Gene Golub SIAM Summer School 2012

Numerical Methods for Wave Propagation
Finite Volume Methods
Lecture 2

Randall J. LeVeque
Applied Mathematics
University of Washington

Outline

This lecture

- Finite difference / finite volume methods
- Godunov's method
- High resolution methods (limiters)
- Two-dimensional methods
- Seismic example

Upwind method for advection

Scalar advection:

$$q_t + uq_x = 0, \qquad u > 0$$

As finite difference method:

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

Gives the explicit method:

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

Stable provided CFL condition satisfied:

$$0 \le \frac{u\Delta t}{\Delta x} \le 1$$

and first order accurate on smooth data.

The CFL Condition (Courant-Friedrichs-Lewy)

Domain of dependence: The solution $q(X,T)$ depends on the data $q(x,0)$ over some set of x values, $x \in \mathcal{D}(X,T)$.

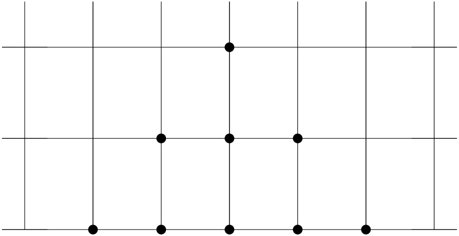
Advection: $q(X,T) = q(X - uT, 0)$ and so $\mathcal{D}(X,T) = \{X - uT\}$.

The CFL Condition: A numerical method can be convergent only if its numerical domain of dependence contains the true domain of dependence of the PDE, at least in the limit as Δt and Δx go to zero.

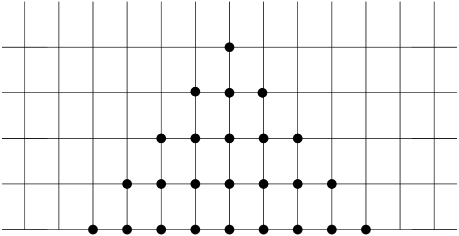
Note: Necessary but not sufficient for stability!

Numerical domain of dependence

With a 3-point explicit method:



On a finer grid with $\Delta t/\Delta x$ fixed:



The CFL Condition

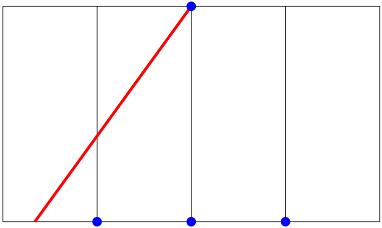
For the method to be stable, the numerical domain of dependence must include the true domain of dependence.

For advection, the solution is constant along characteristics,

$$q(x,t) = q(x - ut, 0)$$

For a 3-point method, CFL condition requires $\left|\frac{u\Delta t}{\Delta x}\right| \leq 1.$

If this is violated:
True solution is determined by data at a point $x - ut$ that is ignored by the numerical method, even as the grid is refined.



The CFL Condition

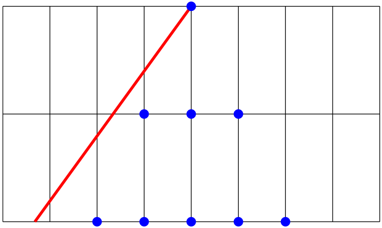
For the method to be stable, the numerical domain of dependence must include the true domain of dependence.

For advection, the solution is constant along characteristics,

$$q(x, t) = q(x - ut, 0)$$

For a 3-point method, CFL condition requires $\left|\frac{u\Delta t}{\Delta x}\right| \leq 1.$

If this is violated:
True solution is determined by data at a point $x - ut$ that is ignored by the numerical method, even as the grid is refined.



The CFL Condition

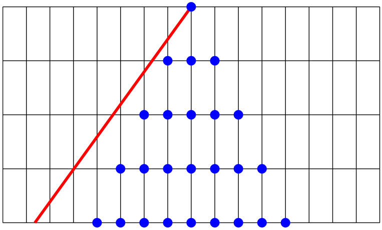
For the method to be stable, the numerical domain of dependence must include the true domain of dependence.

For advection, the solution is constant along characteristics,

$$q(x, t) = q(x - ut, 0)$$

For a 3-point method, CFL condition requires $\left|\frac{u\Delta t}{\Delta x}\right| \leq 1.$

If this is violated:
True solution is determined by data at a point $x - ut$ that is ignored by the numerical method, even as the grid is refined.



The CFL Condition

For the method to be stable, the numerical domain of dependence must include the true domain of dependence.

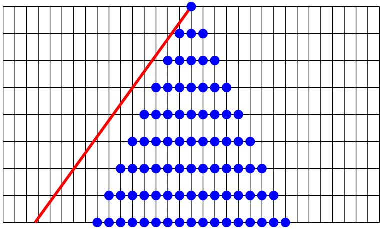
For advection, the solution is constant along characteristics,

$$q(x,t) = q(x - ut, 0)$$

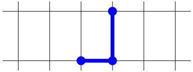
For a 3-point method, CFL condition requires $\left|\frac{u\Delta t}{\Delta x}\right| \leq 1.$

If this is violated:

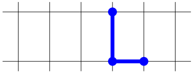
True solution is determined by data at a point $x - ut$ that is ignored by the numerical method, even as the grid is refined.



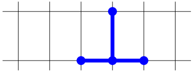
Stencil — CFL Condition



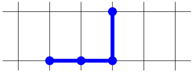
$$0 \le \frac{u\Delta t}{\Delta x} \le 1$$



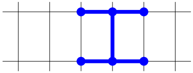
$$-1 \le \frac{u\Delta t}{\Delta x} \le 0$$



$$-1 \le \frac{u\Delta t}{\Delta x} \le 1$$



$$0 \le \frac{u\Delta t}{\Delta x} \le 2$$



$$-\infty < \frac{u\Delta t}{\Delta x} < \infty$$

Upwind method for advection

Scalar advection:

$$q_t + uq_x = 0, \qquad u > 0$$

As finite difference method:

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

Gives the explicit method:

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

Stable provided CFL condition satisfied:

$$0 \le \frac{u\Delta t}{\Delta x} \le 1$$

and first order accurate on smooth data.

Finite differences vs. finite volumes

Finite difference Methods

- Pointwise values $Q_i^n \approx q(x_i, t_n)$
- Approximate derivatives by finite differences
- Assumes smoothness

Finite volume Methods

- Approximate cell averages: $Q_i^n \approx \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x, t_n)\, dx$
- Integral form of conservation law,

$$\frac{\partial}{\partial t} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t)\, dx \quad = \quad f(q(x_{i-1/2}, t)) - f(q(x_{i+1/2}, t))$$

leads to conservation law $q_t + f_x = 0$ but also directly to numerical method.

Finite volume method

$Q_i^n \approx \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t_n)\,dx$

Integral form:

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

Finite volume method

$Q_i^n \approx \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t_n)\,dx$

Integral form:

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

Integrate from t_n to t_{n+1} $\implies$

$$\int q(x,t_{n+1})\,dx = \int q(x,t_n)\,dx + \int_{t_n}^{t_{n+1}} f(q(x_{i-1/2},t)) - f(q(x_{i+1/2},t))\,dt$$

Finite volume method

$Q_i^n \approx \frac{1}{h}\int_{x_{i-1/2}}^{x_{i+1/2}} q(x,t_n)\,dx$

Integral form:

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

Integrate from t_n to t_{n+1} $\implies$

$$\int q(x,t_{n+1})\,dx = \int q(x,t_n)\,dx + \int_{t_n}^{t_{n+1}} f(q(x_{i-1/2},t)) - f(q(x_{i+1/2},t))\,dt$$

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

Numerical flux: $F_{i-1/2}^n \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(q(x_{i-1/2},t))\,dt.$

Upwind method for advection

Flux: $f(q) = uq$

Numerical flux: $F^n_{i-1/2} \approx \dfrac{1}{\Delta t}\displaystyle\int_{t_n}^{t_{n+1}} f(q(x_{i-1/2},t))\,dt.$

If $q(x,t_n)$ is piecewise constant in each cell, then

$$F^n_{i-1/2} = \begin{cases} uQ^n_{i-1} & \text{if } u > 0, \\ uQ^n_i & \text{if } u < 0. \end{cases}$$

Upwind method for advection

Flux: $f(q) = uq$

Numerical flux: $F^n_{i-1/2} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} f(q(x_{i-1/2},t))\,dt$.

If $q(x, t_n)$ is piecewise constant in each cell, then

$$F^n_{i-1/2} = \begin{cases} uQ^n_{i-1} & \text{if } u > 0, \\ uQ^n_i & \text{if } u < 0. \end{cases}$$

This gives the upwind method:

$$Q_i^{n+1} = Q_i^n - \frac{u\Delta t}{\Delta x}(Q_i^n - Q_{i-1}^n) \qquad \text{if } u > 0$$

$$Q_i^{n+1} = Q_i^n - \frac{u\Delta t}{\Delta x}(Q_{i+1}^n - Q_i^n) \qquad \text{if } u < 0$$

Upwind for advection as a finite volume method

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

Advection equation: $f(q) = uq$

$$F_{i-1/2} \approx \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} uq(x_{i-1/2}, t)\, dt.$$

First order upwind:

$$F_{i-1/2} = u^+ Q_{i-1}^n + u^- Q_i^n$$

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

where $u^+ = \max(u, 0), \;\; u^- = \min(u, 0)$.

Generalize upwind to a linear system?

Consider $q_t + Aq_x = 0$. Eigenvalues are wave speeds.

Upwind method if all $\lambda^p > 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_i^n - AQ_{i-1}^n)$$

Generalize upwind to a linear system?

Consider $q_t + Aq_x = 0$. Eigenvalues are wave speeds.

Upwind method if all $\lambda^p > 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_i^n - AQ_{i-1}^n)$$

Upwind method if all $\lambda^p < 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_{i+1}^n - AQ_i^n)$$

Generalize upwind to a linear system?

Consider $q_t + Aq_x = 0$. Eigenvalues are wave speeds.

Upwind method if all $\lambda^p > 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_i^n - AQ_{i-1}^n)$$

Upwind method if all $\lambda^p < 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_{i+1}^n - AQ_i^n)$$

What if some eigenvalues of each sign?

Generalize upwind to a linear system?

Consider $q_t + Aq_x = 0$. Eigenvalues are wave speeds.

Upwind method if all $\lambda^p > 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_i^n - AQ_{i-1}^n)$$

Upwind method if all $\lambda^p < 0$:

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}(AQ_{i+1}^n - AQ_i^n)$$

What if some eigenvalues of each sign?

Diagonalize and apply scalar upwind to each wave family.

Easier ways to accomplish this!

Lax-Wendroff method

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:

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

Second order of smooth solutions but very dispersive!

Discontinuities or steep gradients $\implies$ nonphysical oscillations.

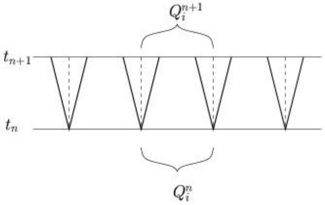
Advection example

Some examples solving the advection equation
with periodic boundary conditions

Using Clawpack and various numerical methods...

www.clawpack.org/g2s3/claw-apps/advection-1d-3/README.html

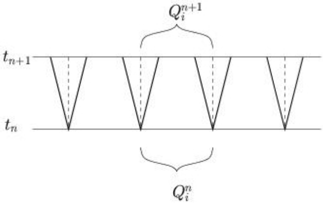
Godunov's Method for $q_t + f(q)_x = 0$



1. Solve Riemann problems at all interfaces, yielding waves $\mathcal{W}^p_{i-1/2}$ and speeds $s^p_{i-1/2}$, for $p = 1,\ 2,\ \ldots,\ m$.

Riemann problem: Original equation with piecewise constant data.

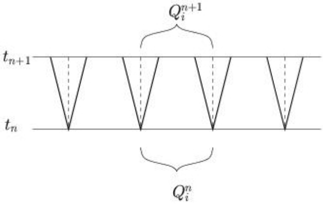
Godunov's Method for $q_t + f(q)_x = 0$



Then either:

1. Compute new cell averages by integrating over cell at t_{n+1},

Godunov's Method for $q_t + f(q)_x = 0$

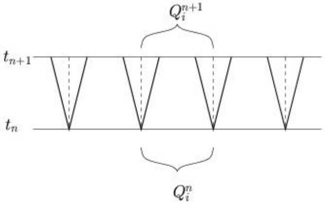


Then either:

1. Compute new cell averages by integrating over cell at t_{n+1},
2. Compute fluxes at interfaces and flux-difference:

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

Godunov's Method for $q_t + f(q)_x = 0$



Then either:

1. Compute new cell averages by integrating over cell at t_{n+1},
2. Compute fluxes at interfaces and flux-difference:

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

3. Update cell averages by contributions from all waves entering cell:

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

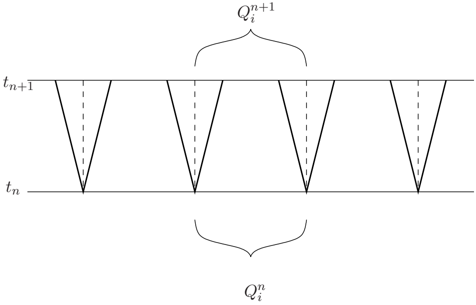
where $\mathcal{A}^\pm \Delta Q_{i-1/2} = \sum_{i=1}^{m} (s_{i-1/2}^p)^\pm \mathcal{W}_{i-1/2}^p$.

Godunov's method with flux differencing

Q_i^n defines a piecewise constant function

$$\tilde{q}^n(x, t_n) = Q_i^n \text{ for } x_{i-1/2} < x < x_{i+1/2}$$

Discontinuities at cell interfaces $\implies$ Riemann problems.



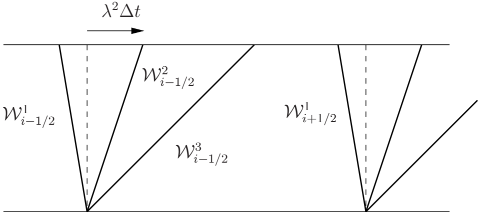
$$\tilde{q}^n(x_{i-1/2}, t) \equiv q^{\vee\!\!|}(Q_{i-1}, Q_i) \quad \text{for } t > t_n.$$

$$F_{i-1/2}^n = \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(q^{\vee\!\!|}(Q_{i-1}^n, Q_i^n))\, dt = f(q^{\vee\!\!|}(Q_{i-1}^n, Q_i^n)).$$

Wave-propagation viewpoint

For linear system $q_t + Aq_x = 0$, the Riemann solution consists of

waves $\mathcal{W}^p$ propagating at constant speed λ^p.



$$Q_i - Q_{i-1} = \sum_{p=1}^{m} \alpha^p_{i-1/2} r^p \equiv \sum_{p=1}^{m} \mathcal{W}^p_{i-1/2}.$$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[\lambda^2 \mathcal{W}^2_{i-1/2} + \lambda^3 \mathcal{W}^3_{i-1/2} + \lambda^1 \mathcal{W}^1_{i+1/2}\right].$$

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 Δt later.

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

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

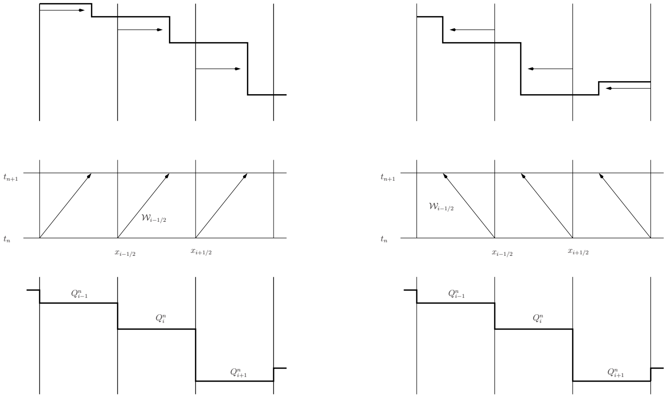
Godunov's method for advection

Q_i^n defines a piecewise constant function

$$\tilde{q}^n(x, t_n) = Q_i^n \text{ for } x_{i-1/2} < x < x_{i+1/2}$$

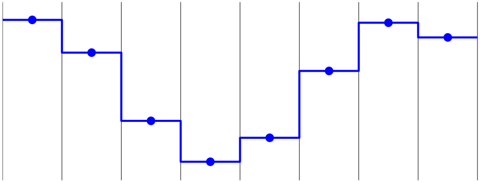
Discontinuities at cell interfaces $\Longrightarrow$ Riemann problems.

$u > 0$ $\qquad\qquad$ $u < 0$

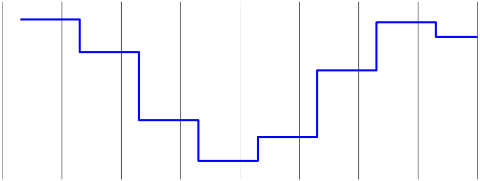


First-order REA Algorithm

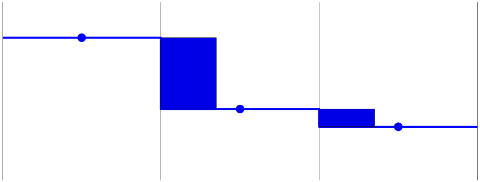
Cell averages and piecewise constant reconstruction:



After evolution:



Cell update



The cell average is modified by

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

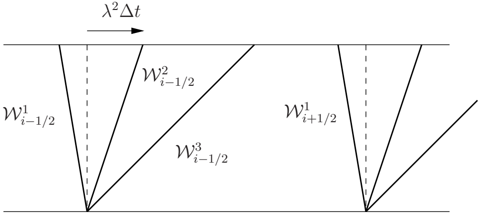
So we obtain the upwind method

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

Wave-propagation viewpoint

For linear system $q_t + Aq_x = 0$, the Riemann solution consists of

waves $\mathcal{W}^p$ propagating at constant speed λ^p.



$$Q_i - Q_{i-1} = \sum_{p=1}^{m} \alpha^p_{i-1/2} r^p \equiv \sum_{p=1}^{m} \mathcal{W}^p_{i-1/2}.$$

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[\lambda^2 \mathcal{W}^2_{i-1/2} + \lambda^3 \mathcal{W}^3_{i-1/2} + \lambda^1 \mathcal{W}^1_{i+1/2}\right].$$

Upwind wave-propagation algorithm

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

or

$$Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x}\left[\mathcal{A}^+\Delta Q_{i-1/2} + \mathcal{A}^-\Delta Q_{i+1/2}\right].$$

where the fluctuations are defined by

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

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

Upwind wave-propagation algorithm

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

where

$$s^+ = \max(s,0), \qquad s^- = \min(s,0).$$

Note: Requires only waves and speeds.

Applicable also to hyperbolic problems not in conservation form.

For $q_t + f(q)_x = 0$, conservative if waves chosen properly,
e.g. using Roe-average of Jacobians.

Great for general software, but only first-order accurate (upwind method for linear systems).

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.

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

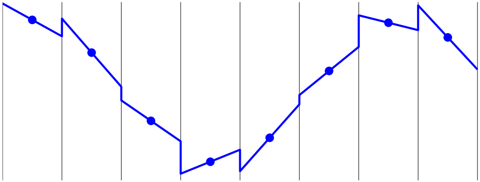
2. Evolve the hyperbolic equation exactly (or approximately) with this initial data to obtain $\tilde{q}^n(x, t_{n+1})$ a time Δt later.

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

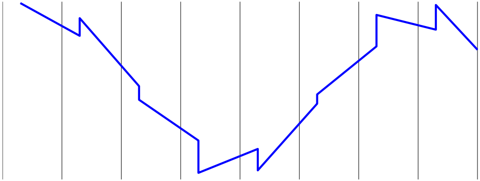
$$Q_i^{n+1} = \frac{1}{\Delta x} \int_{\mathcal{C}_i} \tilde{q}^n(x, t_{n+1})\, dx.$$

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:

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

Choice of slopes:

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

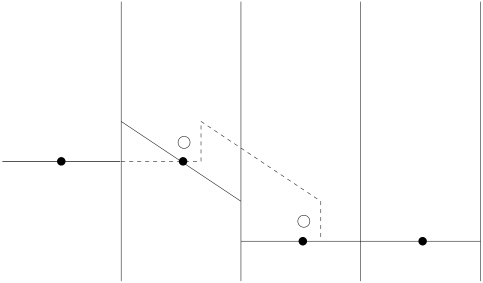
Upwind slope: $\sigma_i^n = \dfrac{Q_i^n - Q_{i-1}^n}{\Delta x}$ (Beam-Warming)

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.

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:

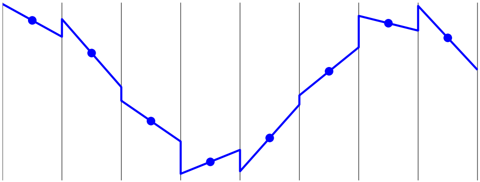
$$\begin{aligned} \sigma_i^n &= \text{minmod}((Q_i^n - Q_{i-1}^n)/\Delta x, \;\; (Q_{i+1}^n - Q_i^n)/\Delta x) \\ &= \left(\frac{Q_{i+1}^n - Q_i^n}{\Delta x}\right)\Phi(\theta_i^n) \end{aligned}$$

where

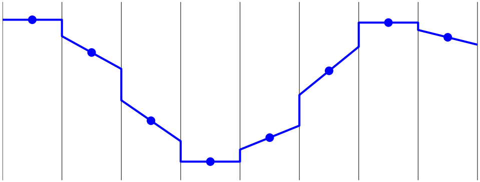
$$\begin{aligned} \theta_i^n &= \frac{Q_i^n - Q_{i-1}^n}{Q_{i+1}^n - Q_i^n} \\ \Phi(\theta) &= \text{minmod}(\theta, 1) \end{aligned}$$

Piecewise linear reconstructions

Lax-Wendroff reconstruction:



Minmod reconstruction:



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:

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

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:

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

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

2d finite volume method for $q_t + f(q)_x + g(q)_y = 0$

Evolution of total mass due to fluxes through cell edges:

$$
\begin{aligned}
\frac{d}{dt} \iint_{\mathcal{C}_{ij}} q(x,y,t)\,dx\,dy = & \int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i+1/2},y,t)\,dy \\
& - \int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i-1/2},y,t)\,dy \\
& + \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x,y_{j+1/2},t)\,dx \\
& - \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x,y_{j-1/2},t)\,dx.
\end{aligned}
$$

2d finite volume method for $q_t + f(q)_x + g(q)_y = 0$

Evolution of total mass due to fluxes through cell edges:

$$\begin{aligned}\frac{d}{dt}\iint_{\mathcal{C}_{ij}} q(x,y,t)\,dx\,dy = &\int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i+1/2},y,t)\,dy \\ &- \int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i-1/2},y,t)\,dy \\ &+ \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x,y_{j+1/2},t)\,dx \\ &- \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x,y_{j-1/2},t)\,dx.\end{aligned}$$

Suggests:

$$\begin{aligned}\frac{\Delta x \Delta y Q_{ij}^{n+1} - \Delta x \Delta y Q_{ij}^{n}}{\Delta t} = &-\Delta y[F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}] \\ &- \Delta x[G_{i,j+1/2}^{n} - G_{i,j-1/2}^{n}],\end{aligned}$$

2d finite volume method for $q_t + f(q)_x + g(q)_y = 0$

$$\Delta x \Delta y Q_{ij}^{n+1} = \Delta x \Delta y Q_{ij}^{n} - \Delta t \Delta y [F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}] \\ - \Delta t \Delta x [G_{i,j+1/2}^{n} - G_{i,j-1/2}^{n}],$$

Where we define numerical fluxes:

$$F_{i-1/2,j}^{n} \approx \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i-1/2}, y, t))\, dy\, dt,$$

$$G_{i,j-1/2}^{n} \approx \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x, y_{j-1/2}, t))\, dx\, dt.$$

2d finite volume method for $q_t + f(q)_x + g(q)_y = 0$

$$\Delta x \Delta y Q_{ij}^{n+1} = \Delta x \Delta y Q_{ij}^{n} - \Delta t \Delta y [F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}] \\ - \Delta t \Delta x [G_{i,j+1/2}^{n} - G_{i,j-1/2}^{n}],$$

Where we define numerical fluxes:

$$F_{i-1/2,j}^{n} \approx \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} f(q(x_{i-1/2}, y, t))\, dy\, dt,$$

$$G_{i,j-1/2}^{n} \approx \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} g(q(x, y_{j-1/2}, t))\, dx\, dt.$$

Rewrite by dividing by $\Delta x \Delta y$:

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}[F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}] \\ - \frac{\Delta t}{\Delta y}[G_{i,j+1/2}^{n} - G_{i,j-1/2}^{n}].$$

2d finite volume method

$$Q_{ij}^{n+1} = Q_{ij}^n - \frac{\Delta t}{\Delta x}[F^n_{i+1/2,j} - F^n_{i-1/2,j}] - \frac{\Delta t}{\Delta y}[G^n_{i,j+1/2} - G^n_{i,j-1/2}].$$

Fluctuation form:

$$Q_{ij}^{n+1} = Q_{ij} - \frac{\Delta t}{\Delta x}(\mathcal{A}^+\Delta Q_{i-1/2,j} + \mathcal{A}^-\Delta Q_{i+1/2,j}) - \frac{\Delta t}{\Delta y}(\mathcal{B}^+\Delta Q_{i,j-1/2} + \mathcal{B}^-\Delta Q_{i,j+1/2}) - \frac{\Delta t}{\Delta x}(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}) - \frac{\Delta t}{\Delta y}(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}).$$

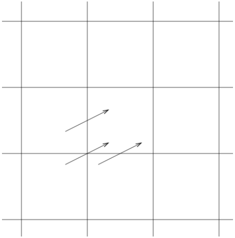
The $\tilde{F}$ and $\tilde{G}$ are correction fluxes to go beyond Godunov's upwind method.

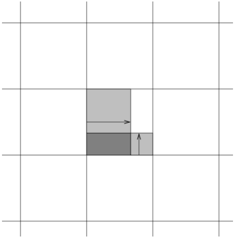
Incorporate approximations to second derivative terms in each direction (q_{xx} and q_{yy}) and mixed term q_{xy}.

Advection: Donor Cell Upwind

With no correction fluxes, Godunov's method for advection is Donor Cell Upwind:

$$Q_{ij}^{n+1} = Q_{ij} - \frac{\Delta t}{\Delta x}[u^+(Q_{ij} - Q_{i-1,j}) + u^-(Q_{i+1,j} - Q_{ij})]$$
$$- \frac{\Delta t}{\Delta y}[v^+(Q_{ij} - Q_{i,j-1}) + v^-(Q_{i,j+1} - Q_{ij})].$$



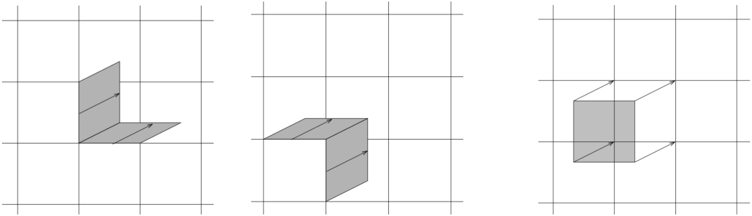


Stable only if $\left|\frac{u\Delta t}{\Delta x}\right| + \left|\frac{v\Delta t}{\Delta y}\right| \leq 1.$

Advection: Corner Transport Upwind (CTU)

Correction fluxes can be added to advect waves correctly.

Corner Transport Upwind:



Stable for $\max\left(\left|\frac{u\Delta t}{\Delta x}\right|, \left|\frac{v\Delta t}{\Delta y}\right|\right) \le 1.$

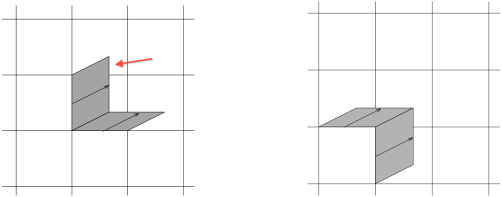
Advection: Corner Transport Upwind (CTU)

Need to transport triangular region from cell (i, j) to $(i, j + 1)$:

$$\text{Area } = \frac{1}{2}(u\Delta t)(v\Delta t) \implies \left(\frac{\frac{1}{2}uv(\Delta t)^2}{\Delta x \Delta y}\right)(Q_{ij} - Q_{i-1,j}).$$

Accomplished by correction flux:

$$\tilde{G}_{i,j+1/2} = -\frac{1}{2}\frac{\Delta t}{\Delta x}\,uv(Q_{ij} - Q_{i-1,j})$$



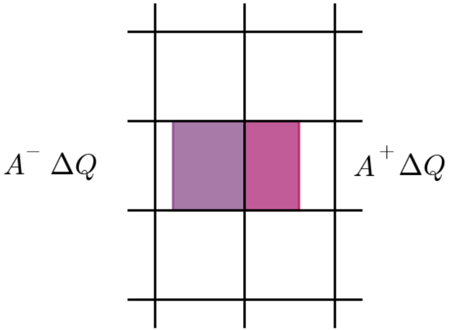
$\frac{\Delta t}{\Delta y}(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2})$ gives approximation to $\frac{1}{2}\Delta t^2 uv q_{xy}$.

$\frac{\Delta t}{\Delta x}(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j})$ gives similar approximation.

Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

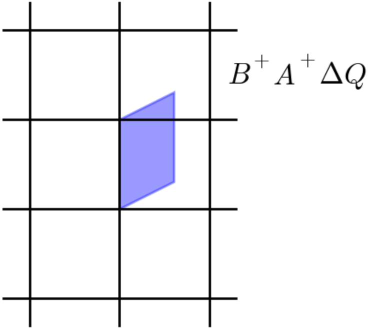
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

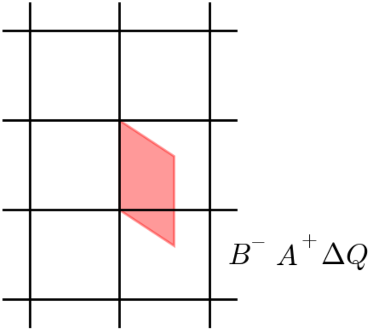
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

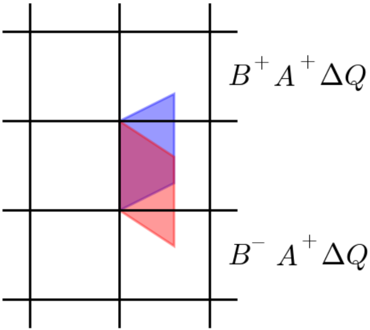
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

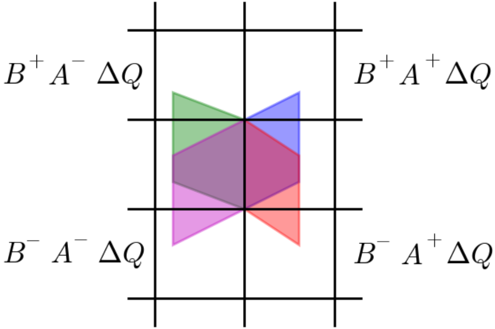
For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Wave propagation algorithm for $q_t + Aq_x + Bq_y = 0$

Decompose $A = A^+ + A^-$ and $B = B^+ + B^-$.

For $\Delta Q = Q_{ij} - Q_{i-1,j}$:



Equations of linear elasticity

$$\begin{aligned}
\sigma^{11}_t - (\lambda + 2\mu) u_x - \lambda v_y &= 0 \\
\sigma^{22}_t - \lambda u_x - (\lambda + 2\mu) v_y &= 0 \\
\sigma^{12}_t - \mu(v_x + u_y) &= 0 \\
\rho u_t - \sigma^{11}_x - \sigma^{12}_y &= 0 \\
\rho v_t - \sigma^{12}_x - \sigma^{22}_y &= 0
\end{aligned}
\qquad
q = \begin{bmatrix} \sigma^{11} \\ \sigma^{22} \\ \sigma^{12} \\ u \\ v \end{bmatrix}$$

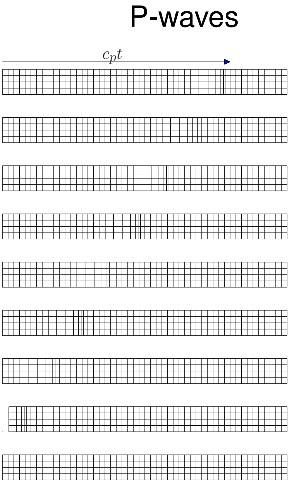
where $\lambda(x, y)$ and $\mu(x, y)$ are Lamé parameters.

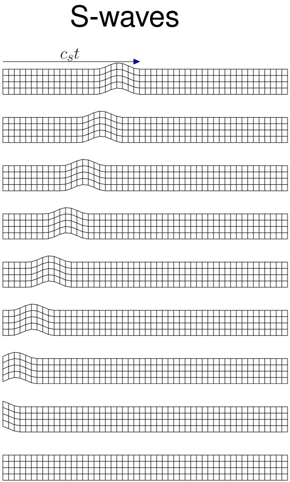
This has the form $q_t + Aq_x + Bq_y = 0$.

The matrix $(A \cos\theta + B \sin\theta)$ has eigenvalues $-c_p,\ -c_s,\ 0,\ c_s,\ c_p$

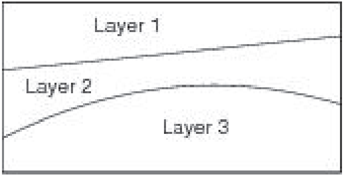
where the P-wave speed and S-wave speed are $c_p = \sqrt{\frac{\lambda+2\mu}{\rho}},\ c_s = \sqrt{\frac{\mu}{\rho}}$

Elastic waves





Seismic waves in layered earth



Layers 1 and 3: $\rho = 2, \;\; \lambda = 1, \;\; \mu = 1, \;\; c_p \approx 1.2, \;\; c_s \approx 0.7$

Layer 2: $\rho = 5, \;\; \lambda = 10, \;\; \mu = 5, \;\; c_p = 2.0, \;\; c_s = 1$

Impulse at top surface at $t = 0$.

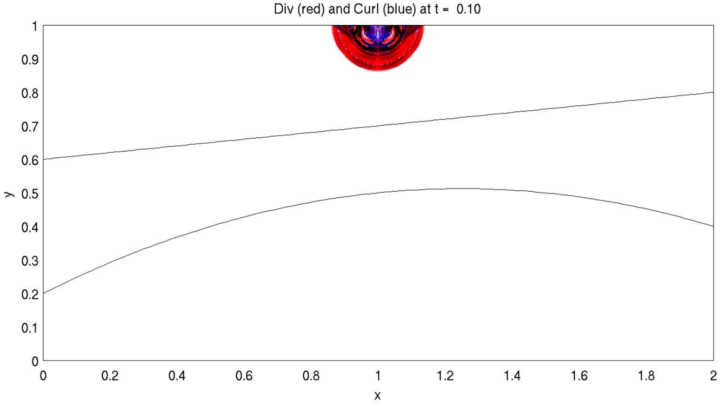
Solved on uniform Cartesian grid (600×300).

Cell average of material parameters used in each finite volume cell.

Extrapolation at computational boundaries.

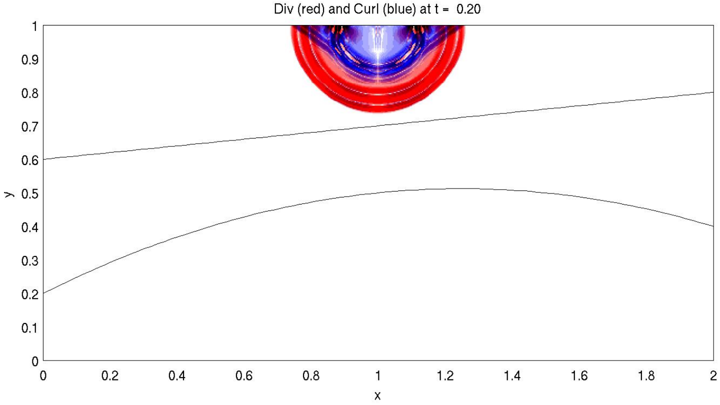
Seismic wave in layered medium

Red = div(u) [P-waves], Blue = curl(u) [S-waves]



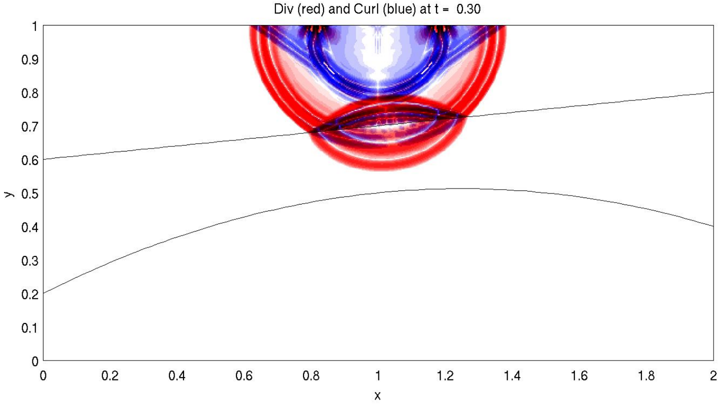
Seismic wave in layered medium

Red = div(u) [P-waves],   Blue = curl(u) [S-waves]



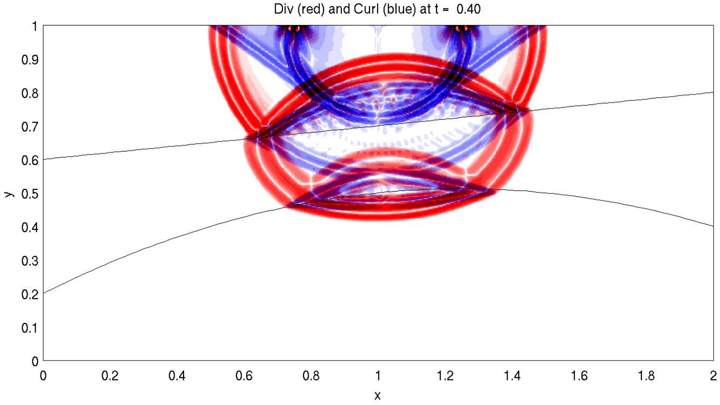
Seismic wave in layered medium

Red = div(u) [P-waves], Blue = curl(u) [S-waves]



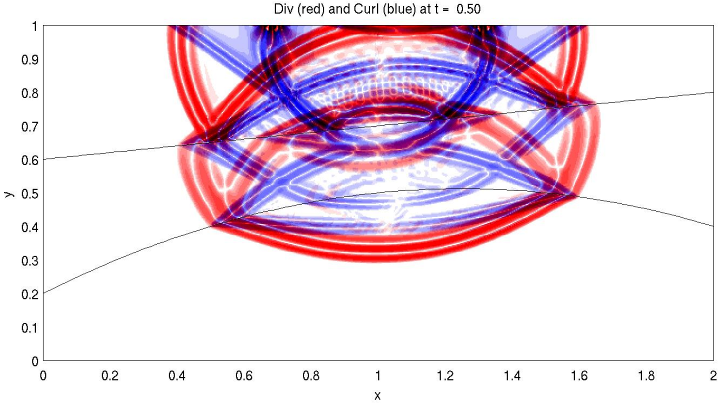
Seismic wave in layered medium

Red = div(u) [P-waves],   Blue = curl(u) [S-waves]



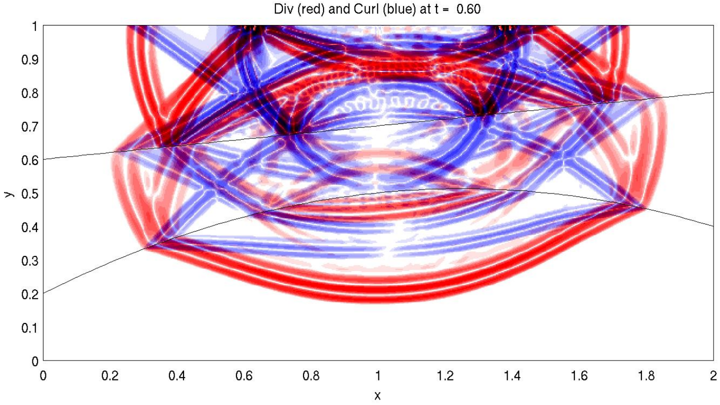
Seismic wave in layered medium

Red = div(u) [P-waves],   Blue = curl(u) [S-waves]



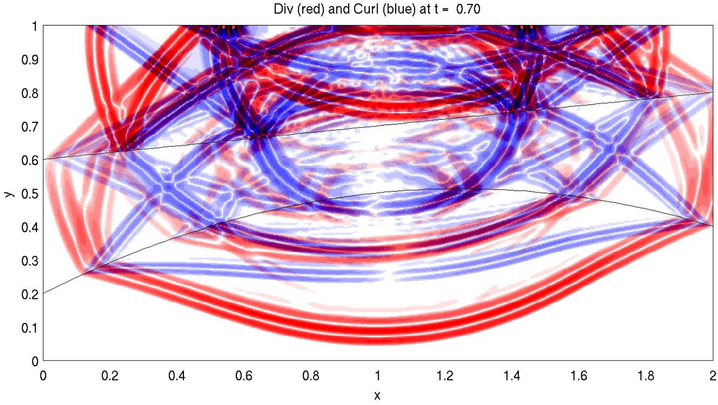
Seismic wave in layered medium

Red = div(u) [P-waves], Blue = curl(u) [S-waves]



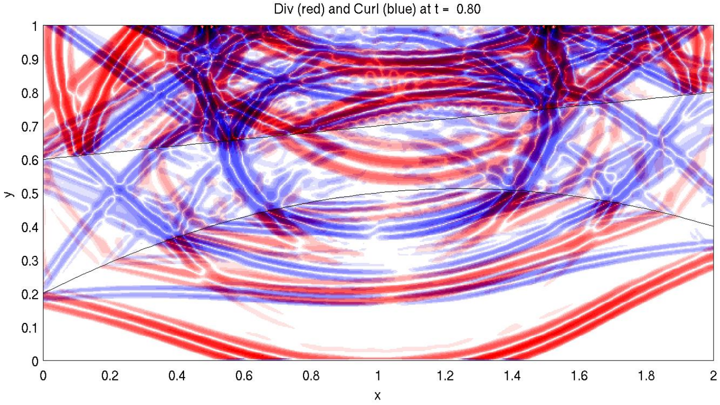
Seismic wave in layered medium

Red = div(u) [P-waves],   Blue = curl(u) [S-waves]



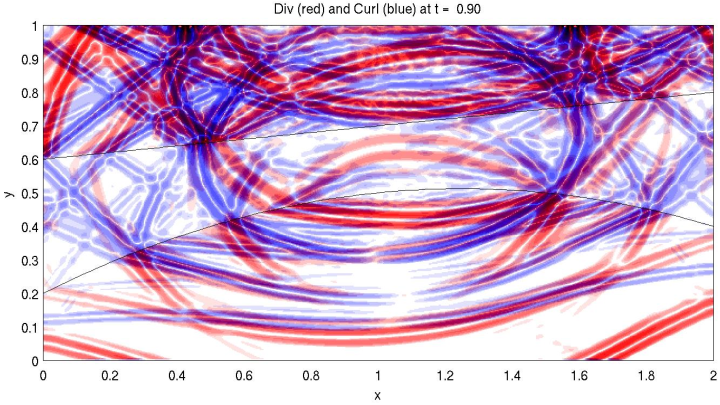
Seismic wave in layered medium

Red = div(u) [P-waves], Blue = curl(u) [S-waves]



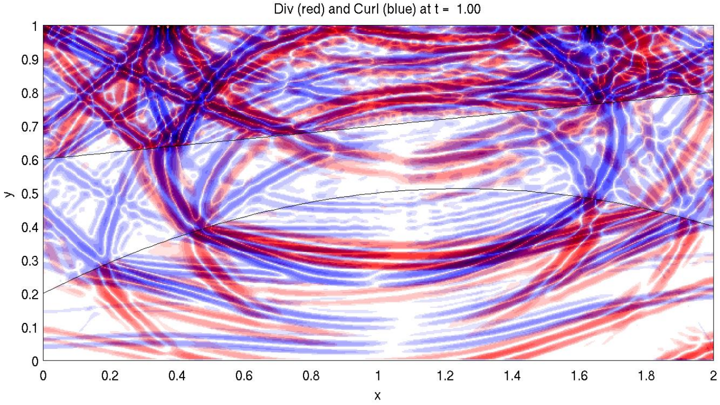
Seismic wave in layered medium

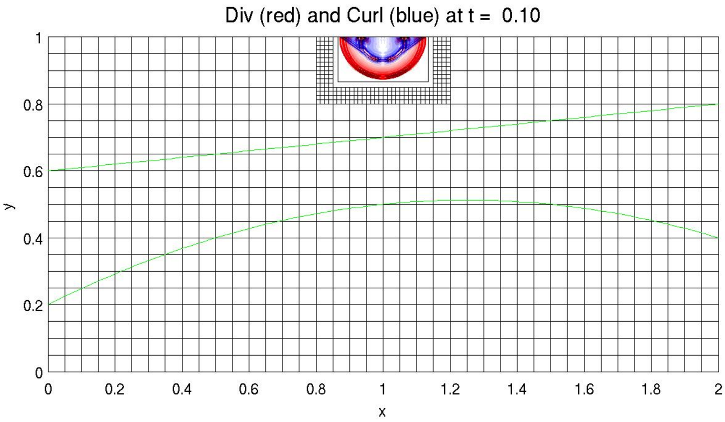
Red = div(u) [P-waves],   Blue = curl(u) [S-waves]



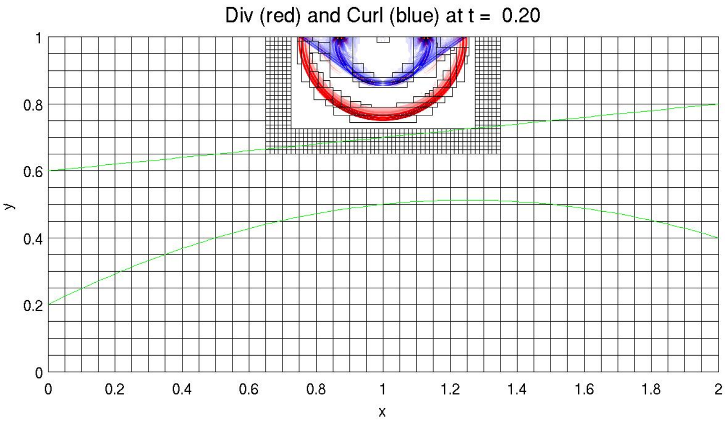
Seismic wave in layered medium

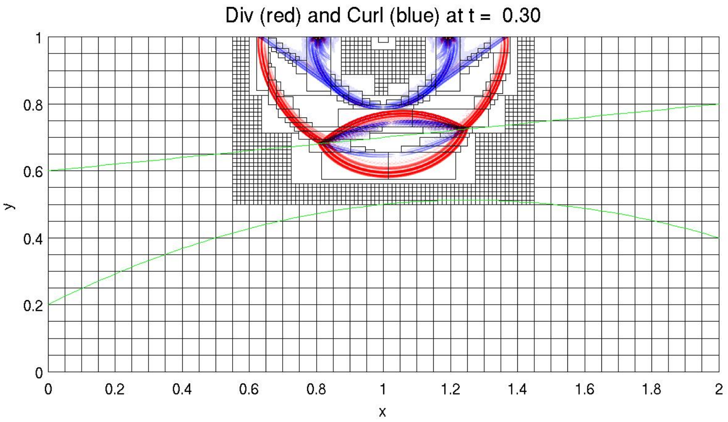
Red = div(u) [P-waves],   Blue = curl(u) [S-waves]

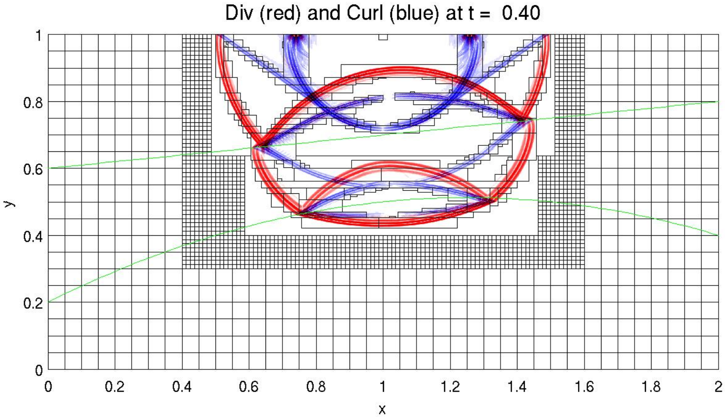




Div (red) and Curl (blue) at t = 0.10
1
0.8
0.6
0.4
0.2
0
y
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x


Div (red) and Curl (blue) at t = 0.20
y
x
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2


Div (red) and Curl (blue) at t = 0.30
1
0.8
0.6
0.4
0.2
0
y
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x



Div (red) and Curl (blue) at t = 0.40
1
0.8
0.6
0.4
0.2
0
y
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2
x

Extra Material

You might want to work through

the following slides on your own!

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 $TV(\tilde{q}^n) \le TV(Q^n)$.

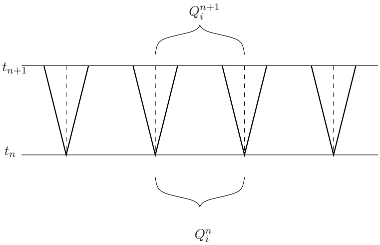
2. Evolve the hyperbolic equation exactly (or approximately) with this initial data to obtain $\tilde{q}^n(x, t_{n+1})$ a time k later.

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.

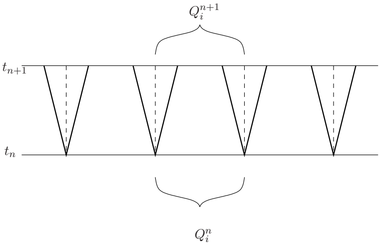
Godunov (upwind) on acoustics



Data at time t_n : $\tilde{q}^n(x, t_n) = Q_i^n$ for $x_{i-1/2} < x < x_{i+1/2}$
Solving Riemann problems for small Δt gives solution:

$$\tilde{q}^n(x, t_{n+1}) = \begin{cases} Q^*_{i-1/2} & \text{if } x_{i-1/2} - c\Delta t < x < x_{i-1/2} + c\Delta t, \\ Q_i^n & \text{if } x_{i-1/2} + c\Delta t < x < x_{i+1/2} - c\Delta t, \\ Q^*_{i+1/2} & \text{if } x_{i+1/2} - c\Delta t < x < x_{i+1/2} + c\Delta t, \end{cases}$$

Godunov (upwind) on acoustics



Data at time t_n : $\tilde{q}^n(x, t_n) = Q_i^n$ for $x_{i-1/2} < x < x_{i+1/2}$
Solving Riemann problems for small Δt gives solution:

$$\tilde{q}^n(x, t_{n+1}) = \begin{cases} Q^*_{i-1/2} & \text{if } x_{i-1/2} - c\Delta t < x < x_{i-1/2} + c\Delta t, \\ Q_i^n & \text{if } x_{i-1/2} + c\Delta t < x < x_{i+1/2} - c\Delta t, \\ Q^*_{i+1/2} & \text{if } x_{i+1/2} - c\Delta t < x < x_{i+1/2} + c\Delta t, \end{cases}$$

So computing cell average gives:

$$Q_i^{n+1} = \frac{1}{\Delta x}\left[c\Delta t Q^*_{i-1/2} + (\Delta x - 2c\Delta t)Q_i^n + c\Delta t Q^*_{i+1/2}\right].$$

Godunov (upwind) on acoustics

$$Q_i^{n+1} = \frac{1}{\Delta x}\left[c\Delta t Q^*_{i-1/2} + (\Delta x - 2c\Delta t)Q_i^n + c\Delta t Q^*_{i+1/2}\right].$$

Solve Riemann problems:

$$Q_i^n - Q_{i-1}^n = \Delta Q_{i-1/2} = \mathcal{W}^1_{i-1/2} + \mathcal{W}^2_{i-1/2} = \alpha^1_{i-1/2} r^1 + \alpha^2_{i-1/2} r^2,$$
$$Q_{i+1}^n - Q_i^n = \Delta Q_{i+1/2} = \mathcal{W}^1_{i+1/2} + \mathcal{W}^2_{i+1/2} = \alpha^1_{i+1/2} r^1 + \alpha^2_{i+1/2} r^2,$$

Godunov (upwind) on acoustics

$$Q_i^{n+1} = \frac{1}{\Delta x}\left[c\Delta t Q^*_{i-1/2} + (\Delta x - 2c\Delta t)Q_i^n + c\Delta t Q^*_{i+1/2}\right].$$

Solve Riemann problems:

$$Q_i^n - Q_{i-1}^n = \Delta Q_{i-1/2} = \mathcal{W}^1_{i-1/2} + \mathcal{W}^2_{i-1/2} = \alpha^1_{i-1/2} r^1 + \alpha^2_{i-1/2} r^2,$$
$$Q_{i+1}^n - Q_i^n = \Delta Q_{i+1/2} = \mathcal{W}^1_{i+1/2} + \mathcal{W}^2_{i+1/2} = \alpha^1_{i+1/2} r^1 + \alpha^2_{i+1/2} r^2,$$

The intermediate states are:

$$Q^*_{i-1/2} = Q_i^n - \mathcal{W}^2_{i-1/2}, \qquad Q^*_{i+1/2} = Q_i^n + \mathcal{W}^1_{i+1/2},$$

Godunov (upwind) on acoustics

$$Q_i^{n+1} = \frac{1}{\Delta x}\left[c\Delta t Q^*_{i-1/2} + (\Delta x - 2c\Delta t)Q_i^n + c\Delta t Q^*_{i+1/2}\right].$$

Solve Riemann problems:

$$Q_i^n - Q_{i-1}^n = \Delta Q_{i-1/2} = \mathcal{W}^1_{i-1/2} + \mathcal{W}^2_{i-1/2} = \alpha^1_{i-1/2}r^1 + \alpha^2_{i-1/2}r^2,$$
$$Q_{i+1}^n - Q_i^n = \Delta Q_{i+1/2} = \mathcal{W}^1_{i+1/2} + \mathcal{W}^2_{i+1/2} = \alpha^1_{i+1/2}r^1 + \alpha^2_{i+1/2}r^2,$$

The intermediate states are:

$$Q^*_{i-1/2} = Q_i^n - \mathcal{W}^2_{i-1/2}, \qquad Q^*_{i+1/2} = Q_i^n + \mathcal{W}^1_{i+1/2},$$

So,

$$\begin{aligned} Q_i^{n+1} &= \frac{1}{\Delta x}\left[c\Delta t(Q_i^n - \mathcal{W}^2_{i-1/2}) + (\Delta x - 2c\Delta t)Q_i^n + c\Delta t(Q_i^n + \mathcal{W}^1_{i+1/2})\right] \\ &= Q_i^n - \frac{c\Delta t}{\Delta x}\mathcal{W}^2_{i-1/2} + \frac{c\Delta t}{\Delta x}\mathcal{W}^1_{i+1/2}. \end{aligned}$$

Godunov (upwind) on acoustics

Solve Riemann problems:

$$Q_i^n - Q_{i-1}^n = \Delta Q_{i-1/2} = \mathcal{W}_{i-1/2}^1 + \mathcal{W}_{i-1/2}^2 = \alpha_{i-1/2}^1 r^1 + \alpha_{i-1/2}^2 r^2,$$

$$Q_{i+1}^n - Q_i^n = \Delta Q_{i+1/2} = \mathcal{W}_{i+1/2}^1 + \mathcal{W}_{i+1/2}^2 = \alpha_{i+1/2}^1 r^1 + \alpha_{i+1/2}^2 r^2,$$

The waves are determined by solving for α from $R\alpha = \Delta Q$:

$$A = \begin{bmatrix} 0 & K \\ 1/\rho & 0 \end{bmatrix}, \qquad R = \begin{bmatrix} -Z & Z \\ 1 & 1 \end{bmatrix}, \qquad R^{-1} = \frac{1}{2Z} \begin{bmatrix} -1 & Z \\ 1 & Z \end{bmatrix}.$$

So

$$\Delta Q = \begin{bmatrix} \Delta p \\ \Delta u \end{bmatrix} = \alpha^1 \begin{bmatrix} -Z \\ 1 \end{bmatrix} + \alpha^2 \begin{bmatrix} Z \\ 1 \end{bmatrix}$$

with

$$\alpha^1 = \frac{1}{2Z}(-\Delta p + Z\Delta u), \qquad \alpha^2 = \frac{1}{2Z}(\Delta p + Z\Delta u).$$

CLAWPACK Riemann solver

The hyperbolic problem is specified by the Riemann solver

- Input: Values of q in each grid cell
- Output: Solution to Riemann problem at each interface.
  - Waves $\mathcal{W}^p \in \mathbb{R}^m$, $p = 1,\ 2,\ \ldots,\ M_w$
  - Speeds $s^p \in \mathbb{R}$, $p = 1,\ 2,\ \ldots,\ M_w$,
  - Fluctuations $\mathcal{A}^-\Delta Q,\ \mathcal{A}^+\Delta Q \in \mathbb{R}^m$

Note: Number of waves M_w often equal to m (length of q), but could be different (e.g. HLL solver has 2 waves).

CLAWPACK Riemann solver

The hyperbolic problem is specified by the Riemann solver

- Input: Values of q in each grid cell
- Output: Solution to Riemann problem at each interface.
  - Waves $\mathcal{W}^p \in \mathbb{R}^m$, $p = 1,\ 2,\ \ldots,\ M_w$
  - Speeds $s^p \in \mathbb{R}$, $p = 1,\ 2,\ \ldots,\ M_w$,
  - Fluctuations $\mathcal{A}^-\Delta Q,\ \mathcal{A}^+\Delta Q \in \mathbb{R}^m$

Note: Number of waves M_w often equal to m (length of q), but could be different (e.g. HLL solver has 2 waves).

Fluctuations:

$$\mathcal{A}^-\Delta Q = \text{Contribution to cell average to left,}$$
$$\mathcal{A}^+\Delta Q = \text{Contribution to cell average to right}$$

For conservation law, $\mathcal{A}^-\Delta Q + \mathcal{A}^+\Delta Q = f(Q_r) - f(Q_l)$

CLAWPACK Riemann solver

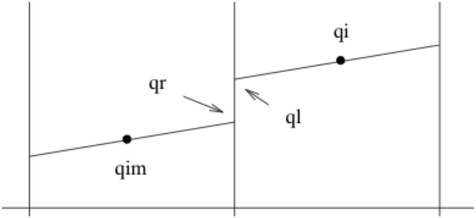
Inputs to `rp1` subroutine:

`ql(i,1:m)` = Value of q at left edge of ith cell,

`qr(i,1:m)` = Value of q at right edge of ith cell,

Warning: The Riemann problem at the interface between cells $i-1$ and i has left state `qr(i-1,:)` and right state `ql(i,:)`.

`rp1` is normally called with `ql = qr = q`,
but designed to allow other methods:



Wave propagation algorithms in 2D

Clawpack requires:

Normal Riemann solver `rpn2.f`
Solves 1d Riemann problem $q_t + Aq_x = 0$
Decomposes $\Delta Q = Q_{ij} - Q_{i-1,j}$ into $\mathcal{A}^+\Delta Q$ and $\mathcal{A}^-\Delta Q$.
For $q_t + Aq_x + Bq_y = 0$, split using eigenvalues, vectors:

$$A = R\Lambda R^{-1} \implies A^- = R\Lambda^- R^{-1}, A^+ = R\Lambda^+ R^{-1}$$

Input parameter `ixy` determines if it's in x or y direction.
In latter case splitting is done using B instead of A.
This is all that's required for dimensional splitting.

Wave propagation algorithms in 2D

Clawpack requires:

Normal Riemann solver `rpn2.f`
Solves 1d Riemann problem $q_t + Aq_x = 0$
Decomposes $\Delta Q = Q_{ij} - Q_{i-1,j}$ into $\mathcal{A}^+\Delta Q$ and $\mathcal{A}^-\Delta Q$.
For $q_t + Aq_x + Bq_y = 0$, split using eigenvalues, vectors:

$$A = R\Lambda R^{-1} \implies A^- = R\Lambda^- R^{-1}, A^+ = R\Lambda^+ R^{-1}$$

Input parameter `ixy` determines if it's in x or y direction.
In latter case splitting is done using B instead of A.
This is all that's required for dimensional splitting.

Transverse Riemann solver `rpt2.f`
Decomposes $\mathcal{A}^+\Delta Q$ into $\mathcal{B}^-\mathcal{A}^+\Delta Q$ and $\mathcal{B}^+\mathcal{A}^+\Delta Q$ by splitting this vector into eigenvectors of B.

(Or splits vector into eigenvectors of A if `ixy=2`.)