

Accuracy, limiters and approximate Riemann solvers

Donna Calhoun

Boise State University

*PASI 2013
Valparaiso, Chile
January 2-13*

Outline

- We solved exactly the Riemann problem for the constant coefficient linear system case, e.g. linear shallow water equations
- We described what is involved in solving the Riemann problem for the nonlinear shallow water wave equations.
- How do these Riemann solvers make it into an actual code?
- Do we actually solve the non-linear problem at every grid cell interface?
- How accurate are these methods?

Riemann problem for linear systems

Solving the Riemann problem for linear problem

$$q_t + A q_x = 0$$

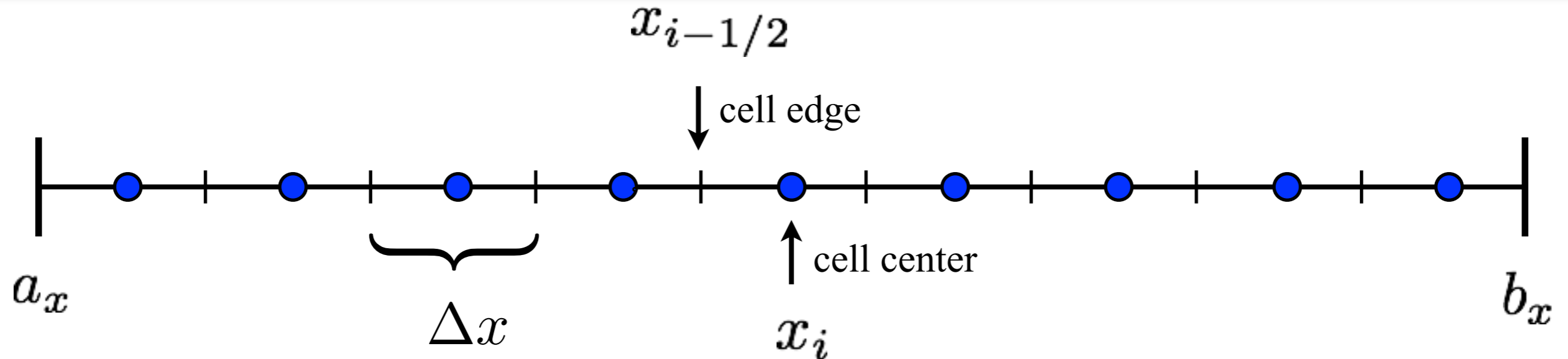
- (1) Compute eigenvalues and eigenvectors of matrix A
- (2) Compute characteristic variables by solving

$$R \alpha = q_r - q_\ell$$

- (3) Use eigenvalues or “speeds” to determine piecewise constant solution

$$\begin{aligned} q(x, t) &= q_\ell + \sum_{p: \lambda^p < x/t} \alpha^p r^p \\ &= q_r - \sum_{p: \lambda^p > x/t} \alpha^p r^p \end{aligned}$$

One dimensional Cartesian grid



Cell centers :

$$x_i = a_x + (i - 1/2)\Delta x, \quad i = 1, 2, \dots, M_x$$

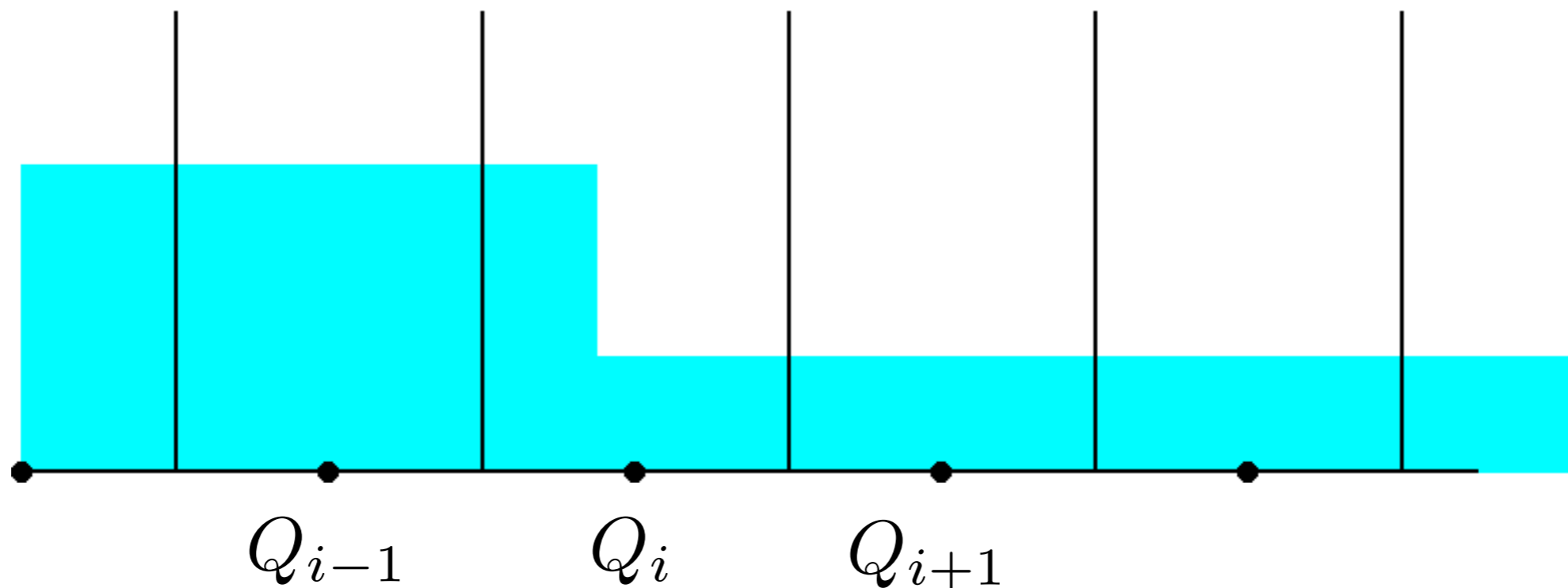
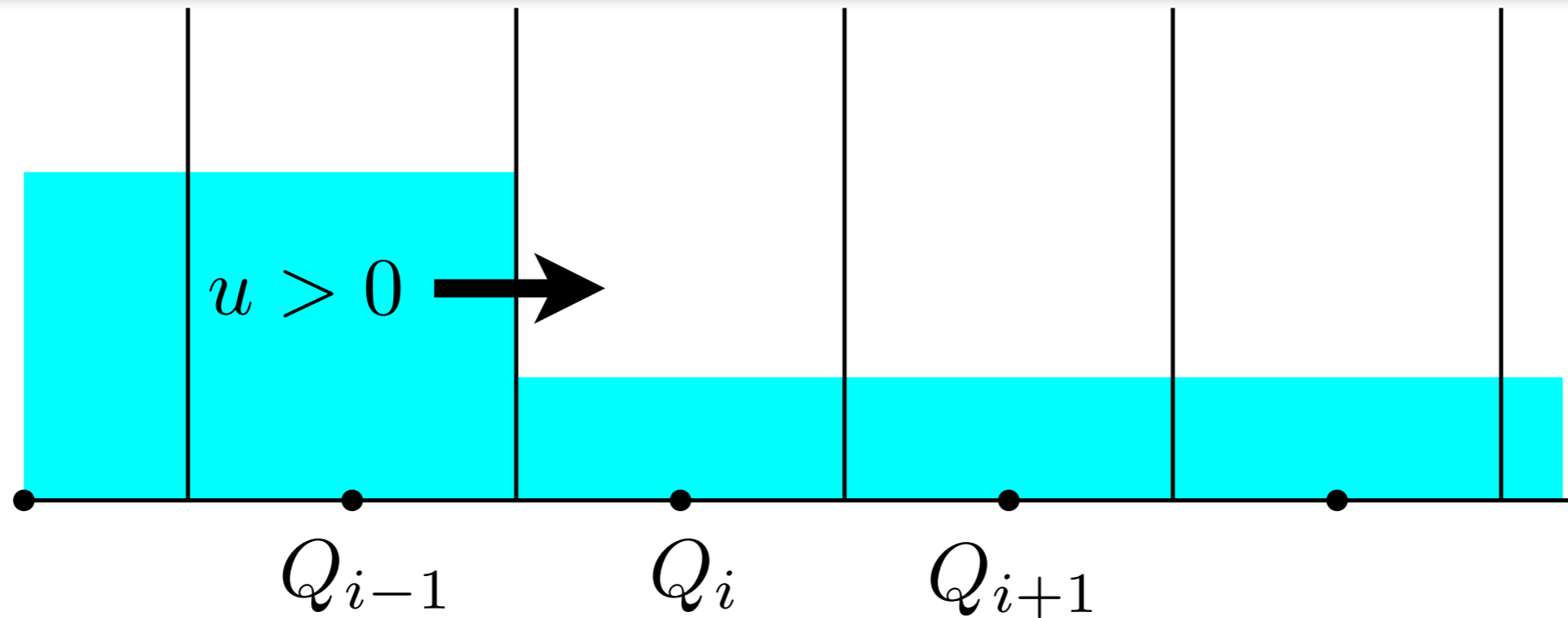
Cell edges :

$$x_{i-1/2} = a_x + (i - 1)\Delta x, \quad i = 1, 2, \dots, M_x + 1$$

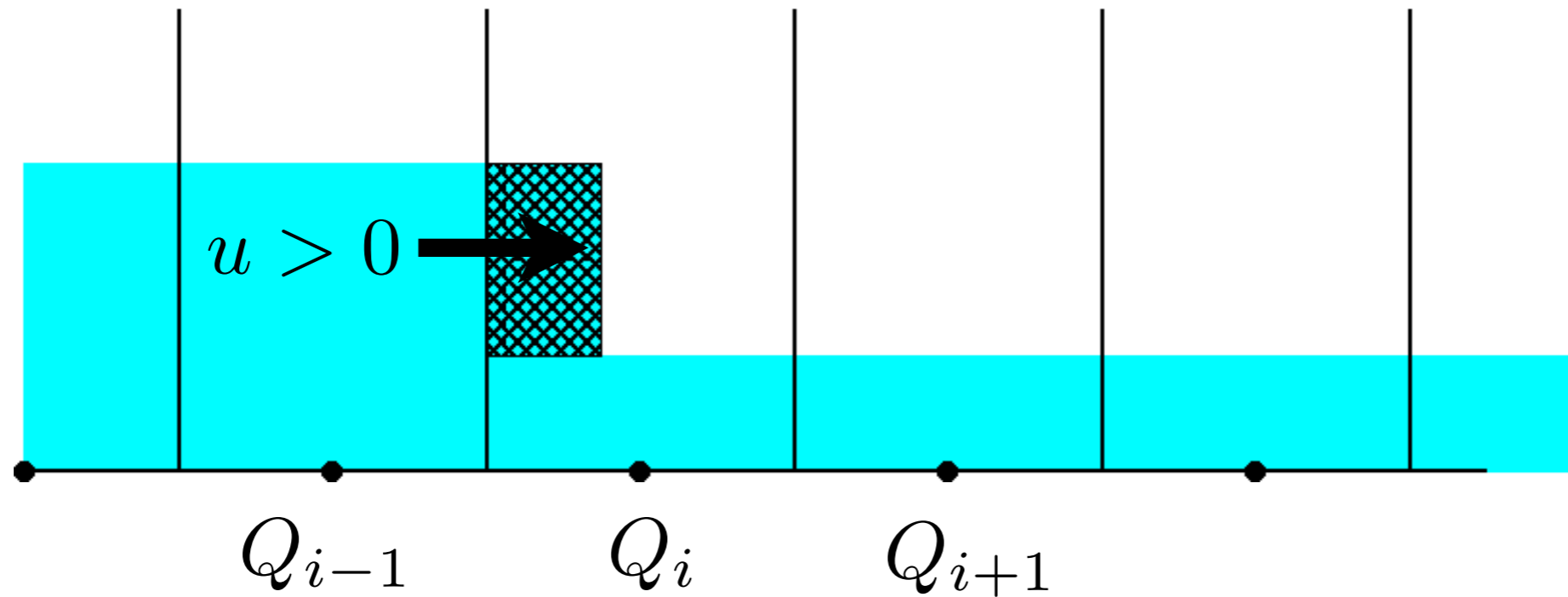
Time step over interval $[0, T]$:

$$t_n = n\Delta t, \quad n = 1, 2, \dots, N_{out}$$

Update cell averages explicitly



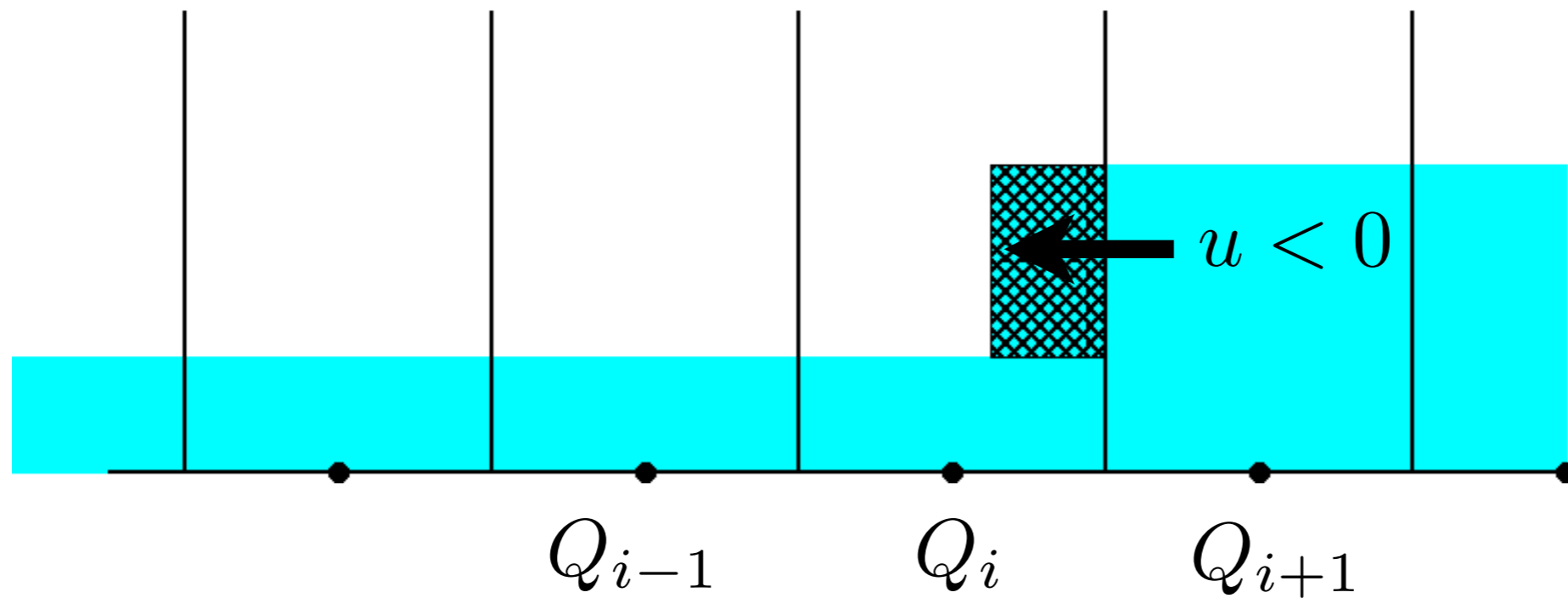
Update cell averages explicitly



In time Δt , mass in cell C_i increases by shaded area :

$$\Delta x Q_i^{n+1} = \Delta x Q_i^n - u \Delta t (Q_i^n - Q_{i-1}^n)$$

Update cell averages explicitly



In time Δt , mass in cell C_i increases by shaded area :

$$\Delta x Q_i^{n+1} = \Delta x Q_i^n - u \Delta t (Q_{i+1}^n - Q_i^n)$$

Wave propagation viewpoint - scalar equation

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(u^+ (Q_i^n - Q_{i-1}^n) + u^- (Q_{i+1}^n - Q_i^n) \right)$$

where

$$u^+ = \max(u, 0), \quad u^- = \min(u, 0)$$

We can define *waves* at each interface as :

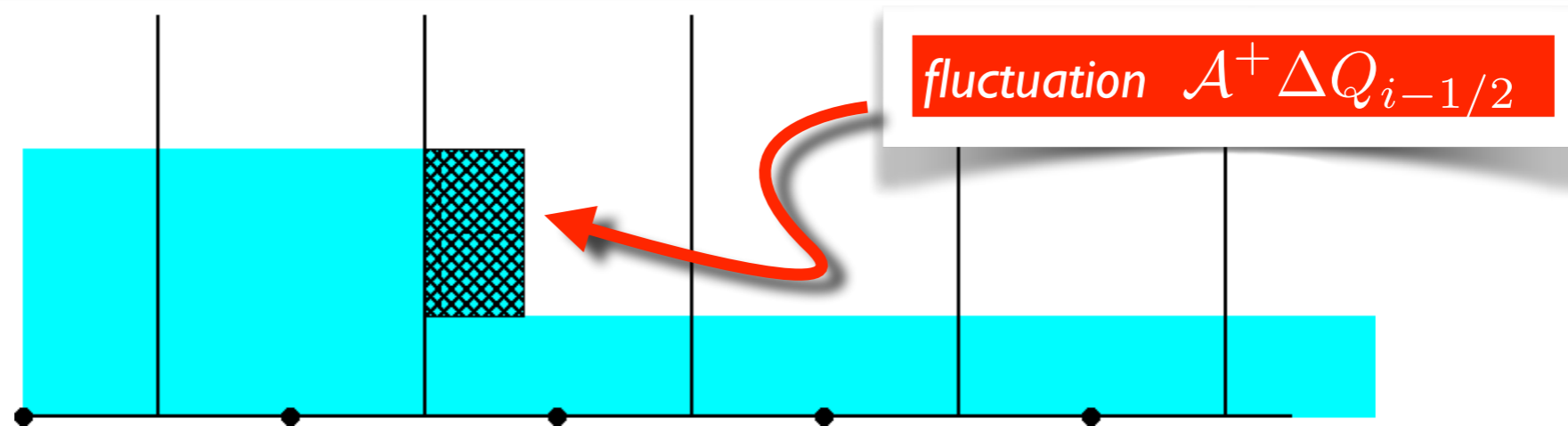
$$\text{Waves : } \mathcal{W}_{i-1/2} \equiv Q_i - Q_{i-1}$$

Our scheme might look like :

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(u^+ \mathcal{W}_{i-1/2} + u^- \mathcal{W}_{i+1/2} \right)$$

“wave propagation algorithm” (R. J. LeVeque)

Wave propagation viewpoint - scalar equation



$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (u^+ \mathcal{W}_{i-1/2} + u^- \mathcal{W}_{i+1/2})$$

We can write this in terms of *fluctuations* :

$$\mathcal{A}^+ \Delta Q_{i-1/2} \equiv u^+ \mathcal{W}_{i-1/2}$$

$$\mathcal{A}^- \Delta Q_{i+1/2} \equiv u^- \mathcal{W}_{i+1/2}$$

The first order term in the update used by Clawpack and GeoClaw

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2})$$


Wave propagation viewpoint - systems

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left(\sum_{p=1}^m (\lambda^p)^+ \mathcal{W}_{i-1/2}^p + \sum_{p=1}^m (\lambda^p)^- \mathcal{W}_{i+1/2}^p \right)$$

where the waves are now defined from an eigenvalue decomposition of the jump in value at each interface

$$\text{Waves : } \mathcal{W}_{i-1/2}^p \equiv \alpha^p r^p$$

Written in terms of *fluctuations* :

$$R\alpha = q_r - q_\ell$$


$$\mathcal{A}^+ \Delta Q_{i-1/2} \equiv \sum_{p=1}^m (\lambda^p)^+ \mathcal{W}_{i-1/2}^p \quad (\lambda^p)^+ = \max(\lambda^p, 0)$$

$$\mathcal{A}^- \Delta Q_{i-1/2} \equiv \sum_{p=1}^m (\lambda^p)^- \mathcal{W}_{i-1/2}^p \quad (\lambda^p)^- = \min(\lambda^p, 0)$$

Clawpack - rp1ad.f

```
sqrtgh = sqrt(grav*h_mean)
do i = 2-mbc,mx+mbc
  do m = 1,2
    delta(m) = qr(i,m) - qr(i-1,m)
  enddo

  # Speeds
  s(i,1) = u_mean - sqrtgh
  s(i,2) = u_mean + sqrtgh

  a1 = (h_mean*delta(2) - sqrtgh*delta(1)) / (2*grav*h_mean)
  a2 = (h_mean*delta(2) + sqrtgh*delta(1)) / (2*grav*h_mean)

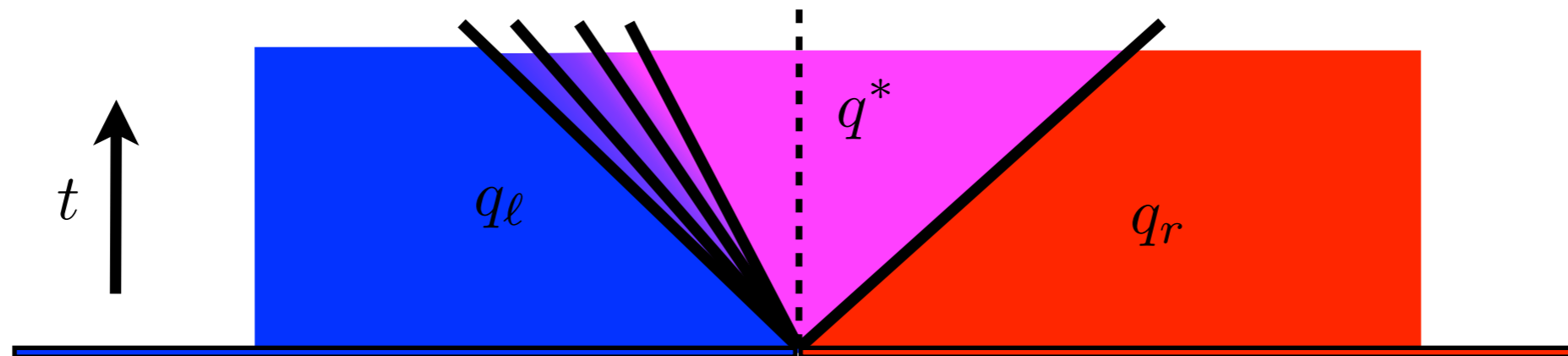
  # Waves
  wave(i,1,1) = -a1*sqrtgh
  wave(i,2,1) = a1*grav
  wave(i,1,2) = a2*sqrtgh
  wave(i,2,2) = a2*grav

  # Fluctuations
  do m = 1,meqn
    amdq(i,m) = 0
    apdq(i,m) = 0
    do mw = 1,mwaves
      amdq(i,m) = amdq(i,m) + min(s(i,mw), 0.d0) * wave(i,m,mw)
      apdq(i,m) = apdq(i,m) + max(s(i,mw), 0.d0) * wave(i,m,mw)
    enddo
  enddo
enddo
enddo
```

The diagram consists of three red boxes with white text, each containing a label. White arrows point from these boxes to specific lines of code in the Fortran program. The 'Speeds' box points to the lines calculating `s(i,1)` and `s(i,2)`. The 'Waves' box points to the lines calculating `amdq(i,m)` and `apdq(i,m)` for the first time step (`mw=1`). The 'Fluctuations' box points to the lines calculating `amdq(i,m)` and `apdq(i,m)` for subsequent time steps (`mw > 1`).

Nonlinear case

Given an exact solution, we can also construct waves, speeds and fluctuations



Waves :

$$\mathcal{W}^1 \equiv q^* - q_l$$

$$\mathcal{W}^2 \equiv q_r - q^*$$

transonic case?

Fluctuations :

$$\mathcal{A}^+ \Delta Q_{i-1/2} \equiv f(q_r) - f(q^*)$$

$$\mathcal{A}^- \Delta Q_{i+1/2} \equiv f(q^*) - f(q_l)$$

Speeds : Shock speed or average speed in a rarefaction

Connection to flux formulation

Numerical fluxes can be written in terms of fluctuations :

$$F_{i-1/2} \equiv f(Q_i) - \mathcal{A}^+ \Delta Q_{i-1/2}$$

or

$$F_{i-1/2} \equiv f(Q_{i-1}) + \mathcal{A}^- \Delta Q_{i-1/2}$$

Flux differences can be expressed in terms of left going and right going fluctuations :

$$F_{i+1/2} - F_{i-1/2} = \mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}$$

Roe linearization for the non-linear case

$$q_t + f(q)_x = 0$$

We solve a linearized system at each cell interface, at each time step

$$q_t + f'(\hat{q})q_x = 0 \quad \longleftrightarrow \quad q_t + A(\hat{q})q_x = 0$$

for “Roe averaged” values \hat{q} .

For conservation, we need \hat{q} to satisfy :

$$f(q_r) - f(q_\ell) = f'(\hat{q})(q_r - q_\ell)$$

P. Roe (JCP, 1981) showed an approach for many important systems.

Roe averaged values for SWE

- Compute Roe averaged values :

$$\hat{h} = \frac{h_\ell + h_r}{2}, \quad \hat{u} = \frac{\sqrt{h_\ell}u_\ell + \sqrt{h_r}u_r}{\sqrt{h_\ell} + \sqrt{h_r}}$$

- Evaluate eigenvalues and eigenvectors at these values.
- Compute waves, speeds and fluctuations as in the linear case. Does not require the nonlinear root-finder.

Roe averages are also available for the Euler equations and other important physical systems.

Other approximate Riemann solvers are available.

Roe solver in Clawpack

```
do i = 2-mbc,mx+mbc
.....
c # compute Roe-averaged quantities:
u_roe = (ur/sqrt(hr) + ul/sqrt(hl))/(sqrt(hl) + sqrt(hr))
h_mean = (hl + hr)/2.d0
sqrtgh_roe = sqrt(grav*h_mean)

c # wave speeds
s(i,1) = u_roe - sqrtgh_roe
s(i,2) = u_roe + sqrtgh_roe

c # compute coeffs in the evector expansion of delta(1),delta(2)
a1 = (-delta(2) + (u_roe + sqrtgh_roe)*delta(1))/(2*sqrtgh_roe)
a2 = (delta(2) - (u_roe - sqrtgh_roe)*delta(1))/(2*sqrtgh_roe)

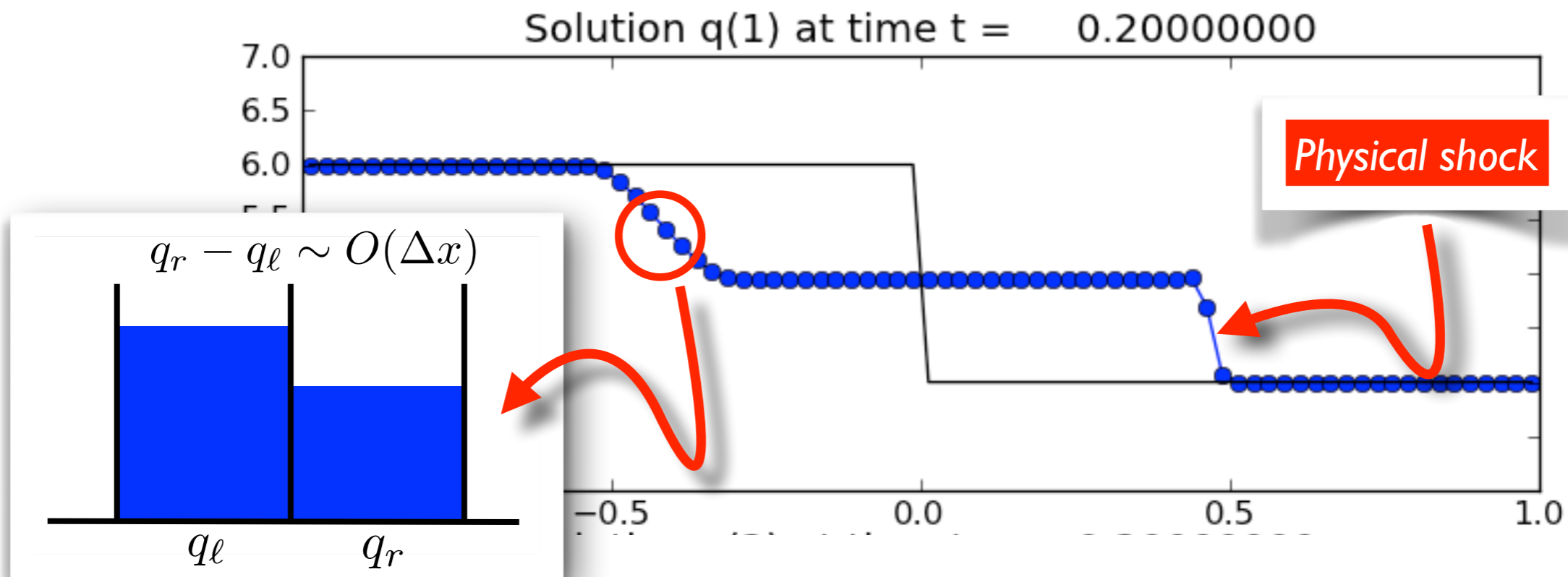
c # finally, compute the waves.
wave(i,1,1) = a1
wave(i,2,1) = a1*(u_roe - sqrtgh_roe)
wave(i,1,2) = a2
wave(i,2,2) = a2*(u_roe + sqrtgh_roe)

.....
do mw=1,mwaves
  amdq(i,m) = amdq(i,m) + min(s(i,mw), 0.d0) * wave(i,m,mw)
  apdq(i,m) = apdq(i,m) + max(s(i,mw), 0.d0) * wave(i,m,mw)
enddo

enddo
```

The diagram illustrates the flow of data in the Roe solver. A red box labeled "Speeds" has an arrow pointing to the calculation of wave speeds $s(i,1)$ and $s(i,2)$. Another red box labeled "Waves" has an arrow pointing to the calculation of wave components $wave(i,m,mw)$. A third red box labeled "Fluctuations" has an arrow pointing to the final update of $amdq$ and $apdq$.

Using Riemann solvers

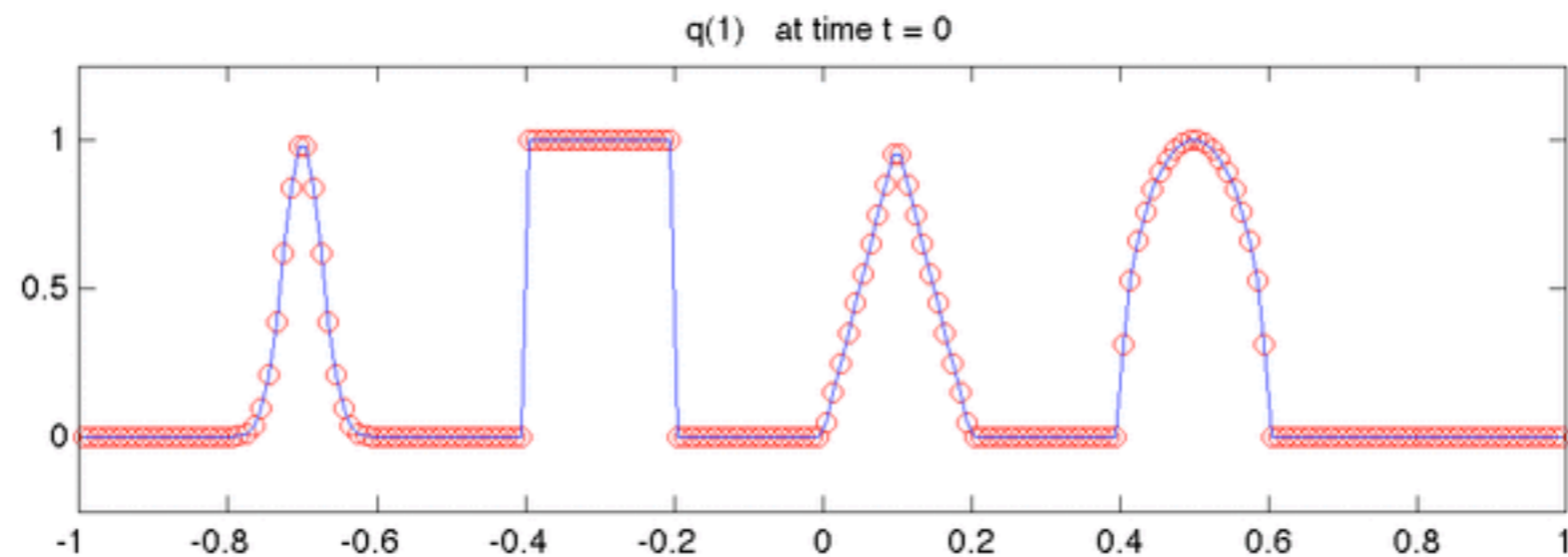


Question : There is only one shock and one rarefaction, but we solve a Riemann problem (either exactly, or approximately) at each cell interface. What happens in the smooth regions?

Answer : In smooth regions, shocks/rarefactions are *weak*. They only have strength on the order of the mesh cell size, i.e.

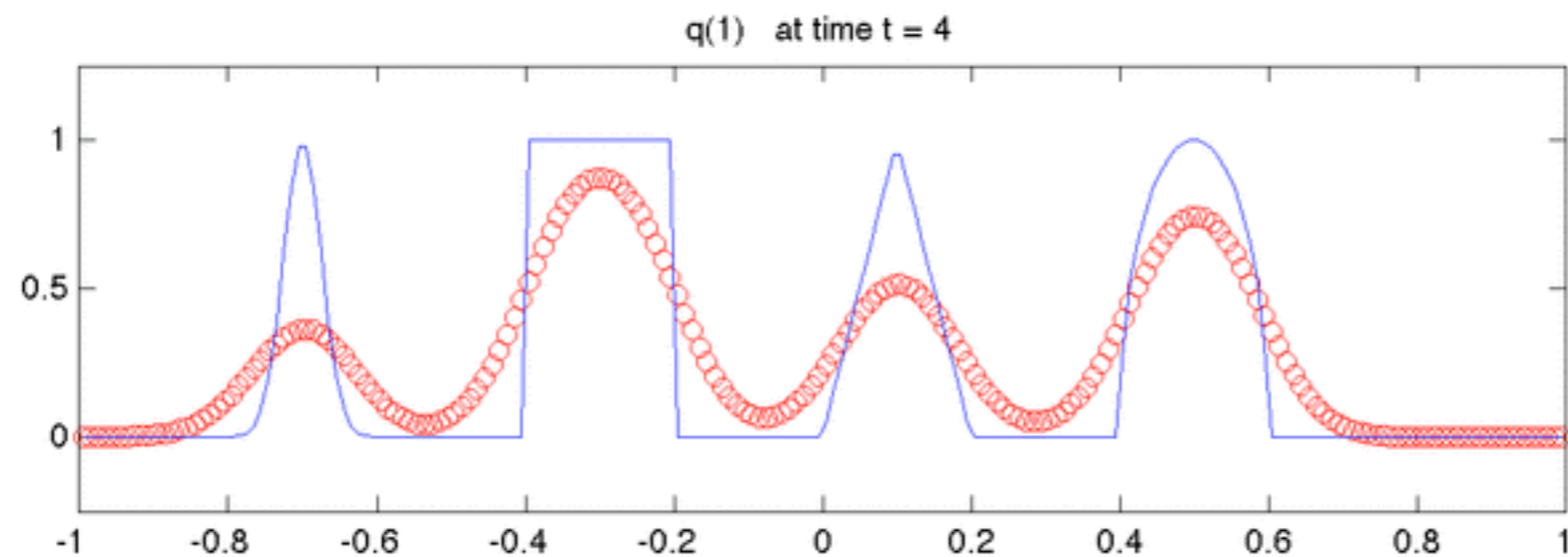
$$q_r - q_l \sim O(\Delta x)$$

Upwind method



First order scheme (200 points)

Upwind method



First order scheme (200 points)

The upwind method

The upwind method

$$\begin{aligned}Q_i^{n+1} &= Q_i^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}) \\ &= Q_i^n - \frac{\Delta t}{\Delta x} \bar{u} (Q_i^n - Q_{i-1}), \quad \text{for } \bar{u} > 0\end{aligned}$$

is only a first order approximation, but gives a good second order approximation to the equation

$$q_t + \bar{u}q_x = \frac{\bar{u}\Delta x}{2} \left(1 - \frac{\bar{u}\Delta t}{\Delta x}\right) q_{xx}$$

modified PDE

The “diffusion term” is proportional to the mesh spacing and the Courant number

Improved accuracy

Why not include these “diffusion” terms in the numerical scheme to get better accuracy?

$$q_t + uq_x = \frac{u\Delta x}{2} \left(1 - \frac{u\Delta t}{\Delta x}\right) q_{xx} \quad u > 0$$
$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} + u \frac{Q_i^n - Q_{i-1}^n}{\Delta x} = \frac{u\Delta x}{2} \left(1 - \frac{u\Delta t}{\Delta x}\right) \left(\frac{Q_{i+1}^n - 2Q_i^n + Q_{i-1}^n}{(\Delta x)^2}\right)$$

Re-arranging terms, we get the second order “Lax-Wendroff” method.

The Lax-Wendroff method

$$Q_i^{n+1} = Q_i^n - \underbrace{\frac{u\Delta t}{\Delta x} (Q_i^n - Q_{i-1}^n)}_{\text{Upwind term}} - \underbrace{\frac{1}{2} \frac{u\Delta t}{\Delta x} \left(1 - \frac{u\Delta t}{\Delta x}\right) ((Q_{i+1}^n - Q_i^n) - (Q_i^n - Q_{i-1}^n))}_{\text{Second order correction}}$$

Upwind term

Second order correction

The Lax-Wendroff method gives a third order approximation to the modified equation

$$q_t + uq_x = -\frac{u(\Delta x)^2}{6} \left(1 - \left(\frac{u\Delta t}{\Delta x}\right)^2\right) q_{xxx}$$

Errors are *dispersive*

Lax Wendroff method

$$Q_i^{n+1} = Q_i^n - \frac{u\Delta t}{\Delta x} (Q_i^n - Q_{i-1}^n) - \frac{1}{2} \frac{u\Delta t}{\Delta x} \left(1 - \frac{u\Delta t}{\Delta x}\right) ((Q_{i+1}^n - Q_i^n) - (Q_i^n - Q_{i-1}^n))$$

Waves

$$Q_i^{n+1} = Q_i^n - \frac{u\Delta t}{\Delta x} (Q_i^n - Q_{i-1}^n) - \frac{1}{2} \frac{u\Delta t}{\Delta x} (\Delta x - u\Delta t) (\sigma_i^n - \sigma_{i-1}^n)$$

slopes

Slopes

Waves

$$\sigma_i^n = \frac{Q_{i+1}^n - Q_i^n}{\Delta x}$$

$$W_{i-1/2}^n = Q_i^n - Q_{i-1}^n$$

Wave propagation viewpoint

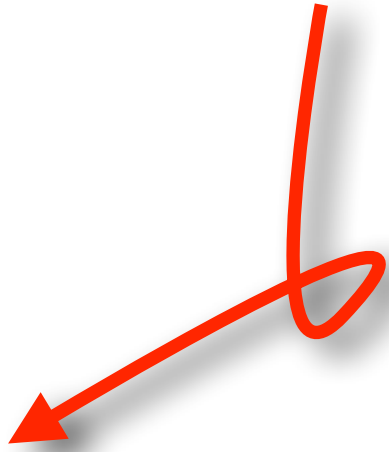
For $u > 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (u \mathcal{W}_{i-1/2}^n) - \frac{1}{2} \frac{u \Delta t}{\Delta x} \left(1 - \frac{u \Delta t}{\Delta x} \right) (\mathcal{W}_{i+1/2}^n - \mathcal{W}_{i-1/2}^n)$$

In wave propagation form, we can write this as :

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}) - \frac{\Delta t}{\Delta x} (\mathcal{F}_{i+1/2} - \mathcal{F}_{i-1/2})$$

where *second order correction terms* are defined as

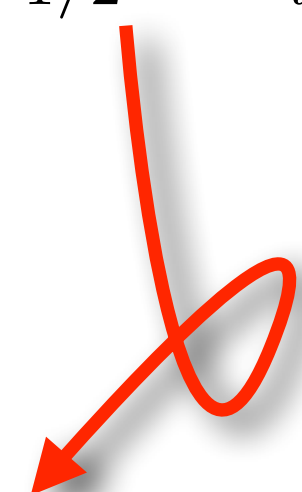
$$\mathcal{F}_{i-1/2} \equiv \frac{1}{2} |u| \left(1 - \frac{|u| \Delta t}{\Delta x} \right) \mathcal{W}_{i-1/2}^n$$


Wave propagation algorithm for systems

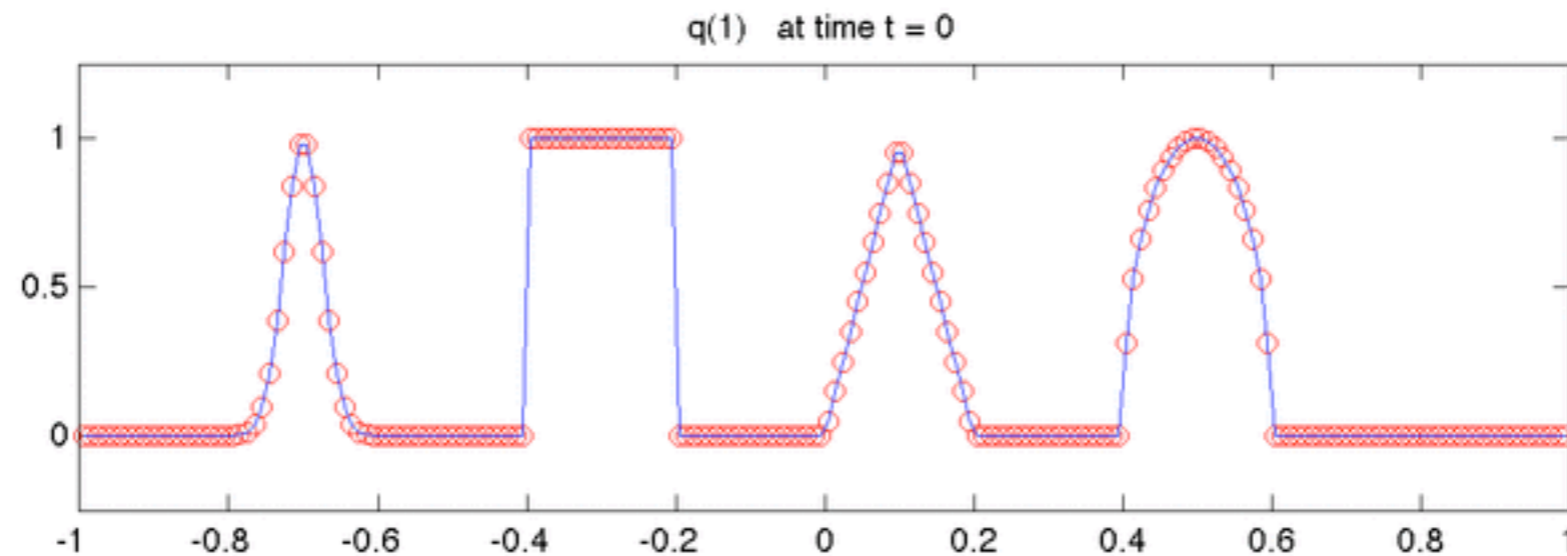
For systems (both linear and nonlinear), we have the update formula

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2}) - \frac{\Delta t}{\Delta x} (\mathcal{F}_{i+1/2} - \mathcal{F}_{i-1/2})$$

where the second order corrections are defined as

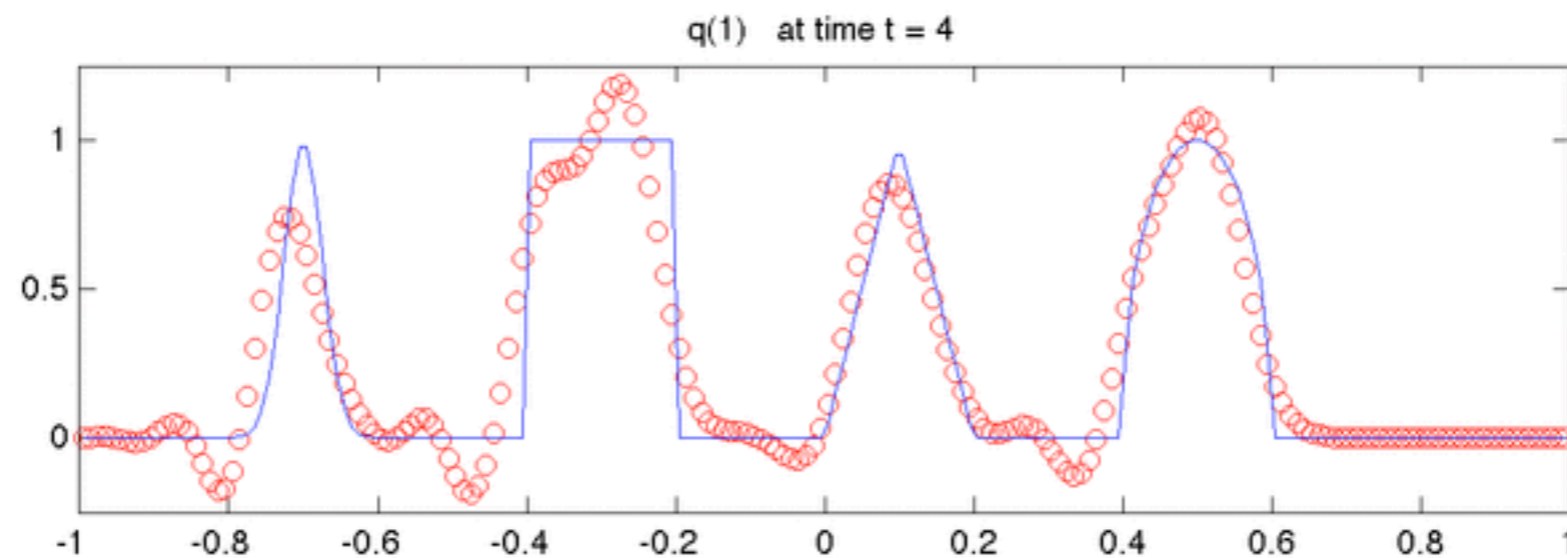
$$\mathcal{F}_{i-1/2} \equiv \frac{1}{2} \sum_{p=1}^m |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda^p| \right) \mathcal{W}_{i-1/2}^p$$


The Lax-Wendroff method



Second order terms included (200 points)

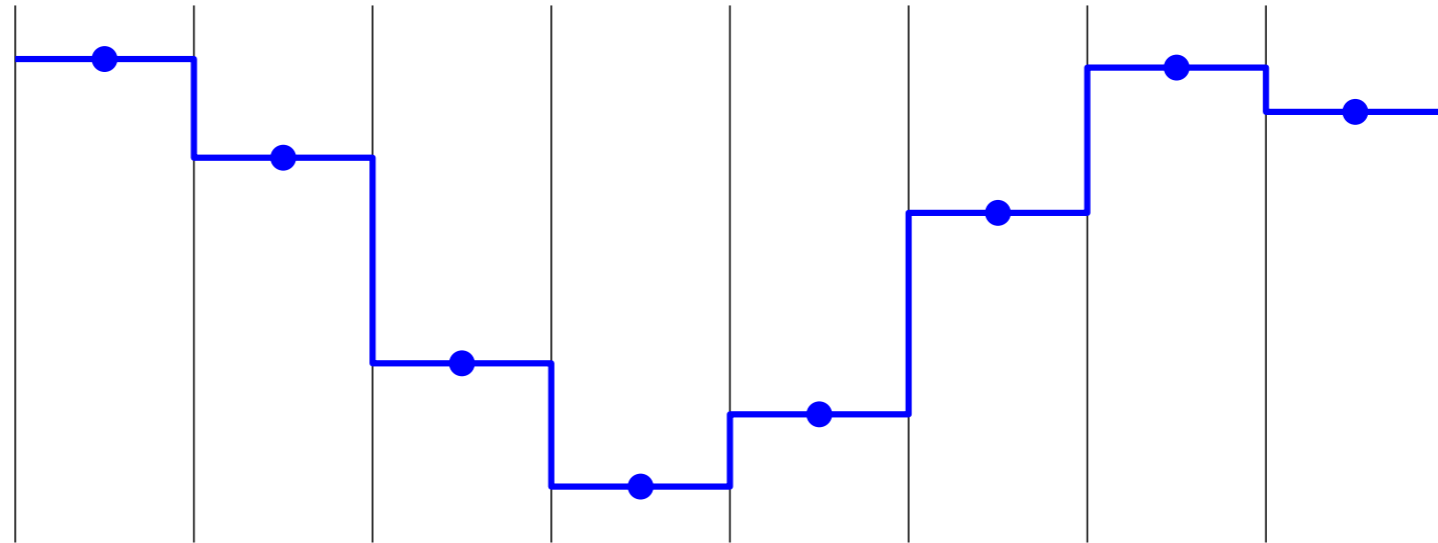
The Lax-Wendroff method



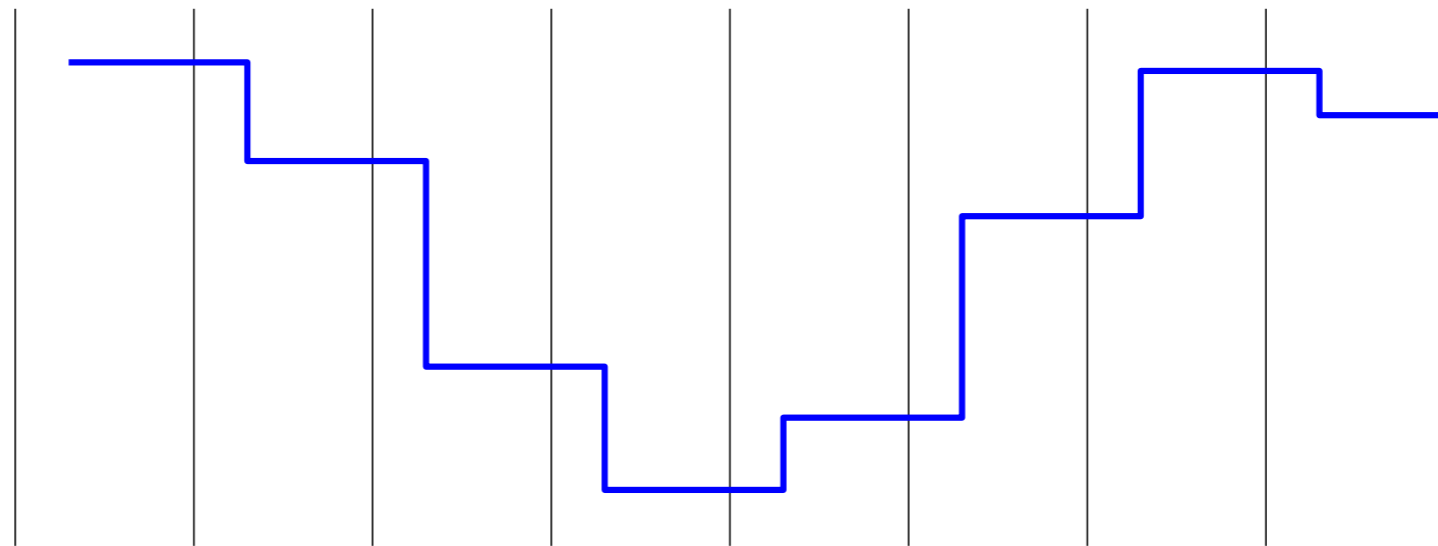
Second order terms included (200 points)

First order REA algorithm

Cell averages and piecewise constant reconstruction:

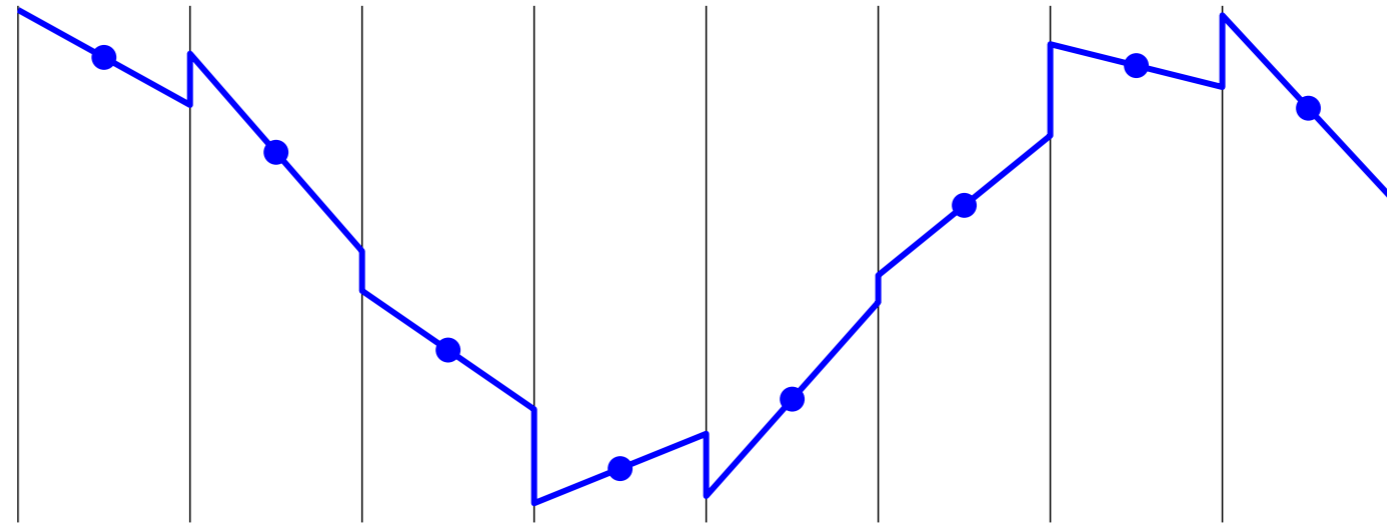


After evolution:

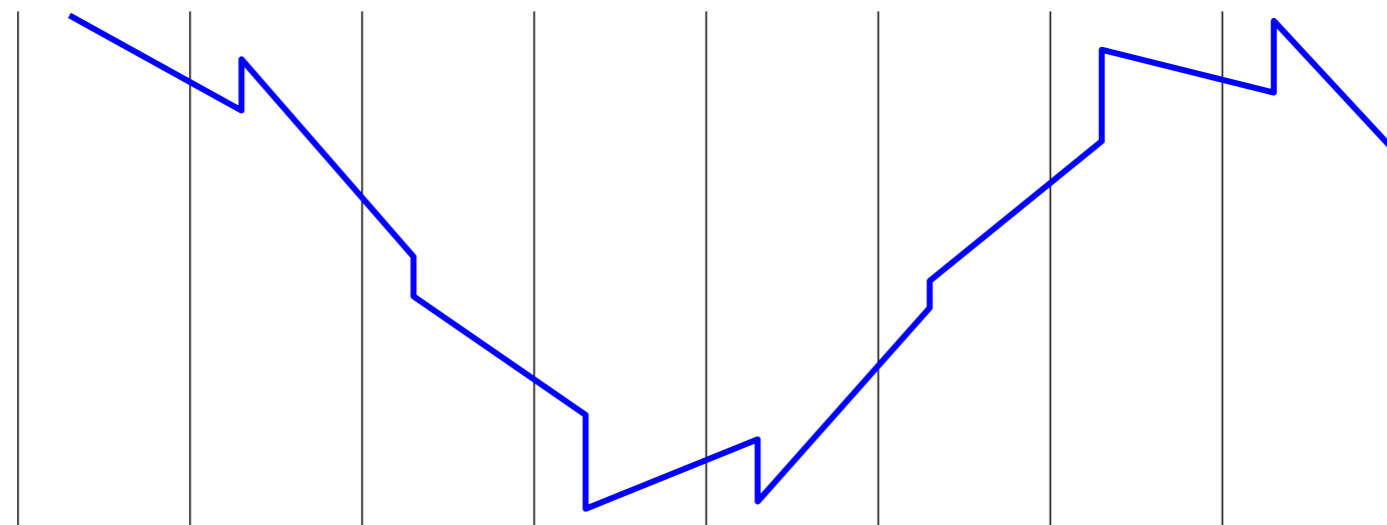


Second order REA algorithm

Cell averages and piecewise linear reconstruction:



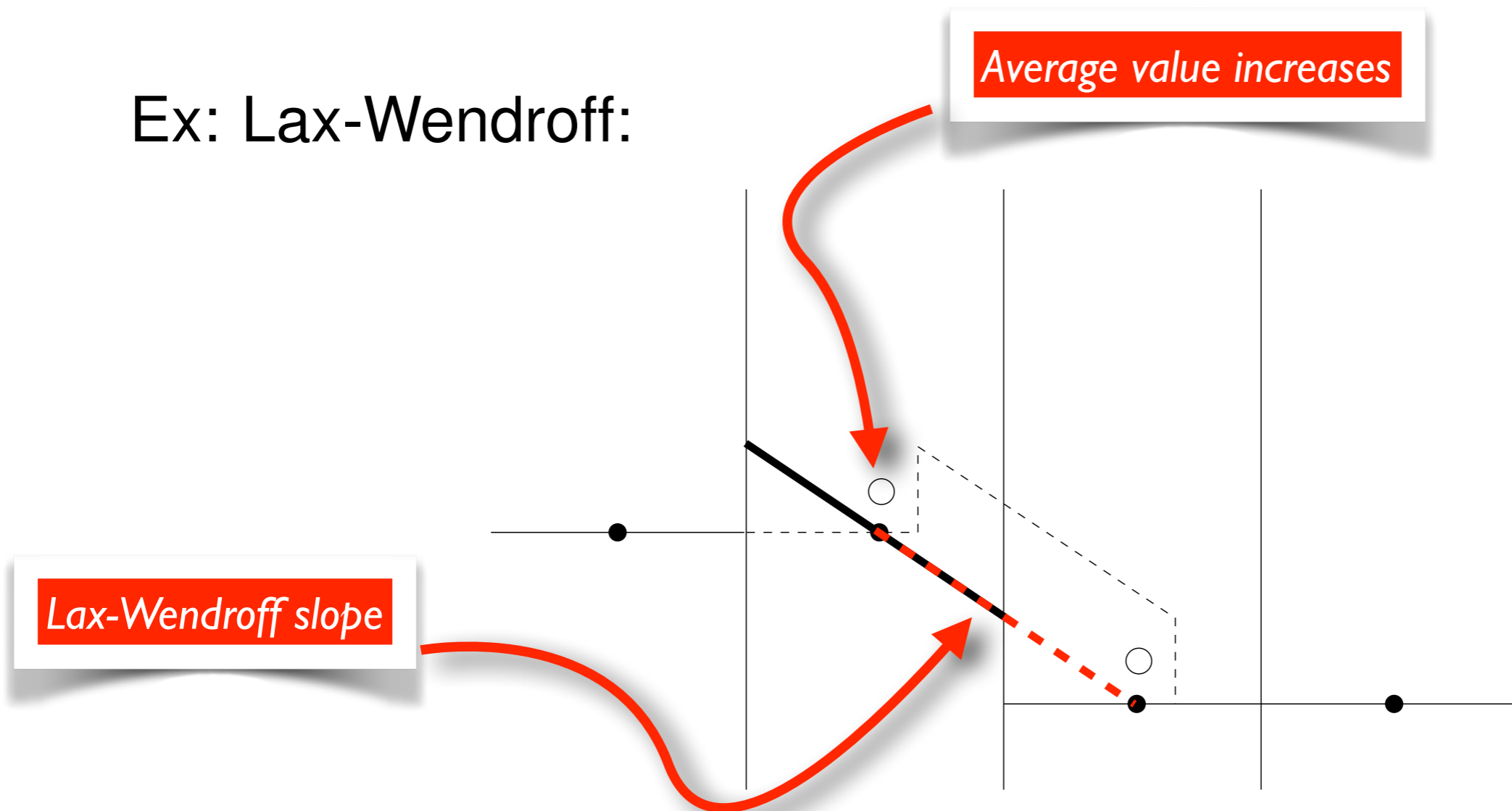
After evolution:



Oscillations

Any of these slope choices will give oscillations near discontinuities.

Ex: Lax-Wendroff:



High resolution methods

Want to use slope where solution is smooth for “second-order” accuracy.

Where solution is not smooth, adding slope corrections gives oscillations.

Limit the slope based on the behavior of the solution.

$$\sigma_i^n = \left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x} \right) \Phi_i^n.$$

$\Phi = 1 \implies$ Lax-Wendroff,

$\Phi = 0 \implies$ upwind.

Minmod slope

$$\text{minmod}(a, b) = \begin{cases} a & \text{if } |a| < |b| \text{ and } ab > 0 \\ b & \text{if } |b| < |a| \text{ and } ab > 0 \\ 0 & \text{if } ab \leq 0 \end{cases}$$

Slope:

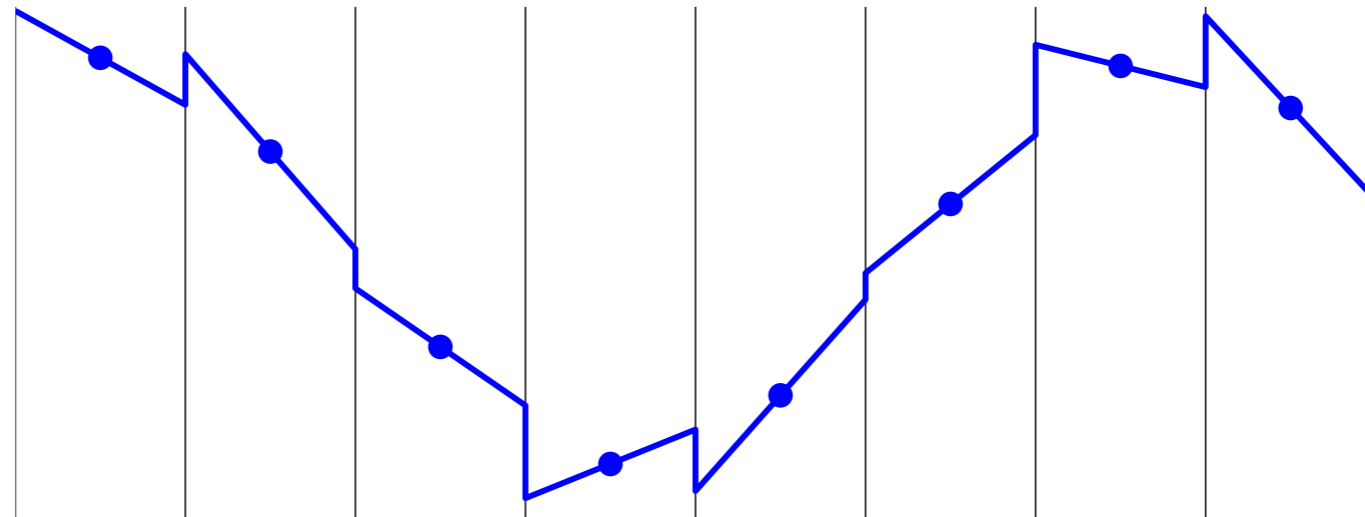
$$\begin{aligned} \sigma_i^n &= \text{minmod}\left(\frac{Q_i^n - Q_{i-1}^n}{\Delta x}, \frac{Q_{i+1}^n - Q_i^n}{\Delta x}\right) \\ &= \left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x}\right) \Phi(\theta_i^n) \end{aligned}$$

where

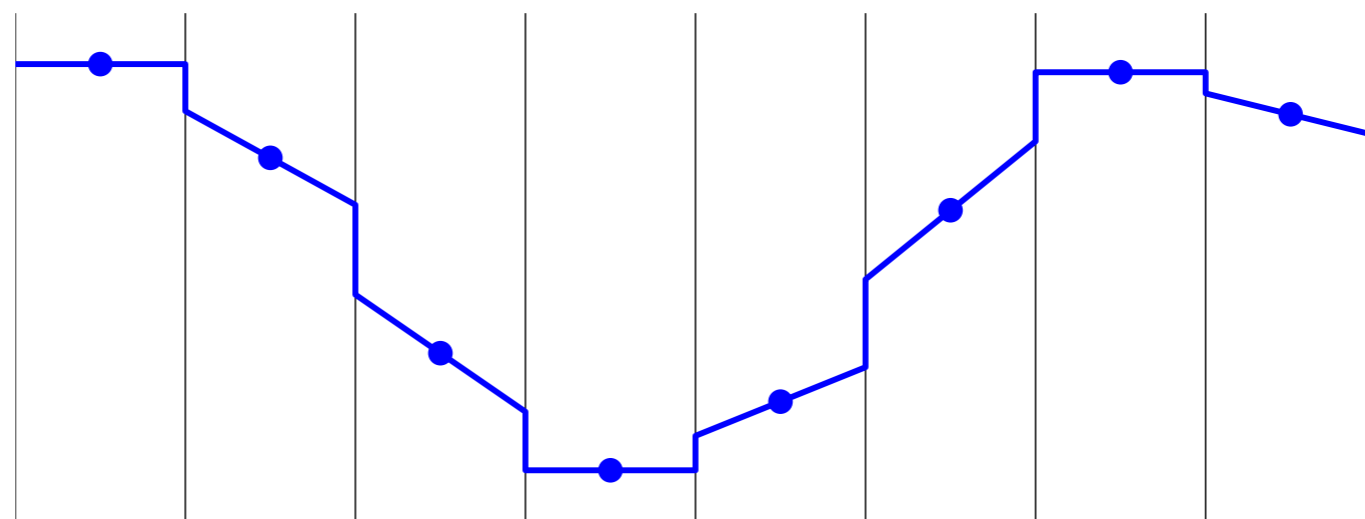
$$\begin{aligned} \theta_i^n &= \frac{Q_i^n - Q_{i-1}^n}{Q_{i+1}^n - Q_i^n} \\ \Phi(\theta) &= \text{minmod}(\theta, 1) \quad 0 \leq \Phi \leq 1 \end{aligned}$$

Minmod reconstruction

Lax-Wendroff reconstruction:



Minmod reconstruction:



Limiters

Linear methods:

$$\text{upwind : } \phi(\theta) = 0$$

$$\text{Lax-Wendroff : } \phi(\theta) = 1$$

$$\text{Beam-Warming : } \phi(\theta) = \theta$$

$$\text{Fromm : } \phi(\theta) = \frac{1}{2}(1 + \theta)$$

*Examples of limiters
available in Clawpack*



High-resolution limiters:

$$\text{minmod : } \phi(\theta) = \text{minmod}(1, \theta)$$

$$\text{superbee : } \phi(\theta) = \max(0, \min(1, 2\theta), \min(2, \theta))$$

$$\text{MC : } \phi(\theta) = \max(0, \min((1 + \theta)/2, 2, 2\theta))$$

$$\text{van Leer : } \phi(\theta) = \frac{\theta + |\theta|}{1 + |\theta|}$$

Hierarchy of methods for systems

Waves

$$\mathcal{W}_{i-1/2}^p \equiv \alpha^p r^p$$


Fluctuations

$$\mathcal{A}^+ \Delta Q_{i-1/2} \equiv \sum_{p=1}^m (\lambda^p)^+ \mathcal{W}_{i-1/2} \quad \mathcal{A}^- \Delta Q_{i-1/2} \equiv \sum_{p=1}^m (\lambda^p)^- \mathcal{W}_{i-1/2}$$

Upwind

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} (\mathcal{A}^+ \Delta Q_{i-1/2} + \mathcal{A}^- \Delta Q_{i+1/2})$$

Lax Wendroff

$$Q_i^{n+1} = Q_i^n - (\text{upwind}) - \frac{\Delta t}{\Delta x} (\mathcal{F}_{i+1/2} - \mathcal{F}_{i-1/2})$$


High Resolution

$$Q_i^{n+1} = Q_i^n - (\text{upwind}) - \frac{\Delta t}{\Delta x} (\tilde{\mathcal{F}}_{i+1/2} - \tilde{\mathcal{F}}_{i-1/2})$$

$$\mathcal{F}_{i-1/2} \equiv \frac{1}{2} \sum_{p=1}^m |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda^p| \right) \mathcal{W}_{i-1/2}^p$$

$$\tilde{\mathcal{F}}_{i-1/2} \equiv \frac{1}{2} \sum_{p=1}^m |\lambda^p| \left(1 - \frac{\Delta t}{\Delta x} |\lambda^p| \right) \Phi \mathcal{W}_{i-1/2}^p$$

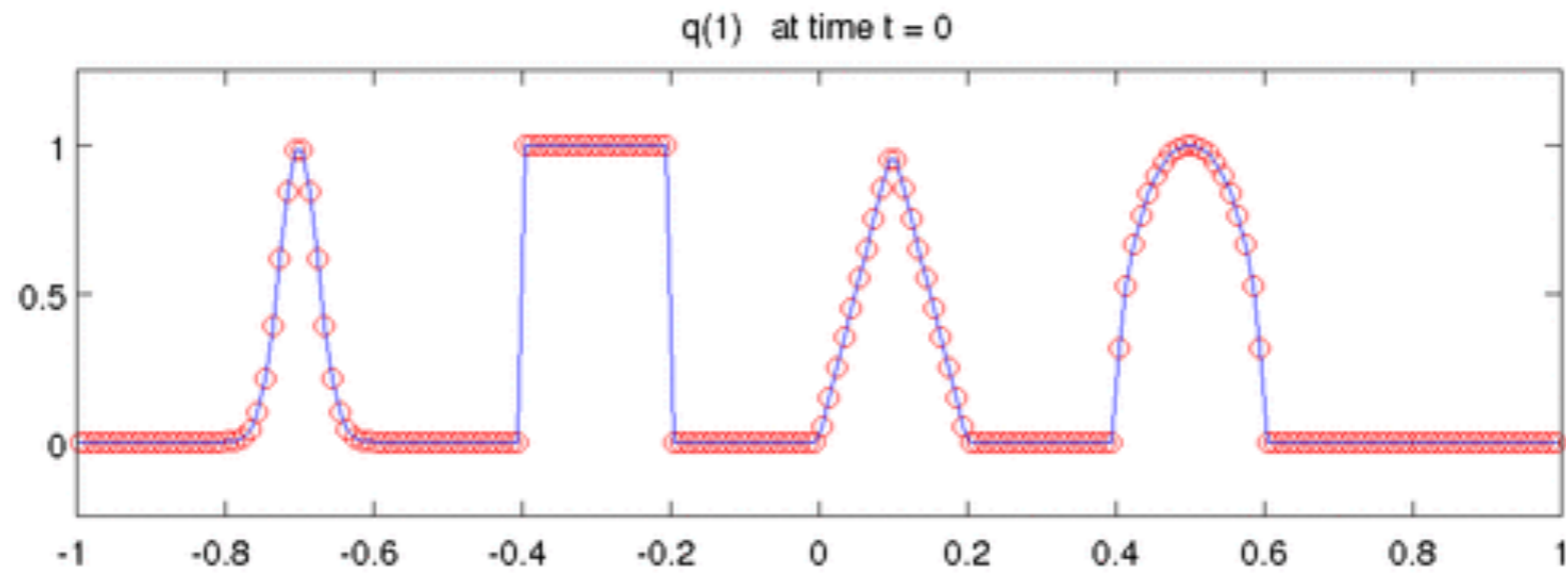
limited wave



High resolution methods

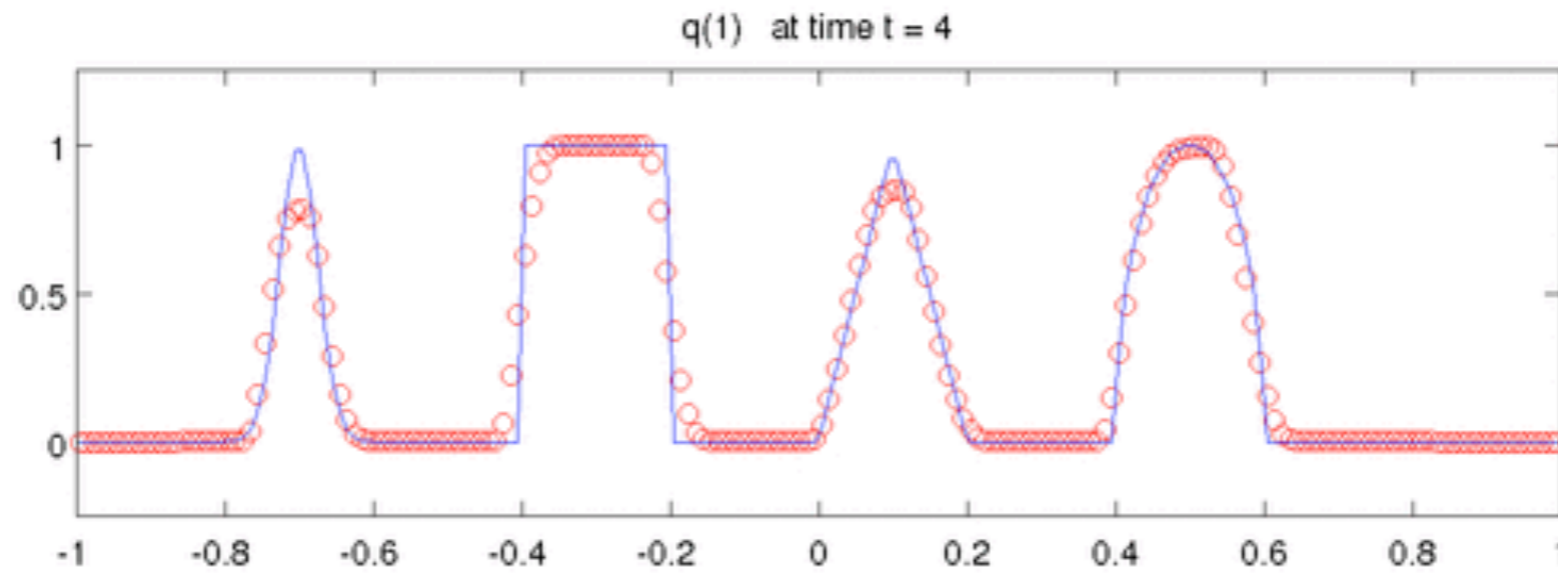
- Wave limiters reduce oscillations near discontinuities, but preserve second order accuracy in smooth regions
- Methods are no longer formally second order accurate, but are “high resolution”.
- Often magnitude of the error is reduced by the use of limiters,
- Useful even for linear problems such as advection
- Clawpack has several limiters available
- Limiters only affect second order correction terms; first order method does not use limiters

High resolution methods



Second order method with limiter (200 points)

High resolution methods



Second order method with limiter (200 points)

What next?

- How do these Riemann solvers make it into an actual code? Clawpack is based on solving Riemann problems.
- Do we actually solve the non-linear problem at every grid cell interface? No! One can use approximate Riemann solvers.
- How accurate are these methods? (second order, or high resolution with limiters)
- Well-balancing
- What do we do in two-dimensions?
- Adaptive mesh refinement?