# Conservation Laws and Finite Volume Methods

## AMath 574
## Winter Quarter, 2011

Randall J. LeVeque
Applied Mathematics
University of Washington

January 26, 2011

# Outline

Today:

- Lax-Wendroff, dispersion
- High resolution methods

Friday:

- Clawpack Plotting

Monday:

- Boundary conditions
- Multi-dimensional

Reading: Chapters 7, 18, 19

Plotting documentation:
**http://www.clawpack.org/users/plotting.html**

# Modified Equations

The upwind method

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

gives a first-order accurate approximation to $q_t + uq_x = 0$.

But it gives a second-order approximation to

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

This is an advection-diffusion equation.

Indicates that the numerical solution will diffuse.

Note: coefficient of diffusive term is $O(\Delta x)$.

# Modified Equations

The upwind method

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

gives a first-order accurate approximation to $q_t + uq_x = 0$.

But it gives a second-order approximation to

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

This is an advection-diffusion equation.

Indicates that the numerical solution will diffuse.

Note: coefficient of diffusive term is $O(\Delta x)$.

Note: No diffusion if $\frac{u\Delta t}{\Delta x} = 1 \quad (Q_i^{n+1} = Q_{i-1}^n$ exactly).

# Lax-Wendroff

Second-order accuracy?

Taylor series:

$$q(x, t+\Delta t) = q(x,t) + \Delta t q_t(x,t) + \frac{1}{2}\Delta t^2 q_{tt}(x,t) + \cdots$$

From $q_t = -Aq_x$ we find $q_{tt} = A^2 q_{xx}$.

$$q(x, t+\Delta t) = q(x,t) - \Delta t A q_x(x,t) + \frac{1}{2}\Delta t^2 A^2 q_{xx}(x,t) + \cdots$$

Replace $q_x$ and $q_{xx}$ by centered differences:

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

# Modified Equation for Lax-Wendroff

The Lax-Wendroff method

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

gives a second-order accurate approximation to $q_t + uq_x = 0$.

But it gives a third-order approximation to

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

This has a dispersive term with $O(\Delta x^2)$ coefficient.

Indicates that the numerical solution will become oscillatory.

# Beam-Warming method

Taylor series for second order accuracy:

$$q(x, t + \Delta t) = q(x, t) - \Delta t A q_x(x, t) + \frac{1}{2}\Delta t^2 A^2 q_{xx}(x, t) + \cdots$$

Replace $q_x$ and $q_{xx}$ by one-sided differences:

$$\begin{aligned} Q_i^{n+1} = Q_i^n &- \frac{\Delta t}{2\Delta x} A(3Q_i^n - 4Q_{i-1}^n + Q_{i-2}^n) \\ &+ \frac{1}{2}\frac{\Delta t^2}{\Delta x^2} A^2(Q_i^n - 2Q_{i-1}^n + Q_{i-2}^n) \end{aligned}$$

# Beam-Warming method

Taylor series for second order accuracy:

$$q(x, t + \Delta t) = q(x, t) - \Delta t A q_x(x, t) + \frac{1}{2}\Delta t^2 A^2 q_{xx}(x, t) + \cdots$$

Replace $q_x$ and $q_{xx}$ by one-sided differences:

$$\begin{aligned} Q_i^{n+1} = Q_i^n &- \frac{\Delta t}{2\Delta x} A(3Q_i^n - 4Q_{i-1}^n + Q_{i-2}^n) \\ &+ \frac{1}{2}\frac{\Delta t^2}{\Delta x^2} A^2(Q_i^n - 2Q_{i-1}^n + Q_{i-2}^n) \end{aligned}$$

CFL condition: $0 \le \lambda^p \le 2$ for all eigenvalues.

This is also the stability limit (von Neumann analysis).

# First-order REA Algorithm

1. Reconstruct a piecewise constant function $\tilde{q}^n(x, t_n)$ defined for all $x$, from the cell averages $Q_i^n$.

$$\tilde{q}^n(x, t_n) = Q_i^n \quad \text{for all } x \in \mathcal{C}_i.$$

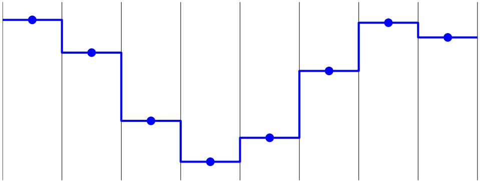
2. Evolve the hyperbolic equation exactly (or approximately) with this initial data to obtain $\tilde{q}^n(x, t_{n+1})$ a time $\Delta t$ later.

3. Average this function over each grid cell to obtain new cell averages

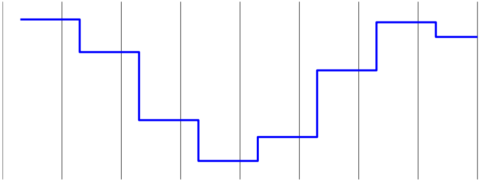
$$Q_i^{n+1} = \frac{1}{\Delta x} \int_{\mathcal{C}_i} \tilde{q}^n(x, t_{n+1})\, dx.$$

# First-order REA Algorithm

Cell averages and piecewise constant reconstruction:



After evolution:



# Second-order REA Algorithm

1. Reconstruct a piecewise linear function $\tilde{q}^n(x, t_n)$ defined for all $x$, from the cell averages $Q_i^n$.

$$\tilde{q}^n(x, t_n) = Q_i^n + \sigma_i^n(x - x_i) \quad \text{for all } x \in \mathcal{C}_i.$$

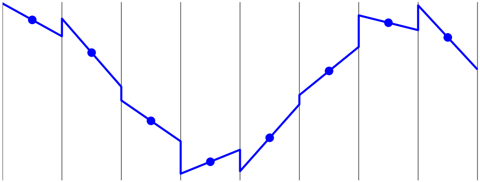
2. Evolve the hyperbolic equation exactly (or approximately) with this initial data to obtain $\tilde{q}^n(x, t_{n+1})$ a time $k$ later.

3. Average this function over each grid cell to obtain new cell averages

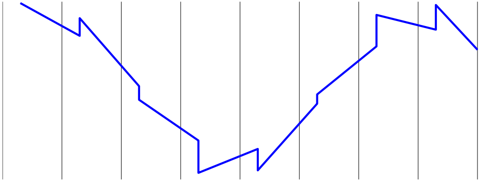
$$Q_i^{n+1} = \frac{1}{\Delta x} \int_{\mathcal{C}_i} \tilde{q}^n(x, t_{n+1})\, dx.$$

# Second-order REA Algorithm

Cell averages and piecewise linear reconstruction:



After evolution:



# Choice of slopes

$$\tilde{Q}^n(x,t_n) = Q_i^n + \sigma_i^n(x - x_i) \qquad \text{for } x_{i-1/2} \le x < x_{i+1/2}.$$

Applying REA algorithm gives:

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

Choice of slopes:

Centered slope: $\sigma_i^n = \dfrac{Q_{i+1}^n - Q_{i-1}^n}{2\Delta x}$ (Fromm)

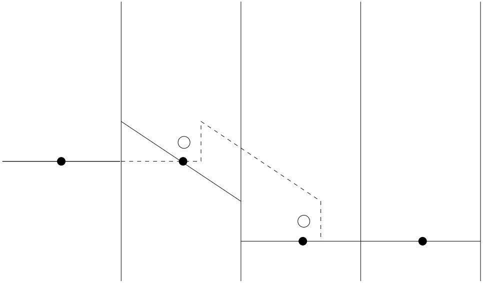
Upwind slope: $\sigma_i^n = \dfrac{Q_i^n - Q_{i-1}^n}{\Delta x}$ (Beam-Warming)

Downwind slope: $\sigma_i^n = \dfrac{Q_{i+1}^n - Q_i^n}{\Delta x}$ (Lax-Wendroff)

# 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.

# 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.

Might also take $1 < \Phi \leq 2$ to sharpen discontinuities.

# Minmod slope

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

Slope:

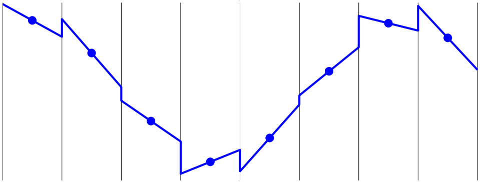
$$\begin{aligned} \sigma_i^n &= \text{minmod}((Q_i^n - Q_{i-1}^n)/\Delta x,\ \ (Q_{i+1}^n - Q_i^n)/\Delta x) \\ &= \left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x}\right)\Phi(\theta_i^n) \end{aligned}$$

where

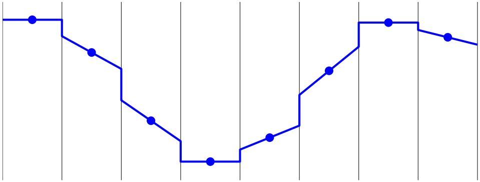
$$\begin{aligned} \theta_i^n &= \frac{Q_i^n - Q_{i-1}^n}{Q_{i+1}^n - Q_i^n} \\ \Phi(\theta) &= \text{minmod}(\theta, 1) \qquad 0 \le \Phi \le 1 \end{aligned}$$

# Piecewise linear reconstructions

Lax-Wendroff reconstruction:



Minmod reconstruction:



# TVD Methods

Total variation:

$$TV(Q) = \sum_i |Q_i - Q_{i-1}|$$

For a function, $TV(q) = \int |q_x(x)|\, dx$.

A method is Total Variation Diminishing (TVD) if

$$TV(Q^{n+1}) \le TV(Q^n).$$

If $Q^n$ is monotone, then so is $Q^{n+1}$.

No spurious oscillations generated.

Gives a form of stability useful for proving convergence, also for nonlinear scalar conservation laws.

# TVD REA Algorithm

1. Reconstruct a piecewise linear function $\tilde{q}^n(x, t_n)$ defined for all $x$, from the cell averages $Q_i^n$.

$$\tilde{q}^n(x, t_n) = Q_i^n + \sigma_i^n(x - x_i) \quad \text{for all } x \in \mathcal{C}_i$$

   with the property that $TV(\tilde{q}^n) \leq TV(Q^n)$.

2. Evolve the hyperbolic equation exactly (or approximately) with this initial data to obtain $\tilde{q}^n(x, t_{n+1})$ a time $k$ later.

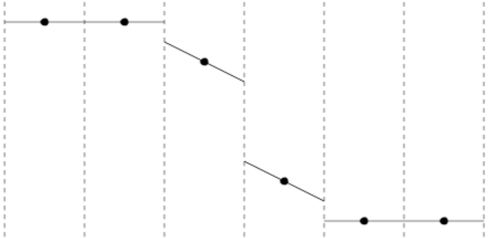
3. Average this function over each grid cell to obtain new cell averages

$$Q_i^{n+1} = \frac{1}{\Delta x} \int_{\mathcal{C}_i} \tilde{q}^n(x, t_{n+1})\, dx.$$

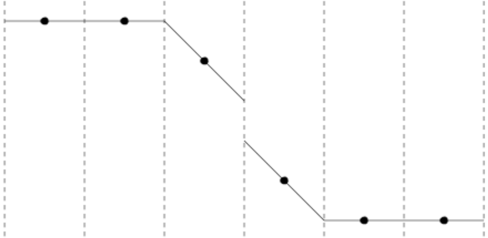
Note: Steps 2 and 3 are always TVD.

# Choosing $1 < \Phi \leq 2$ to sharpen jumps

Minmod reconstruction:



Doubling the slopes is possible without loss of TVD:



# Some popular limiters

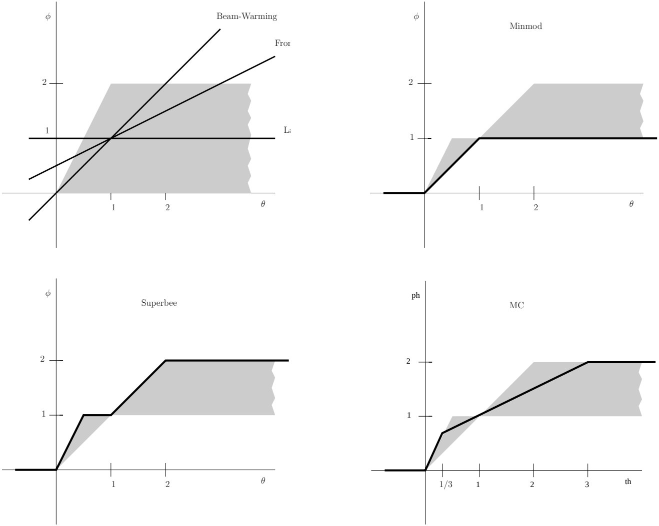
Linear methods:

$$\begin{aligned}
\text{upwind}: &\quad \phi(\theta) = 0 \\
\text{Lax-Wendroff}: &\quad \phi(\theta) = 1 \\
\text{Beam-Warming}: &\quad \phi(\theta) = \theta \\
\text{Fromm}: &\quad \phi(\theta) = \frac{1}{2}(1+\theta)
\end{aligned}$$

High-resolution limiters:

$$\begin{aligned}
\text{minmod}: &\quad \phi(\theta) = \text{minmod}(1,\theta) \\
\text{superbee}: &\quad \phi(\theta) = \max(0,\ \min(1,2\theta),\ \min(2,\theta)) \\
\text{MC}: &\quad \phi(\theta) = \max(0,\ \min((1+\theta)/2,\ 2,\ 2\theta)) \\
\text{van Leer}: &\quad \phi(\theta) = \frac{\theta+|\theta|}{1+|\theta|}
\end{aligned}$$

# Sweby diagram

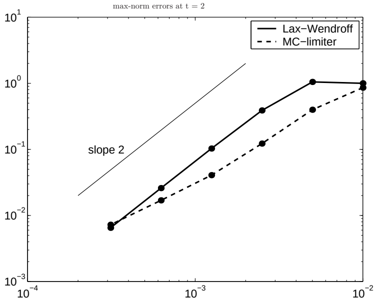


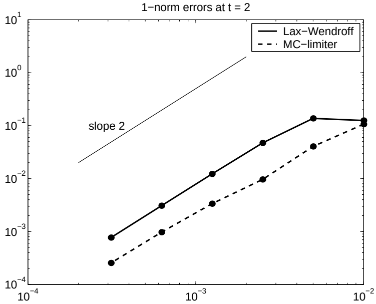
# Order of accuracy isn't everything

Comparison of Lax-Wendroff and a high-resolution method on linear advection equation with smooth data.

The high-resolution method is not formally second-order accurate, but is more accurate on realistic grids.

Crossover in the max-norm is at $2800$ grid points.



# Numerical Experiments

Experiment with the codes available from

`$CLAW/book/chap6/compareadv`
`$CLAW/book/chap6/wavepacket`

Use `clawdata.order = 2` and one of the following:

- `clawdata.mthlim = [0]`: Lax-Wendroff
- `clawdata.mthlim = [1]`: minmod
- `clawdata.mthlim = [2]`: superbee
- `clawdata.mthlim = [3]`: van Leer
- `clawdata.mthlim = [4]`: Monotonized Centered (MC)
- `clawdata.mthlim = [5]`: Beam-Warming

See Figures 6.2 and 6.3 for sample results.

# Slope limiters and flux limiters

Slope limiter formulation for advection:

$$\tilde{Q}^n(x, t_n) = Q_i^n + \sigma_i^n(x - x_i) \qquad \text{for } x_{i-1/2} \le x < x_{i+1/2}.$$

Applying REA algorithm gives:

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

Flux limiter formulation:

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

with flux

$$F_{i-1/2}^n = uQ_{i-1}^n + \frac{1}{2}u(\Delta x - u\Delta t)\sigma_{i-1}^n.$$

# Wave limiters

Let $\mathcal{W}_{i-1/2} = Q_i^n - Q_{i-1}^n$.

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

Lax-Wendroff:

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

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

High-resolution method:

$$\tilde{F}_{i-1/2} = \frac{1}{2}\left(1 - \left|\frac{u\Delta t}{\Delta x}\right|\right)|u|\widetilde{\mathcal{W}}_{i-1/2}$$

where $\widetilde{\mathcal{W}}_{i-1/2} = \Phi_{i-1/2}\mathcal{W}_{i-1/2}$.

# Extension to linear systems

Approach 1: Diagonalize the system to

$$v_t + \Lambda v_x = 0$$

Apply scalar algorithm to each component.

# Extension to linear systems

Approach 1: Diagonalize the system to

$$v_t + \Lambda v_x = 0$$

Apply scalar algorithm to each component.

Approach 2:

Solve the linear Riemann problem to decompose $Q_i^n - Q_{i-1}^n$ into waves.

Apply a wave limiter to each wave.

# Extension to linear systems

Approach 1: Diagonalize the system to

$$v_t + \Lambda v_x = 0$$

Apply scalar algorithm to each component.

Approach 2:

Solve the linear Riemann problem to decompose $Q_i^n - Q_{i-1}^n$ into waves.

Apply a wave limiter to each wave.

For constant-coefficient linear problems these are equivalent.

For nonlinear problems Approach 2 generalizes!

# Extension to linear systems

Approach 1: Diagonalize the system to

$$v_t + \Lambda v_x = 0$$

Apply scalar algorithm to each component.

Approach 2:

Solve the linear Riemann problem to decompose $Q_i^n - Q_{i-1}^n$ into waves.

Apply a wave limiter to each wave.

For constant-coefficient linear problems these are equivalent.

For nonlinear problems Approach 2 generalizes!

Note: Limiters are applied to waves or characteristic components, not to original variables.