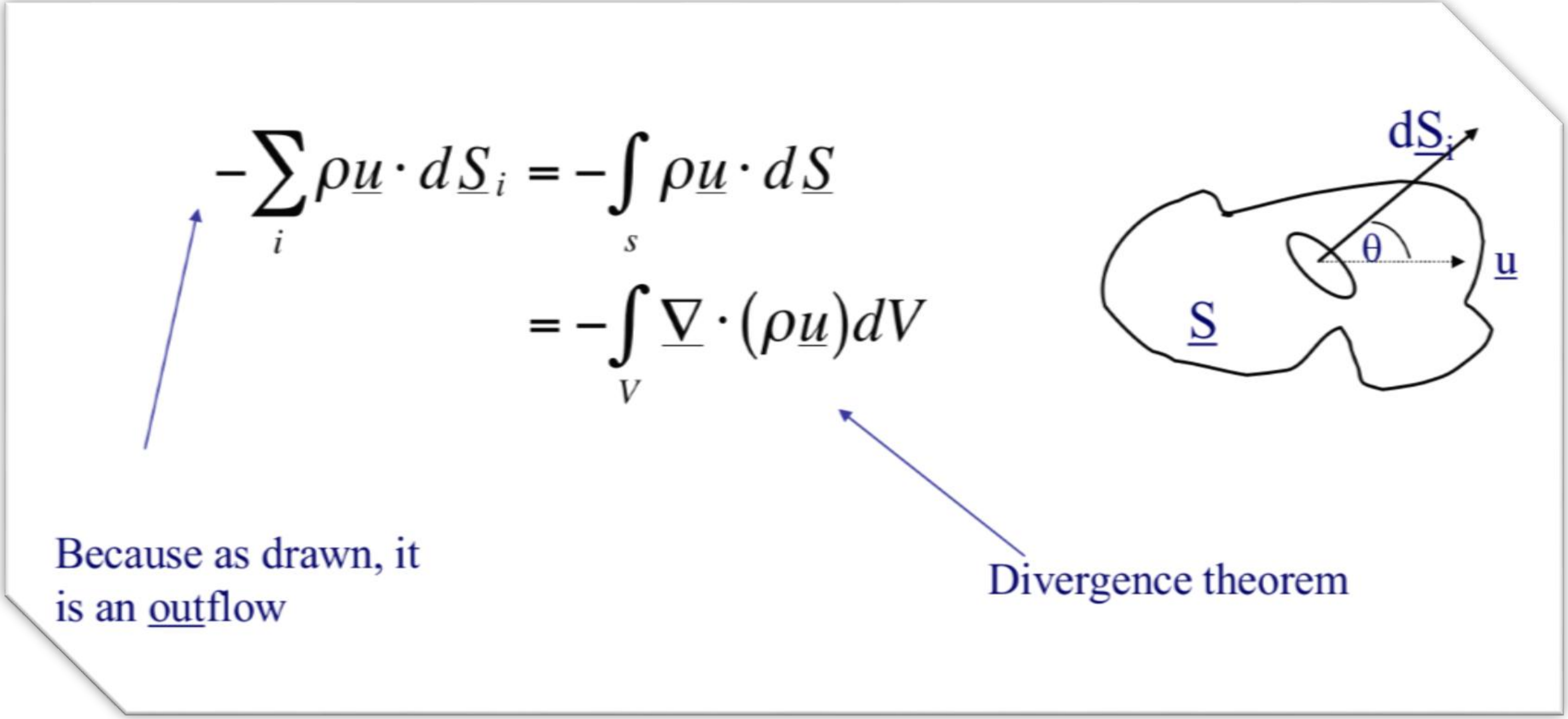
Energy equation

$$\boxed{\frac{\partial}{\partial t}\left(\frac{1}{2}\rho u^2 + \rho e\right) + \underline{\nabla}.\left(\frac{1}{2}\rho u^2 + \rho e\right)\underline{u} = -L - \underline{\nabla}.\left(p\underline{u}\right) - \rho \underline{u}.\underline{\nabla}\psi}$$

We will assume a *polytropic* equation of state: $p = (\gamma - 1)\rho e$

The Euler equations, you say?

Continuity equation

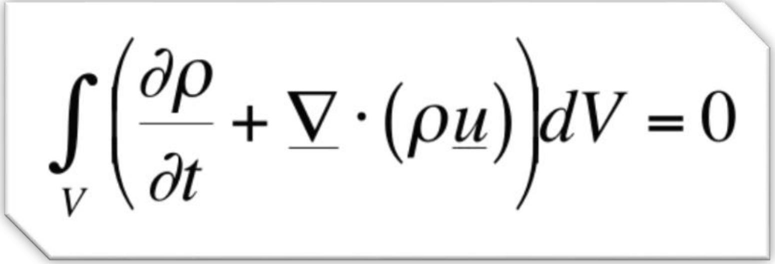


$$-\sum_i \rho\underline{u}\cdot d\underline{S}_i = -\int_s \rho\underline{u}\cdot d\underline{S}$$

$$= -\int_V \underline{\nabla}\cdot(\rho\underline{u})dV$$

Because as drawn, it is an outflow

Divergence theorem

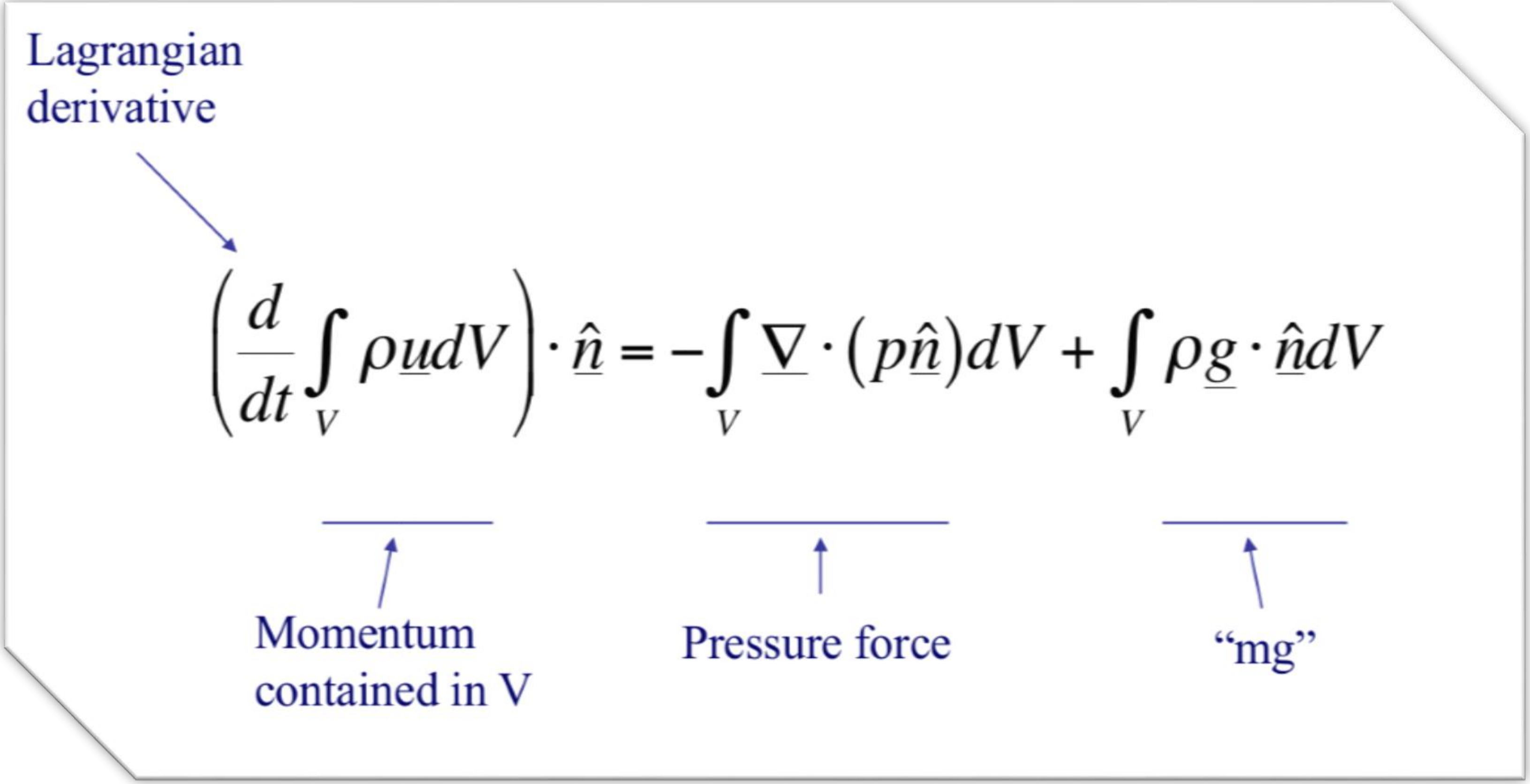


$$\int_V \left(\frac{\partial\rho}{\partial t} + \underline{\nabla}\cdot(\rho\underline{u})\right)dV = 0$$

Source: Moira's fluids course

The Euler equations, you say?

Momentum equation



Lagrangian derivative

$$\left(\frac{d}{dt}\int_V \rho \underline{u}\, dV\right)\cdot \underline{\hat{n}} = -\int_V \underline{\nabla}\cdot(p\underline{\hat{n}})\,dV + \int_V \rho \underline{g}\cdot \underline{\hat{n}}\,dV$$

Momentum contained in V

Pressure force

“mg”

Source: Moira's fluids course

The Euler equations, you say?

Energy equation

$$TdS = \underset{\substack{\nearrow \\ \text{Heat change}}}{dQ} = \underset{\substack{\nwarrow \\ \text{Internal energy change}}}{dU} + pdV$$

$$\Rightarrow \rho\frac{de}{dt} + p\underline{\nabla}\cdot\underline{u} = -L$$

$$\rho\frac{d}{dt}\left(\frac{1}{2}u^2\right) = -\underline{u}\cdot\underline{\nabla}p + \underline{u}\cdot\rho\underline{g}$$

$$\begin{aligned}\rho\frac{d}{dt}\left(\frac{1}{2}u^2 + e\right) &= -L - p\underline{\nabla}\cdot\underline{u} - \underline{u}\cdot\underline{\nabla}p + \underline{u}\cdot\rho\underline{g} \\ &= -L - \underline{\nabla}\cdot p\underline{u} + \underline{u}\cdot\rho\underline{g}\end{aligned}$$

Source: Moira's fluids course

$$\frac{\partial \rho}{\partial t} + \underline{\nabla}.\left(\rho \underline{u}\right) = 0$$

We consider pure hydro, so:

- no gravity
- no energy sources or sinks

$$\rho \frac{\partial \underline{u}}{\partial t} + \boxed{\rho(\underline{u}.\underline{\nabla})\underline{u}} = -\underline{\nabla} p + \cancel{\rho \underline{g}}$$

$$\frac{\partial}{\partial t}\left(\underbrace{\frac{1}{2}\rho u^2 + \rho e}_{\rho e_{tot}}\right) + \underline{\nabla}.\left(\underbrace{\frac{1}{2}\rho u^2 + \rho e}_{\rho e_{tot}}\right)\underline{u} = \cancel{-\ } - \underline{\nabla}.\left(p\underline{u}\right) - \rho \underline{u}.\cancel{\underline{\nabla}\Phi}$$

$$\boxed{\rho(\underline{u}.\underline{\nabla})\underline{u}} = \rho\underline{\nabla}.(\underline{u}\;\underline{u}) - \rho\underline{u}(\underline{\nabla}.\underline{u})$$

$$\frac{\partial\rho}{\partial t} + \underline{\nabla}.(\rho\underline{u}) = \frac{\partial\rho}{\partial t} + \rho(\underline{\nabla}.\underline{u}) + \underline{u}.\underline{\nabla}\rho = 0$$

$$\Rightarrow \quad \underline{\nabla}.\underline{u} = -\frac{1}{\rho}\frac{\partial\rho}{\partial t} - \frac{1}{\rho}\underline{u}.\underline{\nabla}\rho$$

$$\rho(\underline{u}.\underline{\nabla})\underline{u} = \rho\underline{\nabla}.(\underline{u}\;\underline{u}) + \underline{u}\frac{\partial\rho}{\partial t} + \underline{u}\;\underline{u}.\underline{\nabla}\rho$$

$$= \underline{\nabla}.(\rho\underline{u}\;\underline{u}) + \underline{u}\frac{\partial\rho}{\partial t}$$

$$\rho\frac{\partial \underline{u}}{\partial t} + \underline{\nabla}.\left(\rho \underline{u}\,\underline{u}\right) + \underline{u}\frac{\partial \rho}{\partial t} = -\underline{\nabla}p$$

Momentum equation’

$$\boxed{\frac{\partial \rho \underline{u}}{\partial t} + \underline{\nabla}.\left(\rho \underline{u}\,\underline{u}\right) = \boxed{-\underline{\nabla}p}} \longrightarrow \underline{\nabla}p = \frac{\partial p}{\partial x}\underline{i} + \frac{\partial p}{\partial y}\underline{j} + \frac{\partial p}{\partial z}\underline{k}$$

Note for a general tensor

$$\underline{\underline{A}} = A_{11}\underline{i}\,\underline{i} + A_{12}\underline{i}\,\underline{j} + A_{13}\underline{i}\,\underline{k} + \cdots$$

The divergence is given by

$$\underline{\nabla}.\underline{\underline{A}} = \left(\frac{\partial A_{11}}{\partial x} + \frac{\partial A_{21}}{\partial y} + \frac{\partial A_{31}}{\partial z}\right)\underline{i} + \cdots$$

Hence, we can also write the gradient of a scalar as a divergence of a tensor:

$$\underline{\nabla}p = \underline{\nabla}.\left(p\underline{\underline{1}}\right) = \underline{\nabla}.\begin{pmatrix} p & 0 & 0 \\ 0 & p & 0 \\ 0 & 0 & p \end{pmatrix}$$

Hence, we find:

$$\boxed{\frac{\partial \rho \underline{u}}{\partial t} + \underline{\nabla}.\left(\rho \underline{u}\,\underline{u} + p\underline{\underline{1}}\right) = 0}$$

$$\boxed{\frac{\partial \rho \underline{u}}{\partial t} + \underline{\nabla}.\left(\rho \underline{u}\;\underline{u} + p\underline{\underline{1}}\right) = 0}$$

This is a *vector equation*, and is shorthand for:

$$\boxed{\begin{aligned}
&\frac{\partial \rho u_x}{\partial t} + \frac{\partial}{\partial x}(\rho u_x u_x) + \frac{\partial}{\partial y}(\rho u_y u_x) + \frac{\partial}{\partial z}(\rho u_z u_x) + \frac{\partial p}{\partial x} = 0 \\
&\frac{\partial \rho u_y}{\partial t} + \frac{\partial}{\partial x}(\rho u_x u_y) + \frac{\partial}{\partial y}(\rho u_y u_y) + \frac{\partial}{\partial z}(\rho u_z u_y) + \frac{\partial p}{\partial y} = 0 \\
&\frac{\partial \rho u_z}{\partial t} + \frac{\partial}{\partial x}(\rho u_x u_z) + \frac{\partial}{\partial y}(\rho u_y u_z) + \frac{\partial}{\partial z}(\rho u_z u_z) + \frac{\partial p}{\partial z} = 0
\end{aligned}}$$

This gives us the Euler equations in *conservative form*:

$$\boxed{\frac{\partial\rho}{\partial t}+\underline{\nabla}.\left(\rho\underline{u}\right)=0}$$

$$\boxed{\frac{\partial\rho\underline{u}}{\partial t}+\underline{\nabla}.\left(\rho\underline{u}\,\underline{u}+p\underline{\underline{1}}\right)=0}$$

$$\boxed{\frac{\partial\rho e_{tot}}{\partial t}+\underline{\nabla}.\left(\rho e_{tot}\underline{u}+p\underline{u}\right)=0}$$

All equations together can be written in the more compact form

$$\frac{\partial U}{\partial t}+\underline{\nabla}.\underline{F}(U)=0$$

with

$$U=\begin{pmatrix}\rho\\ \rho\underline{u}\\ \rho e_{tot}\end{pmatrix},\quad \underline{F}(U)=\begin{pmatrix}\rho\underline{u}\\ \rho\underline{u}\,\underline{u}+p\underline{\underline{1}}\\ \rho e_{tot}\underline{u}+p\underline{u}\end{pmatrix}$$

We call this a *conservation law*

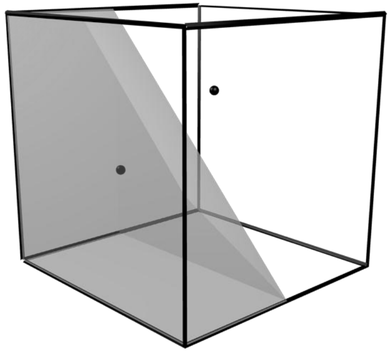
Note that the fluxes $\underline{F}(U)$ can be expressed in terms of the *primitive* variables:

- Density ρ

- Fluid velocity $\underline{u}$

- Pressure $p=(\gamma-1)\rho e=(\gamma-1)\left(\rho e_{tot}-\frac{1}{2}\rho u^2\right)$

Conservation law?



Integrate the conservation law over an arbitrary volume V

$$\iiint\left(\frac{\partial U}{\partial t}+\underline{\nabla}.\underline{F}(U)\right)\mathrm{d}V=0$$

Conserved quantities

We get

if V is independent of t

$$\iiint\frac{\partial U}{\partial t}\mathrm{d}V=\frac{\partial}{\partial t}\iiint U\mathrm{d}V=\frac{\partial}{\partial t}\iiint\begin{pmatrix}\rho\\ \rho\underline{u}\\ \rho e_{tot}\end{pmatrix}\mathrm{d}V\equiv\frac{\partial}{\partial t}\begin{pmatrix}m\\ m\underline{u}\\ E_{tot}\end{pmatrix}=\frac{\partial Q}{\partial t}$$

and

if $\underline{F}(U)$ is continuous over V

$$\iiint\underline{\nabla}.\underline{F}(U)\mathrm{d}V=\iint\underline{F}(U).\mathrm{d}\underline{S}$$

$\underline{S}$ is the boundary surface of V

This means that the change of the conserved quantities Q is given by the fluxes $\underline{F}(U)$ through the surface $\underline{S}$ of the volume V:

$$\frac{\partial Q}{\partial t}=-\iint\underline{F}(U).\mathrm{d}\underline{S}$$

Since this holds for an arbitrary volume, we can apply the same logic to a set of discrete cells i with volumes V_i and boundary surfaces $\underline{S_i}$:

$$\iiint \left(\frac{\partial U}{\partial t} + \underline{\nabla}.\underline{F}(U)\right) \mathrm{d}V_i = 0$$

to get:

$$\frac{\partial Q_i}{\partial t} = -\iint \underline{F}(U).\,\mathrm{d}\underline{S_i} \qquad\qquad Q_i = \begin{pmatrix} m_i \\ m_i\underline{u_i} \\ E_{tot,i} \end{pmatrix}$$

The change in conserved cell quantities Q_i is given by the flux through the boundary surface of the cell.

Since an outflux in cell i corresponds to an influx in another cell j, the entire integration scheme is reduced to an *exchange of fluxes* between cells.

http://woolshed1.blogspot.co.uk/2010/02/sheep-yards-design-and-construction.html

Note that the entire derivation until now was *exact*, if we can guarantee that the fluid quantities are continuous inside a cell, and that the cell volume does not change with time.

$$\frac{\partial Q_i}{\partial t} = -\iint \underline{F}(U).\,\mathrm{d}\underline{S_i}$$

To convert the equation above into a useful numerical method, we will need to make some *approximations*

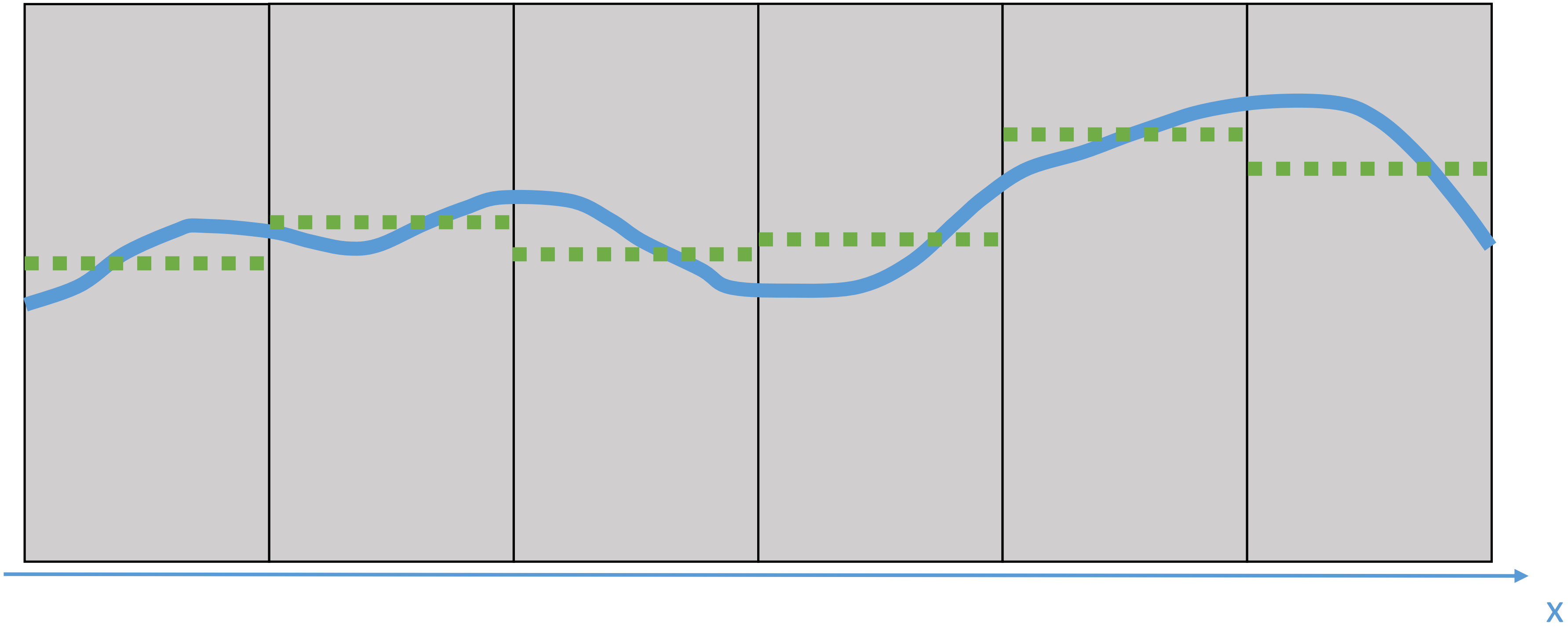
about how to obtain useful fluxes

about how to approximate the surface integral

about how to deal with discontinuities in the fluid quantities
(shocks, contact discontinuities)

about how to discretize time

Obtaining useful fluxes



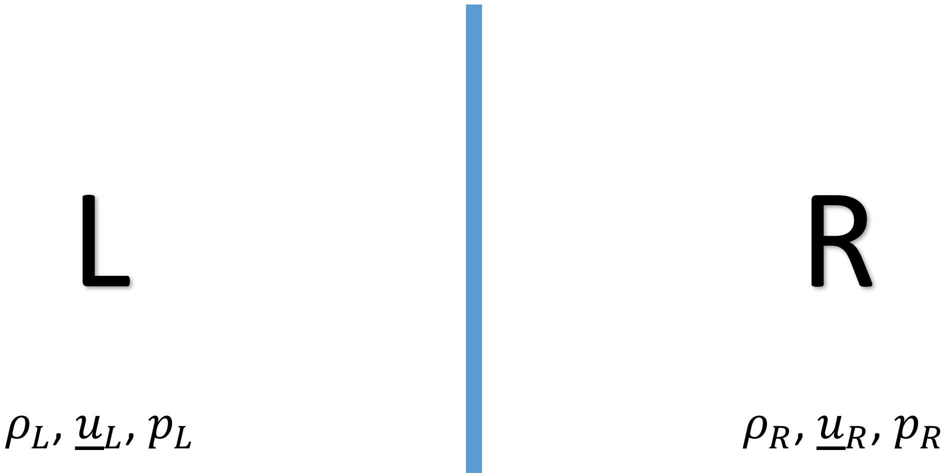
We need to evaluate the fluxes at the boundary between cells.

However, our integration scheme will only be *stable* if we respect the natural flow direction of the fluid, a so called *upwind* scheme.

This means we cannot simply use the average of the fluid quantities on both sides of the boundary.

Obtaining useful fluxes: the Riemann problem

What is the natural flow direction in a two state problem?



L

R

$\rho_L, \underline{u}_L, p_L$

$\rho_R, \underline{u}_R, p_R$

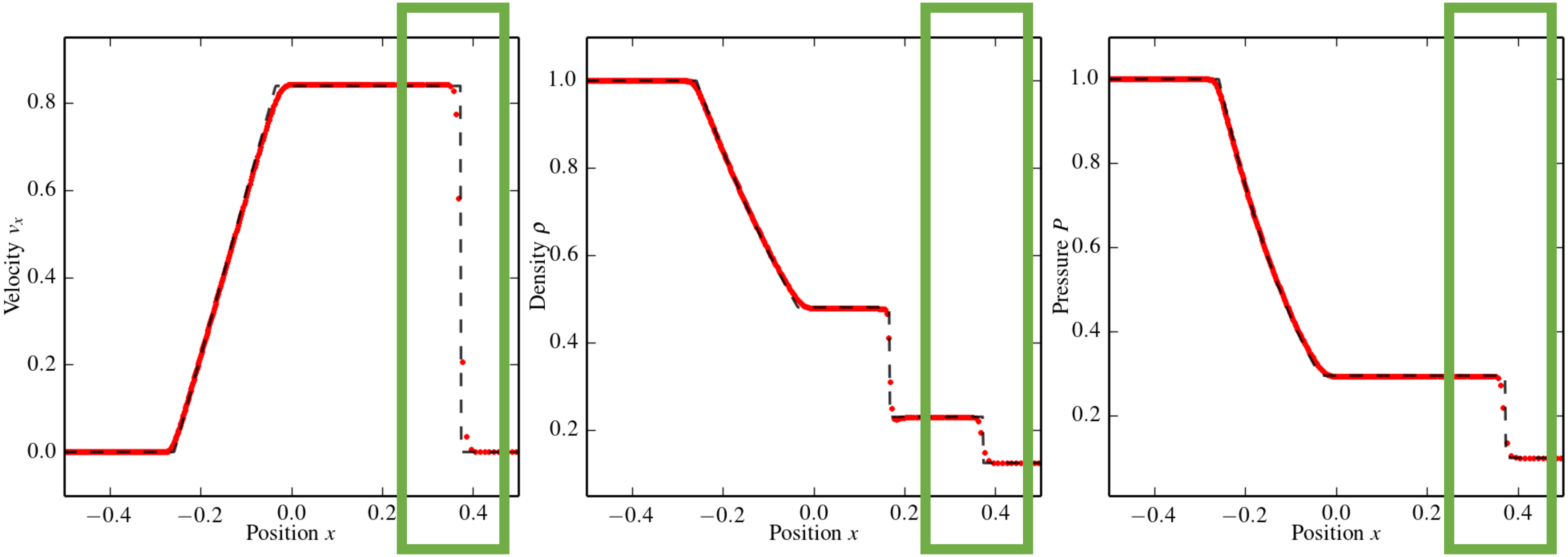
Physical acceptable solutions contain three distinct waves:

- a left rarefaction or shock wave

- a central contact discontinuity

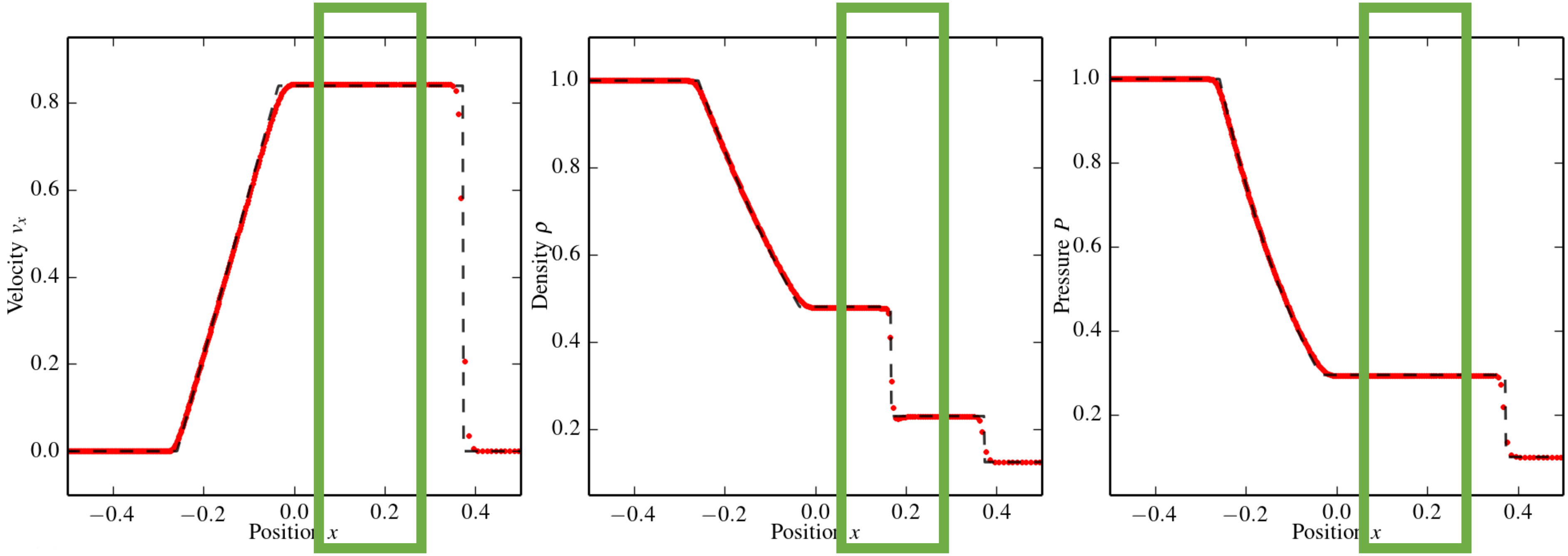
- a right rarefaction or shock wave

Wave types: shock wave



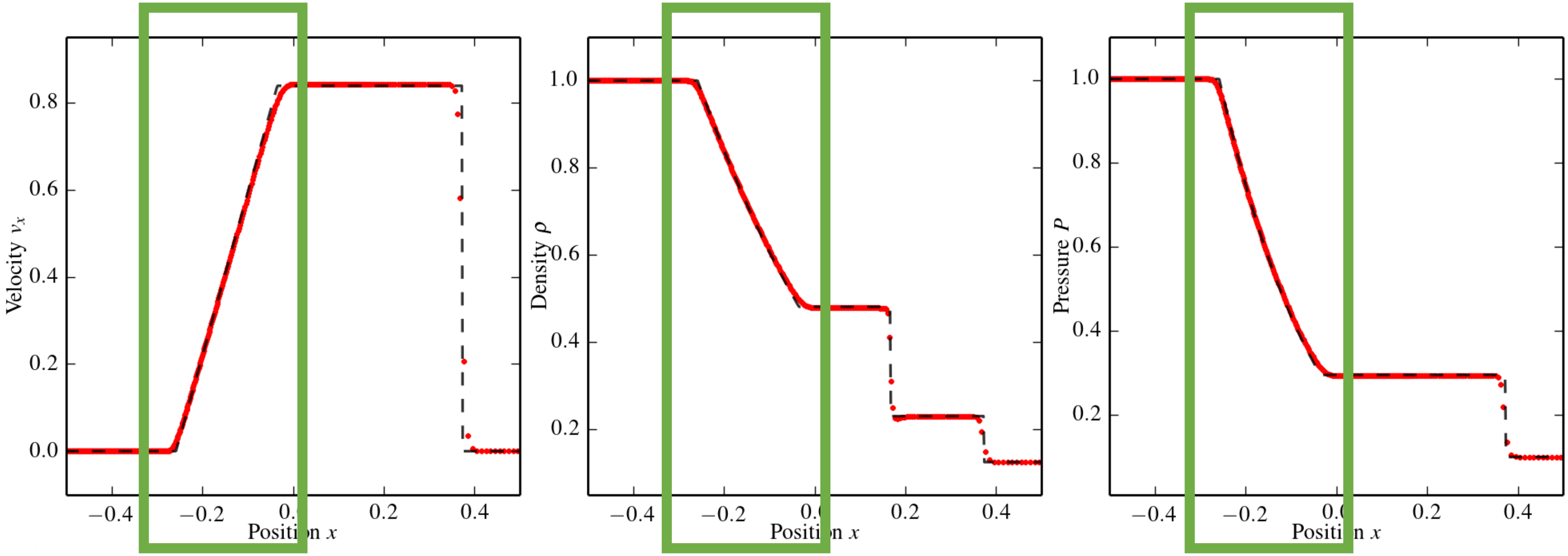
Strong discontinuity in density, pressure and fluid velocity.
Supersonic movement caused by a strong pressure or velocity gradient.

Wave types: contact discontinuity



Discontinuity in density, pressure and fluid velocity are constant.
Moving with the local fluid velocity.

Wave types: rarefaction wave



Continuous change in density, pressure and fluid velocity.
Moving with the sound speed relative w.r.t. the local fluid velocity.
Normal flow, described by the *Euler equations*.

Obtaining useful fluxes: the Riemann problem

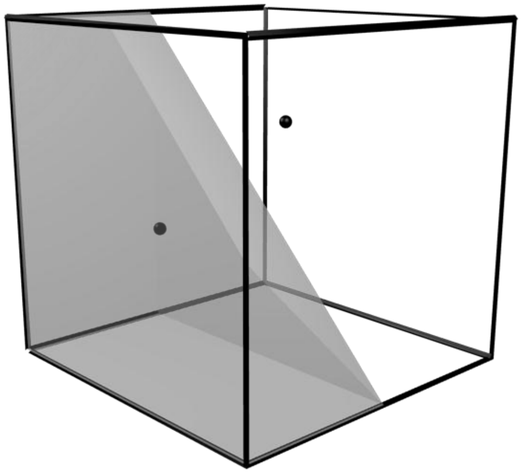
Exact solution to Riemann problem can be found using an iterative procedure

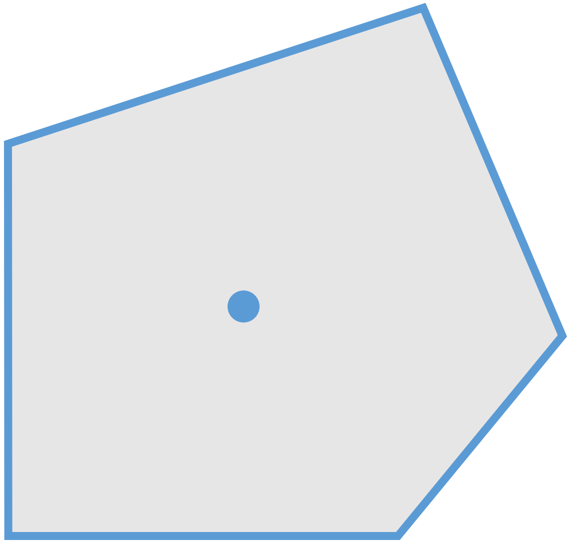
Good approximate solvers exist that do not require an iteration (more efficient)

The solution of the Riemann problem is then used to obtain fluxes that respect the natural direction of the flow.

Since the Riemann problem explicitly deals with discontinuities, our integration scheme can handle discontinuous fluid quantities, as long as they coincide with cell boundaries!

Approximating the surface integral





The integral over the entire cell boundary surface is a sum over boundaries between the cell and its neighbouring cells:

$$\frac{\partial Q_i}{\partial t} = -\iint \underline{F}(U).\,\mathrm{d}\underline{S_i} \approx -\sum_j \underline{F}_{ij}.\,\underline{A}_{ij}$$

Time integration

A first order accurate integration scheme is a simple Euler scheme:

$$\Delta Q_i = -\left(\sum_j \underline{F}_{ij} . \underline{A}_{ij}\right)\Delta t$$

If we use upwind fluxes, this scheme will be conditionally stable, if

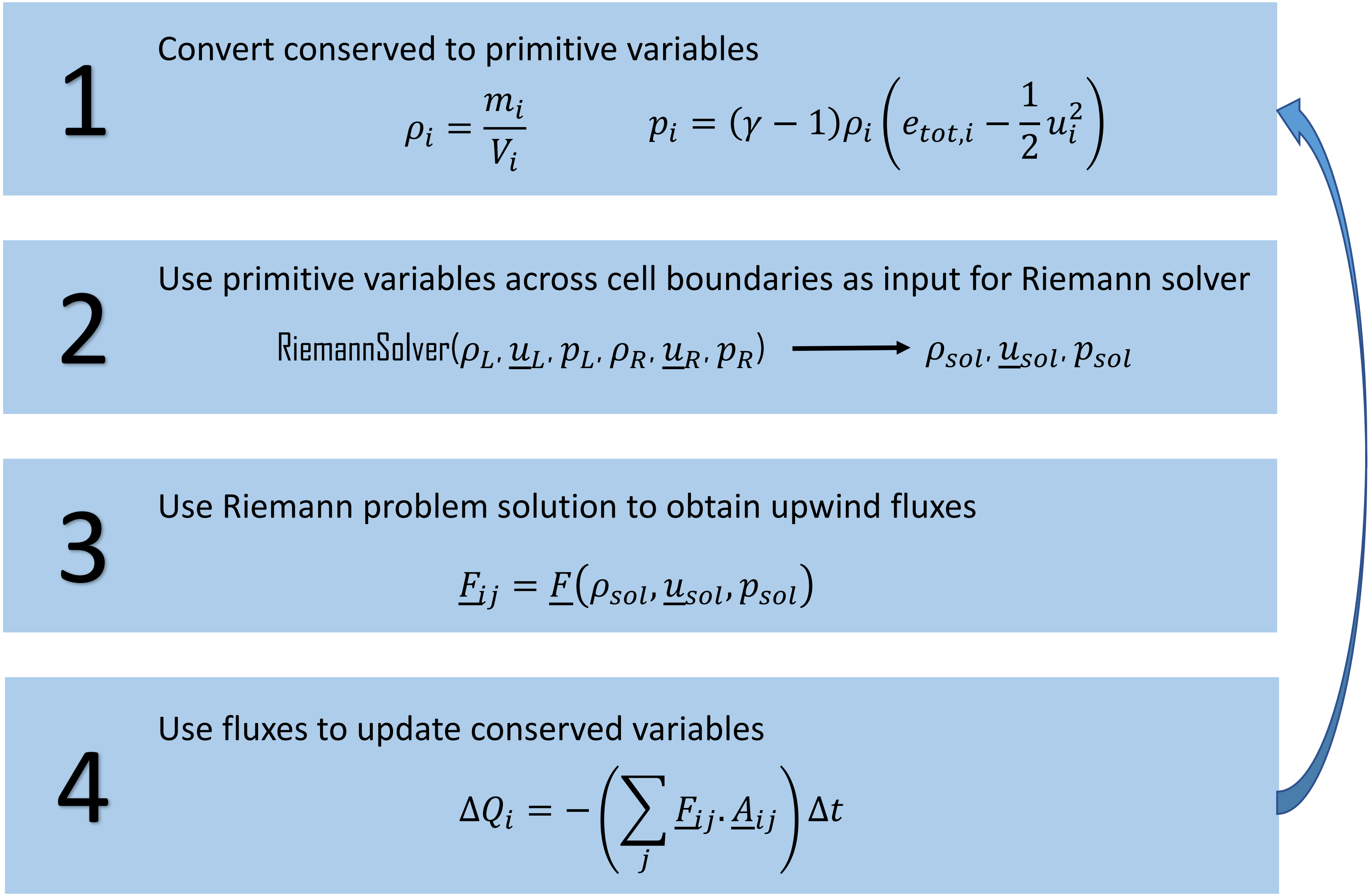
$$\Delta \mathrm{t} < \mathrm{C}_{\mathrm{CFL}} \frac{\Delta x}{v_{signal}}$$

Δx is the size of a single cell

v_{signal} is the velocity of the flow

C_{CFL} is the Courant-Friedrichs-Lewy number, $0 < C_{CFL} < 1$ for a stable solution

Summary: first order finite volume scheme



1

Convert conserved to primitive variables

$$\rho_i = \frac{m_i}{V_i} \qquad p_i = (\gamma - 1)\rho_i\left(e_{tot,i} - \frac{1}{2}u_i^2\right)$$

2

Use primitive variables across cell boundaries as input for Riemann solver

$$\text{RiemannSolver}(\rho_L, \underline{u}_L, p_L, \rho_R, \underline{u}_R, p_R) \longrightarrow \rho_{sol}, \underline{u}_{sol}, p_{sol}$$

3

Use Riemann problem solution to obtain upwind fluxes

$$\underline{F}_{ij} = \underline{F}\left(\rho_{sol}, \underline{u}_{sol}, p_{sol}\right)$$

4

Use fluxes to update conserved variables

$$\Delta Q_i = -\left(\sum_j \underline{F}_{ij}.\underline{A}_{ij}\right)\Delta t$$

Example Python code:

```
# the cell class
class Cell:

    def __init___(self):
        self._volume = 0.
        self._mass = 0.
        self._momentum = 0.
        self._energy = 0.
        self._density = 0.
        self._velocity = 0.
        self._pressure = 0.

        self._right_ngb = None
        self._surface_area = 1.
```

```
# the constant adiabatic index
GAMMA = 5./3.
# the constant time step
timestep = 0.001
```

```
# set up the cells
cells = []
for i in range(100):

    cell = Cell()
    cell._volume = 0.01
    if i < 50:
        cell._mass = 0.01
        cell._energy = 0.01 / (GAMMA - 1.)
    else:
        cell._mass = 0.00125
        cell._energy = 0.001 / (GAMMA - 1.)
    cell._momentum = 0.

    # set the neighbour of the previous cell
    cells[-1]._right_ngb = cell

    cells.append( cell )
```

1

Convert conserved to primitive variables

$$\rho_i = \frac{m_i}{V_i} \qquad p_i = (\gamma - 1)\rho_i \left(e_{tot,i} - \frac{1}{2} u_i^2 \right)$$

Example Python code:

```
for cell in cells:
    volume = cell._volume
    mass = cell._mass
    momentum = cell._momentum
    energy = cell._energy

    density = mass / volume
    velocity = momentum / mass
    pressure = (GAMMA – 1) * (energy / volume – 0.5 * density * velocity * velocity)

    cell._density = density
    cell._velocity = velocity
    cell._pressure = pressure
```

2 Use primitive variables across cell boundaries as input for Riemann solver

$$\text{RiemannSolver}(\rho_L, \underline{u}_L, p_L, \rho_R, \underline{u}_R, p_R) \longrightarrow \rho_{sol}, \underline{u}_{sol}, p_{sol}$$

Example Python code:

```
for cell in cells:
    cell_right = cell._right_ngb

    densityL = cell._density
    velocityL = cell._velocity
    pressureL = cell._pressure

    densityR = cell_right._density
    velocityR = cell_right._velocity
    pressureR = cell_right._pressure

    densitysol, velocitysol, pressuresol = RiemannSolver(densityL, velocityL, pressureL,
                                                         densityR, velocityR, pressureR)

    # ... continued in step 3 ...
```

3

Use Riemann problem solution to obtain upwind fluxes

$$\underline{F}_{ij} = \underline{F}(\rho_{sol}, \underline{u}_{sol}, p_{sol})$$

Example Python code:

```
# for cell in cells:
# ...

flux_mass = densitysol * velocitysol
flux_momentum = densitysol * velocitysol * velocitysol + pressuresol
flux_energy = (pressuresol * GAMMA / (GAMMA – 1) + \
               0.5 * densitysol * velocitysol * velocitysol) * velocitysol

# ... continued in step 4 ...
```

4

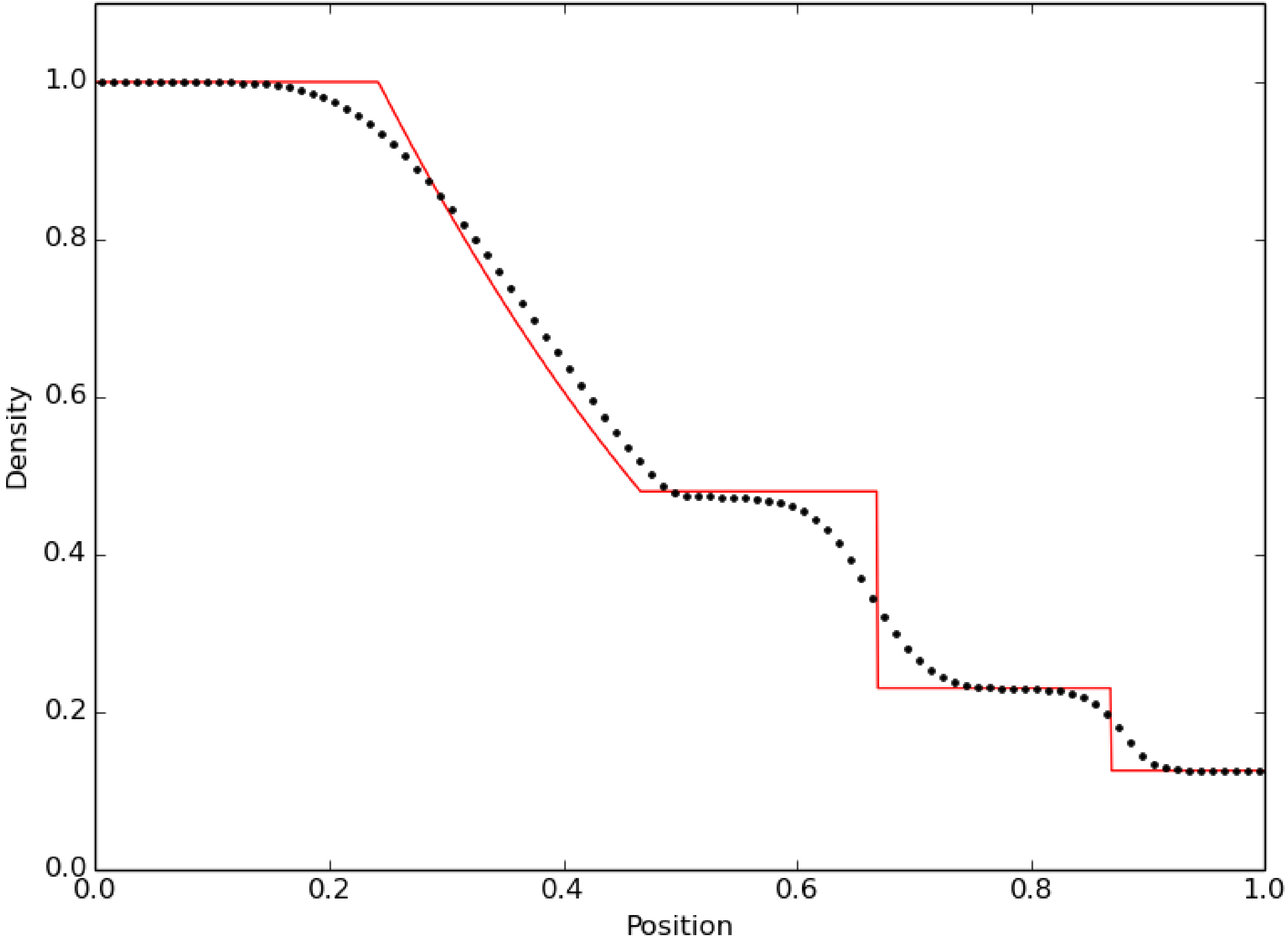
Use fluxes to update conserved variables

$$\Delta Q_i = -\left(\sum_j \underline{F}_{ij} . \underline{A}_{ij}\right) \Delta t$$

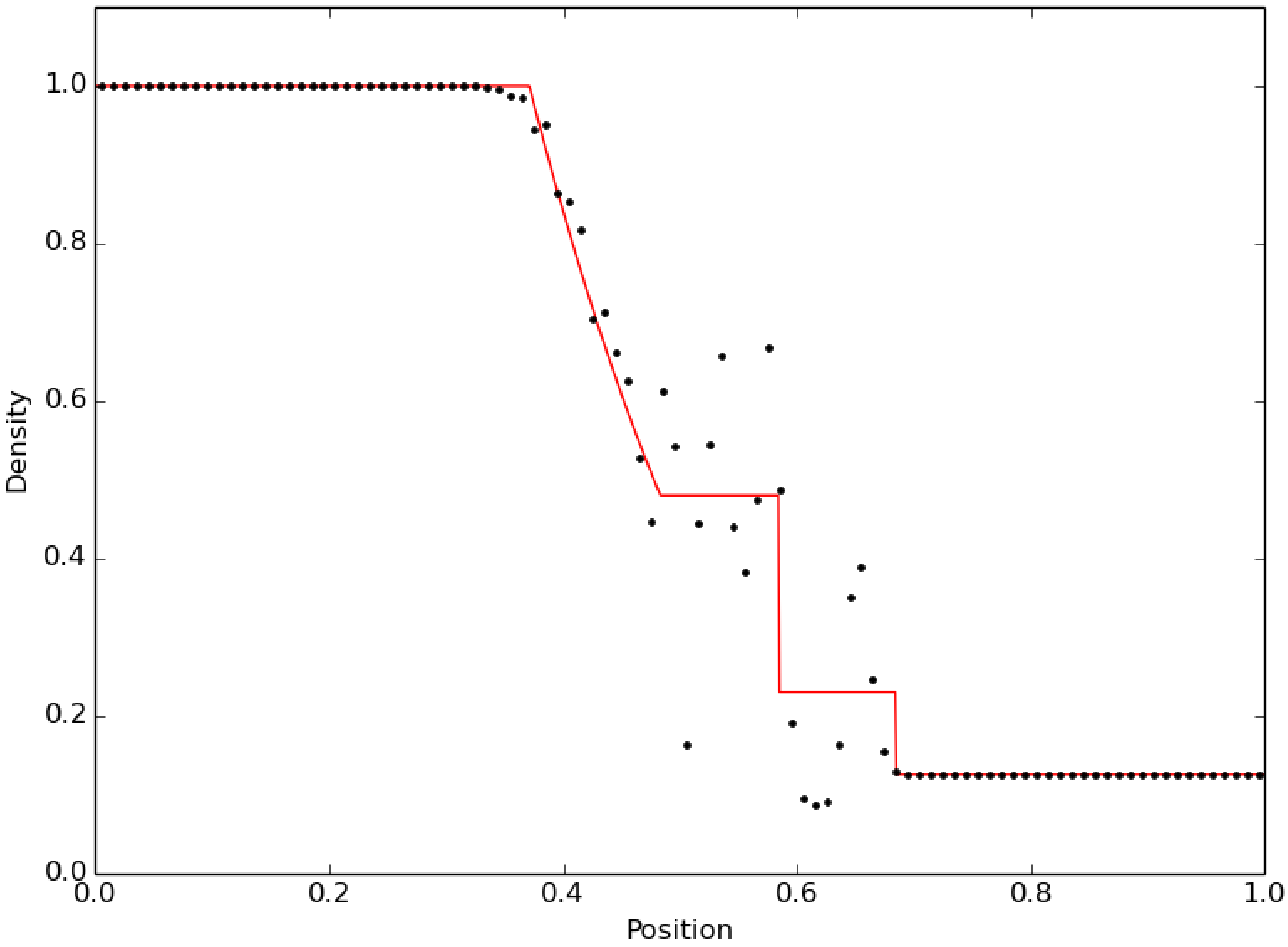
Example Python code:

```
# for cell in cells:
# …

A = cell._surface_area

cell._mass = cell._mass – flux_mass * A * timestep
cell._momentum = cell._momentum – flux_momentum * A * timestep
cell._energy = cell._energy – flux_energy * A * timestep

# flux exchange: what goes out of cell has to go into cell_right
cell_right._mass = cell_right._mass + flux_mass * A * timestep
cell_right._momentum = cell_right._momentum + flux_momentum * A * timestep
cell_right._energy = cell_right._energy + flux_energy * A * timestep
```

Reference solution at t = 0.2 (after 200 steps)



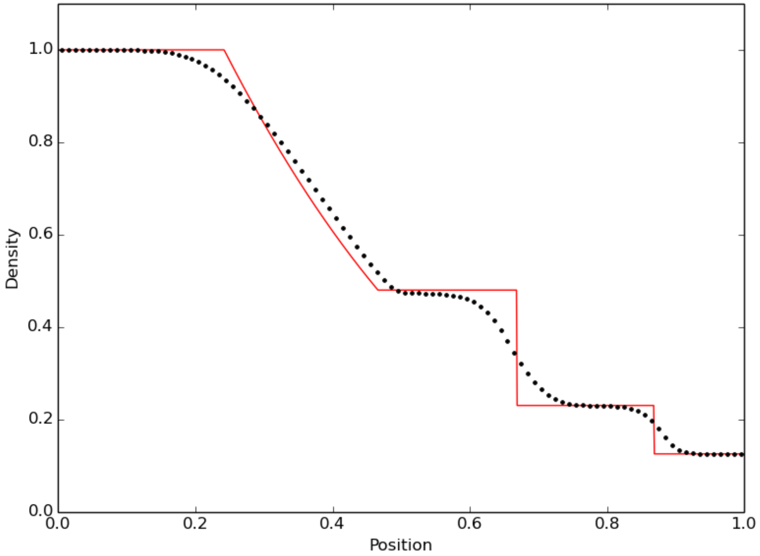
Solution without Riemann solver at t = 0.1



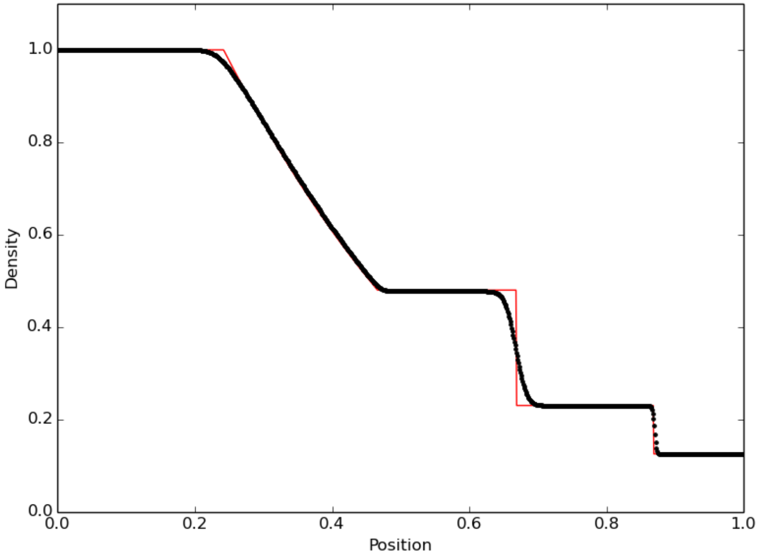
EXTRA: Improving the result

The method we have seen so far is first order in space and time.
This means that the accuracy of the solution improves linearly with the number of cells used to discretize the fluid.

100 cells

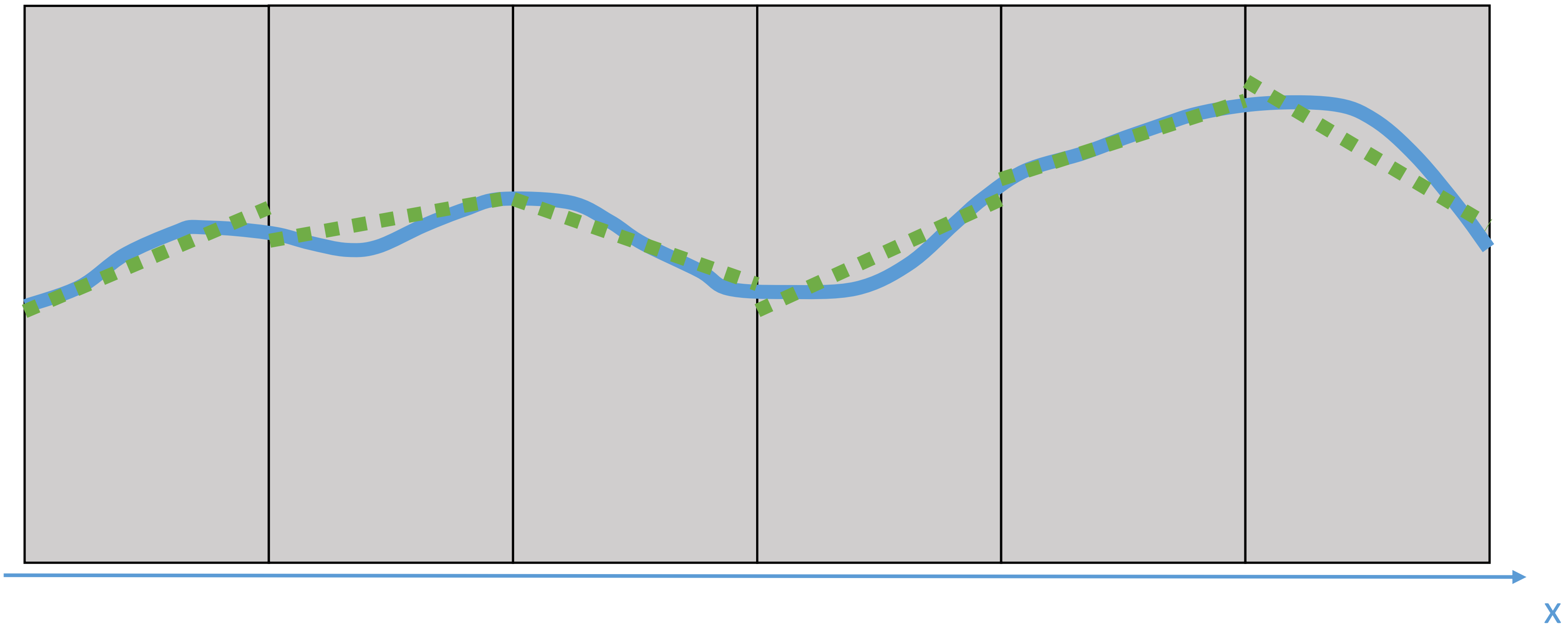


1000 cells



We can do better by using a second order method.

Second order schemes



Use linear interpolation to obtain more accurate values at cell boundaries

Requires the computation of *gradients* across cells

Second order in space

Second order schemes

Gradients can also be used to predict variables forward in time:

$$\rho' = \rho - \frac{1}{2}\Delta t\left(\rho\underline{\nabla}.\underline{u} + \underline{u}.\underline{\nabla}\rho\right)$$

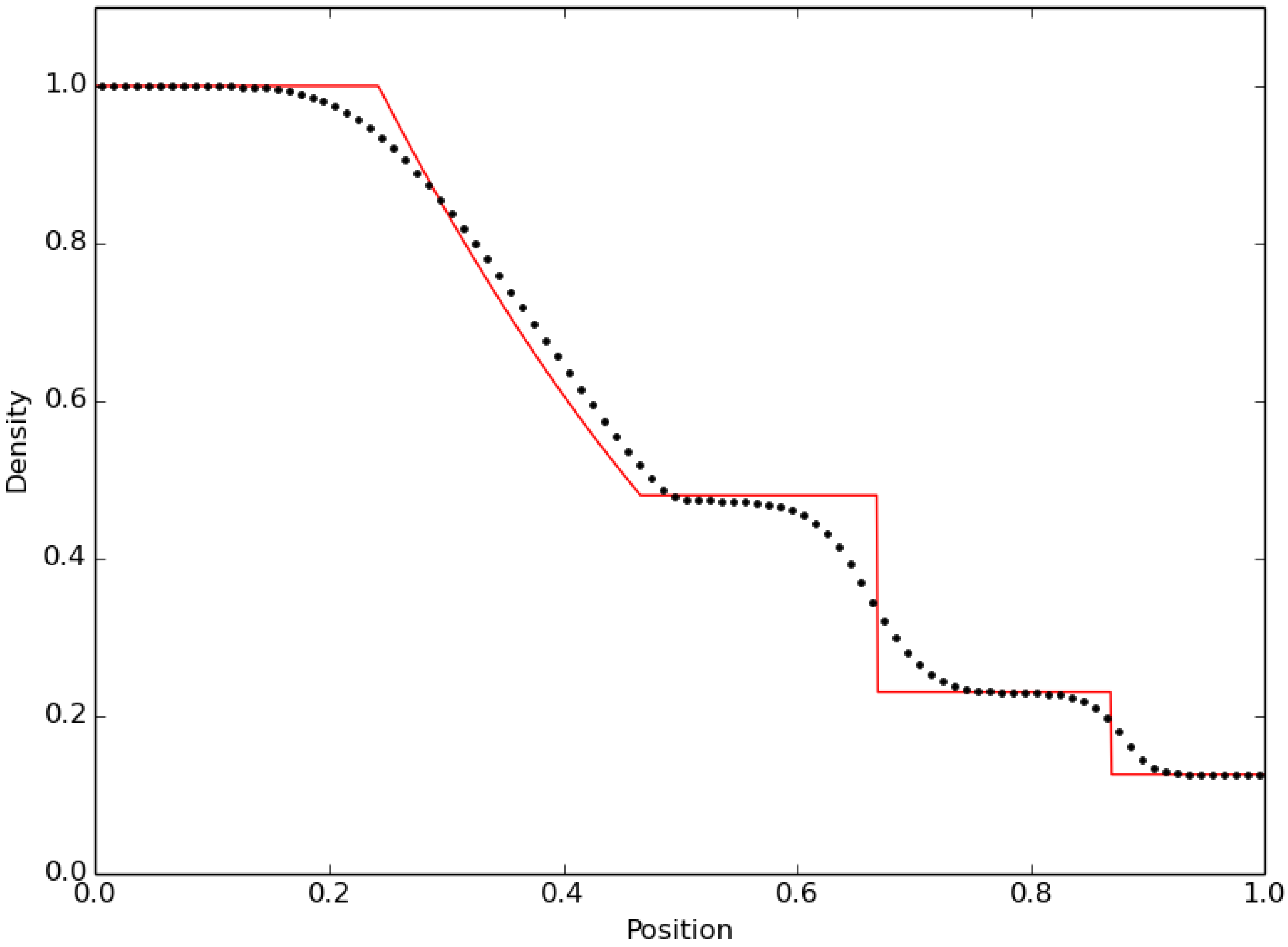
$$\underline{u}' = \underline{u} - \frac{1}{2}\Delta t\left(\underline{u}\,\underline{\nabla}.\underline{u} + \frac{1}{\rho}\underline{\nabla}p\right)$$

(these are just the Euler equations)

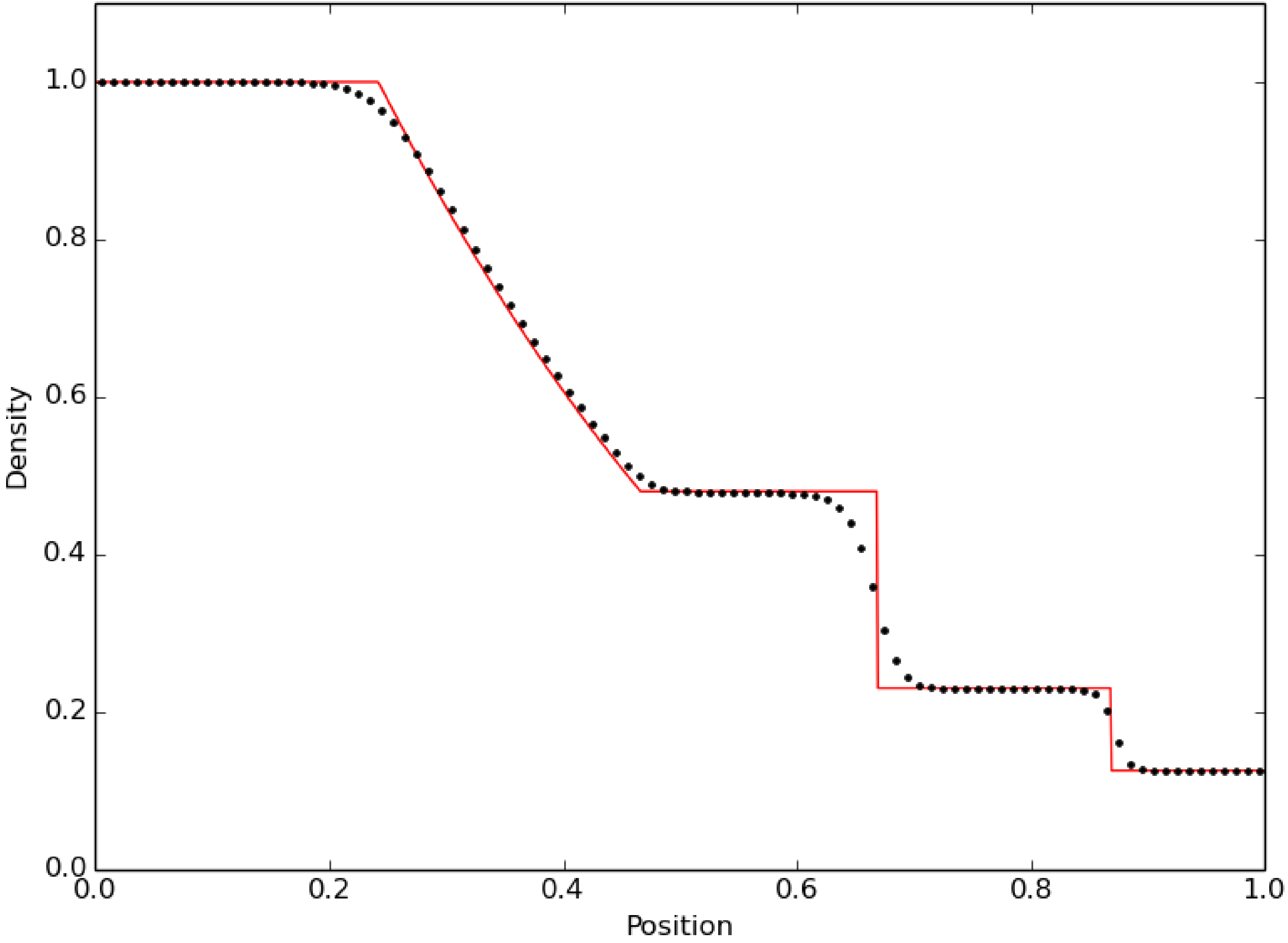
$$p' = p - \frac{1}{2}\Delta t\left(\gamma p\,\underline{\nabla}.\underline{u} + \underline{u}.\underline{\nabla}p\right)$$

Second order in time

Reference solution at t = 0.2 (first order)



Reference solution at t = 0.2 (second order)



Lagrangian schemes

We have imposed no restrictions on the shape of a cell

However, we have required the cell volume to be independent of time

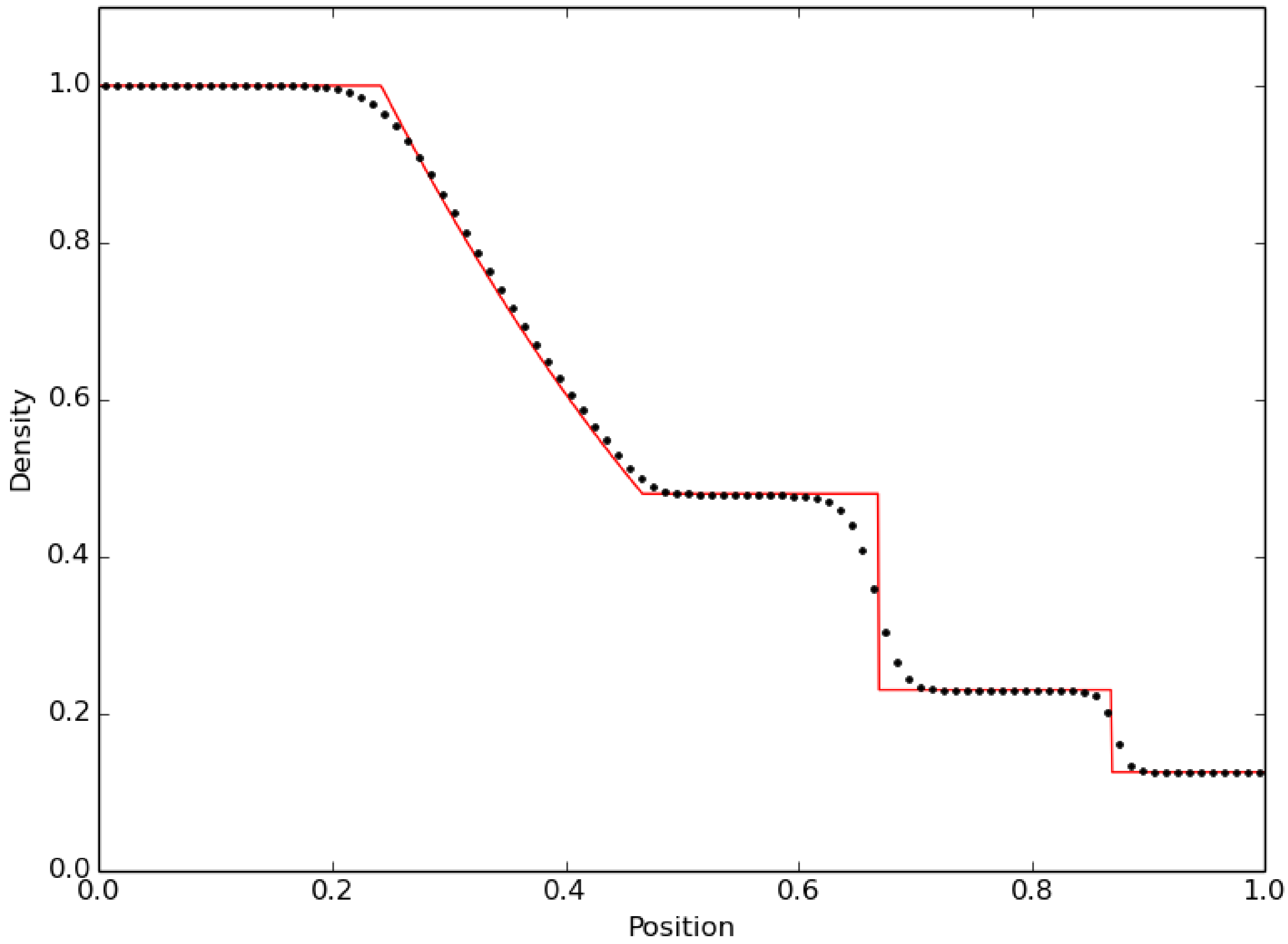
This requirement can be relaxed by adding correction terms to the fluxes:

$$\underline{F}(U) = \begin{pmatrix} \rho(\underline{u} - \underline{w}) \\ \rho\underline{u}(\underline{u} - \underline{w}) + p\underline{\underline{1}} \\ \rho e_{tot}(\underline{u} - \underline{w}) + p\underline{u} \end{pmatrix}$$

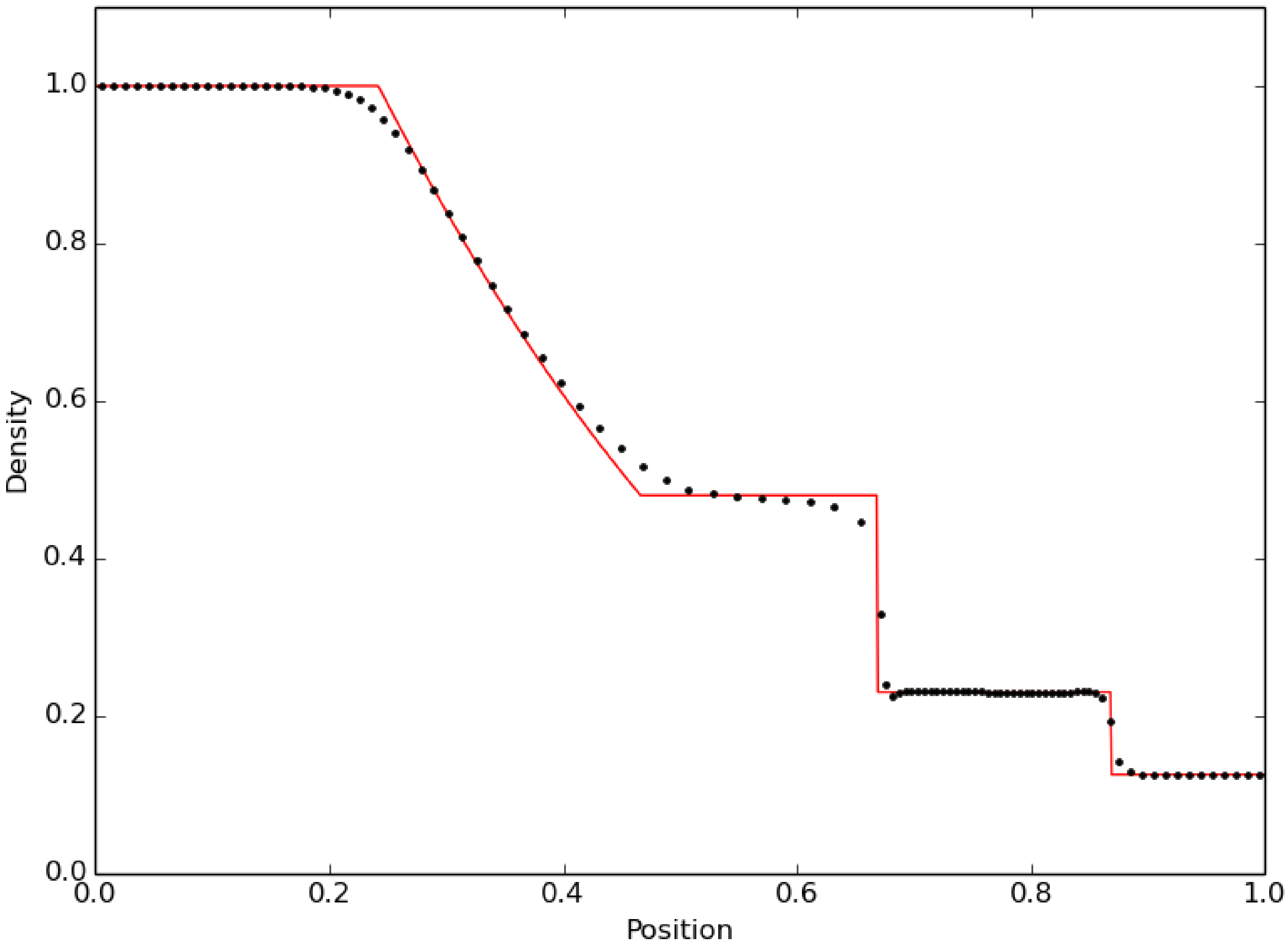
$\underline{w}$ is the velocity of the surface boundary of the cell

This also means we can solve the Riemann problem in the rest frame of the boundary surface, cancelling out large flow velocities that might affect the accuracy of the solver.

Reference solution at t = 0.2 (Eulerian)



Reference solution at t = 0.2 (Lagrangian)


Density
Position