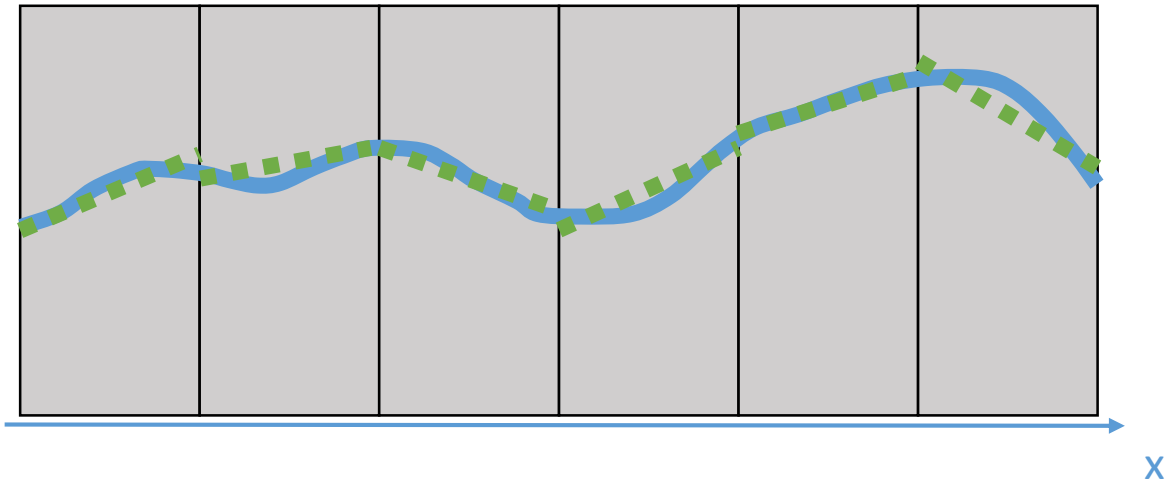
Second order finite volume methods

The details nobody ever really shows

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

Second order in a (1D) nutshell

Do a spatial extrapolation (second order in space)



Do a half time step prediction (second order in time)

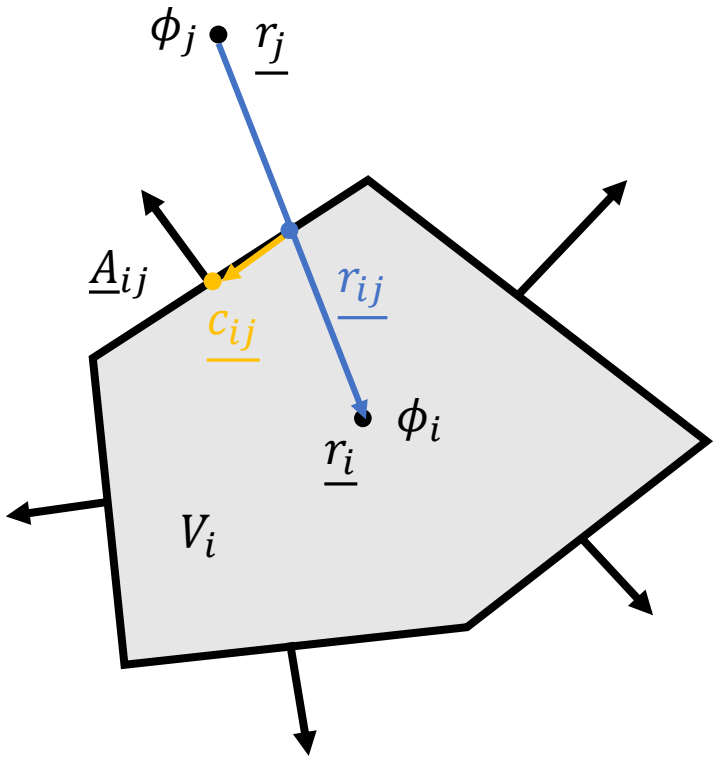
$$\rho' = \rho - \frac{1}{2}\Delta t\left(\rho \underline{\nabla}.\underline{u} + \underline{u}.\underline{\nabla}\rho\right) \qquad \underline{u}' = \underline{u} - \frac{1}{2}\Delta t\left(\underline{u}\,\underline{\nabla}.\underline{u} + \frac{1}{\rho}\underline{\nabla}p\right)$$

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

Part 1

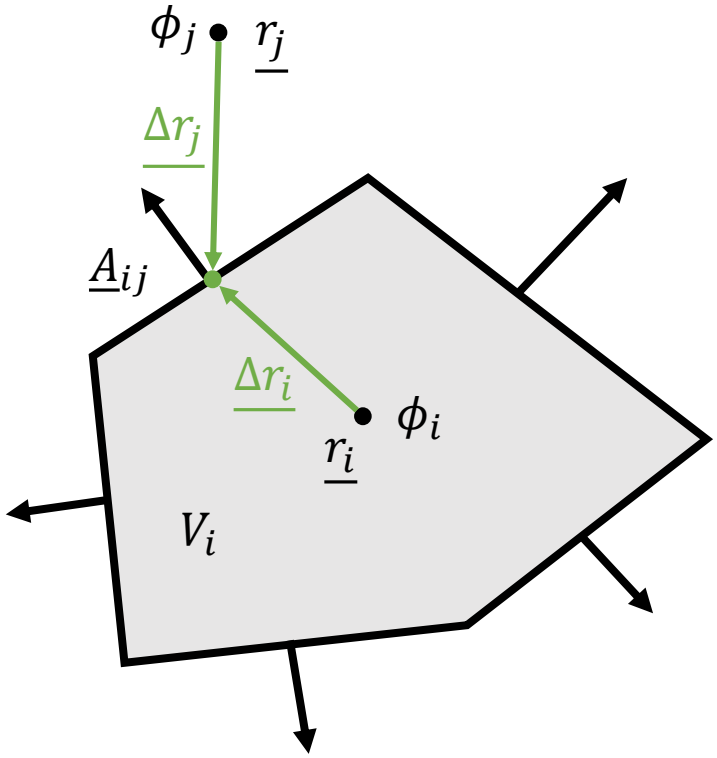
Second order in space

Spatial reconstruction: gradients



$$\langle \underline{\nabla}\phi \rangle_i = \frac{1}{V_i} \sum_{j \neq i} |\underline{A_{ij}}| \left((\phi_j - \phi_i) \frac{\underline{c_{ij}}}{|\underline{r_{ij}}|} - \frac{\phi_i + \phi_j}{2} \frac{\underline{r_{ij}}}{|\underline{r_{ij}}|} \right)$$

Spatial reconstruction



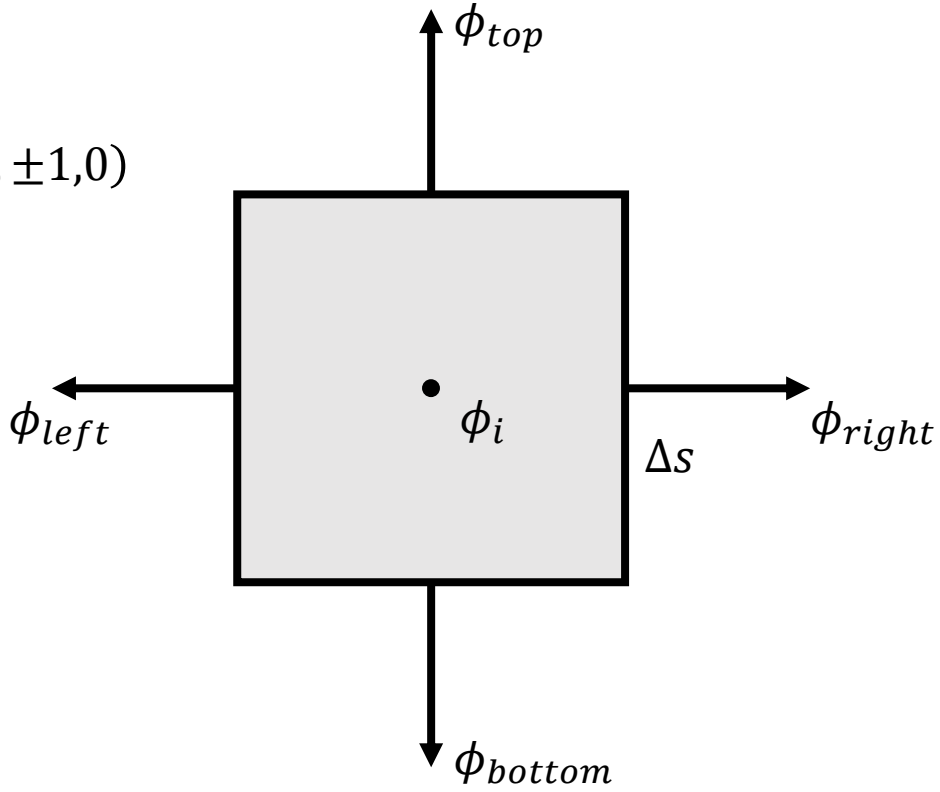
$$\phi_i' = \phi_i + \langle \underline{\nabla} \phi_i \rangle . \underline{\Delta r_i}$$

$$\phi_j' = \phi_j + \langle \underline{\nabla} \phi_j \rangle . \underline{\Delta r_j}$$

Spatial reconstruction: 2D Cartesian grid

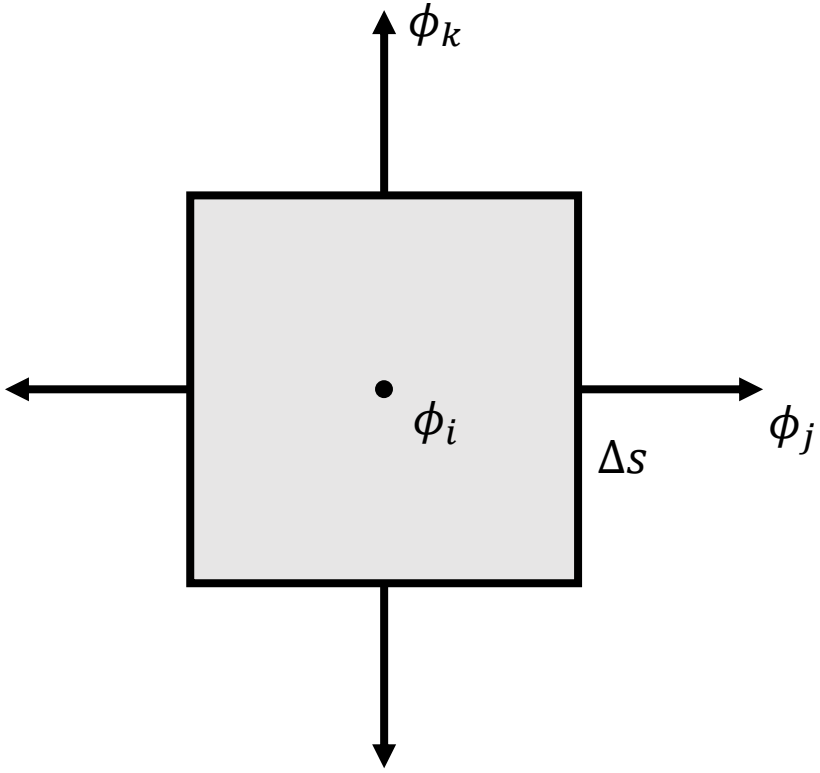
$$\frac{\underline{r_{ij}}}{\left|\underline{r_{ij}}\right|} = (\pm 1,0,0) \text{ or } (0,\pm 1,0)$$

$$\underline{c_{ij}} = 0$$



$$\left\langle \underline{\nabla}\phi \right\rangle_i = \frac{1}{\Delta s}\left(\frac{\phi_{right} - \phi_{left}}{2}, \frac{\phi_{top} - \phi_{bottom}}{2}, 0\right)$$

Spatial reconstruction: 2D Cartesian grid



For an x aligned face:

$$\phi_i' = \phi_i + \frac{\Delta s}{2}\langle \nabla \phi \rangle_{i,x}$$

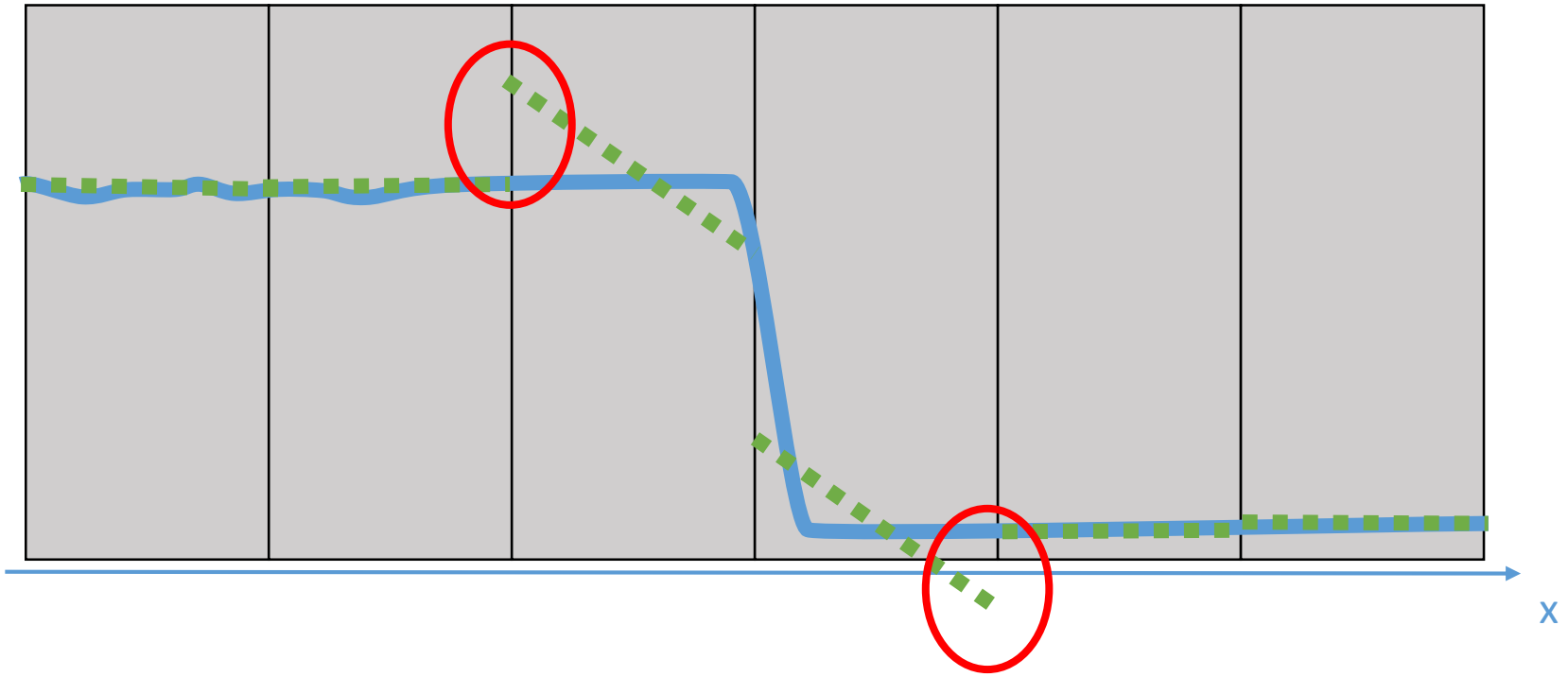
$$\phi_j' = \phi_j - \frac{\Delta s}{2}\langle \nabla \phi \rangle_{j,x}$$

For a y aligned face:

$$\phi_i' = \phi_i + \frac{\Delta s}{2}\langle \nabla \phi \rangle_{i,y}$$

$$\phi_k' = \phi_k - \frac{\Delta s}{2}\langle \nabla \phi \rangle_{k,y}$$

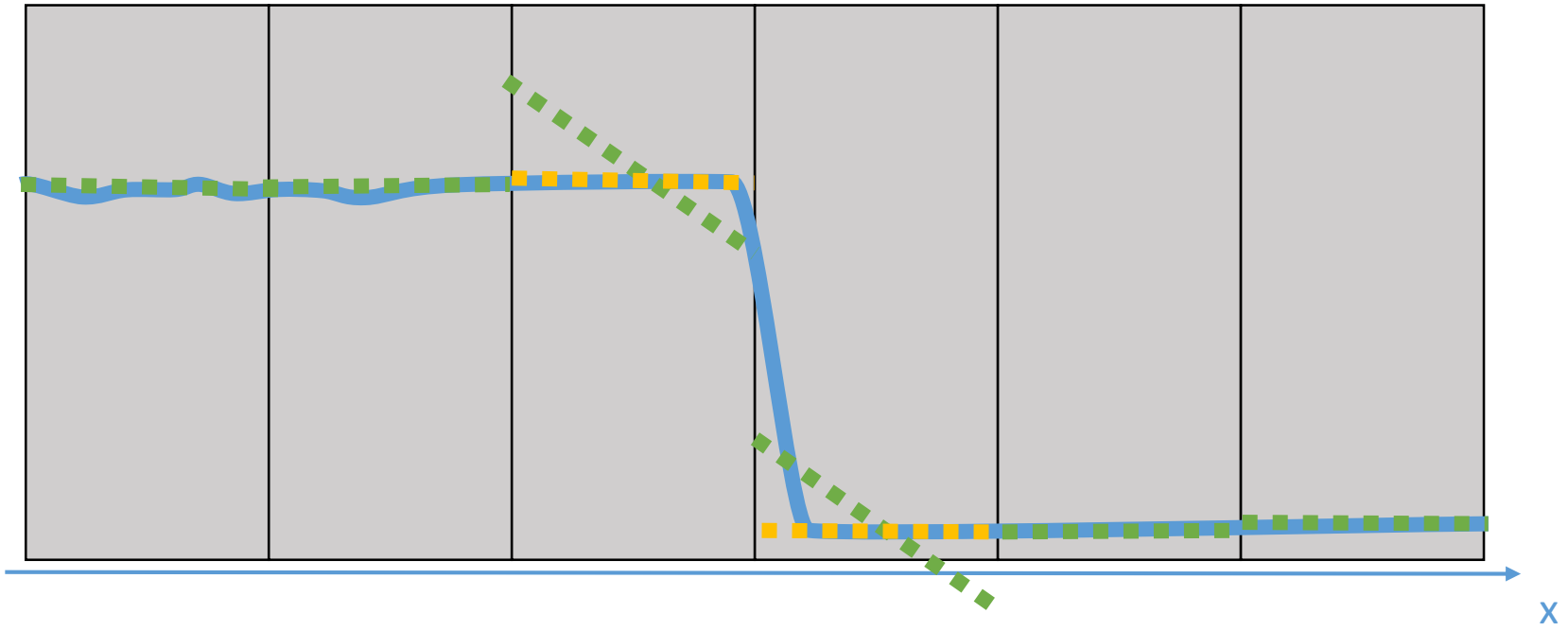
Slope limiters



In the presence of a strong gradient in the actual flow, linear extrapolation can overshoot: new maxima or (unphysical) minima are created at the interfaces

This causes instabilities or even algorithm crashes...

Slope limiters: cell wide



With a cell wide slope limiter, we explicitly make sure all extrapolated values are within the range of the surrounding cells: no new minima or maxima can be created

Slope limiters: cell wide

Compute a slope limited gradient:

$$\underline{\nabla}\phi' = \alpha \underline{\nabla}\phi$$

with

$$\alpha = \min\left(1, \beta \min\left(\frac{\phi_{ngb,max} - \phi}{\phi_{ext,max} - \phi}, \frac{\phi - \phi_{ngb,min}}{\phi - \phi_{ext,min}}\right)\right)$$

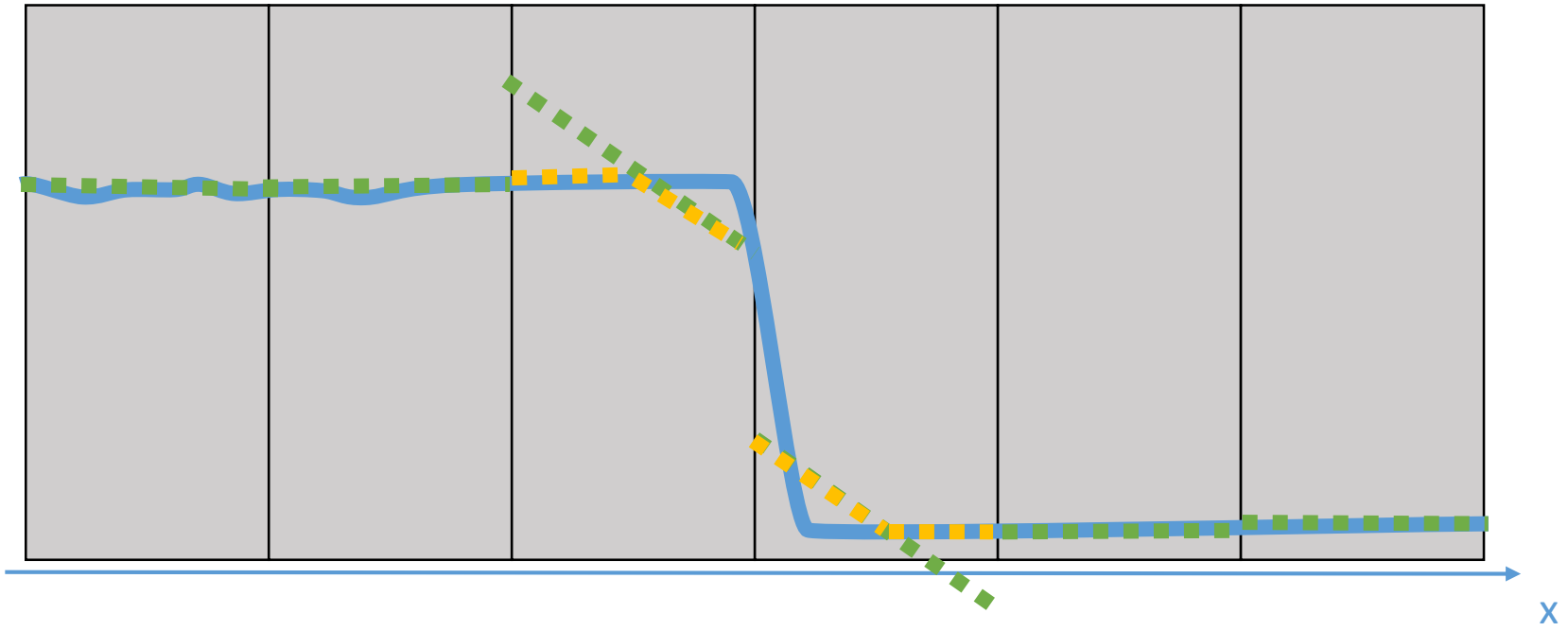
$$\phi_{ngb,max} = \max_j(\phi_j) \qquad \phi_{ext,max} = \max_j\left(\phi + \underline{\nabla}\phi . \underline{\Delta x_j}\right)$$

$$\phi_{ngb,min} = \min_j(\phi_j) \qquad \phi_{ext,min} = \min_j\left(\phi + \underline{\nabla}\phi . \underline{\Delta x_j}\right)$$

where the minima and maxima are taken over all neighbours j of the cell, and $\underline{\Delta x_j}$ is a vector pointing from the cell midpoint to the midpoint of the face between the cell and neighbour j

β is a constant, with $\beta = 0.5$ the most conservative (stable) allowed value

Slope limiters: per face



With a per face slope limiter, we limit the gradient when an overshoot occurs
This means we use different gradients for different interfaces

Slope limiters: per face

Replace the extrapolated value $\phi_{ext,j}$ of variable ϕ at the face with neighbour j with

$$\phi'_{ext,j} = \begin{cases} \phi, & \phi = \phi_j \\ \max\left(\phi_-, \min\left(\overline{\phi_j} + \delta_2, \phi_{ext,j}\right)\right), & \phi < \phi_j \\ \min\left(\phi_+, \max\left(\overline{\phi_j} - \delta_2, \phi_{ext,j}\right)\right), & \phi > \phi_j \end{cases}$$

$$\phi_- = \begin{cases} \phi_{min} - \delta_1, & (\phi_{min} - \delta_1)\phi_{min} > 0 \\ \dfrac{\phi_{min}}{1 + \dfrac{\delta_1}{|\phi_{min}|}}, & (\phi_{min} - \delta_1)\phi_{min} < 0 \end{cases}$$

$$\phi_+ = \begin{cases} \phi_{max} + \delta_1, & (\phi_{max} + \delta_1)\phi_{max} > 0 \\ \dfrac{\phi_{max}}{1 + \dfrac{\delta_1}{|\phi_{max}|}}, & (\phi_{max} + \delta_1)\phi_{max} < 0 \end{cases}$$

$$\overline{\phi_j} = \phi + \frac{\left|\underline{\Delta x_{face}}\right|}{\left|\underline{\Delta x_j}\right|}\left(\phi_j - \phi\right)$$

$$\phi_{min} = \min\left(\phi, \phi_j\right)$$

$$\phi_{max} = \max\left(\phi, \phi_j\right)$$

$$\delta_1 = \psi_1\left|\phi - \phi_j\right|$$

$$\delta_2 = \psi_2\left|\phi - \phi_j\right|$$

$0 \le \psi_1 \le 1$ and $0 \le \psi_2 \le 0.5$ are constants ($\psi_1 = 0.5$ and $\psi_2 = 0.25$ are stable values)

$\underline{\Delta x_{face}}$ is a vector pointing from the cell center to the center of the face

$\underline{\Delta x_j}$ is a vector pointing from the cell center to the center of cell j

Slope limiters

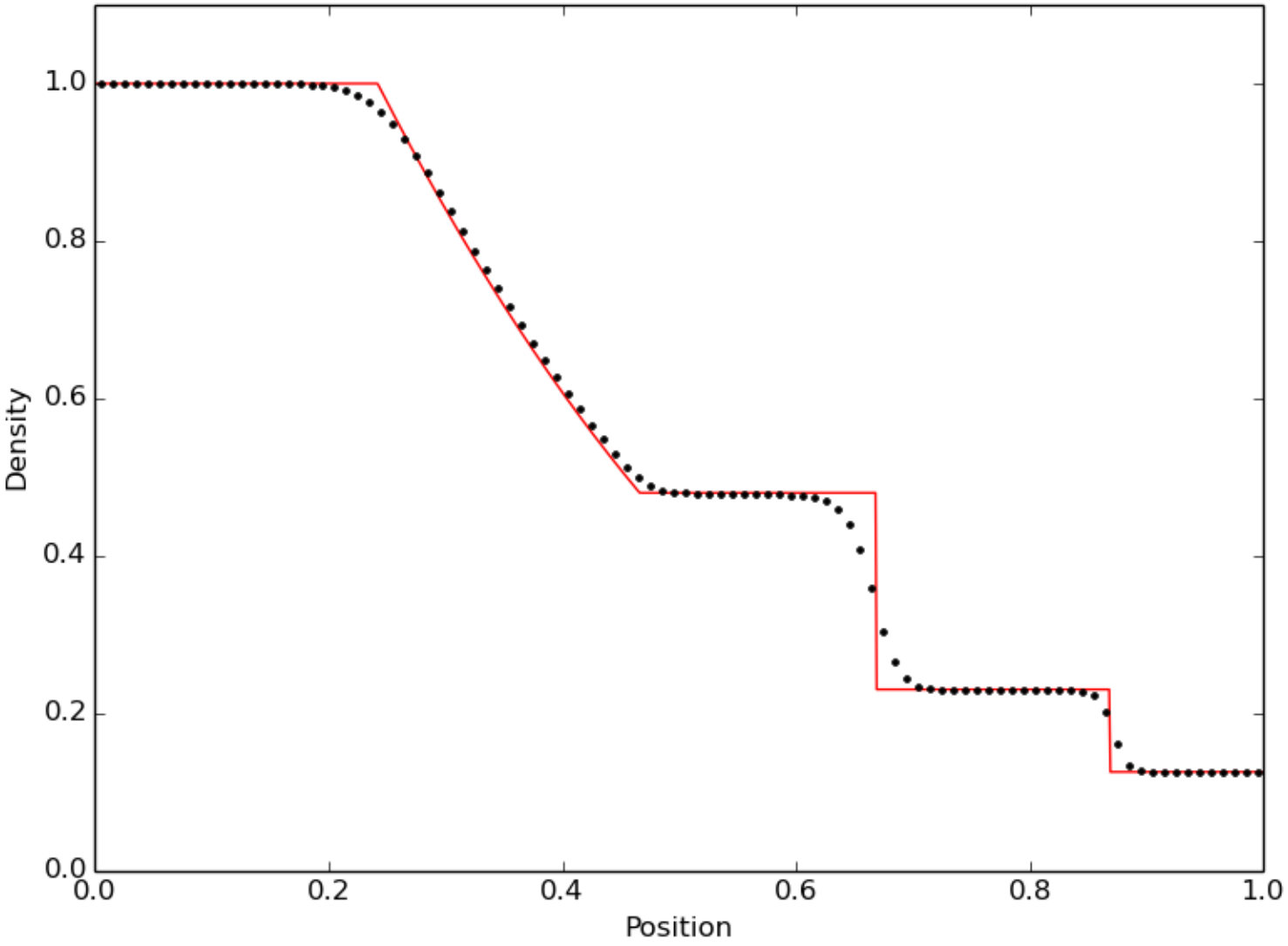
Slope limiters are BAD: they reduce the spatial order of the method

Slope limiters are a NECESSARY evil: they prevent the algorithm from crashing

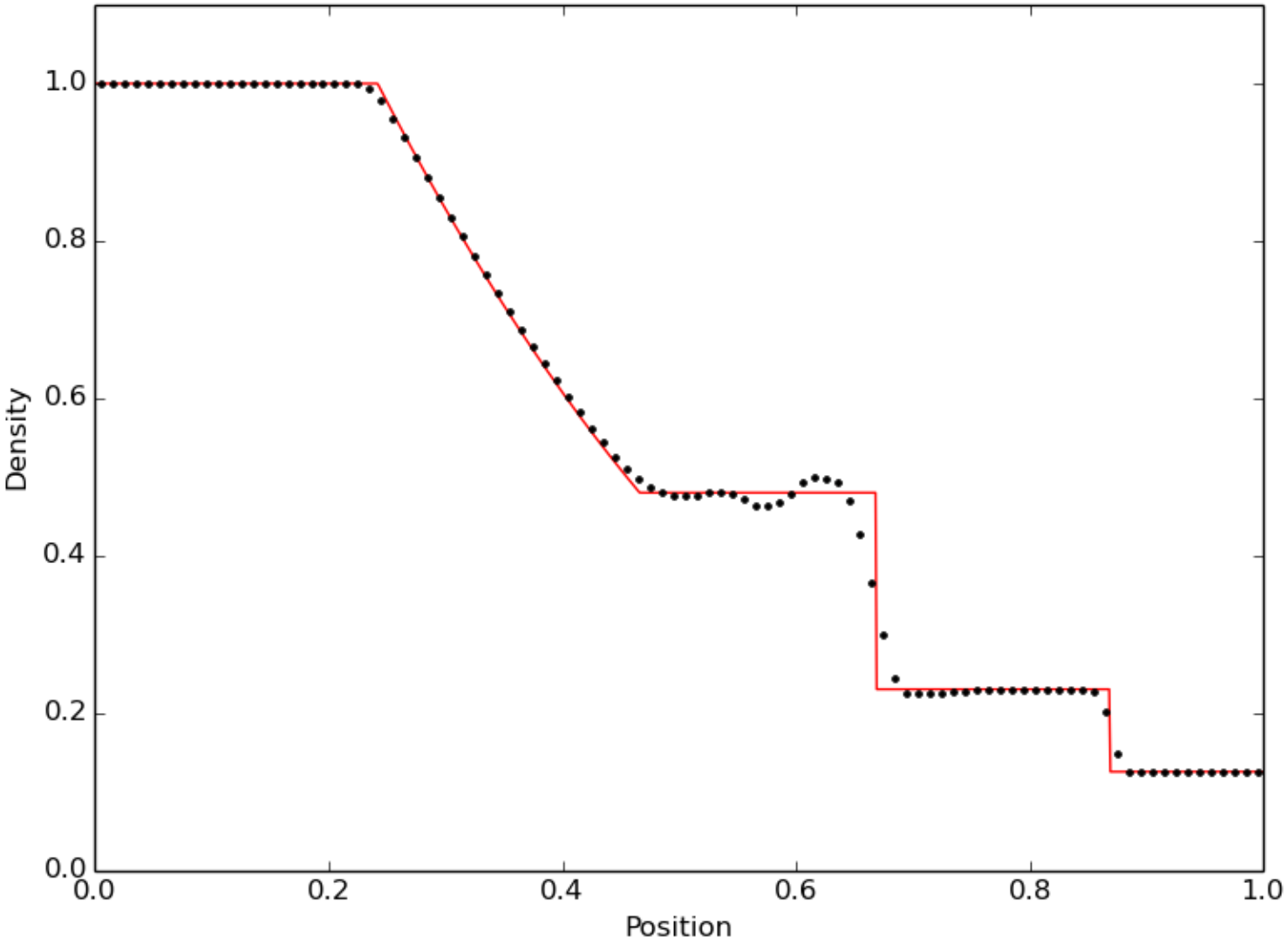
Using a very conservative slope limiter is safer, but less accurate...
Using a less restrictive slope limiter might cause problems...

CONCLUSION: a good second order scheme depends a lot on the choice of slope limiter

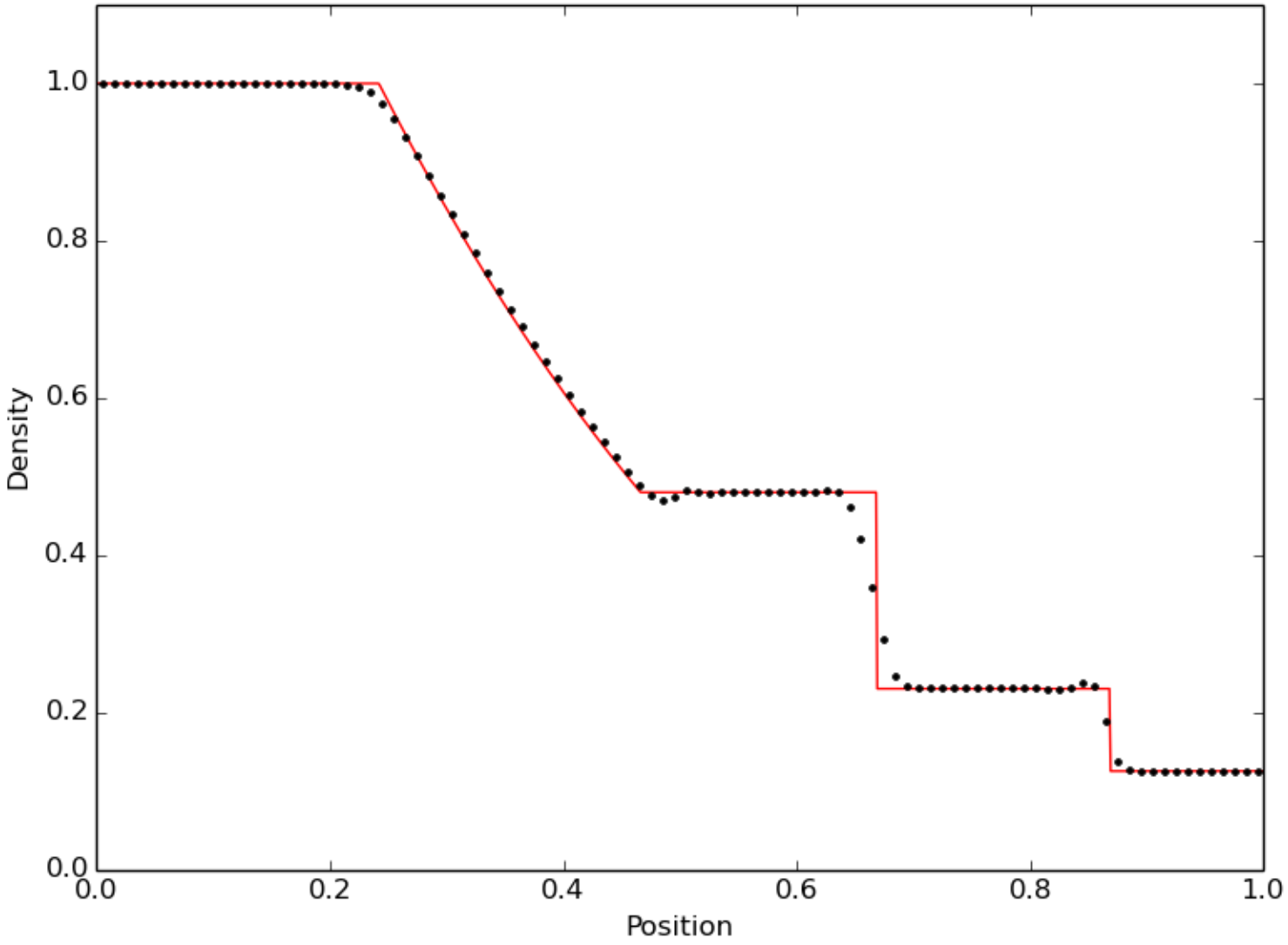
Second order Sod shock, cell wide limiter, conservative and stable



Second order Sod shock, cell wide limiter, less conservative and stable



Second order Sod shock, per face limiter, conservative and stable



Part 2

Second order in time

The Euler equations can be rewritten in terms of the primitive variables
In 1D:

$$\frac{\partial \rho}{\partial t} + u\frac{\partial \rho}{\partial x} + \rho\frac{\partial u}{\partial x} = 0$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0$$

$$\frac{\partial p}{\partial t} + \gamma p\frac{\partial u}{\partial x} + u\frac{\partial p}{\partial x} = 0$$

or:

$$\frac{\partial W}{\partial t} + A(W)\frac{\partial W}{\partial x} = 0 \qquad \text{with} \qquad W = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}, A(W) = \begin{pmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \gamma p & u \end{pmatrix}$$

In 3D:

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

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

$$\frac{\partial p}{\partial t} + \gamma p\underline{\nabla}.\underline{u} + \underline{u}.\underline{\nabla}p = 0$$

or:

$$\frac{\partial W}{\partial t} + A(W)\frac{\partial W}{\partial x} + B(W)\frac{\partial W}{\partial y} + C(W)\frac{\partial W}{\partial z} = 0 \qquad \text{with} \qquad W = \begin{pmatrix} \rho \\ u_x \\ u_y \\ u_z \\ p \end{pmatrix}$$

$$\frac{\partial W}{\partial t} + A(W)\frac{\partial W}{\partial x} + B(W)\frac{\partial W}{\partial y} + C(W)\frac{\partial W}{\partial z} = 0 \qquad \text{with} \qquad W = \begin{pmatrix} \rho \\ u_x \\ u_y \\ u_z \\ p \end{pmatrix}$$

$$A(W) = \begin{pmatrix} u_x & \rho & 0 & 0 & 0 \\ 0 & u_x & 0 & 0 & \frac{1}{\rho} \\ 0 & 0 & u_x & 0 & 0 \\ 0 & 0 & 0 & u_x & 0 \\ 0 & \gamma p & 0 & 0 & u_x \end{pmatrix}$$

$$B(W) = \begin{pmatrix} u_y & \rho & 0 & 0 & 0 \\ 0 & u_y & 0 & 0 & 0 \\ 0 & 0 & u_y & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & u_y & 0 \\ 0 & 0 & \gamma p & 0 & u_y \end{pmatrix} \qquad C(W) = \begin{pmatrix} u_z & \rho & 0 & 0 & 0 \\ 0 & u_z & 0 & 0 & 0 \\ 0 & 0 & u_z & 0 & 0 \\ 0 & 0 & 0 & u_z & \frac{1}{\rho} \\ 0 & 0 & 0 & \gamma p & u_z \end{pmatrix}$$

We can write this more concisely as

$$\frac{\partial W}{\partial t} + \underline{A}(W).\underline{\nabla} W = 0,$$

where $\underline{A}(W)$ is a vector matrix whose components are given by the matrices $A(W)$, $B(W)$ and $C(W)$

This can be used to obtain a first order accurate half time step prediction:

$$\Delta W = -\frac{1}{2}\Delta t \underline{A}(W).\underline{\nabla} W$$

The gradients $\underline{\nabla} W$ are the same gradients we use in the spatial reconstruction

In 1D, the full equations are

$$\rho' = \rho - \frac{1}{2}\Delta t\left(\rho\frac{\partial u_x}{\partial x} + u_x\frac{\partial \rho}{\partial x}\right)$$

$$u_x' = u_x - \frac{1}{2}\Delta t\left(u_x\frac{\partial u_x}{\partial x} + \frac{1}{\rho}\frac{\partial p}{\partial x}\right)$$

$$p' = p - \frac{1}{2}\Delta t\left(\gamma p\frac{\partial u_x}{\partial x} + u_x\frac{\partial p}{\partial x}\right)$$

In 2D, the full equations are

$$\rho' = \rho - \frac{1}{2}\Delta t\left(\rho\frac{\partial u_x}{\partial x} + \rho\frac{\partial u_y}{\partial y} + u_x\frac{\partial \rho}{\partial x} + u_y\frac{\partial \rho}{\partial y}\right)$$

$$u_x' = u_x - \frac{1}{2}\Delta t\left(u_x\frac{\partial u_x}{\partial x} + u_x\frac{\partial u_y}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x}\right)$$

$$u_y' = u_y - \frac{1}{2}\Delta t\left(u_y\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_y}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y}\right)$$

$$p' = p - \frac{1}{2}\Delta t\left(\gamma p\frac{\partial u_x}{\partial x} + \gamma p\frac{\partial u_y}{\partial y} + u_x\frac{\partial p}{\partial x} + u_y\frac{\partial p}{\partial y}\right)$$

In 3D, the full equations are

$$\rho' = \rho - \frac{1}{2}\Delta t\left(\rho\frac{\partial u_x}{\partial x} + \rho\frac{\partial u_y}{\partial y} + \rho\frac{\partial u_z}{\partial z} + u_x\frac{\partial \rho}{\partial x} + u_y\frac{\partial \rho}{\partial y} + u_z\frac{\partial \rho}{\partial z}\right)$$

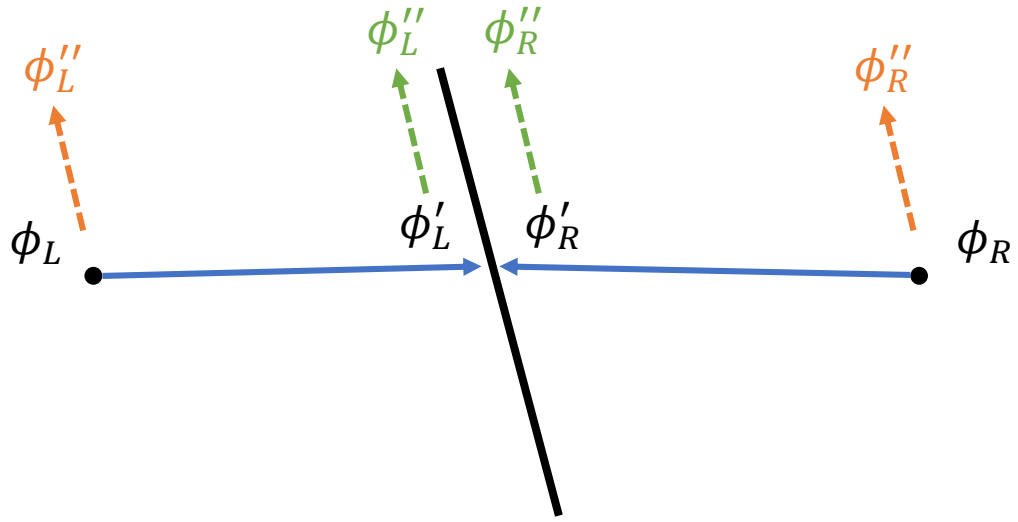
$$u_x' = u_x - \frac{1}{2}\Delta t\left(u_x\frac{\partial u_x}{\partial x} + u_x\frac{\partial u_y}{\partial y} + u_x\frac{\partial u_z}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial x}\right)$$

$$u_y' = u_y - \frac{1}{2}\Delta t\left(u_y\frac{\partial u_x}{\partial x} + u_y\frac{\partial u_y}{\partial y} + u_y\frac{\partial u_z}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial y}\right)$$

$$u_z' = u_z - \frac{1}{2}\Delta t\left(u_z\frac{\partial u_x}{\partial x} + u_z\frac{\partial u_y}{\partial y} + u_z\frac{\partial u_z}{\partial z} + \frac{1}{\rho}\frac{\partial p}{\partial z}\right)$$

$$p' = p - \frac{1}{2}\Delta t\left(\gamma p\frac{\partial u_x}{\partial x} + \gamma p\frac{\partial u_y}{\partial y} + \gamma p\frac{\partial u_z}{\partial z} + u_x\frac{\partial p}{\partial x} + u_y\frac{\partial p}{\partial y} + u_z\frac{\partial p}{\partial z}\right)$$

Time prediction step



The time prediction can be carried out before or after the extrapolation step

However, the cleanest approach is to keep them independent of each other, which means both steps use the central values

Note that a cell wide slope limiter will also affect the time prediction step, while a per face limiter can be limited to the spatial reconstruction