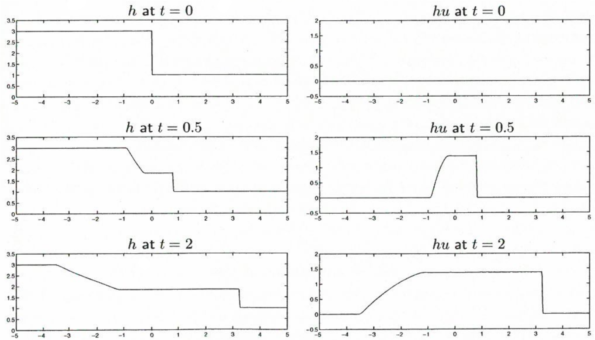
Finite volume methods for advection equations and hyperbolic systems

version 4, June 2023

Ed Bueler, UAF



Outline

0. overview and scope
1. scalar advection equation
2. linear systems and Riemann solvers
3. high-resolution methods (slope-limiters)
4. nonlinear conservation laws
5. advection again, but 2D spatial

overview

- numerical solutions of systems of first-order, time-dependent PDEs
- hyperbolic PDEs:

1. scalar advection
$$q_t + a q_x = 0$$

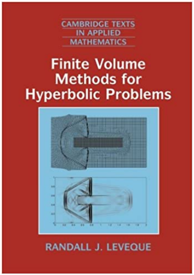
2. linear systems
$$\mathbf{q}_t + A\,\mathbf{q}_x = 0$$

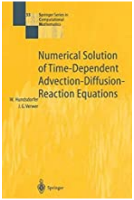
5. scalar advection in 2D
$$q_t + (uq)_x + (vq)_y = 0$$

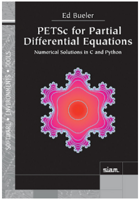
4. conservation-law systems
$$\mathbf{q}_t + \mathbf{F}(t,x,\mathbf{q})_x = \mathbf{g}(t,x,\mathbf{q})$$

- finite volume (FV) discretizations
    - a genuine introduction to FV methods
- section 3 is about "high-resolution" flux discretizations

- R. J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002
- W. Hundsdorfer and J. G. Verwer, *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Springer, 2003
- E. Bueler, *PETSc for Partial Differential Equations*, SIAM Press, 2020







slides, movies, and codes

- for these slides (as `.pdf`), and the movies (as `.mp4`), see the `fvolume/` directory in this Github repository:

  `https://github.com/bueler/slide-teach`

- for the C codes which generated all the results, see the `c/riemann/` directory in this Github repository:

  `https://github.com/bueler/p4pdes-next`

- C programming is NOT required for appreciating these notes!

visual example 1: merely a numerical solution

- before getting to numerical solutions, two show-and-tell movies
- consider advection equation for scalar density $q(t, x)$:

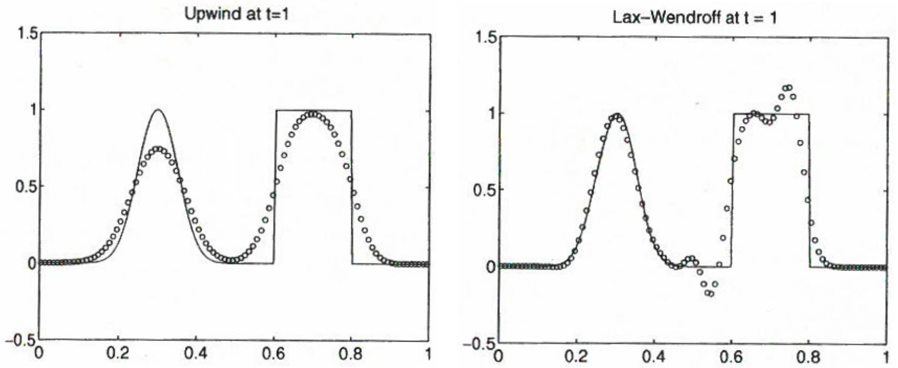
$$q_t + a\, q_x = 0$$

  with speed $a = 1$, initial condition $q(0, x)$ known, and periodic boundary conditions on $0 \le x \le 1$
- movie of numerical solution for $0 \le t \le 1$
  - initial shape is transported rightward, from initial position back to same position

SHOW ADVECTION MOVIE $\leftarrow$ `movies/advection.mp4`

visual example 1: numerical *and exact* solutions

- was it clear what the movie showed?
- figures below are better: they show numerical and exact solutions



visual example 2: merely a numerical solution

- shallow water equations:

$$
\begin{aligned}
h_t + (hu)_x &= 0 \\
(hu)_t + \left(hu^2 + \frac{g}{2}h^2\right)_x &= 0
\end{aligned}
$$

  - coupled
  - hyperbolic
  - nonlinear
- suppose initial condition is a “hump” on $x \in [-5, 5]$
  - $h(0, x) = ae^{-bx^2}$, $u(0, x) = 0$
  - vertical displacement in the center of the domain
  - simplest model for a tsunami generated in middle of ocean
- movie of numerical solution for $0 \leq t \leq 3$

SHOW SHALLOW WATER “HUMP” MOVIE $\leftarrow$ `movies/hump.mp4`

multiple roles for exact solutions

- exact solutions are rare but valuable!
- in these slides, exact solutions have two roles:
  1. for *verifying* simulations
     - measure norm of difference between exact and numerical solutions
     - precise "mathematical engineering" of numerical solvers
  2. as *"Riemann solvers"* for hyperbolic systems
     - used locally in constructing the numerical scheme
     - solutions for discontinuous initial conditions
     - explaining this mystery is a major purpose of my talk!

my context: high performance PDEs

- I am interested in high performance solutions of PDEs
- all examples in these slides use fast C code
  - but more complicated than Matlab or Python
- my C codes call the Portable Extensible Toolkit for Scientific computing:

  `https://petsc.org`

  - a mathematical library for high-performance computing from the Department of Energy's Argonne National Laboratory
- the run which generated the previous shallow water movie, with 1000×361 (space$\times$time) grid, completed in 0.3 seconds
- speed is more critical for problems with 2D and 3D space
  - parallelizability also important there

please ask questions

- the rest of the talk is about the math not the movies
  - but with many figures to explain concepts
- PLEASE ask lots of questions, about any topic here at all
  - slowing me down is a *good* thing!
  - I'll try to watch the zoom chat, too
- feel free to email any time after the talk:
  - `elbueler@alaska.edu`

Outline

one-way advection

- next few slides should be an easy review
- consider the *scalar advection PDE* for $q(t,x)$:

$$q_t + a q_x = 0$$

- if a is constant and we have a smooth initial condition $q(t,0) = f(x)$ then the solution is

$$q(t,x) = f(x - at),$$

  because, by the chain rule

$$q_t + a q_x = -af'(x-at) + af'(x-at) = 0$$

- the solution $q(t,x) = f(x - at)$ is a “movie”: the graph of $f(x)$ is translated distance by at, to the right, in time t
  - even if a and/or t are negative
  - a is the speed of the motion

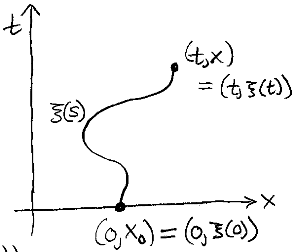
solution by characteristics

- but what about with variable speed $a(t, x)$?

$$q_t + a(t,x)q_x = 0, \qquad q(0,x) = f(x)$$

- we need the idea of a *characteristic curve* (ODE):

$$\frac{d\xi}{ds} = a(s, \xi(s))$$



- for a solution $q(t, x)$, we calculate $\left[\xi' = \frac{d\xi}{ds}\right]$

$$\begin{aligned}\Big(q(s,\xi(s))\Big)' &= q_t(s,\xi(s)) + q_x(s,\xi(s))\xi'(s) \\ &= q_t(s,\xi(s)) + q_x(s,\xi(s))a(s,\xi(s)) \\ &= 0,\end{aligned}$$

- conclude: $q(t, \xi(t)) = q(0, \xi(0))$
- *solution by characteristics*: $q(t, x)$ has the same value as $f(x_0)$ if $\xi(s)$ is a characteristic curve that ends at (t, x) and starts at $(0, x_0)$

solution by characteristics 2

- the method can be extended to the nonlinear equation

$$q_t + a(t,x)q_x = g(t,x,q), \qquad q(0,x) = f(x)$$

  - *a* is the speed of the characteristic curve
  - *g* is the *source term*
  - *idea*: the solution *q* changes at rate *g* along the characteristic
  - if $g = 0$ then *q* is constant along the characteristic

- now we have a pair of ODEs to solve:

$$\begin{aligned} \xi'(s) &= a(s,\xi(s)) \\ \omega'(s) &= g(s,\xi(s),\omega(s)) \end{aligned}$$

- *solution by characteristics*:
  i) from 1st ODE, find characteristic $\xi(s)$ through $(0, x_0)$ and (t, x)
  ii) solve 2nd ODE with initial condition $\omega(0) = f(x_0)$
  iii) then $q(t,x) = \omega(t)$

- main idea about advection PDEs:

information travels along the characteristics

upwind scheme for one-way advection

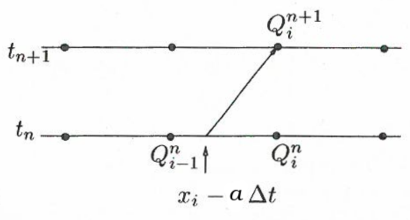
- we may apply the *upwind* scheme to $q_t + aq_x = 0$ (case $a > 0$):

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} + a\frac{Q_i^n - Q_{i-1}^n}{\Delta x} = 0$$

- equivalently, solve for the new value at t_{n+1}:

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

where $\ell(x)$ linearly interpolates between (x_{i-1}, Q_{i-1}^n) and (x_i, Q_i^n)

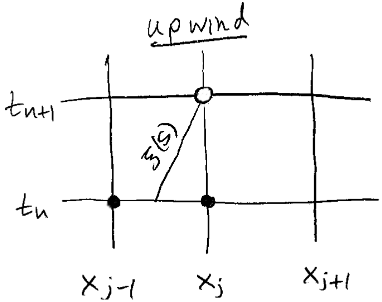


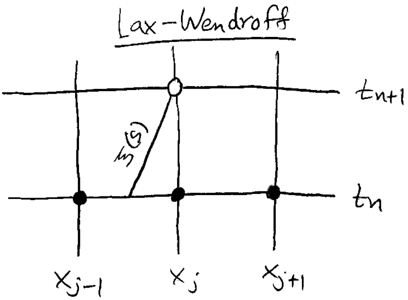
- interpolate ℓ where the characteristic through (x_i, Q_i^{n+1}) hits the t_n line
- except we must require *interpolation* instead of extrapolation: $\dfrac{|a|\Delta t}{\Delta x} \leq 1$

upwind and Lax-Wendroff schemes

- while upwind uses linear interpolation using two points, ...
- the *Lax-Wendroff* scheme uses quadratic interpolation with three points
- formulas: if $\nu = a\Delta t/\Delta x$ then

$$
\begin{aligned}
Q_i^{n+1} &= \nu Q_{i-1}^n + (1-\nu)\, Q_i^n && \text{upwind} \\
Q_i^{n+1} &= \tfrac{1}{2}\nu(1+\nu) Q_{i-1}^n + \left(1-\nu^2\right) Q_i^n + \tfrac{1}{2}\nu(1-\nu) Q_{i+1}^n && \text{LW}
\end{aligned}
$$





four derivations of Lax-Wendroff

- by the way ...
- there are multiple interpretations of Lax-Wendroff:
  1. quadratic interpolation (previous)
  2.
  3.
  4.

four derivations of Lax-Wendroff

- by the way . . .
- there are multiple interpretations of Lax-Wendroff:
  1. quadratic interpolation (previous)
  2. use spatial difference on $O(\Delta t^2)$ term in Taylor series
  3.
  4.

- by the way ...
- there are multiple interpretations of Lax-Wendroff:
  1. quadratic interpolation (previous)
  2. use spatial difference on $O(\Delta t^2)$ term in Taylor series
  3. half steps of Lax-Friedrichs followed by leap-frog
  4.

four derivations of Lax-Wendroff

- by the way ...
- there are multiple interpretations of Lax-Wendroff:
    1. quadratic interpolation (previous)
    2. use spatial difference on $O(\Delta t^2)$ term in Taylor series
    3. half steps of Lax-Friedrichs followed by leap-frog
    4. *revealed later in this talk*

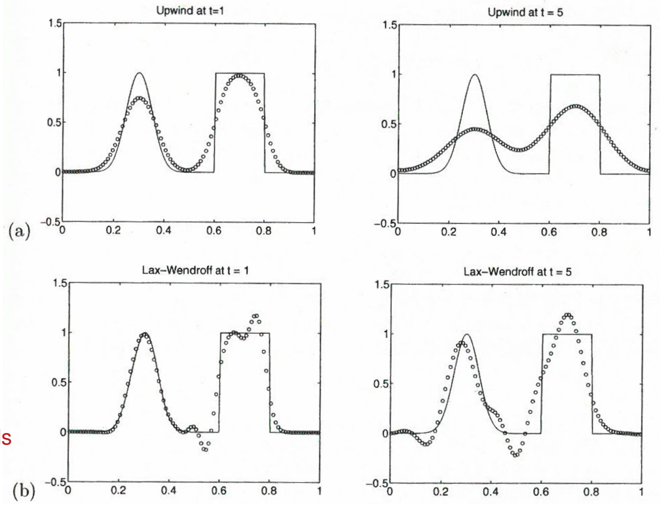
stability

- in a very early paper, Courant, Friedrichs, and Lewy (1928) gave a criterion for stability of numerical methods on hyperbolic PDEs
- CFL criterion: the characteristic through the new location (t_{n+1}, x_i) must be in the numerical domain of dependence of the scheme at t_n, i.e.

$$\nu_{CFL} = \frac{|a|\Delta t}{\Delta x} \le 1 \qquad \iff \qquad \Delta t \le \frac{\Delta x}{|a|}$$

- we need CFL so that upwind and Lax-Wendroff formulas are *interpolations* not *extrapolations*
  - the errors from extrapolation would propagate forward as exponential growth, i.e. *unstably*
- CFL is a *necessary* condition for stability (and thus for convergence)
- CFL applies to any finite difference, finite volume, or finite element scheme for a hyperbolic PDE

- suppose same problem as in earlier movie: $q_t + aq_x = 0$, $a = 1$, $0 \le x \le 1$, and periodic boundary conditions



issues:

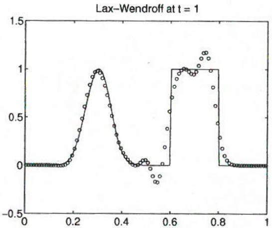
oversmoothing

spurious oscillation

loss of original bounds

how to do better? ...the history

- upwind and Lax-Wendroff methods were obvious technology by $\sim$1960
- but the results suffered from three diseases:
  - oversmoothing (upwind)
  - spurious oscillation (Lax-Wendroff)
  - loss of original solution bounds (Lax-Wendroff)



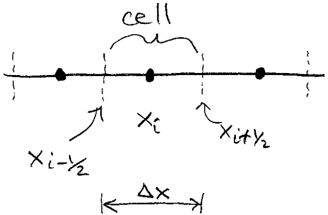
- in 1960–90s these diseases were mostly fixed ...how?:
  - reading Godunov (1959)
  - new “finite volume” thinking
  - new “Riemann solver” interpretation of the upwind method
  - new “slope-limiting” or “flux-limiting” to avoid oscillations
- this talk: make sense of these 1990s buzzwords!

the finite volume (FV) idea

- assume the problem is in flux-conservation form

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

  - $f(q) = aq$ for scalar advection $q_t + aq_x = 0$
- put on a grid $\{x_i\}$ with spacing Δx



- *cell* $=$ *finite volume*
- suppose x_i is the center of a cell, and integrate over it:

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q_t + f(q)_x \, dx = 0$$

$$\frac{d}{dt}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\, dx + f\left(q(t,x_{i+\frac{1}{2}})\right) - f\left(q(t,x_{i-\frac{1}{2}})\right) = 0$$

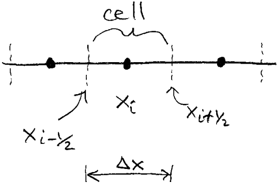
numerical quantities in an FV method

- define: $$Q_i(t) \approx \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\,dx$$
    - $Q_i(t)$ represents a cell average, not a point value
    - versus in a finite difference scheme: $Q_i(t) \approx q(t,x_i)$
    - make sure to distinguish $q(t,x_i)$ (exact) and $Q_i(t)$ (numerical)
- let $F_{i+\frac{1}{2}}(t) \approx f\left(q(t,x_{i+\frac{1}{2}})\right)$ be the flux at *cell face*
- so exact statement from last slide,

$$\frac{d}{dt}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\,dx + f\left(q(t,x_{i+\frac{1}{2}})\right) - f\left(q(t,x_{i-\frac{1}{2}})\right) = 0,$$

becomes a numerical scheme:

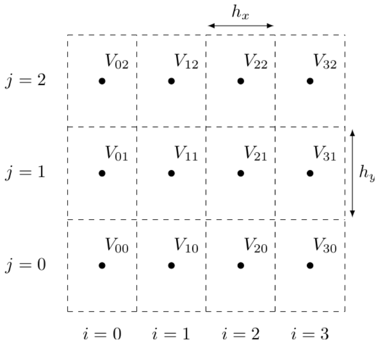
$$\Delta x \frac{dQ_i}{dt} + F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} = 0$$



- this is only a spatial discretization

just an illustration: FV in 2D

- the name “finite volumes” makes more sense in 2D or 3D
  - figure shows a structured 2D FV grid
- the “grid” in an FV scheme is often *shown* as cell centers (dots)
- . . . the solution is actually represented by an *average value* Q_{ij} *per cell*
- each cell V_{ij} is a domain of integration
- the scheme relates the cell average to the fluxes F on faces (dashed lines)
- conservation holds because each cell face has one flux value



an almost-complete FV scheme

- also put a grid on t
- use forward Euler in time
- we have this derivation of a scheme:

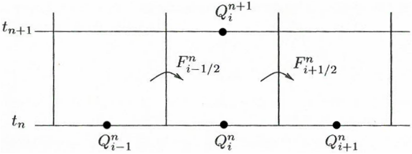
$$q_t + f(q)_x = 0$$

$$\frac{d}{dt}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\,dx + f\left(q(t,x_{i+\frac{1}{2}})\right) - f\left(q(t,x_{i-\frac{1}{2}})\right) = 0$$

$$\Delta x \frac{dQ_i}{dt} + F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}} = 0$$

$$\Delta x \left(\frac{Q_i^{n+1} - Q_i^n}{\Delta t}\right) + F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}} = 0$$

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} + \frac{F^n_{i+\frac{1}{2}} - F^n_{i-\frac{1}{2}}}{\Delta x} = 0$$

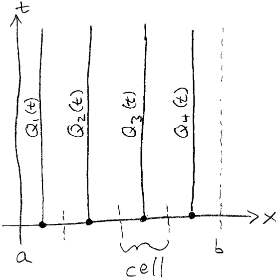


- the red equation is a computable numerical scheme *if* we have a scheme for approximating the face fluxes $F^n_{i+\frac{1}{2}}$

method-of-lines (MOL) thinking

- *but* we should not commit to forward Euler so soon!
- remember: $Q_i(t) \approx \dfrac{1}{\Delta x} \displaystyle\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\,dx$ and $F_{i+\frac{1}{2}}(t) \approx f\left(q(t, x_{i+\frac{1}{2}})\right)$
- our basic FV scheme is an ODE system in time:

$$\frac{dQ_i}{dt} + \frac{F_{i+\frac{1}{2}} - F_{i+\frac{1}{2}}}{\Delta x} = 0$$



- write as:

$$\frac{d\mathbf{Q}}{dt} = \mathbf{G}(t, \mathbf{Q}), \quad \mathbf{Q}(t) = \begin{bmatrix} Q_1(t) \\ \vdots \\ Q_J(t) \end{bmatrix}$$

- why? because good black-box ODE solvers are available
  - . . . it is not 1970 anymore, people!

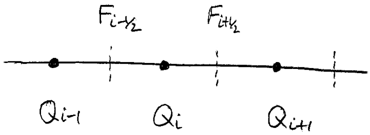
upwind as the “donor cell” method

- in FV language, upwind for $q_t + aq_x = 0$ is 3 steps:
  - i) integrate over the spatial cell (= derive the FV MOL scheme):

$$\frac{dQ_i}{dt} + \frac{F_{i+\frac{1}{2}} - F_{i-\frac{1}{2}}}{\Delta x} = 0$$

  - ii) compute flux $F(q) = aq$ from the upwind *donor cell*:

$$F_{i+\frac{1}{2}} = \begin{cases} aQ_i, & a \geq 0, \\ aQ_{i+1}, & a < 0 \end{cases}$$



  - iii) forward Euler for time stepping:

$$\frac{Q_i^{n+1} - Q_i^n}{\Delta t} + a \frac{\begin{Bmatrix} Q_i^n - Q_{i-1}^n & [a \geq 0] \\ Q_{i+1}^n - Q_i^n & [a < 0] \end{Bmatrix}}{\Delta x} = 0$$

- for ii) we will do better (“high-resolution” methods; “slope-limiters”)
  - but how to interpret “donor cell” for hyperbolic *systems*?
- for iii) we can already do better (Runge-Kutta, Matlab ODE solvers, . . .)

Outline

linear systems: examples

- consider a *linear, constant-coefficient, homogeneous system*:

$$\mathbf{q}_t + A\,\mathbf{q}_x = 0$$

  - $\mathbf{q}(t,x) \in \mathbb{R}^d$ vector-valued solution
  - $A \in \mathbb{R}^{d\times d}$ square matrix

$d = 2$ example: *acoustics* ($=$ classical 2nd-order wave equation)

$$\mathbf{q} = \begin{bmatrix} p \\ u \end{bmatrix}, \; A = \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix} \quad \implies \quad \begin{aligned} p_t + K_0 u_x &= 0 \\ u_t + \tfrac{1}{\rho_0} p_x &= 0 \end{aligned}$$

$d = 2$ example: <u>linearized</u> *shallow water equations*

$$\mathbf{q} = \begin{bmatrix} h \\ hu \end{bmatrix}, \; A = \begin{bmatrix} 0 & 1 \\ -u_0^2 + gh_0 & 2u_0 \end{bmatrix} \implies \begin{gathered} h_t + (hu)_x = 0 \\ (hu)_t + (-u_0^2 + gh_0)h_x + 2u_0(hu)_x = 0 \end{gathered}$$

example: acoustics

- $p(t,x)$ is gas pressure, $u(t,x)$ is gas velocity
- assume pressure/velocity variations are small, and density ρ_0 and compressiblity K_0 are constant
- thus linear, constant-coefficient first-order PDE system:

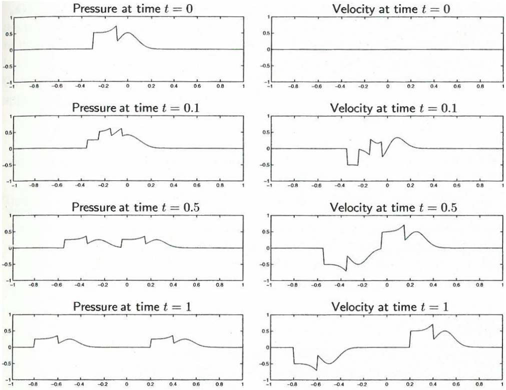
$$\begin{aligned} p_t + K_0 u_x &= 0 \\ u_t + \frac{1}{\rho_0} p_x &= 0 \end{aligned}$$

- or $\mathbf{q}_t + A\mathbf{q}_x = 0$ where

$$\mathbf{q} = \begin{bmatrix} p \\ u \end{bmatrix}, \quad A = \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix}$$

- or in 2nd-order form with $c^2 = \frac{K_0}{\rho_0}$:

$$\begin{aligned} p_{tt} &= c^2 p_{xx} \\ u_{tt} &= c^2 u_{xx} \end{aligned}$$



linear system we can already handle

- consider boring decoupled system:

$$\mathbf{q} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}, \; A = \begin{bmatrix} a & 0 & 0 \\ 0 & b & 0 \\ 0 & 0 & c \end{bmatrix} \quad \Longrightarrow \quad \begin{aligned} u_t + a u_x &= 0 \\ v_t + b v_x &= 0 \\ w_t + c w_x &= 0 \end{aligned}$$

- *method*: upwind on each equation independently

- *claim*: for general A, changing the basis should put us in this boring situation
- notation: bold for column vectors $\mathbf{u} \in \mathbb{R}^d$
  - inner product: $\mathbf{u}^\top \mathbf{v} = \langle \mathbf{u}, \mathbf{v} \rangle$

Definition

a first-order system $\mathbf{q}_t + A\mathbf{q}_x = 0$ is *hyperbolic* if A is diagonalizable and all of the eigenvalues of A are real

- *diagonalizable* = there is a basis of $\mathbb{R}^d$ consisting of eigenvectors of A
- consider *left eigenvectors* of A, namely vectors $\mathbf{w}_k \in \mathbb{R}^d$ so that

$$\mathbf{w}_k^\top A = \lambda_k \mathbf{w}_k^\top$$

  - λ_k are *eigenvalues*, real numbers if the system is hyperbolic
  - $\mathbf{w}_k$ are column vectors so $\mathbf{w}_k^\top$ are row vectors
  - $\mathbf{w}_k$ are also right eigenvectors of $A^\top$: $\quad A^\top \mathbf{w}_k = \lambda_k \mathbf{w}_k$
  - the left/right eigen*values* are the same

eigenvectors decouple hyperbolic systems

- assume the system $\mathbf{q}_t + A\mathbf{q}_x = 0$ is hyperbolic
- decouple it by multiplying by $\mathbf{w}_k^\top$:

$$\begin{aligned}\mathbf{w}_k^\top \mathbf{q}_t + \mathbf{w}_k^\top A\mathbf{q}_x &= 0\\ \mathbf{w}_k^\top \mathbf{q}_t + \lambda_k \mathbf{w}_k^\top \mathbf{q}_x &= 0\end{aligned}$$

- define scalar functions (inner products)

$$v_k(t,x) = \mathbf{w}_k^\top \mathbf{q}(t,x)$$

- these scalar functions satisfy decoupled advection equations:

$$(v_k)_t + \lambda_k (v_k)_x = 0$$

- solve these one-way advection problems by characteristics:

$$v_k(t,x) = v_k(0, x - \lambda_k t)$$

- *note*: matrix A must be constant for this calculation

- the functions $v_k(t,x)$ are called the *Riemann invariants*:

$$v_k(t,x) = \mathbf{w}_k^\top \mathbf{q}(t,x) = v_k(0, x - \lambda_k t) = \mathbf{w}_k^\top \mathbf{q}(0, x - \lambda_k t)$$

- but how to write $\mathbf{q}(t,x)$ if we have $v_k(t,x)$?
  - expand in basis $\mathbf{w}_k$, with scalar coefficients $c_k(t,x)$:

$$\mathbf{q}(t,x) = \sum_{k=1}^{d} c_k(t,x)\mathbf{w}_k$$

  - note $\qquad v_\ell(t,x) = \mathbf{w}_\ell^\top \mathbf{q}(t,x) = \sum_{k=1}^{d} c_k(t,x)\mathbf{w}_\ell^\top \mathbf{w}_k$
  - define matrix $B \in \mathbb{R}^{d\times d}$ with entries $B_{\ell k} = \mathbf{w}_\ell^\top \mathbf{w}_k$:
  - B is invertible so solve:

$$B\mathbf{c} = \mathbf{v} \qquad \iff \qquad \sum_{k=1}^{d} B_{\ell k} c_k(t,x) = \mathbf{w}_\ell^\top \mathbf{q}(0, x - \lambda_\ell t)$$

- red equations combine into a computable solution

left eigenvectors, transposes, and MATLAB

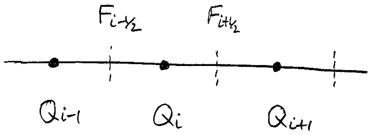
- left eigenvectors for A are the same as right eigenvectors for $A^\top$
- in MATLAB, find left eigenvectors $\mathbf{w}_k$ by applying `eig()` to `A'` $= A^\top$:

```
>> A = [...; ...; ...];      % input square matrix A
>> [X,D] = eig(A');
>> lamk = D(k,k);            % eigenvalue
>> wk = X(:,k);              % left eigenvector
```

Riemann solver

- key idea: in a FV scheme, at t_n we have two different numerical values on either side of the cell face at $x_{i+1/2}$:

$$\mathbf{Q}_i = \mathbf{Q}_L, \quad \mathbf{Q}_{i+1} = \mathbf{Q}_R$$

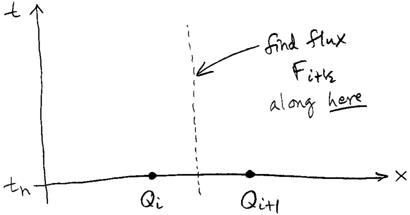


- on the other hand: $\mathbf{f}(\mathbf{q}) = A\mathbf{q}$ is a function of $\mathbf{q}$, so to get flux $\mathbf{F}_{i+1/2}$ we must know the solution *on* the face: $\mathbf{q}(t, x_{i+1/2})$ for $t > t_n$
- a *Riemann solver* solves the following problem:

find $\mathbf{q}(t, x_{i+1/2})$ for $t > t_n$ given

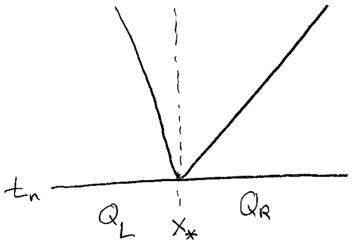
$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0$$

$$\mathbf{q}(t_n, x) = \begin{cases} \mathbf{Q}_L, & x < x_{i+1/2} \\ \mathbf{Q}_R, & x > x_{i+1/2} \end{cases}$$

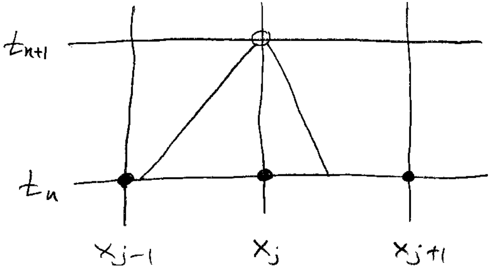


forward versus backward characteristics

- thus when constructing numerical schemes for hyperbolic problems there are two ways of thinking about characteristics:
  i) Riemann solvers generate a flux at x_* at times $t > t_n$
  ii) FD methods (e.g. Lax-Wendroff) find (x_j, t_{n+1}) solution by going back to t_n



Riemann solver view



finite difference (FD) view

- the Riemann solver view makes it easier to
  - generalize to nonlinear systems
  - work with the MOL equations (because no time-step choice)

Riemann solver for the acoustics problem

- for example, ...
- recall the acoustics problem $\mathbf{q}_t + A\mathbf{q}_x = 0$:

$$\mathbf{q} = \begin{bmatrix} p \\ u \end{bmatrix}, \; A = \begin{bmatrix} 0 & K_0 \\ \frac{1}{\rho_0} & 0 \end{bmatrix} \qquad \Longrightarrow \qquad \begin{aligned} p_t + K_0 u_x &= 0 \\ u_t + \tfrac{1}{\rho_0} p_x &= 0 \end{aligned}$$

- eigen-decomposition of A:

$$\lambda_1 = -c_0, \quad \mathbf{w}_1 = \begin{bmatrix} 1 \\ -Z_0 \end{bmatrix} \qquad\qquad \lambda_2 = +c_0, \quad \mathbf{w}_2 = \begin{bmatrix} 1 \\ +Z_0 \end{bmatrix}$$

  where $c_0 = \sqrt{K_0/\rho_0}$ and $Z_0 = \rho_0 c_0$
- thus the Riemann invariants $v_k(t,x) = \mathbf{w}_k^\top \mathbf{q}(t,x)$ are:

$$v_1(t,x) = p(t,x) - Z_0 u(t,x), \qquad v_2(t,x) = p(t,x) + Z_0 u(t,x)$$

Riemann solver for the acoustics problem 2

- denote $x_* = x_{i+1/2}$ and let $\mathbf{Q}_L = \begin{bmatrix} p_L \\ u_L \end{bmatrix}$, $\mathbf{Q}_R = \begin{bmatrix} p_R \\ u_R \end{bmatrix}$
- $\lambda_1 < 0$ so v_1 is left-going, so we solve forward from $t = t_n$:

$$\begin{aligned} p(t, x_*) - Z_0 u(t, x_*) &= v_1(t, x_*) = v_1(t_n, x_* - \lambda_1(t - t_n)) \\ &= p_R - Z_0 u_R \end{aligned}$$

- $\lambda_2 > 0$ so v_2 is right-going:

$$\begin{aligned} p(t, x_*) + Z_0 u(t, x_*) &= v_2(t, x_*) = v_2(t_n, x_* - \lambda_2(t - t_n)) \\ &= p_L - Z_0 u_L \end{aligned}$$

- solve for the solution at the face, namely $\mathbf{q}(t, x_*)$:

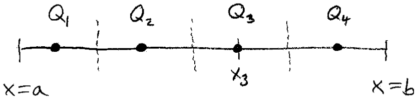
$$\begin{aligned} p(t, x_*) &= \frac{1}{2}\left(p_L + p_R + Z_0(u_L - u_R)\right) \\ u(t, x_*) &= \frac{1}{2}\left(\frac{1}{Z_0}(p_L - p_R) + u_L + u_R\right) \end{aligned}$$

- thus the face flux at $x_* = x_{i+1/2}$ is computable; this is the Riemann solver:

$$\mathbf{F}_{i+1/2}(t) = A\mathbf{Q}(t, x_*) = \begin{bmatrix} K_0 u(t, x_{i+1/2}) \\ \frac{1}{\rho_0} p(t, x_{i+1/2}) \end{bmatrix}$$

flux boundary conditions, and the grid

- boundary conditions at $x = a, b$
- easiest if think in terms of the value of the *flux* there,
- …thus we set up the grid to have cell faces at $x = a, b$



$$x_j = a + (j - 1/2)\Delta x \quad \text{where} \quad \Delta x = \frac{b-a}{J}$$

- for the acoustics problem on the next slide, suppose
  - reflecting condition on left: $u(t, a) = 0$
  - outflow condition on right: $v_1(t, b) = 0$
- modify the Riemann solvers at $x = a$ and $x = b$ accordingly
  - inflow versus outflow condition is clear in the scheme

- I used all these ideas in a C+PETSc program for the acoustics problem:

$$\begin{aligned} p_t + K_0 u_x &= 0 \\ u_t + \frac{1}{\rho_0} p_x &= 0 \end{aligned}$$

- see `c/riemann/riemann.c` at `github.com/bueler/p4pdes-next`

SHOW ACOUSTICS MOVIE $\leftarrow$ `movies/acoustics.mp4`

summary of linear hyperbolic systems and Riemann solvers

- if a linear system $\mathbf{q}_t + A\,\mathbf{q}_x = 0$ in $\mathbb{R}^d$ is *hyperbolic* then (by definition) it can be decoupled ($\mathbf{w}_k^\top A = \lambda_k \mathbf{w}_k$) into d scalar (real) advection problems
- the solutions of these advection problems, forward from time t_n, are the *Riemann invariants*:

$$v_k(t,x) = \mathbf{w}_k^\top \mathbf{q}(t,x) = v_k(t_n, x - \lambda_k(t - t_n))$$

- now write in conservation form: $\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0$ where $\mathbf{f}(\mathbf{q}) = A\mathbf{q}$
- in the FV *method-of-lines (MOL)* view we only integrate in space:

$$\frac{d\mathbf{Q}_j}{dt} + \frac{\mathbf{F}_{i+1/2} - \mathbf{F}_{i-1/2}}{\Delta x} = 0$$

- a *Riemann solver* finds the face fluxes $\mathbf{F}_{i+1/2} = A\mathbf{Q}^*$ by solving the *Riemann problem*, with $\mathbf{Q}_L, \mathbf{Q}_R$ on sides of the face, to find $\mathbf{Q}^*$ on the face
    - this uses the Riemann invariants

- can we generalize to nonlinear systems by using $A = \mathbf{f}'(\mathbf{q})$?

Outline

Godunov's barrier theorem

- can we do better than Lax-Wendroff, even for scalar advection?
- can we have high accuracy without oscillations?
  - upwinding is only first-order accurate, but it avoids oscillations
  - Lax-Wendroff is second-order but it generates oscillations beyond the range of the initial condition
- rough answer: NO

Theorem (*Godunov, 1959*)

A monotonicity-preserving linear scheme for $q_t + aq_x = 0$ cannot have second-order (or higher) local truncation error in x.

- "monotonicity-preserving" means (essentially) that the scheme does not add oscillations
- Godunov's barrier created modern hyperbolic PDE solvers
  - upwinding, Lax-Friedrichs, Lax-Wendroff, leapfrog are the old technology
  - "high-resolution" schemes of the 1970–90s overcame the barrier

Godunov's barrier theorem

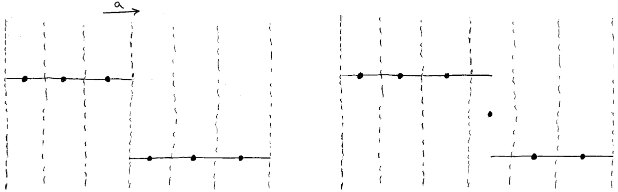
- can we do better than Lax-Wendroff, even for scalar advection?
- can we have high accuracy without oscillations?
  - upwinding is only first-order accurate, but it avoids oscillations
  - Lax-Wendroff is second-order but it generates oscillations beyond the range of the initial condition
- rough answer: ~~NO~~ yes

Theorem (*Godunov, 1959*)

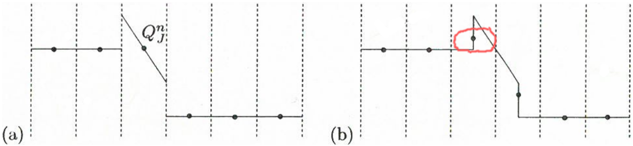
A monotonicity-preserving linear scheme for $q_t + aq_x = 0$ cannot have second-order (or higher) local truncation error in x.

- "monotonicity-preserving" means (essentially) that the scheme does not add oscillations
- Godunov's barrier created modern hyperbolic PDE solvers
  - upwinding, Lax-Friedrichs, Lax-Wendroff, leapfrog are the old technology
  - "high-resolution" schemes of the 1970–90s overcame the barrier   how?

- if we want to kill oscillation then another view is helpful
- consider: $q_t + aq_x = 0$ for $a > 0$, with cell values $\{Q_i^n\}$
- apply 1st-order upwinding Euler step from t_n (left) to t_{n+1} (right):



  - right figure: new cell averages (dots) after evolving exact solution (lines)
  - upwinding uses constant values in cells (*see next slide*)
- versus Lax-Wendroff, which uses downwind slope:



  - note the overshoot, which we want to avoid

- at time t_n we only have the discrete unknowns $\{Q_j^n\}$
- but we can "reconstruct" a linear function $\tilde{q}(t_n, x)$ on each cell:

$$\tilde{q}(t_n, x) = Q_i^n + \sigma_i^n (x - x_i)$$

  - σ_i^n is the *slope* in the cell $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$
  - for any σ_i^n, the cell average remains Q_i^n

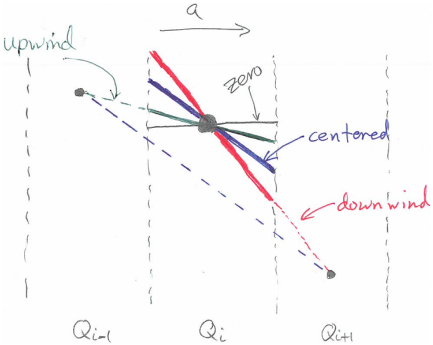
- possibilities for slope when $a > 0$:

zero: $\sigma_i = 0$

downwind: $\sigma_i = \dfrac{Q_{i+1} - Q_i}{\Delta x}$

upwind: $\sigma_i = \dfrac{Q_i - Q_{i-1}}{\Delta x}$

centered: $\sigma_i = \dfrac{Q_{i+1} - Q_{i-1}}{2\Delta x}$



- WARNING: upwind *scheme* uses zero slope

from reconstruction to flux

- for slope σ_i^n we get a model (reconstruction) of the solution in the cell:

$$\tilde{q}(t_n, x) = Q_i^n + \sigma_i^n (x - x_i)$$

- this gives solution estimates at the cell faces $x_{i-1/2}, x_{i+1/2}$:

$$Q_i^L = \tilde{q}(t_n, x_{i-1/2}) = Q_i^n - \sigma_i^n \frac{\Delta x}{2}$$

$$Q_i^R = \tilde{q}(t_n, x_{i+1/2}) = Q_i^n + \sigma_i^n \frac{\Delta x}{2}$$

  - use these to compute the fluxes at cell faces in an explicit method
- for example, in the scalar advection equation $F(q) = aq$:

$$F_{i+\frac{1}{2}} = \begin{cases} aQ_i^R, & a \ge 0 \\ aQ_{i+1}^L, & a < 0 \end{cases}$$

  - similarly for $F_{i-\frac{1}{2}}$
  - compare the "upwind as the donor cell method" slide

slope limiter idea

- we should use a nonzero slope σ_i to get higher accuracy
- ... *except* when the slope would put the reconstruction out of range of the three values $\{Q_{i-1}, Q_i, Q_{i+1}\}$
- computing σ_i by a *slope limiter* avoid the out-of-range problem
- for example, the minmod *slope limiter*:

$$\sigma_i = \operatorname{minmod}\left\{\frac{Q_i - Q_{i-1}}{\Delta x}, \frac{Q_{i+1} - Q_i}{\Delta x}\right\}$$

  - by definition, for real numbers a, b of the same sign:

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

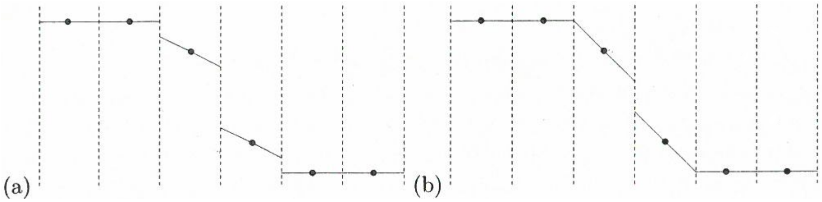
  - in words: $\operatorname{minmod}\{a, b\}$ is closest to zero of a and b unless they differ in sign; then it is zero
  - if Q_i is the extrema of the three values then $\sigma_i = 0$

slope limiter idea 2

- an alternative is the *MC slope limiter*:

$$\sigma_i = \operatorname{minmod}\left\{\frac{Q_{i+1}-Q_{i-1}}{2\Delta x}, 2\operatorname{minmod}\left\{\frac{Q_i-Q_{i-1}}{\Delta x}, \frac{Q_{i+1}-Q_i}{\Delta x}\right\}\right\}$$

- in pictures:

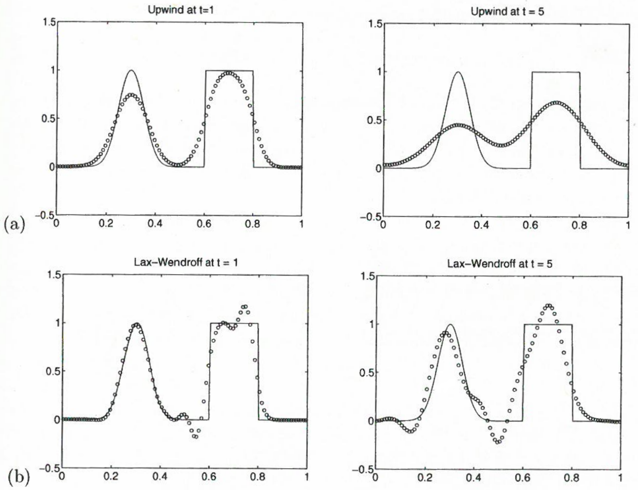


minmod limiter   MC limiter

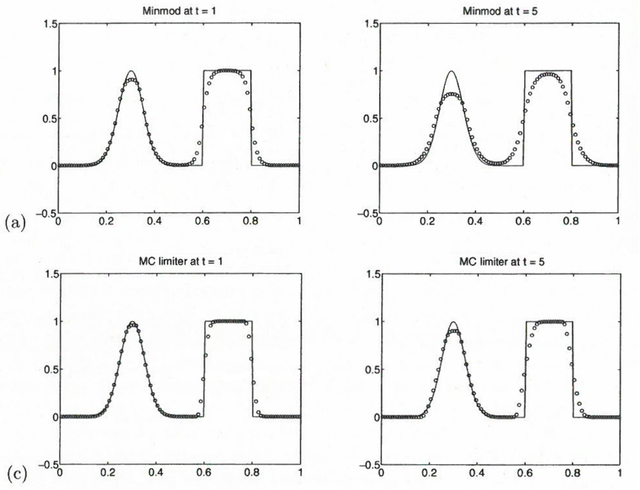
  - can you draw-in the downwind slopes (Lax-Wendroff)? . . . get overshoot

- historical comment: the theory of *total variation diminishing* (TVD) schemes, circa 1990s, arises from these pictures

- consider scalar advection again: $q_t + aq_x = 0$, $a = 1$, periodic b.c.s

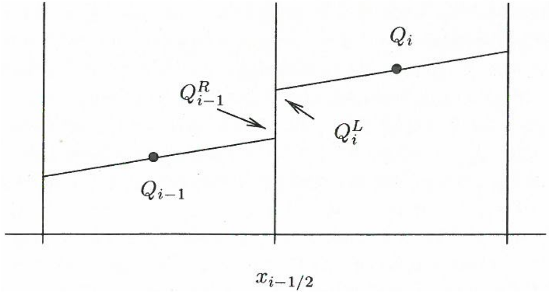


- consider scalar advection again: $q_t + aq_x = 0$, $a = 1$, periodic b.c.s



MOL: slope limiting and Riemann solvers

- suppose we want to use limited slopes and Riemann solvers in a method-of-lines (MOL) framework … how to do this?
- *answer*: apply the slope calculation and slope-limiter as usual and then compute the flux $F_{i-1/2}$ in terms of this picture:



- the left and right values at the face, Q_{i-1}^R and Q_i^L, are the ones used in the Riemann solver to compute the solution value at the face for $t > t_n$
- thereby get the flux $F_{i-1/2}(t)$ for $t > t_n$

Outline

- a *conservation law* is a first-order PDE system, for $\mathbf{q}(t,x) \in \mathbb{R}^d$, with a given *flux function* **f** and *source function* **g**:

$$\mathbf{q}_t + \mathbf{f}(t,x,\mathbf{q})_x = \mathbf{g}(t,x,\mathbf{q})$$

  - linear conservation law: $\mathbf{f}(\mathbf{q}) = A\mathbf{q}$
  - if **f** depends on $\nabla \mathbf{q}$ (e.g. heat equation) then the system is *not* hyperbolic

- scalar, nonlinear, and hyperbolic examples:
  - *Burger's equation* with $f(q) = \frac{1}{2}q^2$ and $g = 0$:

$$q_t + \left(\frac{1}{2}q^2\right)_x = 0 \qquad \iff \qquad q_t + q q_x = 0$$

  - *nonlinear traffic model* with $f(q) = u_{\max}(1-q)q$ and $g = 0$:

$$q_t + \left(u_{\max}(1-q)q\right)_x = 0$$

- *shallow water equations* with height $h(t,x)$ and velocity $u(t,x)$:

$$\mathbf{q} = \begin{bmatrix} h \\ hu \end{bmatrix}, \ \mathbf{f}(\mathbf{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{g}{2}h^2 \end{bmatrix} \quad \Longrightarrow \quad \begin{aligned} h_t + (hu)_x &= 0 \\ (hu)_t + \left(hu^2 + \tfrac{g}{2}h^2\right)_x &= 0 \end{aligned}$$

- *Euler equations for an ideal gas*:

$$\mathbf{q} = \begin{bmatrix} \rho \\ \rho u \\ E \end{bmatrix}, \ \mathbf{f}(\mathbf{q}) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ (E+p)u \end{bmatrix} \quad \Longrightarrow \quad \begin{aligned} \rho_t + (\rho u)_x &= 0 \\ (\rho u)_t + \left(\rho u^2 + p\right)_x &= 0 \\ E_t + \left((E+p)u\right)_x &= 0 \end{aligned}$$

  - variables are density $\rho(t,x)$, velocity $u(t,x)$, and energy density $E(t,x)$
  - the pressure p is found from an equation of state, for example for a polytropic ideal gas ($\gamma \approx 1.4$ for air):

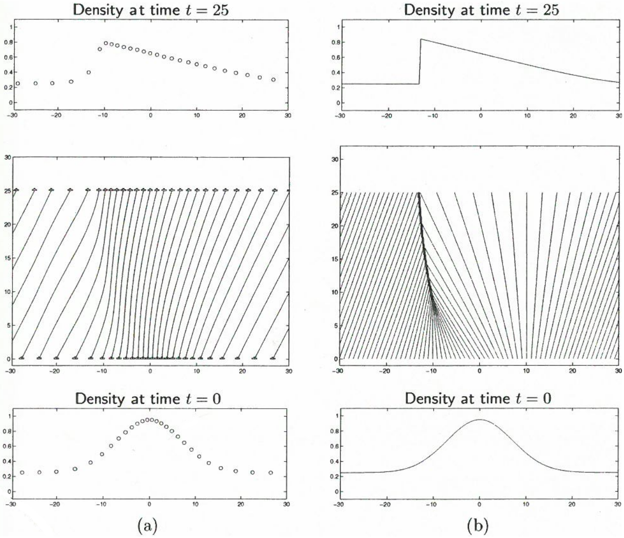
$$E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho u^2$$

scalar conservation law: traffic model

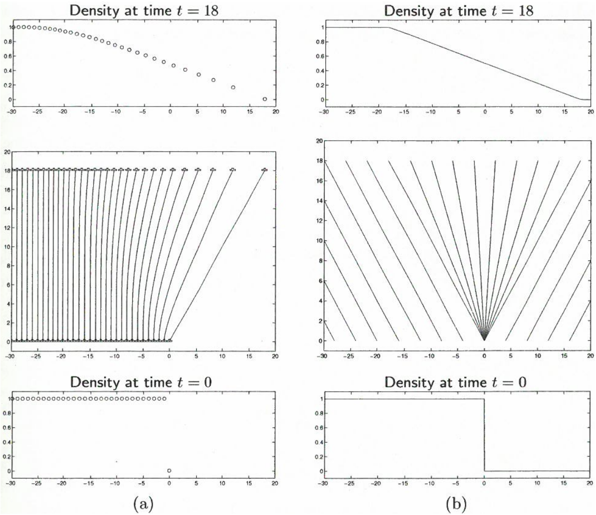
- for this traffic model, $q(t,x)$ is the density of cars, $0 \le q \le 1$
- cars move at speed
$$U(q) = u_{\text{max}}(1-q)$$
  - they slow down when the density is high
- the flux of cars is
$$f(q) = U(q)q = u_{\text{max}}(1-q)q$$
- but note that
$$f'(q) = u_{\text{max}}(1-2q)$$
- scalar nonlinear conservation laws are nonlinear advection problems:
$$q_t + f'(q)q_x = 0$$
- the solution is constant along characteristics, but the characteristic speed depends on the solution:
$$a = f'(q) = u_{\text{max}}(1-2q)$$

traffic: speed versus speed

- speed of car is $U(q) = u_{\max}(1 - q)$
- speed of characteristic is $f'(q) = u_{\max}(1 - 2q)$



- previous slide shows formation of a *shock wave* from a smooth hump
- the model can also form *rarefaction waves*



recall the FV-MOL-Riemann solver idea

- for problems in flux-conservation form: $q_t + f(q)_x = 0$
- put a grid on x, with x_i at cell center
- integrate over the cell:

$$\frac{d}{dt} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\,dx + f\left(q(t, x_{i+\frac{1}{2}})\right) - f\left(q(t, x_{i-\frac{1}{2}})\right) = 0$$

- interpret discrete unknowns as cell averages (FV):

$$Q_i(t) = \frac{1}{\Delta x} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} q(t,x)\,dx$$

- get a big ODE system (MOL) for solution by an ODE black box:

$$\frac{dQ_i}{dt} + \frac{F_{i+1/2} - F_{i-1/2}}{\Delta x} = 0$$

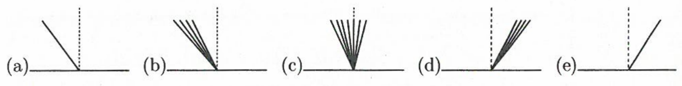
- a (slope-limited) Riemann solver will compute the face fluxes $F_{i+1/2}$

Riemann problem for scalar conservation laws

- the Riemann problem addresses a discontinuity at $x_* = x_{i+1/2}$:

$$q_t + f'(q)q_x = 0, \qquad q(t_n, x) = \begin{cases} Q_L & x < x_* \\ Q_R & x > x_* \end{cases}$$

  - the goal is to compute Q_* on the face at x_*
  - and thereby compute the flux $F_{i+1/2}$
- for scalar nonlinear conservation laws there are several possibilities:



  - (a), (e) are shocks while (b), (d) are rarefaction waves
    - (a),(b) left-going or (d),(e) right-going, as shown
    - in all these cases $Q_* = Q_L$ or $Q_* = Q_R$
  - (c) is a *transonic rarefaction wave*: Q_* satisfies $f'(Q_*) = 0$
  - these are all the possibilities if $f(q)$ is convex or concave

Riemann solver for scalar conservation laws

- the flux solution of the Riemann problem for a scalar conservation law:

$$F_{i+1/2} = \begin{cases} \min\limits_{Q_L \le q \le Q_R} f(q) & \text{if } Q_L \le Q_R \\ \max\limits_{Q_R \le q \le Q_L} f(q) & \text{if } Q_L \ge Q_R \end{cases}$$

  - formula (12.4) in LeVeque (2002)
- see `c/riemann/riemann.c` at `github.com/bueler/p4pdes-next`

SHOW TRAFFIC SHOCK MOVIE $\leftarrow$ `movies/traffic.mp4`

shallow water equations

- $h(t,x)$ is water surface height, $u(t,x)$ is horizontal water velocity
  - assuming $h > 0$ throughout
- conservation law $\quad \mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0$ with

$$\mathbf{q} = \begin{bmatrix} h \\ hu \end{bmatrix}, \qquad \mathbf{f}(\mathbf{q}) = \begin{bmatrix} hu \\ hu^2 + \frac{g}{2}h^2 \end{bmatrix}$$

- eigen-decomposition of $\quad A = \mathbf{f}'(\mathbf{q}) = \begin{bmatrix} 0 & 1 \\ -u^2 + gh & 2u \end{bmatrix}$:

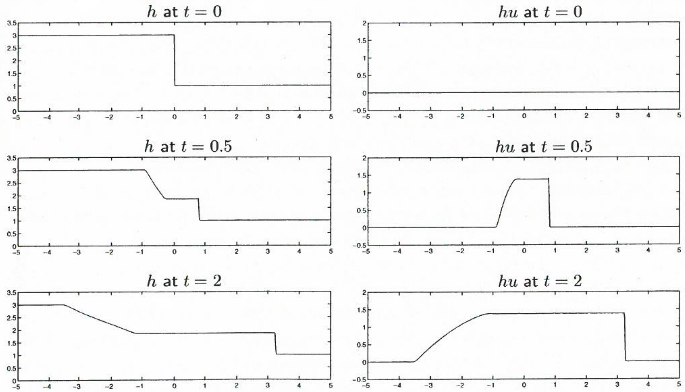
$$\lambda_1 = u - \sqrt{gh}, \quad \mathbf{w}_1 = \begin{bmatrix} 1 \\ u - \sqrt{gh} \end{bmatrix}$$

$$\lambda_2 = u + \sqrt{gh}, \quad \mathbf{w}_2 = \begin{bmatrix} 1 \\ u + \sqrt{gh} \end{bmatrix}$$

  - main idea: the eigen-decomposition depends on $\mathbf{q}$
  - the speeds always bracket the water velocity: $\lambda_1 < u < \lambda_2$
    - it is possible that both λ_i are negative, or both are positive
  - these *gravity waves* travel at speed at least $\sqrt{gh}$

shallow water equations: illustration of a Riemann problem

- what does a Riemann problem look like?
  - recall: $\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0, \quad \mathbf{q}(t_n, x) = \{\mathbf{Q}_L, \mathbf{Q}_R\}$
- in the case where $\mathbf{Q}_L = [3, 0]^\top$, $\mathbf{Q}_R = [1, 0]^\top$ it is a "dam break":



  - nontrivial wave structure with left-going rarefaction and right-going shock

Riemann solver options

- how should the Riemann solver work?
- exact (harder):
  - fully-solve the nonlinear Riemann problem

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = 0, \quad \mathbf{q}(t_n, x) = \{\mathbf{Q}_L, \mathbf{Q}_R\}$$

  - evaluate $\mathbf{Q}_*$ as the solution $\mathbf{q}(t, x_*)$ at $x_* = x_{i+1/2}$ for $t > t_n$
  - get $\mathbf{F}_{i+1/2} = \mathbf{f}(\mathbf{Q}_*)$
- average-and-linearization (easier):
  - compute $\mathbf{Q}_0$ as an average of $\mathbf{Q}_L$ and $\mathbf{Q}_R$
  - let $\hat{A} = \mathbf{f}'(\mathbf{Q}_0)$
  - solve Riemann problem for linear system

$$\mathbf{q}_t + \hat{A}\mathbf{q}_x = 0, \quad \mathbf{q}(t_n, x) = \{\mathbf{Q}_L, \mathbf{Q}_R\}$$

    - use the Riemann invariants $v_k(t, x) = \mathbf{w}_k^\top \mathbf{q}(t, x)$ from $\mathbf{w}_k^\top \hat{A} = \lambda_k \mathbf{w}_k^\top$
  - evaluate $\mathbf{Q}_*$ as the solution $\mathbf{q}(t, x_*)$ at $x_* = x_{i+1/2}$ for $t > t_n$
  - get $\mathbf{F}_{i+1/2} = \mathbf{f}(\mathbf{Q}_*)$

Roe approximate solver

- for the easier average-and-linearization route, almost all steps are the same as for a linear system (e.g. acoustics)
- but how to "compute $\mathbf{Q}_0$ as an average of $\mathbf{Q}_L$ and $\mathbf{Q}_R$"?
  1. low-accuracy approximation results from a simple average:

$$\mathbf{Q}_0 = \frac{1}{2}\left(\mathbf{Q}_L + \mathbf{Q}_R\right) \implies \hat{A} = \mathbf{f}'(\mathbf{Q}_0)$$

  2. different low-accuracy approximation from:

$$\hat{A} = \frac{1}{2}\left(\mathbf{f}'(\mathbf{Q}_L) + \mathbf{f}'(\mathbf{Q}_R)\right)$$

     - not a good idea; no guarantee $\hat{A}$ is hyperbolic

  3. higher-accuracy idea of Roe (1981), called *Roe averaging*:

$$\begin{aligned} \hat{h} &= \tfrac{1}{2}(h_L + h_R) \\ \hat{u} &= \frac{\sqrt{h_L}\, u_L + \sqrt{h_R}\, u_R}{\sqrt{h_L} + \sqrt{h_R}} \end{aligned} \implies \mathbf{Q}_0 = \begin{bmatrix} \hat{h} \\ \hat{h}\hat{u} \end{bmatrix} \implies \hat{A} = \mathbf{f}'(\mathbf{Q}_0)$$

shallow water: my results

- put together the above tools . . .
- see `c/riemann/riemann.c` at `github.com/bueler/p4pdes-next`

SHOW SHALLOW WATER “DAM” MOVIE $\leftarrow$ `movies/dam.mp4`

summary

- for hyperbolic conservation-law systems $\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{g}$
  - linear examples $\mathbf{f}(\mathbf{q}) = A\mathbf{q}$: acoustics, elasticity, Maxwell's equations
  - nonlinear examples: shallow water, compressible gas

  a preferred numerical approach since the 1990s is a

  high-resolution Godunov method
- consists of three things:
  1. finite volume thinking
     - conservation law is spatially integrated
     - discrete unknowns $\mathbf{Q}_i$ are cell averages
     - flux is needed at faces between cells: $\mathbf{F}_{i+1/2}$
  2. Riemann solvers
     - the "Riemann problem" considers different cell values on each side of a face
     - a Riemann solver determines the local wave structure (rarefaction, shock)
       - provided by the user; exact or approximate (e.g. Roe)
       - for a system, uses eigen-expansion of $A = \mathbf{f}'(\mathbf{q})$
       - compute the flux on the face forward in time
  3. slope (or flux) limiters
     - based on first-order upwinding ($=$ donor-cell $=$ "classical Godunov")
     - to achieve higher-order without oscillations:
       - solution is "reconstructed" with slope in each cell
       - slope is (nonlinearly) limited
       - the method reverts to first-order upwinding at extrema

Outline

- for this section, the problem is a scalar conservation law in 2D:

$$q_t + \nabla \cdot \mathbf{F}(q) = 0$$

  - the solution is a scalar $q(t, x, y)$ but the flux is a vector $\mathbf{F}(q)$
  - q might be a (conserved) density like mass or energy
  - *advection* is when $\mathbf{F}(q) = \mathbf{a}q$, where $\mathbf{a}(x, y) = \langle u(x, y), v(x, y)\rangle$ is velocity:

$$q_t + (uq)_x + (vq)_y = 0$$

- we want the solution in a region $\Omega \subset \mathbb{R}^2$, for some times $0 \le t \le T$

finite volumes in 2D

- how does the finite volumes (FV) method work in 2D?
  - the domain on which you solve the PDE is cut into finitely-many cells
    - 1D cells are just intervals
    - 2D cells are polygons (e.g. triangles or rectangles, etc.)
    - 3D cells are polyhedra (e.g. tetrahedra or cubes, etc.)
  - the method enforces the integrated version of conservation on the cell
  - there is one unknown and one equation per cell

scalar conservation laws in 2D: integral form

- the FV method always starts with the same manipulations ...
- suppose $V \subset \mathbb{R}^2$ is a *finite volume* (cell) in the x, y plane
    - the cell has volume (area) $|V|$
- integrate the conservation law over the cell:

$$\int_V q_t \, dxdy + \int_V \nabla \cdot \mathbf{F}(q) \, dxdy = 0$$

- let $Q_V(t)$ be the average of the solution over the cell:

$$Q_V(t) := \frac{1}{|V|} \int_V q(t, x, y) \, dxdy$$

- apply divergence theorem:

$$\frac{dQ_V}{dt} + \frac{1}{|V|} \int_{\partial V} \mathbf{F} \cdot \mathbf{n} \, ds = 0$$

    - $\partial V =$ boundary of V

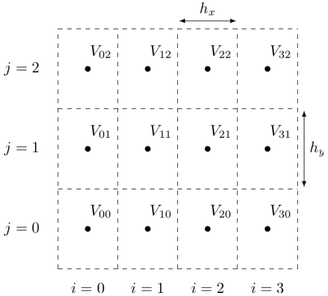
structured finite volumes in 2D

- PDE $q_t + \nabla \cdot \mathbf{F}(q) = 0$ has become an ODE for Q_V:

$$\frac{dQ_V}{dt} + \frac{1}{|V|} \int_{\partial V} \mathbf{F} \cdot \mathbf{n}\, ds = 0$$

- suppose region Ω is a rectangle: $\Omega = [a,b] \times [c,d]$
- a *structured* FV method cuts Ω into a grid of indexed cells V_{ij}
  - rectangular cells have dimensions h_x, h_y and area $|V_{ij}| = h_x h_y$
- apply the equation for each cell:

$$\frac{dQ_{ij}}{dt} + \frac{1}{h_x h_y} \int_{\partial V_{ij}} \mathbf{F} \cdot \mathbf{n}\, ds = 0$$



- get a system of ODEs: $\dfrac{d\mathbf{Q}}{dt} = \mathbf{G}(t, \mathbf{Q})$
- to actually construct $\mathbf{G}$ for this system we need a method for computing the flux $\mathbf{F}$ on the boundary of each V_{ij}

advection in 2D

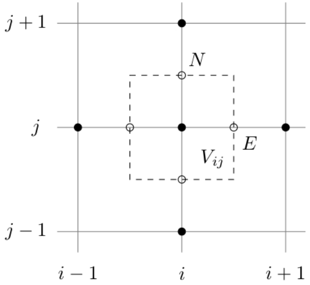
- consider advection with $\mathbf{F}(q) = \mathbf{a}q$ where $\mathbf{a} = \langle u, v \rangle$:

$$\frac{dQ_{ij}}{dt} + \frac{1}{h_x h_y} \int_{\partial V_{ij}} \langle uq, vq \rangle \cdot \mathbf{n}\, ds = 0$$

- the boundary ∂V_{ij} has 4 sides (faces) N, E, S, W
- thus need 4 integrals:

$$\int_{\partial V_{ij}} \langle uq, vq \rangle \cdot \mathbf{n}\, ds = \int_N + \int_E + \int_S + \int_W$$

  - actually, only compute N, E faces ...
- simplest integral (*quadrature*) choice uses the value at the midpoint of the face
  - open circles in figure



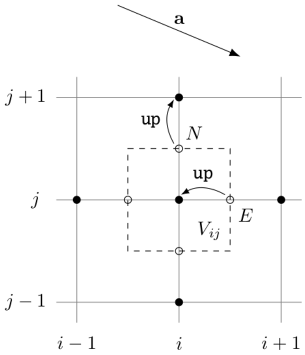
- consider advection with $\mathbf{F}(q) = \mathbf{a}q$ where $\mathbf{a} = \langle u, v \rangle$
- the simplest (Godunov) "donor cell" upwinding method decides which side contributes the solution value Q when computing the flux
- for example, at $N = (x_i, y_j + \frac{h_y}{2})$ the normal direction is $\hat{\mathbf{y}} = \langle 0, 1 \rangle$:

$$\begin{aligned}\int_N \langle uq, vq \rangle \cdot \mathbf{n}\, ds &= \int_N \langle uq, vq \rangle \cdot \langle 0, 1 \rangle \; ds \\ &= \int_N vq\, ds \\ &\approx h_x\, v(N) \tilde{Q} \quad \text{[midpoint rule]}\end{aligned}$$

where

$$\tilde{Q} = \begin{cases} Q_{i,j} & v(N) \geq 0 \\ Q_{i,j+1} & v(N) < 0 \end{cases}$$

- such decisions are made at all 4 faces
  - figure: at N and E faces for given $\mathbf{a}$



higher-resolution schemes in 2D

- recall that the Riemann problem has an initial condition of two values Q_L, Q_R on the sides of a face
- important principle for 2D and 3D FV schemes:

  the Riemann problem on each face is treated as 1D, normal to the face

  - e.g. for advection, first-order upwinding looks only at the component of velocity normal to the face
- two flavors of "high-resolution" schemes:
  1. *slope limiters* regard the solution in each cell as not constant, and then modify the slopes to avoid oscillations
  2. *flux limiters* compute the flux based on first-order upwinding, with an added term based on a higher-order flux, but limited to avoid oscillations

  - these 2 modern routes are (essentially) equivalent in 1D
  - flux limiters may be more natural in 2D and 3D?

a flux-limiter scheme

- recall that at the N face of cell V_{ij}, first-order (donor-cell) upwinding was

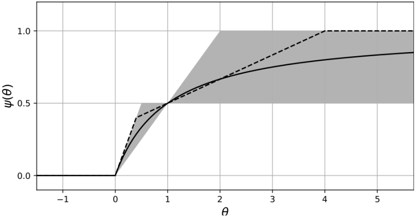
$$\int_N \langle uq, vq\rangle \cdot \mathbf{n}\, ds \approx h_x\, v(N)\tilde Q \quad \text{where} \quad \tilde Q = \begin{cases} Q_{i,j} & v(N) \ge 0 \\ Q_{i,j+1} & v(N) < 0 \end{cases}$$

- normal flux at the center of N face is: $f_N = v(N)\tilde Q$
- with a flux-limiter ψ we instead have

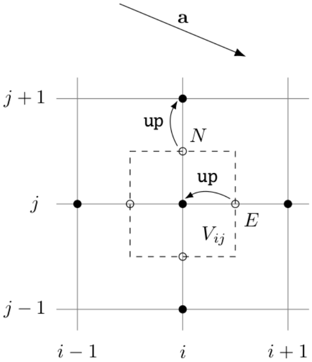
$$f_N = v(N) \begin{cases} Q_{i,j} + \psi(\theta_j)(Q_{i,j+1} - Q_{i,j}) & v(N) \ge 0 \\ Q_{i,j+1} + \psi(1/\theta_{j+1})(Q_{i,j} - Q_{i,j+1}) & v(N) < 0 \end{cases}$$

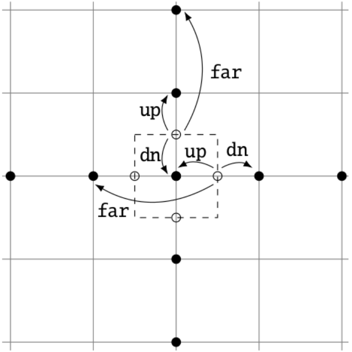
where $\theta_j = \dfrac{Q_{i,j} - Q_{i,j-1}}{Q_{i,j+1} - Q_{i,j}}$

- $\psi(\theta)$ is a curve in the famous Sweby (1984) shaded region
    - solid: van Leer (1974)
    - dashed: Koren (1993)



- make decisions at N, E faces ... this suffices
- compare: first-order upwinding (left) versus the flux-limiter case (right) using "downwind" (**`dn`**) and "far" points (**`far`**)
  - a flux- or slope-limiter is *stencil expanding*



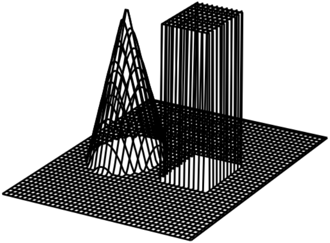


results

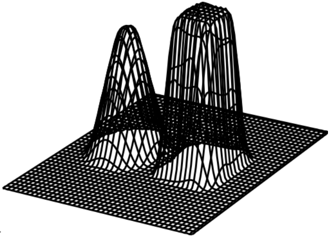
- suppose domain is square $\Omega = (-1,1) \times (-1,1)$ and equation $q_t + \nabla \cdot \mathbf{F}(q) = 0$ is advection ($\mathbf{F}(q) = \mathbf{a}q$) with a rotation velocity field:

$$\mathbf{a}(x,y) = \langle y, -x \rangle$$

- solving for $0 \le t \le 2\pi$ should rotate initial condition back to itself
- results on a 40×40 grid with the Koren flux limiter and an adaptive, 2nd-order Runge-Kutta ODE solver ($\approx$ `ode23`):



$\longrightarrow$

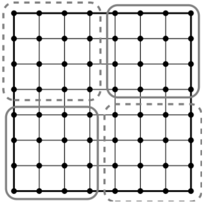


  - good results like this would require *much* finer grid using Lax-Wendroff (1000×1000?)

live movie, computed in parallel

- see `c/ch11/advect.c` at `github.com/bueler/p4pdes`
  - uses C and PETSc and MPI
- runs here use Koren flux-limiter and SSP time-stepping
  - *strong stability-preserving* (SSP) is better than RK
- movie from parallel run with 4 processors:

```
$ tmpg -n 4 ./advect -adv_problem rotation \
    -da_refine 3 -ts_max_time 6.283185 \
    -ts_monitor_solution draw -draw_size 600,600 \
    -ts_type ssp -adv_limiter koren
```



- parallel run with 12 processes in 90 seconds:
  - $N = 640 \times 640 \times 4022 = 1.6 \times 10^9$, a billion-point space-time grid
  - . . . but no graphics, which would take forever

```
$ tmpg -n 12 ./advect -adv_problem rotation -da_refine 7 \
    -ts_max_time 6.283185 -ts_type ssp -adv_limiter koren
```

SHOW LIVE