FYS-GEO 4500

Lecture Notes #8
Source Terms and Multidimensions

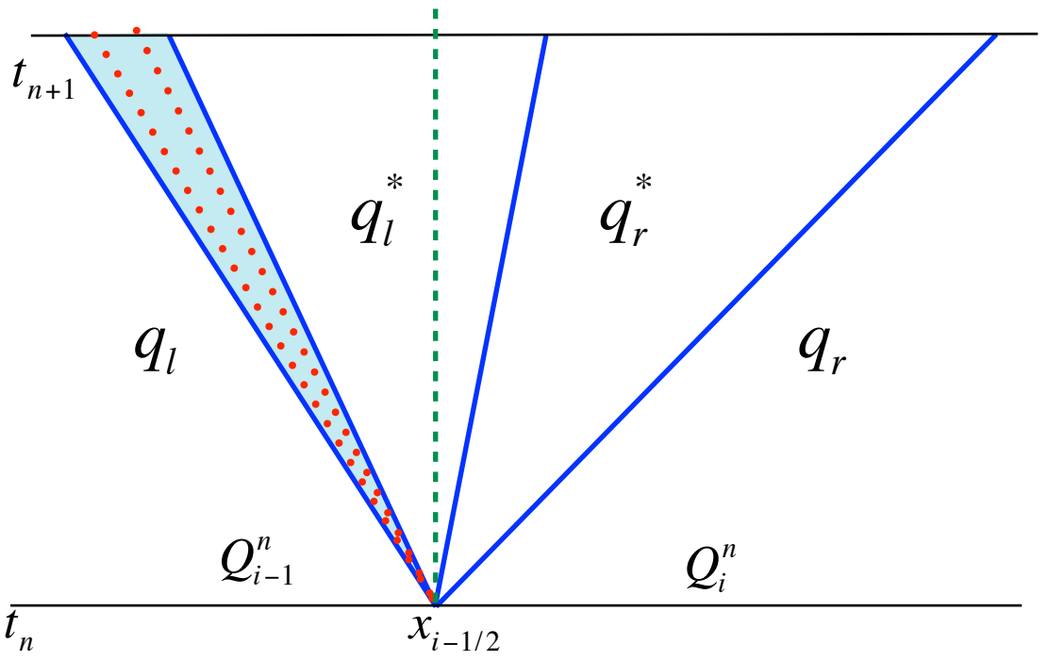
Where we are today

	date	Topic	Chapter in LeVeque
1	1.sep. 2011	introduction to conservation laws, Clawpack	1 & 2
2	15.sep. 2011	the Riemann problem, characteristics	3 & 5
3	22.sep. 2011	finite volume methods for linear systems, high resolution	4 & 6
4	29.sep. 2011	boundary conditions, accuracy, variable coeff.	7,8, part 9
5	6.okt. 2011	nonlinear conservation laws, finite volume methods	11 & 12
6	13.okt. 2011	nonlinear equations & systems	13 & 14
	20.okt. 2011	no lecture	
7	27.okt. 2011	finite volume methods for nonlinear systems	15,16,17
8	3.nov. 2011	multidimensional systems and source terms, etc.	18, 19, 20, 21
	10.nov. 2011	no lecture	
9	17.nov. 2011	waves in elastic media	22
10	24.nov. 2011	unfinished business: capacity functions, source terms, project plans	
11	1.des. 2011	student presentations	
	8.des. 2011	no lecture	
12	15.des. 2011	FINAL REPORTS DUE	

The exact Riemann solver for the nonlinear problem is expensive, and most of it is not necessary!

$$u = \varphi_l(p) = \begin{cases} u_l + \dfrac{2c_l}{\gamma-1}\left[1-\left(p/p_l\right)^{\frac{\gamma-1}{2\gamma}}\right] & \text{if} \quad p \le p_l \\ u_l + \dfrac{2c_l}{\sqrt{2\gamma(\gamma-1)}}\left[\dfrac{1-p/p_l}{\sqrt{1+\beta p/p_l}}\right] & \text{if} \quad p \ge p_l \end{cases}$$

$$u = \varphi_r(p) = \begin{cases} u_r - \dfrac{2c_r}{\gamma-1}\left[1-\left(p/p_r\right)^{\frac{\gamma-1}{2\gamma}}\right] & \text{if} \quad p \le p_r \\ u_r - \dfrac{2c_r}{\sqrt{2\gamma(\gamma-1)}}\left[\dfrac{1-p/p_r}{\sqrt{1+\beta p/p_r}}\right] & \text{if} \quad p \ge p_r \end{cases}$$



Solving $\varphi_l(p_m) = \varphi_r(p_m)$ yields (p^*, u^*), then $\rho_l^* = \left(\dfrac{1+\beta p^*/p_l}{p^*/p_l+\beta}\right)\rho_l$; $\rho_r^* = \left(\dfrac{1+\beta p^*/p_r}{p^*/p_r+\beta}\right)\rho_r$

In the rarefaction fan, $u(\xi) = \dfrac{(\gamma-1)u_l + 2(c_l+\xi)}{\gamma+1}$.

$$\rho(\xi) = \left(\frac{\rho_l^\gamma}{\gamma p_l}\left(u(\xi)-\xi\right)^2\right)^{1/(\gamma-1)}$$

$$p(\xi) = \left(\frac{p_l}{\rho_l^\gamma}\right)\rho(\xi)^\gamma$$

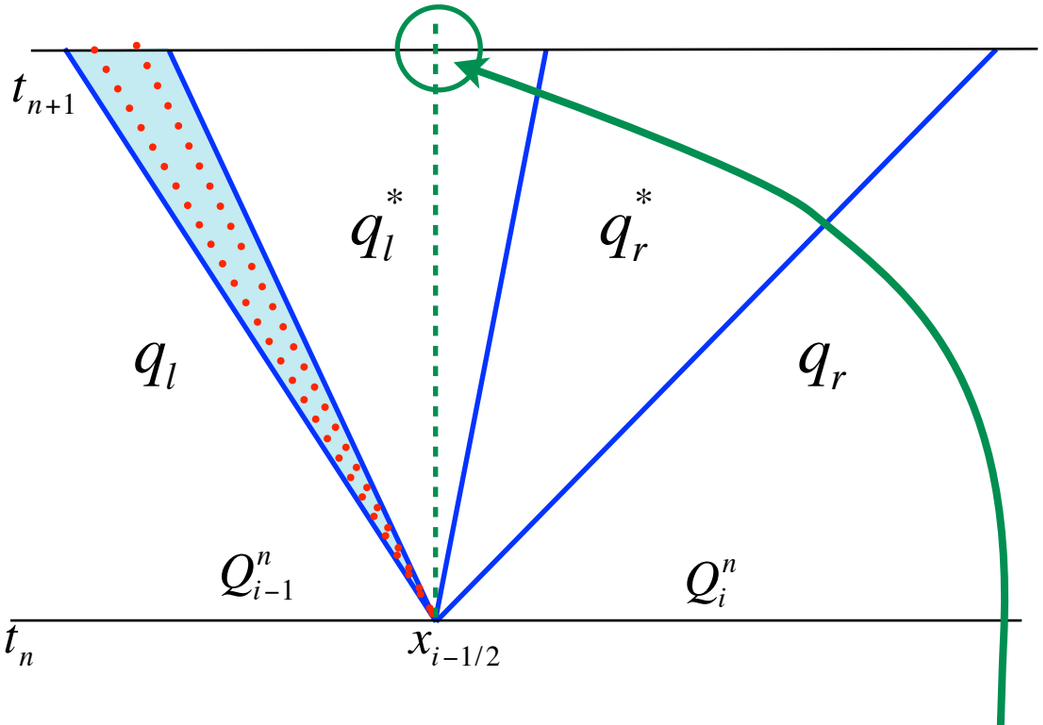
These are useful for exact solutions of certain problems.

But in simulations we only need the solution at the cell interface!

The exact Riemann solver for the nonlinear problem is expensive, and most of it is not necessary!

$$u = \varphi_l(p) = \begin{cases} u_l + \dfrac{2c_l}{\gamma-1}\left[1-\left(p/p_l\right)^{\frac{\gamma-1}{2\gamma}}\right] & \text{if} \quad p \le p_l \\ u_l + \dfrac{2c_l}{\sqrt{2\gamma(\gamma-1)}}\left[\dfrac{1-p/p_l}{\sqrt{1+\beta p/p_l}}\right] & \text{if} \quad p \ge p_l \end{cases}$$

$$u = \varphi_r(p) = \begin{cases} u_r - \dfrac{2c_r}{\gamma-1}\left[1-\left(p/p_r\right)^{\frac{\gamma-1}{2\gamma}}\right] & \text{if} \quad p \le p_r \\ u_r - \dfrac{2c_r}{\sqrt{2\gamma(\gamma-1)}}\left[\dfrac{1-p/p_r}{\sqrt{1+\beta p/p_r}}\right] & \text{if} \quad p \ge p_r \end{cases}$$



Solving $\varphi_l(p_m) = \varphi_r(p_m)$ yields (p^*, u^*), then $\rho_l^* = \left(\dfrac{1+\beta p^*/p_l}{p^*/p_l+\beta}\right)\rho_l$; $\rho_r^* = \left(\dfrac{1+\beta p^*/p_r}{p^*/p_r+\beta}\right)\rho_r$

In the rarefaction fan, $u(\xi) = \dfrac{(\gamma-1)u_l + 2(c_l+\xi)}{\gamma+1}$.

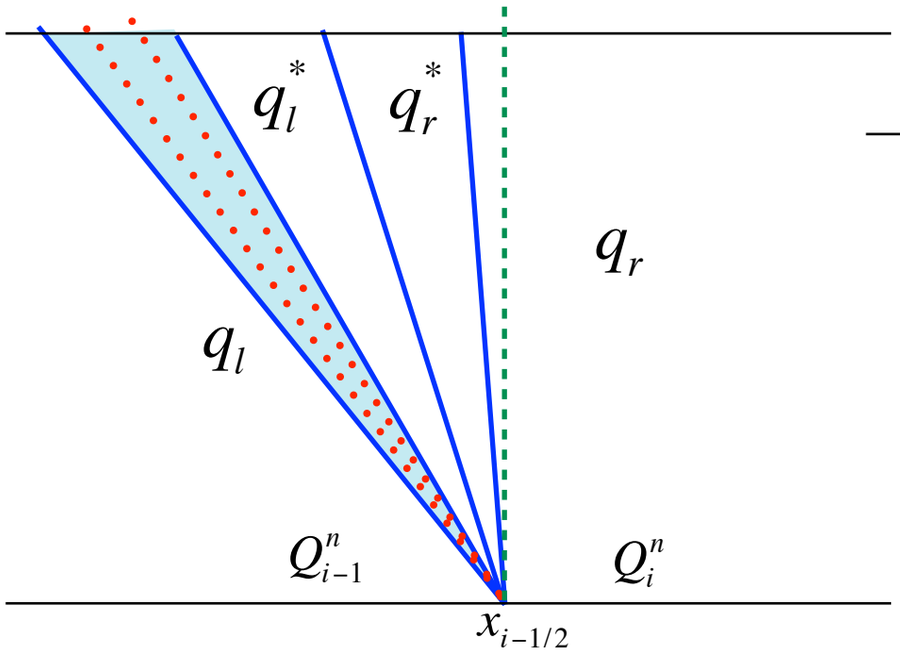
$$\rho(\xi) = \left(\frac{\rho_l^\gamma}{\gamma p_l}\left(u(\xi)-\xi\right)^2\right)^{1/(\gamma-1)}$$

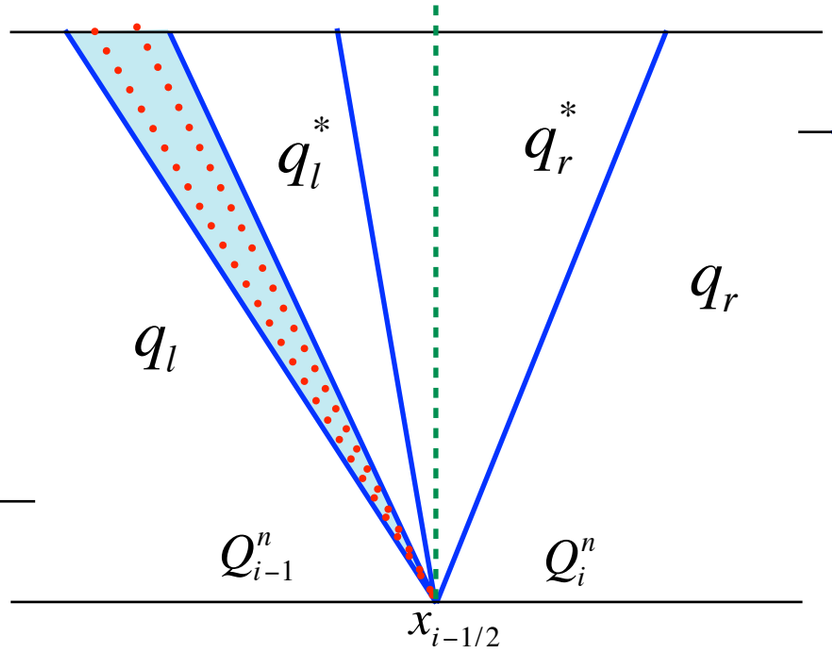
$$p(\xi) = \left(\frac{p_l}{\rho_l^\gamma}\right)\rho(\xi)^\gamma$$

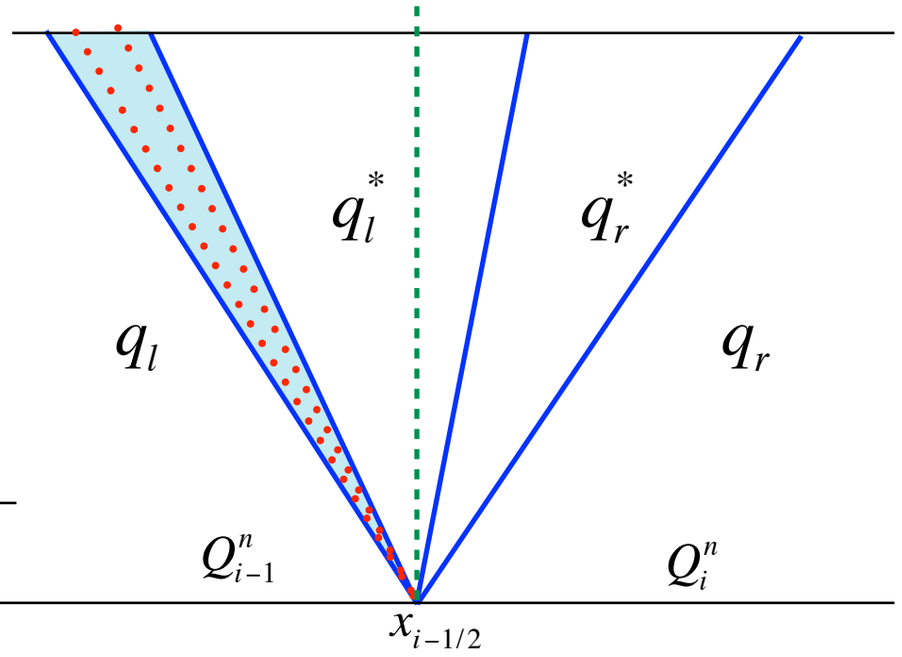
These are useful for exact solutions of certain problems.

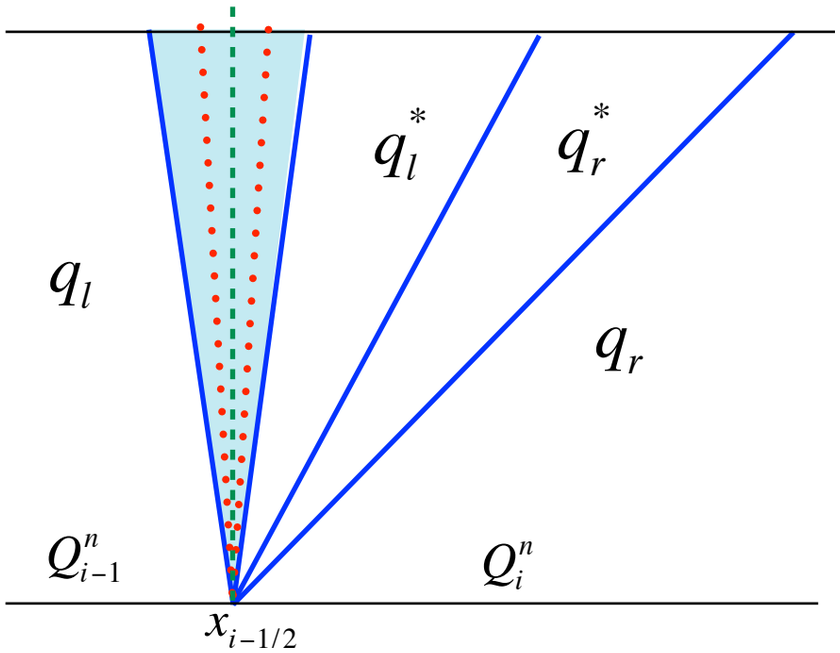
But in simulations we only need the solution at the cell interface!

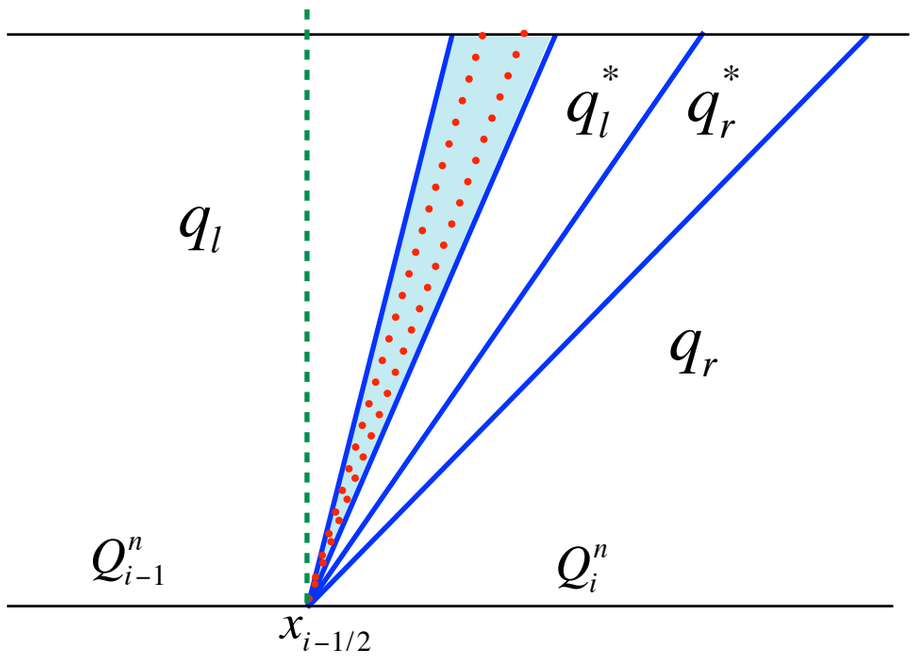
But we do need to know where the cell interface lies with respect to the waves, and compute accordingly











Wave propagation for nonlinear systems

An **approximate** Riemann solver is typically used to get the wave decomposition

$$Q_i - Q_{i-1} = \sum_{p=1}^{m} \mathcal{W}_{i-1/2}^p,$$

where the wave $\mathcal{W}_{i-1/2}^p$ propagates at a speed $s_{i-1/2}^p$.

If we define $\hat{A}_{i-1/2} = \hat{A}(Q_i^n, Q_{i-1}^n)$ as a linearised approximation to $f'(q)$ valid in the neighbourhood of (Q_i, $Q_{i\text{-}1}$),

then we can solve the simpler linear Riemann problem at that cell interface for the linearised equation:

$$q_t + \hat{A}_{i-1/2} q_x = 0,$$

to obtain

$$\mathcal{W}_{i-1/2}^p = \alpha_{i-1/2}^p \hat{r}_{i-1/2}^p, \quad s_{i-1/2}^p = \hat{\lambda}_{i-1/2}^p .$$

Roe's approximate Riemann solver

Roe suggested these constraints for $\hat{A}$:

1. $\hat{A}_{rl}(q_r - q_l) = f(q_r) - f(q_l)$ Cf. Rankine-Hugoniot condition.
2. $\hat{A}$ is diagonalisable with real eigenvalues.
3. $\hat{A}_{rl} \to f'(\overline{q})$ smoothly as $q_l, q_r \to \overline{q}$.

A single shock is captured exactly because (1.) is essentially the Rankine-Hugoniot jump condition.

$f(q_r) - f(q_l) = s(q_r - q_l)$ implies that $q_r - q_l$ is an eigenvector of $\hat{A}$.

It is a good approximation for weak waves, or smooth flow.

The wave-propagation algorithm is also conservative since

$$\mathcal{A}^- \Delta Q_{i-1/2} + \mathcal{A}^+ \Delta Q_{i-1/2} = \sum_p s^p_{i-1/2} \mathcal{W}^p_{i-1/2} = \hat{A} \sum_p \mathcal{W}^p_{i-1/2} .$$

Roe solver for the Euler equations

The eigensystem of the Euler equations for a polytropic gas is:

$$\lambda^1 = u - c \qquad \lambda^2 = u \qquad \lambda^3 = u + c$$

$$r^1 = \begin{bmatrix} 1 \\ u - c \\ H - uc \end{bmatrix} \quad r^2 = \begin{bmatrix} 1 \\ u \\ \frac{1}{2}u^2 \end{bmatrix} \quad r^3 = \begin{bmatrix} 1 \\ u + c \\ H + uc \end{bmatrix}$$

These need to be evaluated at the Roe-averaged state, so we need the Roe averages for u, H, c. These are:

$$\hat{u} = \frac{\sqrt{\rho_{i-1}}u_{i-1} + \sqrt{\rho_i}u_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}}$$

$$\hat{H} = \frac{\sqrt{\rho_{i-1}}H_{i-1} + \sqrt{\rho_i}H_i}{\sqrt{\rho_{i-1}} + \sqrt{\rho_i}}$$

$$\hat{c} = \sqrt{(\gamma - 1)\left(\hat{H} - \frac{1}{2}\hat{u}^2\right)}$$

Roe solver for the Euler equations

Then the wave decomposition between the left and right states is

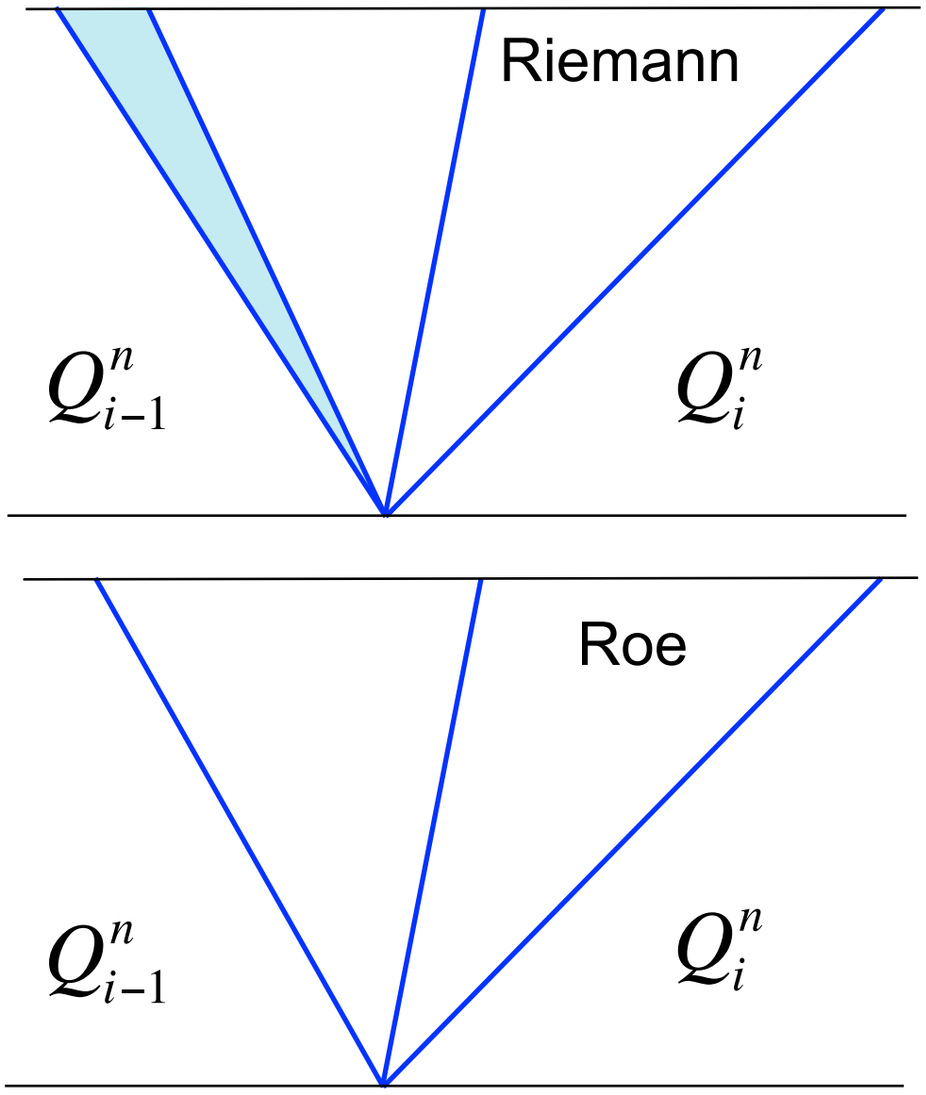
$$\delta \equiv Q_i - Q_{i-1} = \alpha^1 \hat{r}^1 + \alpha^2 \hat{r}^2 + \alpha^3 \hat{r}^3$$

where

$$\alpha^2 = (\gamma - 1)\frac{\left(\hat{H} - \hat{u}^2\right)\delta^1 + \hat{u}\delta^2 - \delta^3}{\hat{c}^2}$$

$$\alpha^3 = \frac{\delta^2 + (\hat{c} - \hat{u})\delta^1 - \hat{c}\alpha^2}{2\hat{c}}$$

$$\alpha^1 = \delta^1 - \alpha^2 - \alpha^3$$



But note that, while the Riemann solution consists of three waves, one of which is a rarefaction fan, the Roe solution only consists of three waves. In most cases this does not matter, since the desired solution at $x/t=0$ will be the same intermediate state. In the case of a transonic rarefaction a modification (in the form of an entropy fix) is necessary.

Roe solver for the Euler equations

Then the wave decomposition between the left and right states is

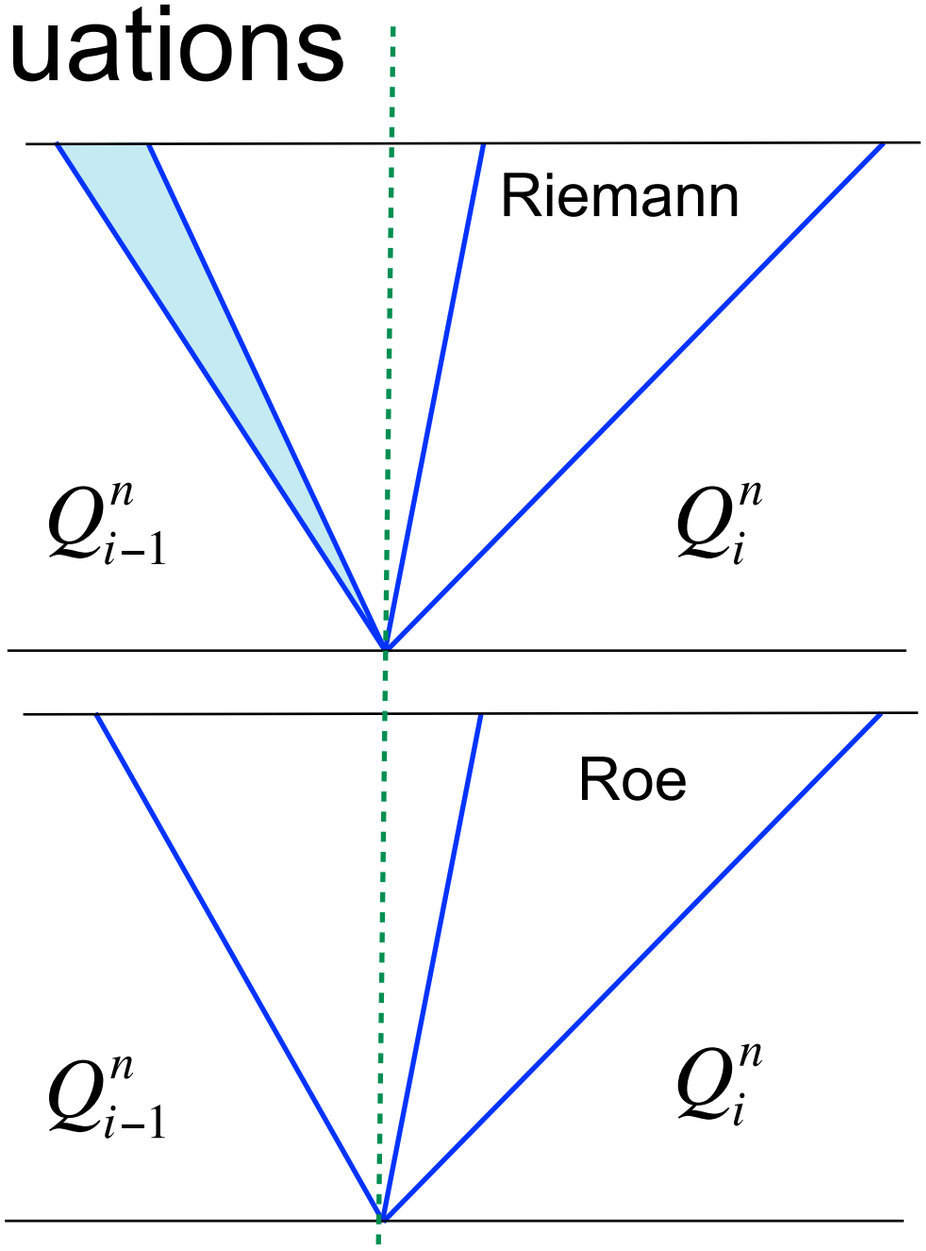
$$\delta \equiv Q_i - Q_{i-1} = \alpha^1 \hat{r}^1 + \alpha^2 \hat{r}^2 + \alpha^3 \hat{r}^3$$

where

$$\alpha^2 = (\gamma - 1)\frac{\left(\hat{H} - \hat{u}^2\right)\delta^1 + \hat{u}\delta^2 - \delta^3}{\hat{c}^2}$$

$$\alpha^3 = \frac{\delta^2 + (\hat{c} - \hat{u})\delta^1 - \hat{c}\alpha^2}{2\hat{c}}$$

$$\alpha^1 = \delta^1 - \alpha^2 - \alpha^3$$



But note that, while the Riemann solution consists of three waves, one of which is a rarefaction fan, the Roe solution only consists of three waves. In most cases this does not matter, since the desired solution at $x/t=0$ will be the same intermediate state. In the case of a transonic rarefaction a modification (in the form of an entropy fix) is necessary.

Entropy fix for transonic rarefactions

Suppose there is a transonic rarefaction in the k wave:

$$\lambda_l^k < 0 < \lambda_r^k, \quad q_l^k = Q_{i-1} + \sum_{p=1}^{k-1} \mathcal{W}^p, \quad q_r^k = q_l^k + \mathcal{W}^k$$

The method proposed by Harten and Hyman, modified slightly by Leveque, and implemented in Clawpack, is the following. Define

$$\beta = \frac{\lambda_k^r - \hat{\lambda}_k}{\lambda_k^r - \lambda_k^l}$$

where $\hat{\lambda}_k$ is the Roe-averaged eigenvalue for this wave. Then in computing the fluctuations

$$\mathcal{A}^- \Delta Q_{i-1/2} = \sum_p \left(s_{i-1/2}^p\right)^- \mathcal{W}_{i-1/2}^p, \quad \mathcal{A}^+ \Delta Q_{i-1/2} = \sum_p \left(s_{i-1/2}^p\right)^+ \mathcal{W}_{i-1/2}^p$$

for the speed of the k wave use

$$\left(\hat{\lambda}^k\right)^- = \beta \lambda_l^k, \quad \left(\hat{\lambda}^k\right)^+ = (1-\beta)\lambda_r^k$$

Ch 16: Nonclassical hyperbolic problems

Chapter 16 covers some situations that may be of interest.

nonconvex flux functions, e.g. oil-reservoir simulations

flow in porous media

nearly singular equations

phase changes, van der Waals gases

nonconservative transport

This chapter is rich in possibilities for projects, so is worth skimming.

Source Terms
(Chapter 17 in Leveque)

Source terms

Many of the situations we will want to study, especially in geophysics, are conservation laws with source terms:

$$q_t + f(q)_x = \psi(q)$$

Of course this equation arises from the more fundamental integral form:

$$\frac{\partial}{\partial t}\int_{x_1}^{x_2} q(x,t)\,dx = f\Big(q\big(x_1,t\big)\Big) - f\Big(q\big(x_2,t\big)\Big) + \int_{x_1}^{x_2} \psi\Big(q\big(x,t\big)\Big)dx$$

Examples:

- external forces, for example gravity
- reacting flow (combustion, dissolution, exsolution)
- conductive or radiative heat transfer
- drag, viscosity
- varying depth in shallow-water equations
- varying pipe shape in Euler equations
- systems with symmetries (geometrical source terms, see section 18.9)

A given system may have more than one of these sources!

Fractional-Step Methods

In the system $q_t + f(q)_x = \psi(q)$:

If the homogeneous part $q_t + f(q)_x = 0$ is hyperbolic, and the source term part can be expressed as $q_t = \psi\left(q(x,t),x\right)$ (that is, without derivatives of q), then it is possible to alternate between solving the homogeneous hyperbolic equation and the source term equation.

Using the example of an advection-reaction equation, Leveque shows in some detail how fractional-step and “unsplit” methods relate to one another.

Fractional step methods are easier to implement in code, they are generally faster than unsplit methods, they do nearly as well, and are readily extended to high resolution.

However, there is a *splitting error* one must be aware of.

Fractional Step Methods

The system $q_t + f(q)_x = \psi(q)$ is split into two parts:

A-part: Use a high-resolution method to solve $q_t + f(q)_x = 0$

B-part: Use a high-order method to solve $q_t = \psi\left(q\right)$

There are two popular ways of doing this:

Godunov splitting: (first-order accurate at best) `method(5)=1`

full time step Δt on the A-part

full time step Δt on the B-part

Strang splitting: (second-order accurate at best) `method(5)=2`

time step $\Delta t / 2$ on the B-part

time step Δt on the A-part

time step $\Delta t / 2$ on the B-part

Although the order of accuracy suffers, Godunov splitting is often preferred because it is usually faster, depending on the complexity of the source term.

Fractional Step Methods

The system $q_t + f(q)_x = \psi(q)$ is split into two parts:

A-part: Use a high-resolution method to solve $q_t + f(q)_x = 0$

B-part: Use a high-order method to solve $q_t = \psi(q)$

There are two popular ways of doing this:

Godunov splitting: (first-order accurate at best)

full time step Δt on the A-part

full time step Δt on the B-part

Strang splitting: (second-order accurate at best)

time step $\Delta t / 2$ on the B-part

time step Δt on the A-part

time step $\Delta t / 2$ on the B-part

in Clawpack

`method(5)=1`

`method(5)=2`

Although the order of accuracy suffers, Godunov splitting is often preferred because it is usually faster, depending on the complexity of the source term.

Operator splitting for a general PDE: how the splitting error arises

A general form for a linear partial differential equation is

$$q_t = \left(\mathcal{A} + \mathcal{B}\right) q.$$

(For a conservation law in quasilinear form, $\mathcal{A} = f'(q)\partial_x$; $\mathcal{B} = \psi(\,)$.)

Further time derivatives can be computed by

$$q_{tt} = \left(\mathcal{A} + \mathcal{B}\right) q_t = \left(\mathcal{A} + \mathcal{B}\right)^2 q,$$

and we can form the Taylor series expansion

$$q\left(x, \Delta t\right) = q\left(x, 0\right) + \Delta t \left(\mathcal{A} + \mathcal{B}\right) q\left(x, 0\right) + \tfrac{1}{2} \Delta t^2 \left(\mathcal{A} + \mathcal{B}\right)^2 q\left(x, 0\right) + \cdots$$

which we can write in short-hand (*solution operator form*) as

$$q\left(x, \Delta t\right) = e^{\Delta t \left(\mathcal{A} + \mathcal{B}\right)} q\left(x, 0\right).$$

Using *operator splitting* we compute

$$\widehat{q}\left(x, \Delta t\right) = e^{\Delta t \mathcal{B}} e^{\Delta t \mathcal{A}} q\left(x, 0\right).$$

The difference between these two, in second order, is

$$q\left(x, \Delta t\right) - \widehat{q}\left(x, \Delta t\right) = \tfrac{1}{2} \Delta t^2 \left(\mathcal{A}\mathcal{B} - \mathcal{B}\mathcal{A}\right) q\left(x, 0\right) + \mathcal{O}\left(\Delta t^3\right)$$

which is zero *only when the differential operators commute*.

Multidimensional problems can also use a kind of operator splitting

The two-dimensional conservation law $q_t + f(q)_x + g(q)_y = 0$

is similar to the one-dimensional conservation law with a source term, and can be solved in a similar way. In Clawpack (see $CLAW/2d/lib/claw2.f):

`method(3)=-1` dimensional splitting, Godunov style
`method(3)=-2` dimensional splitting, Strang style

This is the easiest way to extend good 1-D methods to 2-D and 3-D, and it is usually effective and efficient but it misses corner transport (more later).

Unsplit methods for multidemsions are also available in Clawpack:

`method(3)= 0` no transverse propagation, only normal waves
`method(3)=+1` transverse propagation without correction waves
`method(3)=+2` transverse propagation with correction waves

The dimensionally split Godunov method

The two steps of a dimensionally-split Godunov method are:

$$Q_{ij}^{*} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j} \right)$$

$$Q_{ij}^{n+1} = Q_{ij}^{*} - \frac{\Delta t}{\Delta y}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2}^{*} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}^{*} \right)$$

For the linear system, $q_t + Aq_x + Bq_y = 0$, the fluctuations are

$$\mathcal{A}^{\pm}\Delta Q_{i-1/2,j} = A^{\pm}\left(Q_{ij} - Q_{i-1,j} \right)$$

$$\mathcal{B}^{\pm}\Delta Q_{i-1/2,j} = B^{\pm}\left(Q_{ij} - Q_{i,j-1} \right)$$

where $A^{\pm} = R^{x}\left(\Lambda^{x}\right)^{\pm}\left(R^{x}\right)^{-1}$ and $B^{\pm} = R^{y}\left(\Lambda^{y}\right)^{\pm}\left(R^{y}\right)^{-1}$.

Multidimensional Hyperbolic Problems (Chapter 18 in Leveque)

The world has more than one dimension!

A conservation law in two dimensions:

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

and in three dimensions:

$$q_t + f(q)_x + g(q)_y + h(q)_z = 0$$

where $f(q)$, $g(q)$, and $h(q)$ are the fluxes in the x, y, and z directions respectively.

More generally we write:

$$q_t + \vec{\nabla} \cdot \vec{f}(q) = 0$$

where $\vec{f}(q)$ is a vector function representing the flux of q, and the divergence operator is defined as

$$\text{for } \vec{f} = \left[f, g, h \right], \quad \vec{\nabla} \cdot \vec{f} = \frac{\partial f}{\partial x} + \frac{\partial g}{\partial y} + \frac{\partial h}{\partial z} = f_x + g_y + h_z$$

A word about notation

In general we will refer to spatial vectors and their components as:

position $\vec{x} = [x, y, z]$

velocity $\vec{u} = \left[u(x,y,z,t), v(x,y,z,t), w(x,y,z,t)\right]$

flux $\vec{f}(q) = [f(q), g(q), h(q)]$

vector of matrices $\vec{A} = [A, B, C]$

unit normal vector $\vec{n} = \left[n^x, n^y, n^z\right]$

Hyperbolicity

The linear system in one dimension, $q_t + Aq_x = 0$, is hyperbolic if the matrix A is diagonalisable, with real eigenvalues.

In two dimensions, we need a stronger condition. For the system

$$q_t + Aq_x + Bq_y = 0$$

to be hyperbolic, not only must the matrices A and B be diagonalisable, with real eigenvalues, but so must every projection of these matrices in all spatial directions.

Defining a unit vector $\vec{n} = (n^x, n^y)$, we define

$$\breve{A} = n^x A + n^y B$$

and require that this combination be diagonalisable, with real eigenvalues for any choice of $\vec{n}$. Problem 18.3 is an example in which each matrix is separately hyperbolic, but the combination is not! **For extra credit, do this.**

This requirement is easily extended to a three-dimensional system.

Coupling among dimensions

We cannot in general decouple a linear multidimensional system into separate advection equations as we could in the one-dimensional case.

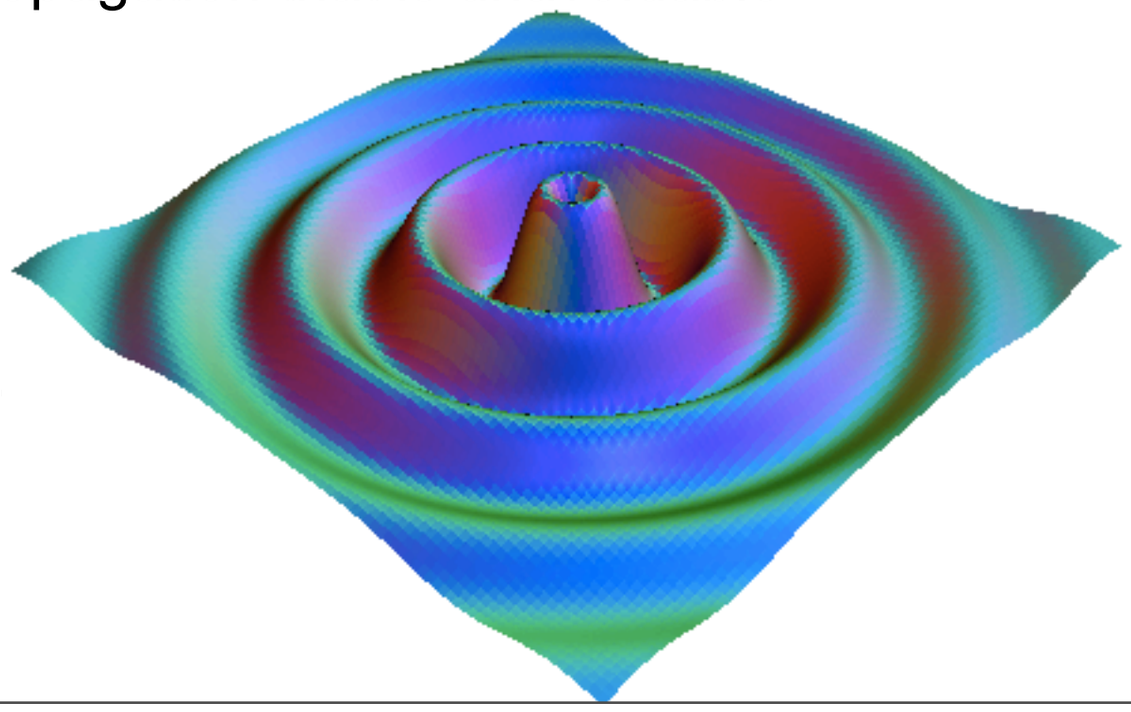
We can diagonalise each matrix separately:

$$A = R^x \Lambda^x (R^x)^{-1}, \quad B = R^y \Lambda^y (R^y)^{-1}$$

but unless $BA = AB$, performing these operations does not produce decoupling.

The point is that hyperbolic equations in two or more dimensions produce waves that can travel in *any* direction, not just in the x or y directions. Thus there is coupling between the x and the y propagation that is unavoidable except under special circumstances.

Dimensional splitting may not adequately deal with the coupling between dimensions. Effectively, the order of accuracy is reduced.



Compressible barotropic flow in 2 dimensions

With the velocity vector defined as $\vec{u} = \left[u, v\right]$

the equations of compressible flow in 2 dimensions are:

$$\rho_t + (\rho u)_x + (\rho v)_y = 0$$

$$\left(\rho u\right)_t + \left(\rho u^2 + p\right)_x + (\rho uv)_y = 0$$

$$\left(\rho v\right)_t + (\rho uv)_x + \left(\rho v^2 + p\right)_y = 0$$

We can write these as $q_t + f(q)_x + g(q)_y = 0$

by putting

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \end{bmatrix}, f(q) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \end{bmatrix}, g(q) = \begin{bmatrix} \rho u \\ \rho uv \\ \rho v^2 + p \end{bmatrix}$$

The Jacobians for 2-d compressible flow

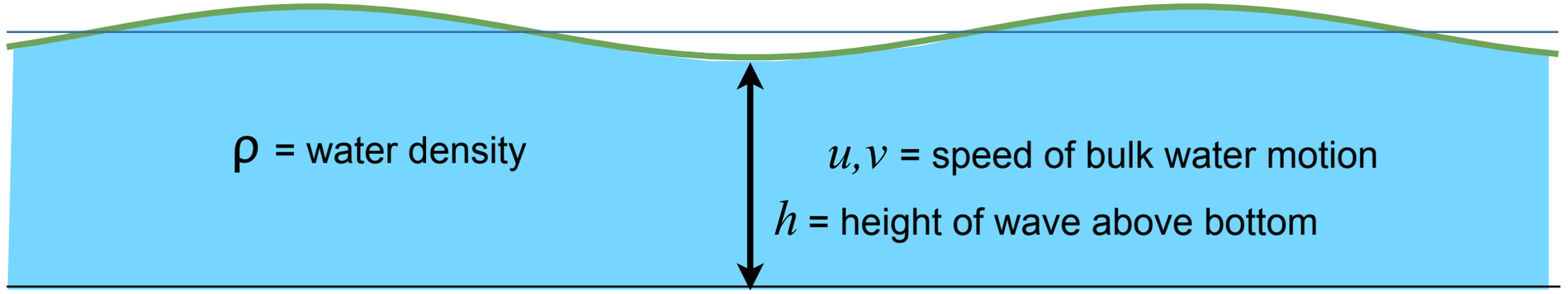
There are now two Jacobian matrices, for the quasilinear form

$$q_t + f'(q)q_x + g'(q)q_y = 0$$

with the barotropic equation of state $p = P(\rho)$ these are:

$$f'(q) = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + P'(\rho) & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad g'(q) = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + P'(\rho) & 0 & 2v \end{bmatrix}$$

Shallow-water equations in 2 dimensions



In two dimensions the shallow-water equations are:

$$h_t + (hu)_x + (hv)_y = 0$$

$$(hu)_t + \left(hu^2 + \frac{1}{2} g h^2 \right)_x + (huv)_y = 0$$

$$(hv)_t + (huv)_x + \left(hv^2 + \frac{1}{2} g h^2 \right)_y = 0$$

and the flux Jacobians are (similar to barotropic compressible flow):

$$f'(q) = \begin{bmatrix} 0 & 1 & 0 \\ -u^2 + gh & 2u & 0 \\ -uv & v & u \end{bmatrix}, \quad g'(q) = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ -v^2 + gh & 0 & 2v \end{bmatrix}$$

Eigensystem for shallow-water equations

The Jacobian matrix $f'(q)$ has these eigenvalues and eigenvectors:

$$\lambda^{x1} = u - c, \qquad \lambda^{x2} = u, \qquad \lambda^{x3} = u + c, \quad c = \sqrt{gh}$$

$$r^{x1} = \begin{bmatrix} 1 \\ u - c \\ v \end{bmatrix}, r^{x2} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, r^{x3} = \begin{bmatrix} 1 \\ u + c \\ v \end{bmatrix}$$

and the other Jacobian matrix $g'(q)$ has similar pattern, reversing u and v:

$$\lambda^{y1} = v - c, \qquad \lambda^{y2} = v, \qquad \lambda^{y3} = v + c$$

$$r^{y1} = \begin{bmatrix} 1 \\ u \\ v - c \end{bmatrix}, r^{y2} = \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix}, r^{y3} = \begin{bmatrix} 1 \\ u \\ v + c \end{bmatrix}$$

Notice the eigenvectors are different, but in general we have two nonlinear fields with the eigenvalues $\vec{n} \cdot \vec{u} \pm c$ and the linearly degenerate field $\vec{n} \cdot \vec{u}$.

The Euler equations in 3 dimensions

In 2 dimensions, we have 4 equations, and in 3 dimensions 5 equations:

$$q_t + f(q)_x + g(q)_y + h(q)_z = 0$$

$$q = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ E \end{bmatrix}, f(q) = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ (E+p)u \end{bmatrix}, g(q) = \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ (E+p)v \end{bmatrix}, h(q) = \begin{bmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ (E+p)w \end{bmatrix}$$

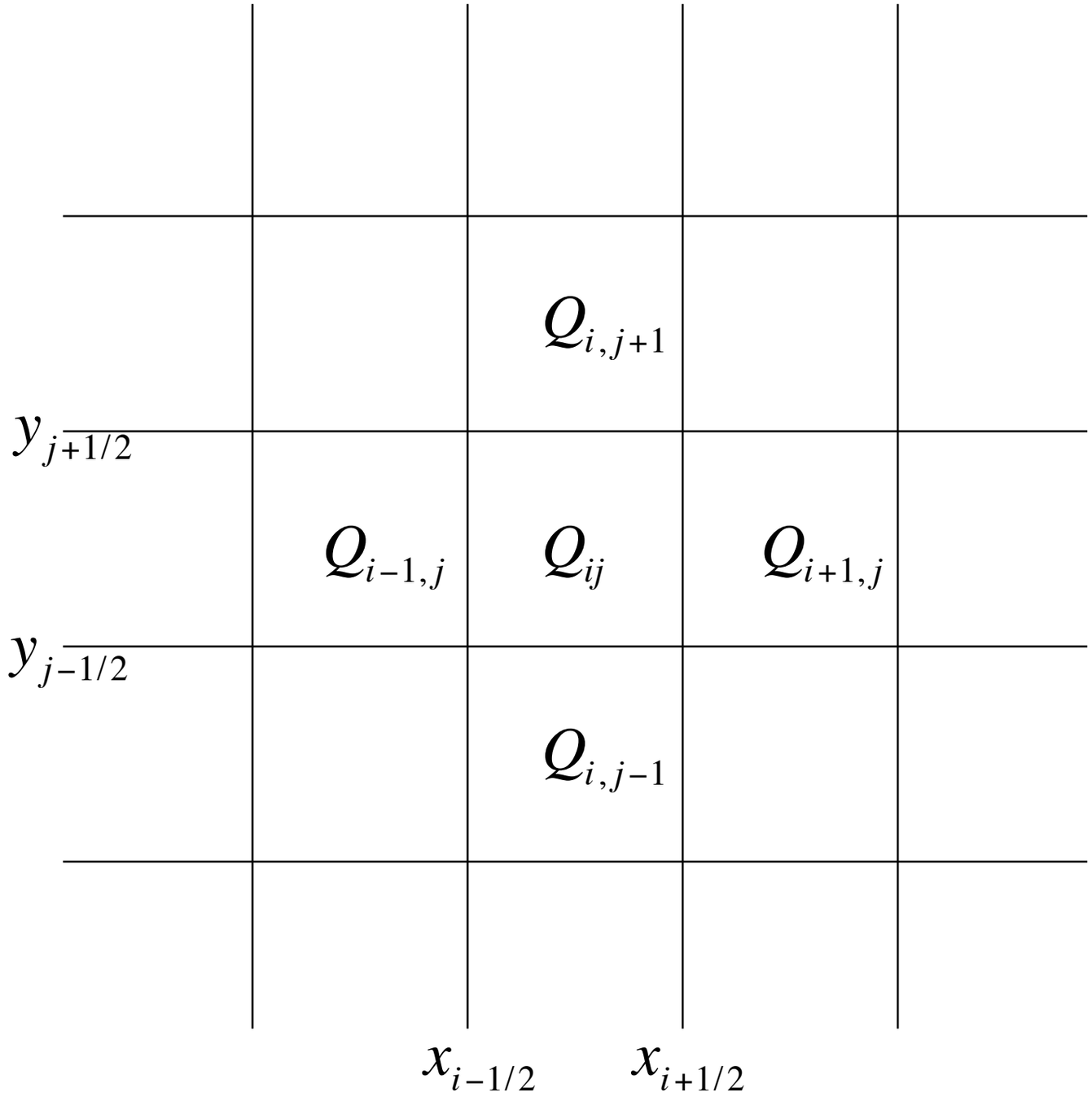
The energy is $E = \rho e + \frac{1}{2}\rho \vec{u}\cdot\vec{u} = \rho e + \frac{1}{2}\rho\left(u^2 + v^2 + w^2\right)$ and this system still needs to be supplemented with an equation of state $e = e(p,\rho)$.

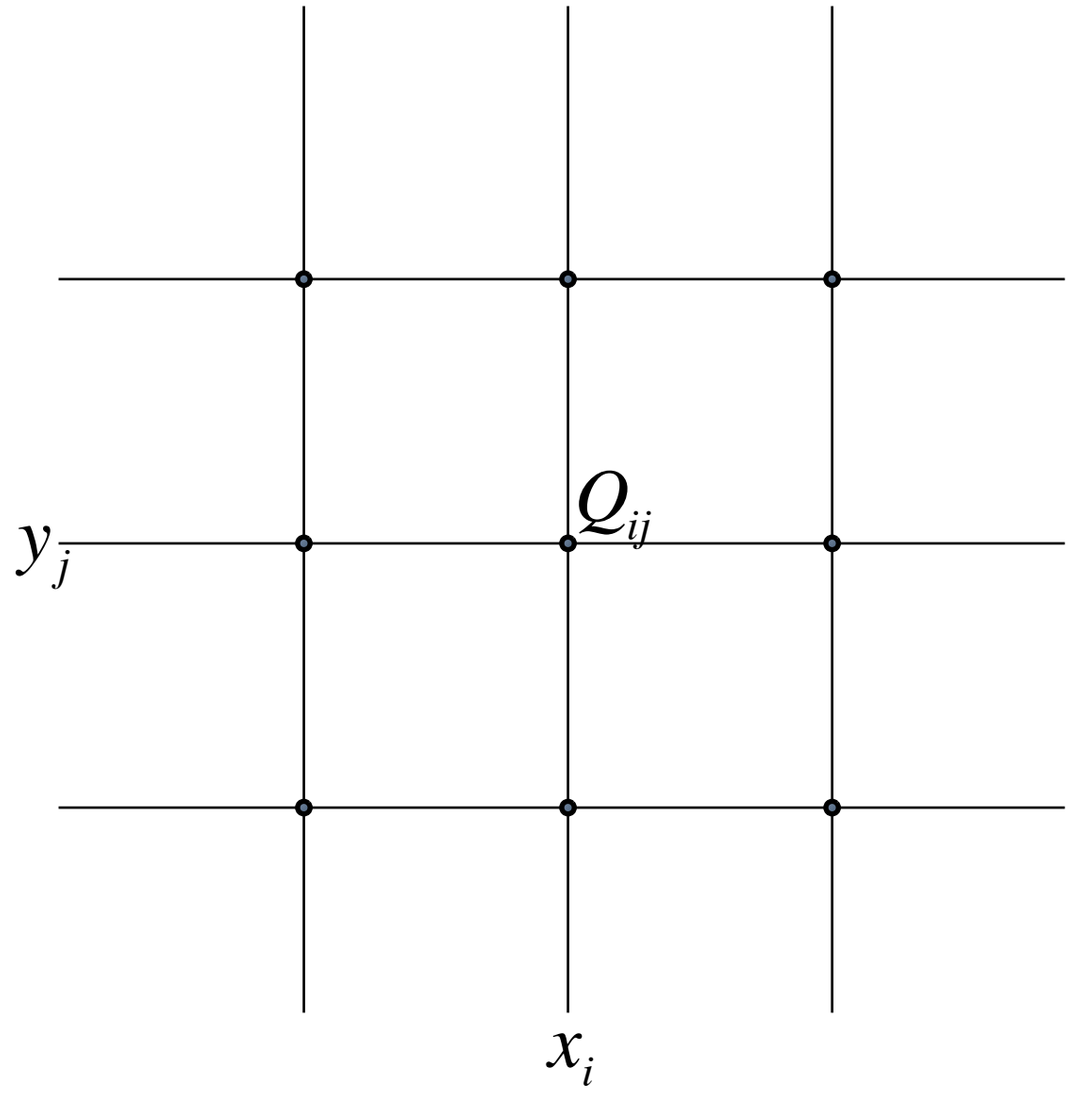
Again there will be eigenvalues of the form $\vec{n}\cdot\vec{u}$ and $\vec{n}\cdot\vec{u} \pm c$, where $c = \sqrt{\frac{dp}{d\rho}}$.

There are two nonlinear acoustic fields and one linearly degenerate field.

Multidimensional Numerical Methods
(Chapter 19 in Leveque)

Finite Volume vs Finite Difference





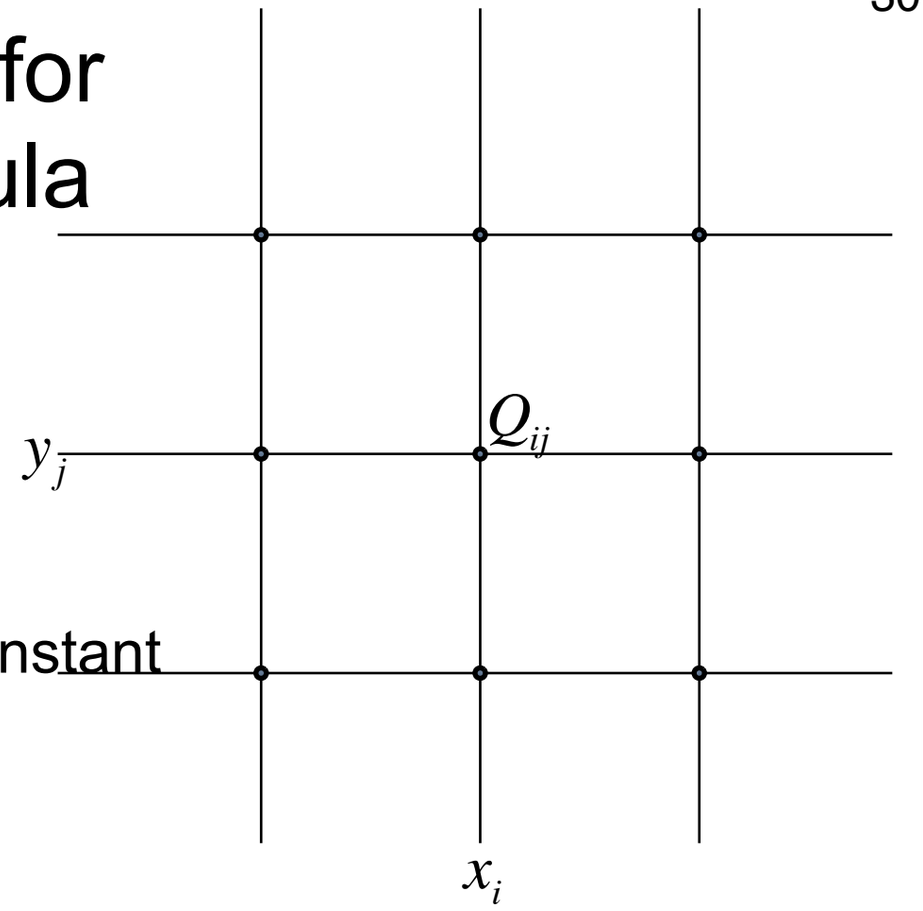
The values are considered to be averages within cells:

$$Q_{ij}^n \approx \frac{1}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} q(x,y,t_n)\,dx\,dy$$

The values are considered to be evaluated on a grid:

$$Q_{ij}^n \approx q(x_i, y_j, t_n)$$

The finite-difference form is useful for developing the Lax-Wendroff formula via a Taylor-series expansion



We consider first the linear system, with A and B constant (though noncommutative) matrices:

$$q_t + Aq_x + Bq_y = 0$$

We make a Taylor-series expansion of q at the point (x_i, y_j) at time $t_n + \Delta t$ in terms of its value at time t_n:

$$q(x_i, y_j, t_n + \Delta t) = q + \Delta t q_t + \frac{1}{2}\Delta t^2 q_{tt} + \ldots$$

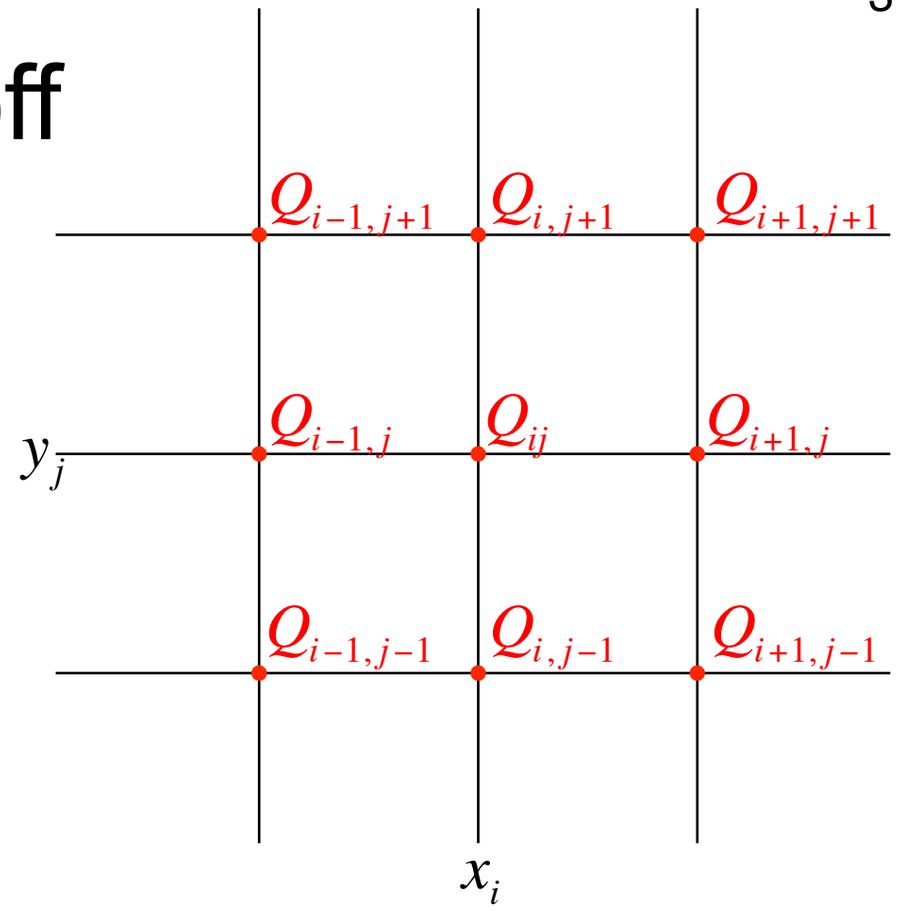
$$= q - \Delta t(Aq_x + Bq_y) + \frac{1}{2}\Delta t^2 (A^2 q_{xx} + ABq_{yx} + BAq_{xy} + B^2 q_{yy}) + \ldots$$

Two-dimensional Lax-Wendroff

The pure double derivatives are approximated as in one dimension:

$$q_{xx} \approx \left(\frac{1}{\Delta x}\right)^2 \left(Q^n_{i-1,j} - 2Q^n_{i,j} + Q^n_{i+1,j}\right)$$

$$q_{yy} \approx \left(\frac{1}{\Delta y}\right)^2 \left(Q^n_{i,j-1} - 2Q^n_{i,j} + Q^n_{i,j+1}\right)$$



and there are also cross-double-derivative terms:

$$q_{xy} = q_{yx} \approx \frac{1}{4\Delta x \Delta y}\left[\left(Q^n_{i+1,j+1} - Q^n_{i-1,j+1}\right) - \left(Q^n_{i+1,j-1} - Q^n_{i-1,j-1}\right)\right]$$

giving the method:

$$Q^{n+1}_{ij} = Q^n_{ij} - \frac{\Delta t}{2\Delta x} A\left(Q^n_{i+1,j} - Q^n_{i-1,j}\right) - \frac{\Delta t}{2\Delta y} B\left(Q^n_{i,j+1} - Q^n_{i,j-1}\right)$$

$$+ \frac{\Delta t^2}{2\Delta x^2} A^2\left(Q^n_{i+1,j} - 2Q^n_{ij} + Q^n_{i-1,j}\right) + \frac{\Delta t^2}{2\Delta y^2} B^2\left(Q^n_{i,j+1} - 2Q^n_{ij} + Q^n_{i,j-1}\right)$$

$$+ \frac{\Delta t^2}{8\Delta x \Delta y}(AB + BA)\left[\left(Q^n_{i+1,j+1} - Q^n_{i-1,j+1}\right) - \left(Q^n_{i+1,j-1} - Q^n_{i-1,j-1}\right)\right]$$

Again, Lax-Wendroff is a good starting point, but we need to do better…

In a *finite volume* approach, using upwind biasing and flux limiting we can achieve second-order accuracy and high-resolution in multi-dimensions as we did in one dimension.

Note: the Riemann problem at a cell edge is essentially one-dimensional; we can do the problem as before, and bring in multidimensional information to improve the solution.

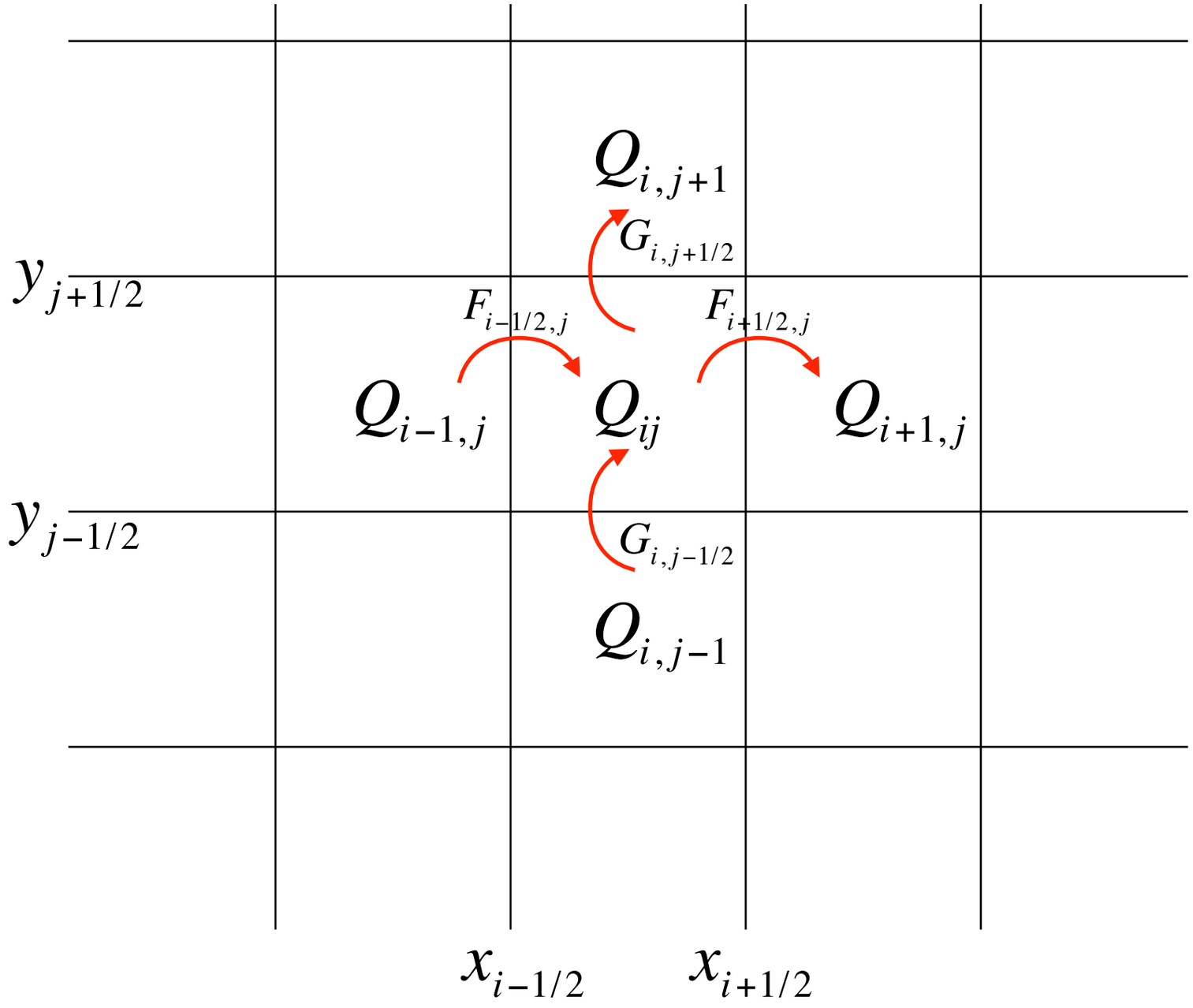
You can

use a full flux-differencing (dimensionally unsplit) approach (better) or

split the problem into a sequence of one-dimensional problems (easier, but you have to take care of the corners)

For conservation laws, finite volume methods are natural

We update the conserved quantity by keeping track of the fluxes into and out of each cell (and sources and sinks, as relevant)



The fully discrete flux differencing method to update Q for the next time step is:

$$Q_{ij}^{n+1} \approx Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(F_{i+1/2,j}^{n} - F_{i-1/2,j}^{n}\right) - \frac{\Delta t}{\Delta y}\left(G_{i,j+1/2}^{n} - G_{i,j-1/2}^{n}\right)$$

The fluxes are found by integrating along the edges

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

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

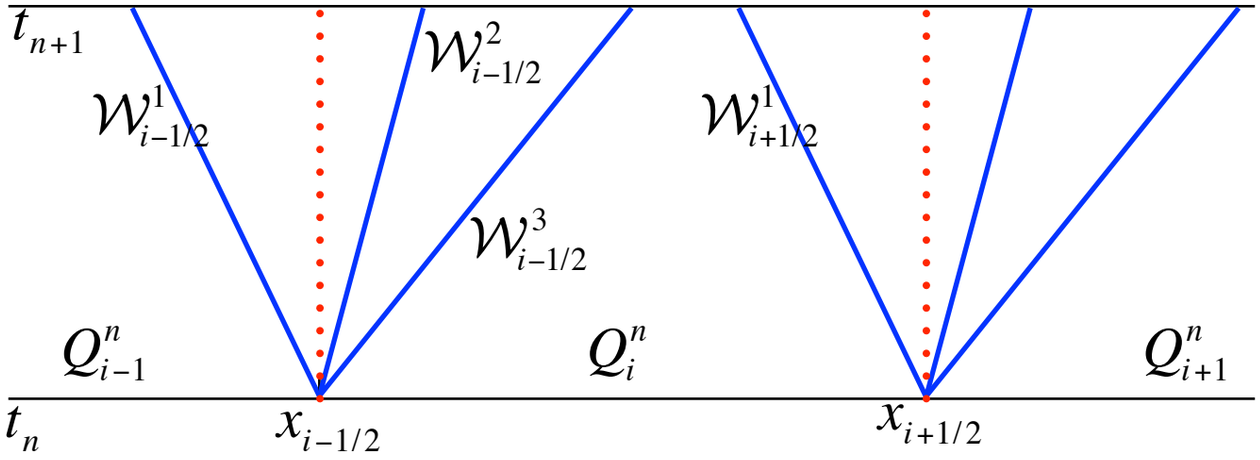
With a Taylor expansion we approximate these, to second-order in time, as:

$$F_{i-1/2,j} \approx Aq(x_{i-1/2}, y_j, t_n) - \frac{\Delta t}{2} A^2 q_x(x_{i-1/2}, y_j, t_n) - \frac{\Delta t}{2} ABq_y(x_{i-1/2}, y_j, t_n)$$

$$G_{i,j-1/2} \approx Bq(x_i, y_{j-1/2}, t_n) - \frac{\Delta t}{2} B^2 q_y(x_i, y_{j-1/2}, t_n) - \frac{\Delta t}{2} BAq_x(x_i, y_{j-1/2}, t_n)$$

These fluxes can be used to re-interpret the Lax-Wendroff formula as a finite volume method.

The Godunov method is simpler



Here we simply find the Riemann solution that propagates with zero speed (straight up the time axis) and evaluate the flux functions at these values:

$$F_{i-1/2,j} = f(Q^{\downarrow}_{i-1/2,j})$$

$$G_{i,j-1/2} = f(Q^{\downarrow}_{i,j-1/2})$$

$Q^{\downarrow}_{i-1/2,j}$ is obtained by solving the Riemann problem for $q_t + f(q)_x = 0$ and

$Q^{\downarrow}_{i,j-1/2}$ is obtained by solving the Riemann problem for $q_t + g(q)_y = 0$.

We can adopt the fluctuation form

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}\right)$$

$$- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right)$$

Here the high-resolution corrections in the second line can be omitted for the pure Godunov method, or they can be included with appropriate flux limiters for high-resolution techniques.

Dimensional Splitting - Godunov Splitting

We can split a multidimensional problem into a sequence of one-dimensional steps, for example for the two-dimensional linear problem:

$$q_t + Aq_x + Bq_y = 0$$

x sweeps: $q_t + Aq_x = 0$

y sweeps: $q_t + Bq_y = 0$

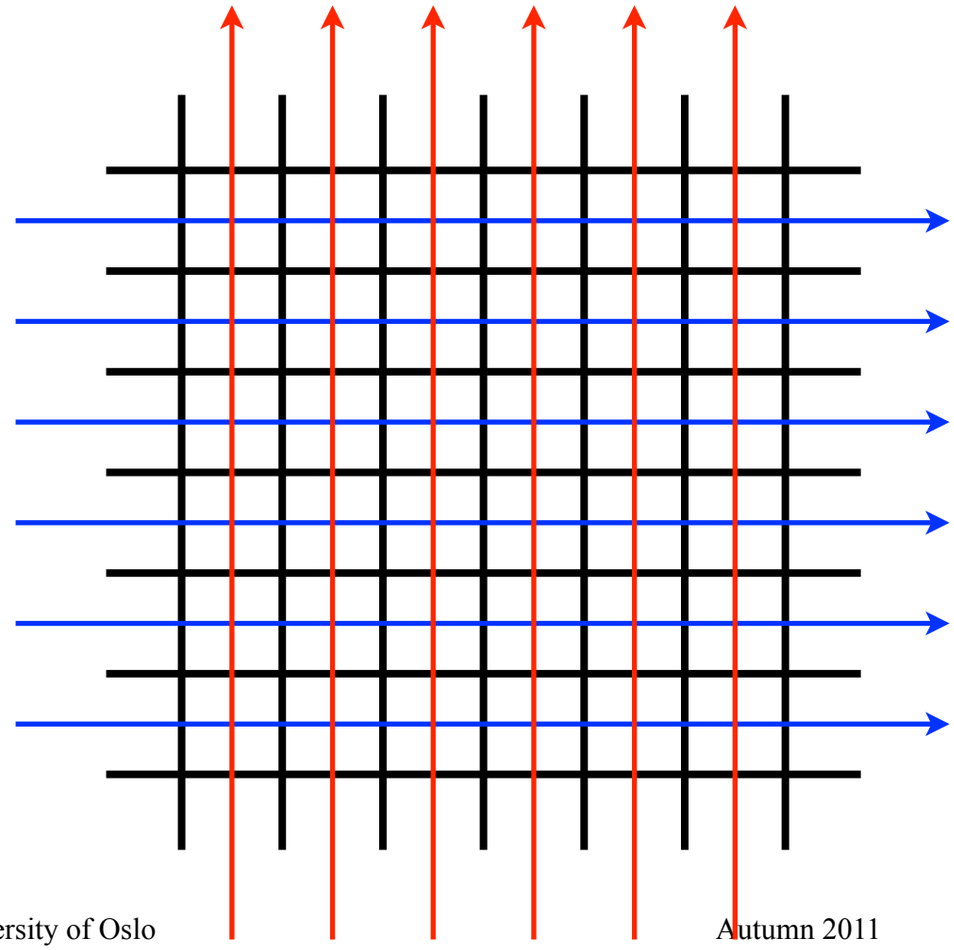
In the x sweeps, we march along in i, keeping j fixed, and update

$$Q_{ij}^{*} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(F_{i+1/2,j}^{n} - F_{i,j-1/2}^{n} \right)$$

In the y sweeps, we march along in j, keeping i fixed, and update

$$Q_{ij}^{n+1} = Q_{ij}^{*} - \frac{\Delta t}{\Delta y}\left(G_{i,j+1/2}^{*} - G_{i,j-1/2}^{*} \right)$$

Alternating the order of the sweeps every time step is equivalent to Strang splitting



Multidimensional Scalar Equations
(Chapter 20 in Leveque)

High-resolution methods for scalar hyperbolic equations in two dimensions

Start once again with the constant-velocity advection equation, this time with a velocity $\vec{u} = \left[u, v \right]$ with both x and y components. We consider both positive for now.

$$q_t + uq_x + vq_y = 0$$

Then of course the solution is the unchanged original function, simply translated with time at the velocity $\vec{u}$. But now it's moving at an angle.

A Taylor series expansion gives us

$$q(x,y,t_{n+1}) = q(x,y,t_n) - u\Delta t q_x - v\Delta t q_y$$

$$+ \frac{1}{2}(\Delta t)^2 \left[u^2 q_{xx} + vuq_{xy} + uvq_{yx} + v^2 q_{yy} \right] + \ldots$$

Donor-cell upwind method

The donor-cell upwind method is simply

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

$$- \frac{\Delta t}{\Delta y}\left[v^{+}\left(Q_{i,j}^{n} - Q_{i,j-1}^{n}\right) + v^{-}\left(Q_{i,j+1}^{n} - Q_{i,j}^{n}\right)\right]$$

with $u^{+} = \max(u,0),\ \ u^{-} = \min(u,0)$ as before. This can be written in the fluctuation form

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}\right)$$

$$- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right)$$

with

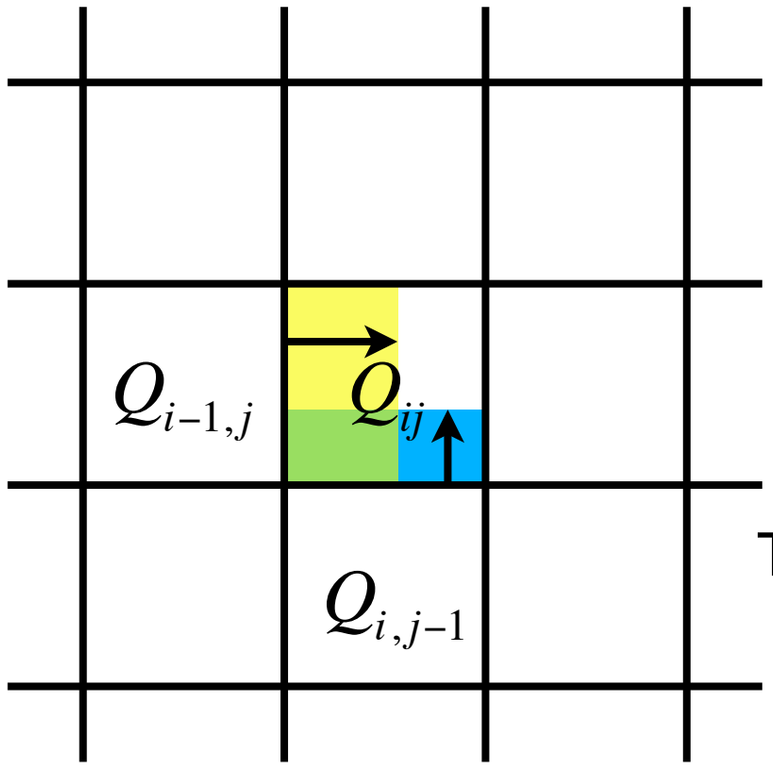
$$\mathcal{A}^{+}\Delta Q_{i-1/2,j} = u^{\pm}\left(Q_{ij} - Q_{i-1,j}\right), \quad \mathcal{B}^{+}\Delta Q_{i,j-1/2} = v^{\pm}\left(Q_{i,j} - Q_{i,j-1}\right)$$

and without the corrective limited fluxes, i.e. $\tilde{F} = \tilde{G} = 0$

The naive donor-cell upwind method misses the contribution from the corner

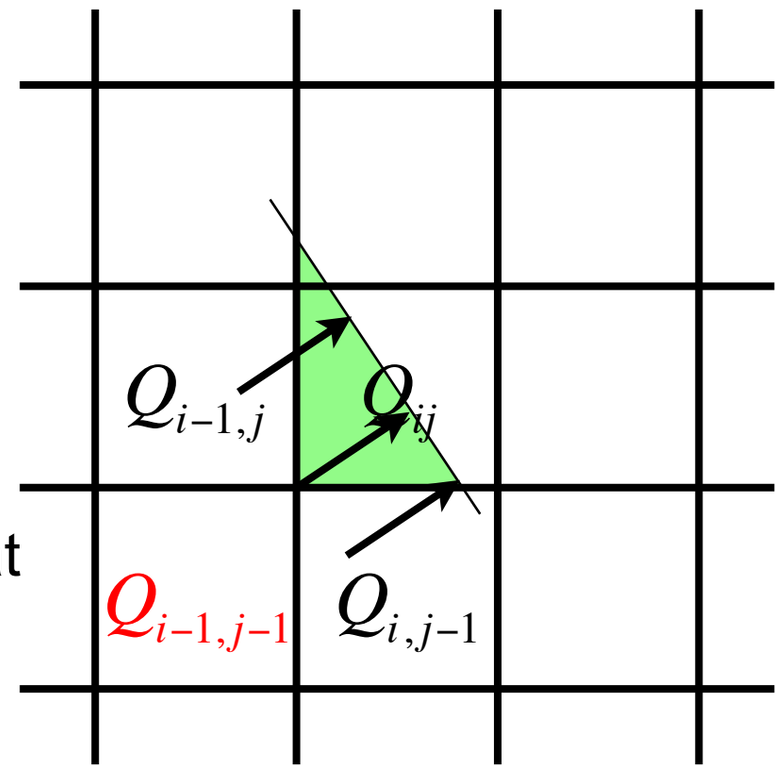
$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left[u^{+}\left(Q_{i,j}^{n} - Q_{i-1,j}^{n}\right) + u^{-}\left(Q_{i+1,j}^{n} - Q_{i,j}^{n}\right)\right]$$

$$- \frac{\Delta t}{\Delta y}\left[v^{+}\left(Q_{i,j}^{n} - Q_{i,j-1}^{n}\right) + v^{-}\left(Q_{i,j+1}^{n} - Q_{i,j}^{n}\right)\right]$$



$\vec{u}$

The true velocity is at an angle to the grid.



This method will have stability problems. Dimensional splitting (doing an intermediate update with a sweep in one direction, followed by a sweep in the other direction) may help.

A better approach is to use the REA technique

The Godunov REA scheme can be easily extended to two (or three) dimensions:

1. Reconstruct a piece-wise linear function from the cell averages.

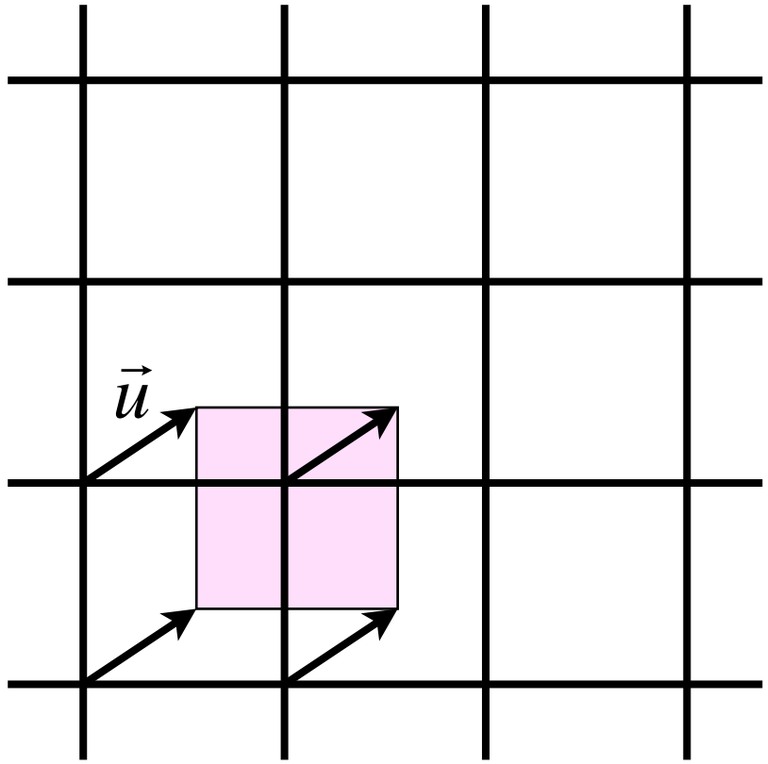
$$q^n(x,y,t_n) = Q_{ij}^n \text{ for } x,y \text{ in cell } i,j$$

2. Evolve the advection equation exactly with these data to obtain a later-time function.

$$\tilde{q}^n(x,y,t_{n+1})$$

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

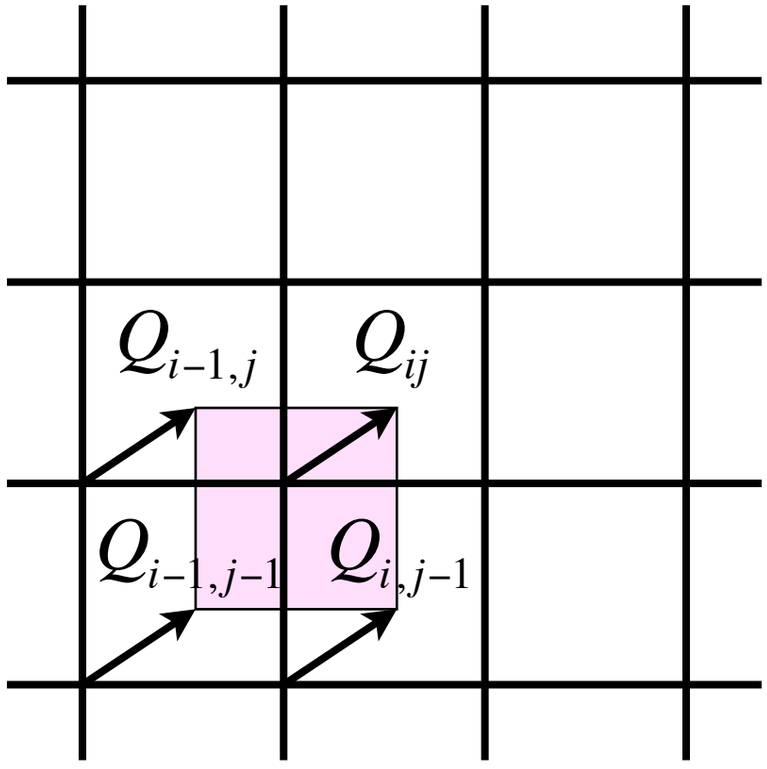
$$Q_{ij}^{n+1} = \frac{1}{\Delta x \Delta y} \int_{y_{j-1/2}}^{y_{j+1/2}} \int_{x_{i-1/2}}^{x_{i+1/2}} \tilde{q}^n(x,t_{n+1})\,dx\,dy$$



Colella's corner-transport upwind method

Following the REA procedure results in:

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

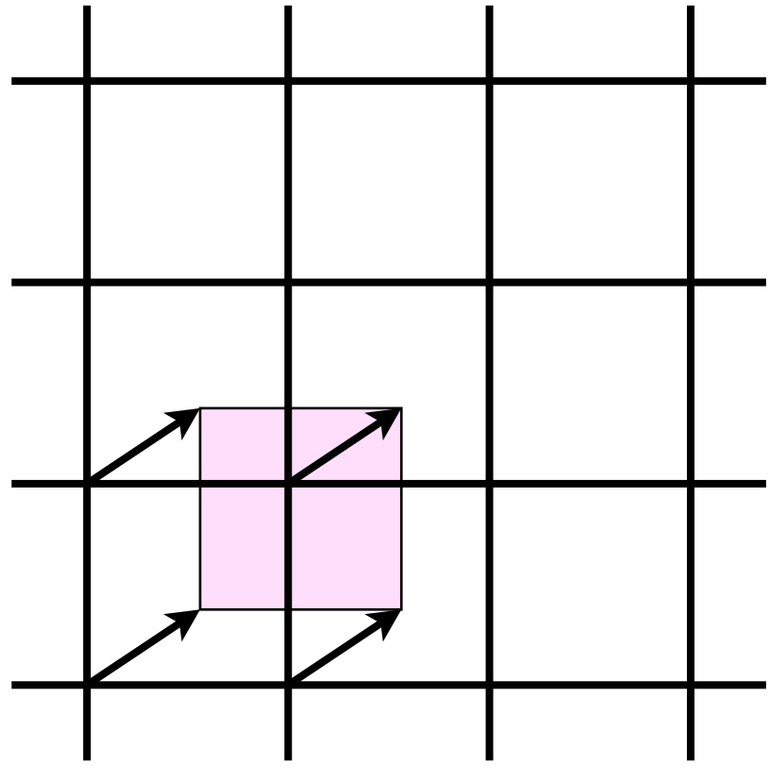


This expression is rearranged and simplified from Leveque's Eqn 20.10

This is still just first-order accurate, but can be improved by introducing limited slopes, just as we did for 1-dimension.

Let's look at it in still a different way.

Wave-propagation implementation of CTU

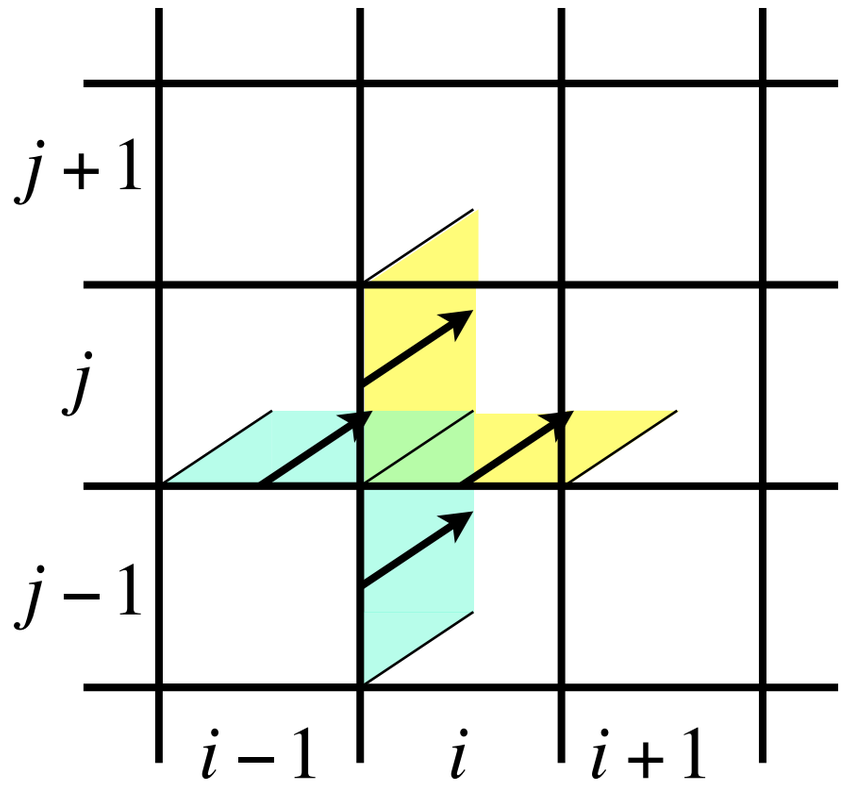


In a sense, we turn CTU inside out.

Consider transport from the edges (faces), not the corners.

From the interface between $(i-1, j)$ and (i, j) a wave goes into both (i, j) and $(i, j+1)$: and four distinct waves affect (i, j).

You can work out how much each wave contributes to each cell from simple geometry. Just sum up the areas of the little triangles and squares!



This interpretation leads to the fluctuation format

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}\right)$$

$$- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right)$$

with the fluctuations defined as for Lax-Wendroff:

$$\mathcal{A}^{\pm}\Delta Q_{i-1/2,j} = u^{\pm}\left(Q_{ij} - Q_{i-1,j}\right), \quad \mathcal{B}^{\pm}\Delta Q_{i,j-1/2} = v^{\pm}\left(Q_{i,j} - Q_{i,j-1}\right)$$

but now with the correction fluxes:

$$\tilde{F}_{i-1/2,j} = -\frac{1}{2}\frac{\Delta t}{\Delta y}uv(Q_{i-1,j} - Q_{i-1,j-1}),$$

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

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

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

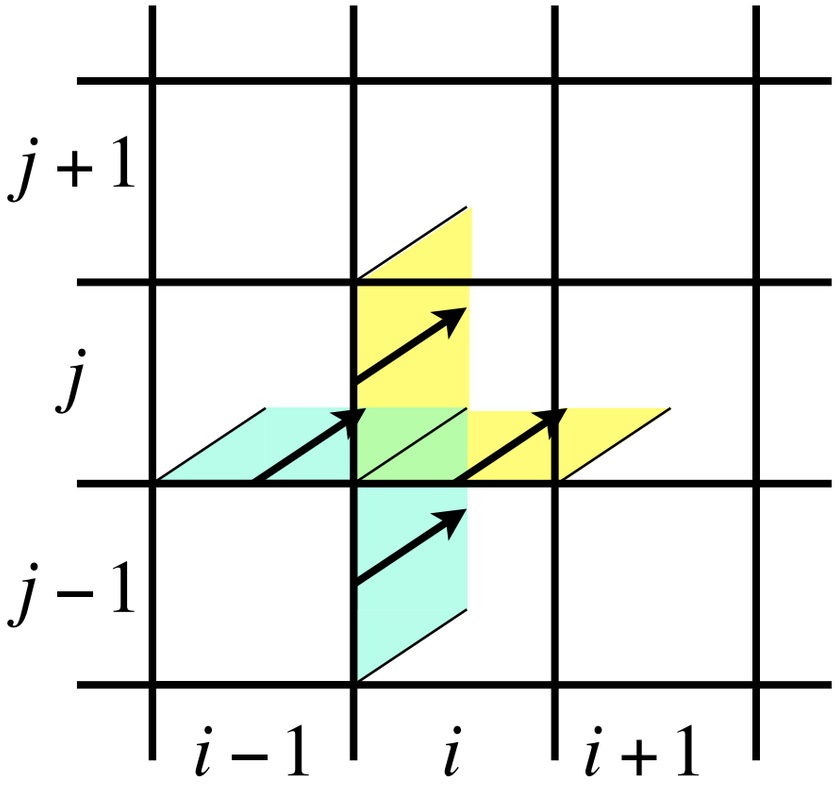
Generalise to variable-velocity advection

The fluctuations generalise easily, using the cell-interface velocities:

$$\mathcal{A}^{\pm}\Delta Q_{i-1/2,j} = u^{\pm}_{i-1/2,j}\left(Q_{ij} - Q_{i-1,j}\right), \quad \mathcal{B}^{\pm}\Delta Q_{i,j-1/2} = v^{\pm}_{i,j-1/2}\left(Q_{i,j} - Q_{i,j-1}\right)$$

But now the fluxes must be rewritten for the transverse velocities, taking into account that the velocities may be either negative or positive. The correction fluxes are set to zero to start each time step, then built up from consideration of the signs of the velocities. There are four possibilities in each direction.

First set $\tilde{F}_{i-1/2,j} := 0$ and $\tilde{G}_{i,j-1/2} := 0 \qquad \forall i,j$



Then do an x-sweep to update

$$\tilde{G}_{i-1,j-1/2} := \tilde{G}_{i-1,j-1/2} - \frac{1}{2}\frac{\Delta t}{\Delta x} u^{-}_{i-1/2,j} v^{-}_{i-1,j-1/2}(Q_{ij} - Q_{i-1,j}),$$

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

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

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

Transverse fluxes are also needed

A transverse Riemann solver takes the right- and left-going fluctuations

$$\mathcal{A}^{\pm}\Delta Q_{i-1/2,j} = u^{\pm}_{i-1/2,j}(Q_{ij} - Q_{i-1,j})$$

and makes the four transverse fluctuations

$$\mathcal{B}^{-}\mathcal{A}^{+}\Delta Q_{i-1/2,j} = v^{-}_{i,j-1/2}u^{+}_{i-1/2,j}\left(Q_{ij} - Q_{i-1,j}\right)$$

$$\mathcal{B}^{+}\mathcal{A}^{+}\Delta Q_{i-1/2,j} = v^{+}_{i,j+1/2}u^{+}_{i-1/2,j}\left(Q_{ij} - Q_{i,j-1}\right)$$

$$\mathcal{B}^{-}\mathcal{A}^{-}\Delta Q_{i-1/2,j} = v^{-}_{i,j-1/2}u^{-}_{i-1/2,j}\left(Q_{ij} - Q_{i-1,j}\right)$$

$$\mathcal{B}^{+}\mathcal{A}^{-}\Delta Q_{i-1/2,j} = v^{+}_{i,j+1/2}u^{-}_{i-1/2,j}\left(Q_{ij} - Q_{i,j-1}\right)$$

These are then used to update the correction fluxes $\tilde{G}$

Similarly a y-sweep to update the $\tilde{F}$ terms, also including the transverse fluxes.

Then to High-Resolution

For the extension to high-resolution, all that is needed now is to add additional correction terms to $\tilde{F}$ and $\tilde{G}$:

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

$$\tilde{G}_{i,j-1/2} := \tilde{G}_{i,j-1/2} + \frac{1}{2}\left|v_{i,j-1/2}\right|\left(1 - \frac{\Delta t}{\Delta y}\left|v_{i,j-1/2}\right|\right)\tilde{\mathcal{W}}_{i,j-1/2}$$

Once again, $\tilde{\mathcal{W}}_{i,j-1/2}$ represents the appropriately slope-limited version of the wave.

For the advection equation the waves are simply

$$\mathcal{W}_{i-1/2,j} = Q_{ij} - Q_{i-1,j}$$

$$\mathcal{W}_{i,j-1/2} = Q_{ij} - Q_{i,j-1}$$

Then to nonlinear conservation laws:

The scheme just developed works for nonlinear conservation laws as well.
The waves are

$$\mathcal{W}_{i-1/2,j} = Q_{ij} - Q_{i-1,j}$$

$$\mathcal{W}_{i,j-1/2} = Q_{ij} - Q_{i,j-1}$$

the speeds are

$$s_{i-1/2,j} = \begin{cases} \left[f(Q_{ij}) - f(Q_{i-1,j}) \right] / (Q_{ij} - Q_{i-1,j}) & \text{if } Q_{i-1,j} \neq Q_{ij} \\ f'(Q_{ij}) & \text{if } Q_{i-1,j} = Q_{ij} \end{cases}$$

$$s_{i,j-1/2} = \begin{cases} \left[g(Q_{ij}) - g(Q_{i,j-1}) \right] / (Q_{ij} - Q_{i,j-1}) & \text{if } Q_{i,j-1} \neq Q_{ij} \\ g'(Q_{ij}) & \text{if } Q_{i,j-1} = Q_{ij} \end{cases}$$

the fluctuations are

$$\mathcal{A}^{\pm}\Delta Q_{i-1/2,j} = s^{\pm}_{i-1/2,j}\mathcal{W}_{i-1/2,j}, \quad \mathcal{B}^{\pm}\Delta Q_{i,j-1/2} = s^{\pm}_{i,j-1/2}\mathcal{W}_{i,j-1/2}$$

and the second-order correction terms are

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

$$\tilde{G}_{i,j-1/2} := \tilde{G}_{i,j-1/2} + \frac{1}{2}\left|s_{i,j-1/2}\right|\left(1 - \frac{\Delta t}{\Delta y}\left|s_{i,j-1/2}\right|\right)\tilde{\mathcal{W}}_{i,j-1/2}$$

with a similar, but slightly different recipe for the transverse fluxes…

Multidimensional Systems of Equations
(Chapter 21 in Leveque)

Constant coefficient linear system

We look first at the linear system, with A and B constant (though noncommutative) matrices:

$$q_t + Aq_x + Bq_y = 0$$

and we will adopt the fluctuation form for the update:

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}\right)$$

$$- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right)$$

so we will need the fluctuations themselves and the correction fluxes.

Wave-propagation approach

The algorithm goes as follows:

1. Initialize $\tilde{F}_{i-1/2,j} := 0$ and $\tilde{G}_{i,j-1/2} := 0 \qquad \forall i, j$

2. Sweep through in x, solving each Riemann problem. Get the waves and speeds and compute the fluctuations. For the constant-coefficient linear case, the fluctuations are

$$\mathcal{A}^{\pm}\Delta Q_{i-1/2,j} = \sum_{p=1}^{m}\left(s^{p}_{i-1/2,j}\right)^{\pm}\mathcal{W}^{p}_{i-1/2,j} = A^{\pm}\Delta Q_{i-1/2,j}$$

3. With slope-limited waves, the correction fluxes are updated:

$$\tilde{F}_{i-1/2,j} := \tilde{F}_{i-1/2,j} + \frac{1}{2}\sum_{p=1}^{m}\left|s^{p}_{i-1/2,j}\right|\left(1 - \frac{\Delta t}{\Delta x}\left|s^{p}_{i-1/2,j}\right|\right)\tilde{\mathcal{W}}^{p}_{i-1/2,j}$$

4. As in the scalar case, right-going transverse fluctuations are defined:

$$\mathcal{B}^{\pm}\mathcal{A}^{+}\Delta Q_{i-1/2,j} = B^{\pm}A^{+}(Q_{ij} - Q_{i-1,j})$$

Wave-propagation approach, continued

5. Use the fluctuations $\mathcal{B}^{\pm}\mathcal{A}^{+}\Delta Q_{i-1/2,j}$ to update the correction fluxes above and below the current cell:

$$\tilde{G}_{i,j+1/2} := \tilde{G}_{i,j+1/2} - \frac{\Delta t}{2\Delta x}\mathcal{B}^{+}\mathcal{A}^{+}\Delta Q_{i-1/2,j},\ \tilde{G}_{i,j-1/2} := \tilde{G}_{i,j-1/2} - \frac{\Delta t}{2\Delta x}\mathcal{B}^{-}\mathcal{A}^{+}\Delta Q_{i-1/2,j}$$

6. Now left-going transverse fluctuations are defined and used to update the correction fluxes above and below the *previous* cell:

$$\tilde{G}_{i-1,j+1/2} := \tilde{G}_{i-1,j+1/2} - \frac{\Delta t}{2\Delta x}\mathcal{B}^{+}\mathcal{A}^{-}\Delta Q_{i-1/2,j},\ \tilde{G}_{i-1,j-1/2} := \tilde{G}_{i-1,j-1/2} - \frac{\Delta t}{2\Delta x}\mathcal{B}^{-}\mathcal{A}^{-}\Delta Q_{i-1/2,j}$$

7. Sweep through in y, solving each Riemann problem as in steps 2-6. Get the waves and speeds and compute the fluctuations $\mathcal{B}^{\pm}\Delta Q_{i,j-1/2}$. Split these waves transversely to modify the $\tilde{F}$ correction fluxes.

8. Finally, apply the updating formula:

$$Q_{ij}^{n+1} = Q_{ij}^{n} - \frac{\Delta t}{\Delta x}\left(\mathcal{A}^{+}\Delta Q_{i-1/2,j} + \mathcal{A}^{-}\Delta Q_{i+1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\mathcal{B}^{+}\Delta Q_{i,j-1/2} + \mathcal{B}^{-}\Delta Q_{i,j+1/2}\right)$$

$$- \frac{\Delta t}{\Delta x}\left(\tilde{F}_{i+1/2,j} - \tilde{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\tilde{G}_{i,j+1/2} - \tilde{G}_{i,j-1/2}\right)$$

Så enkelt er det!

We’re basically done.

Leveque then goes on to describe how the two-dimensional algorithm is implemented in Clawpack, and illustrates examples for the acoustic equations (as a linear system) and the shallow-water equations (as a nonlinear system).

There are some examples you can run in the code you have already.

There’s just one small thing more to be concerned with, before we do our projects, and that’s some new aspects of boundary conditions that apply in multi-dimensional systems.

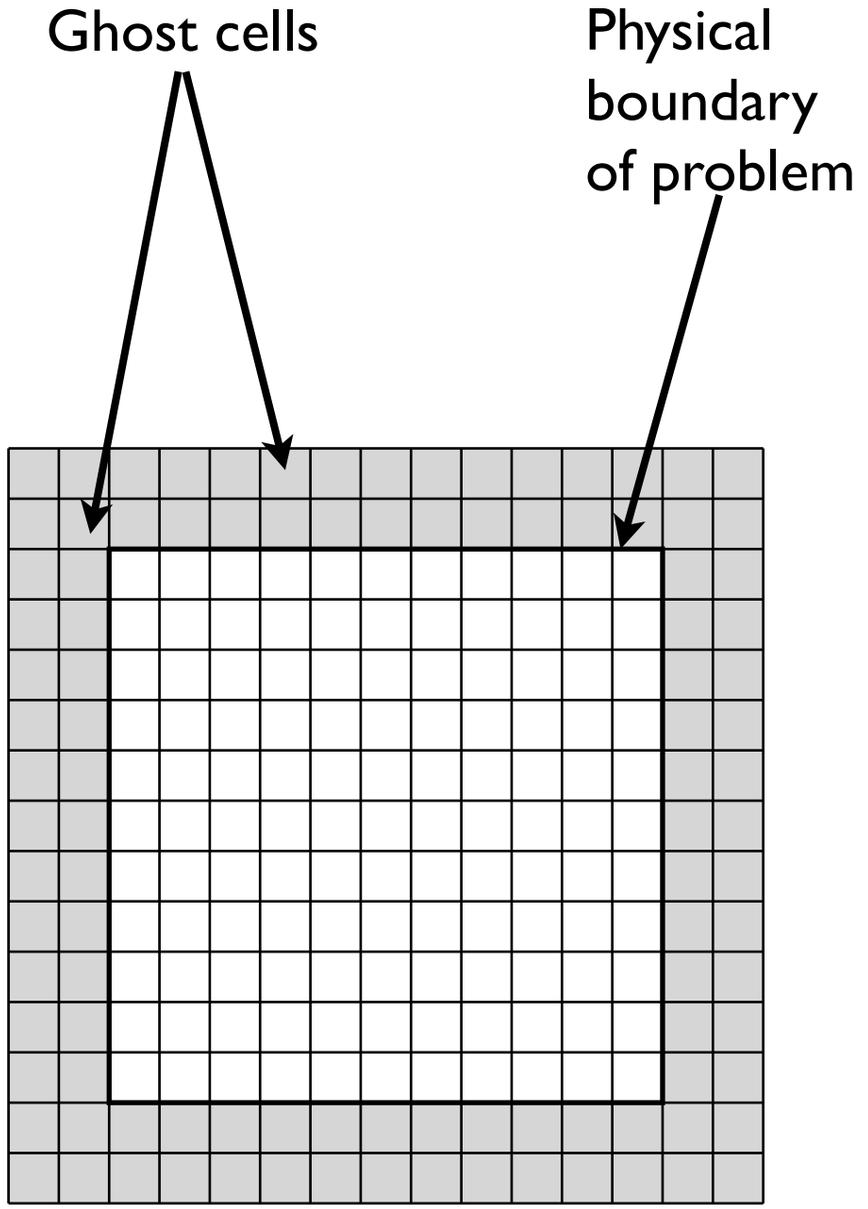
Boundary conditions

Boundary conditions are handled in two or more dimensions in much the same way as they are in one dimension - by using ghost cells in all dimensions. Note that the corners need to be taken care of and properly filled.

Periodic: simply copy the data from the other side of the grid (doubly periodic maps a torus!).

For **Solid Walls**, it is the velocity *normal* to the wall that is negated in the ghost cells (the x velocity along the right and left edges and the y velocity along the bottom and top edges) while the other quantities are copied from the neighboring interior cells. In particular, any *tangential* velocity (*free slip*) is allowed.

These boundary conditions, and simple extrapolation, are implemented in `clawpack/2d/lib/bc2.f`

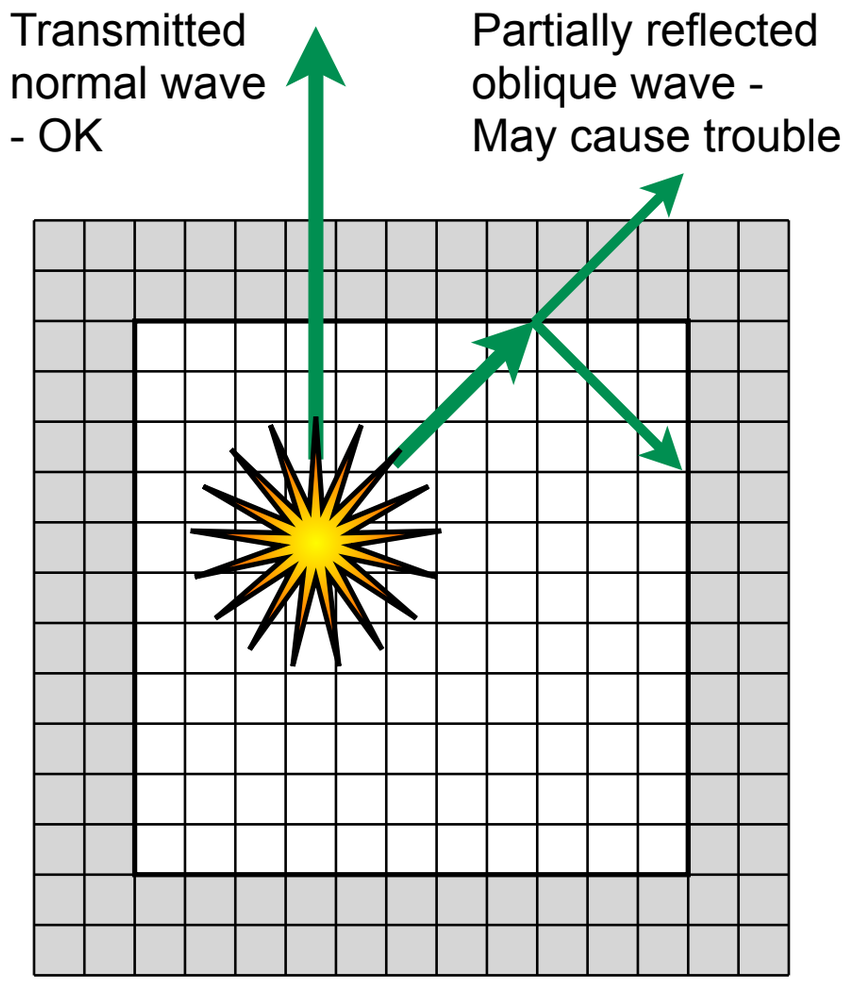


A two-dimensional grid with a border of two ghost cells on all boundaries

Boundary conditions - Extrapolation

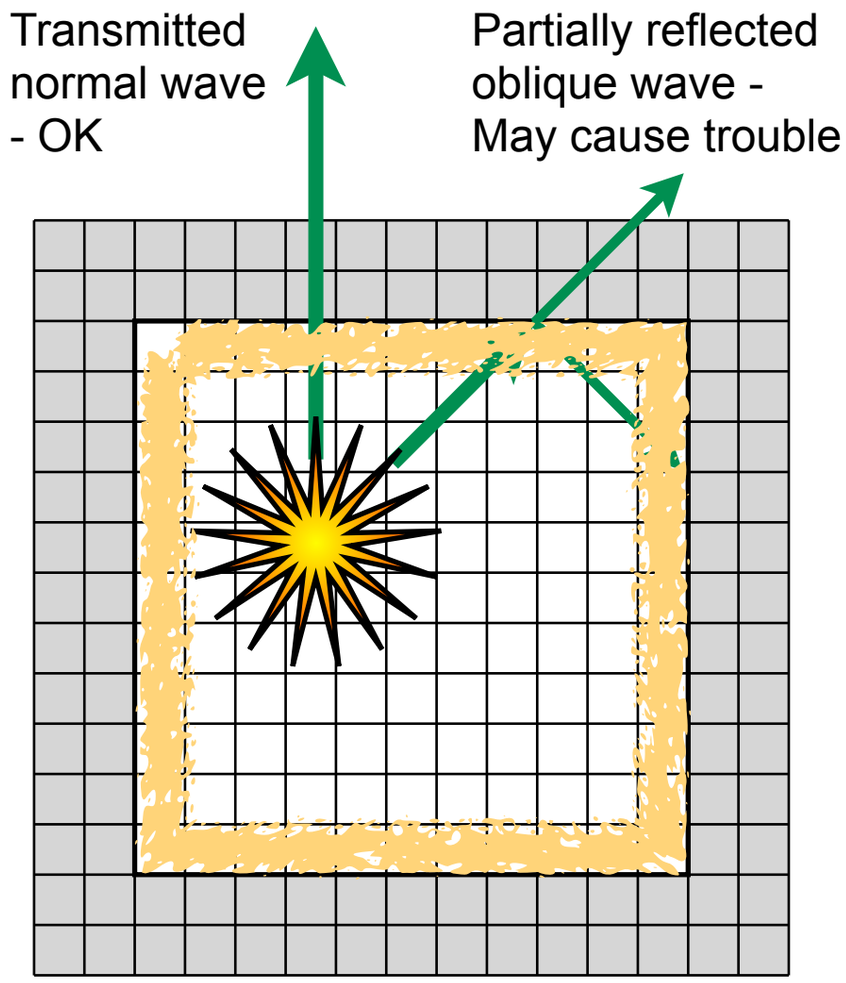
Extrapolation of outgoing waves is also included in `clawpack/2d/lib/bc2.f`, and often works fairly well. It works perfectly for waves moving normal to the boundaries, but may cause spurious reflections for waves moving *obliquely*, especially in the corners. See the discussion in Leveque section 21.8.5.

The simplest solution is to make the computational box large enough that reflections don't affect the region of interest.



Boundary conditions - Extrapolation

Extrapolation of outgoing waves is also included in `clawpack/2d/lib/bc2.f`, and often works fairly well. It works perfectly for waves moving normal to the boundaries, but may cause spurious reflections for waves moving *obliquely*, especially in the corners. See the discussion in Leveque section 21.8.5.

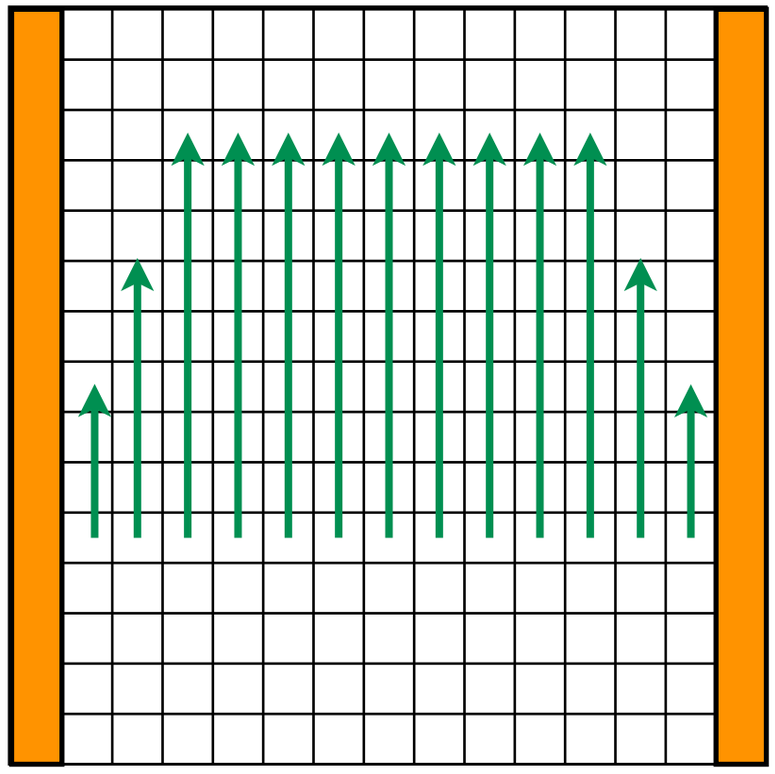


The simplest solution is to make the computational box large enough that reflections don't affect the region of interest.

Another possibility is to fill a portion of the interior adjacent to the boundary with material of high impedance so that waves passing through are attenuated. This is tricky, and may result in loss of hyperbolicity. Also this absorbing material will diffuse into the interior and may affect the solution.

No-Slip Boundaries

Real fluids experience wall friction and form a thin *boundary layer* next to a wall in which the tangential velocity rapidly approaches zero. In this boundary layer, viscosity is important, and there is therefore a loss of hyperbolicity in this region. You can implement no-slip conditions by negating the tangential velocity in the ghost cells, but be careful to watch for non-physical spreading of the resultant vorticity beyond the boundary layer.



What we've accomplished

We started with the one-dimensional advection equation, and developed a Godunov-style REA approach tailored to it;

We realised that systems of linear hyperbolic partial differential equations could be broken down into independent advection equations;

We learned how Riemann problems can be used to advance the solution of such problems from one time step to the next;

We learned that the procedure can be generalised to nonlinear hyperbolic equations and systems of equations;

We learned (perhaps) to accept the compromise of *approximate* Riemann solvers for nonlinear systems;

And we've extended what we've learned in one dimension to (at least) two.

Now a tiny bit more about the practicalities of the code -

Project recipe:

Write down all the equations of the problem in conservative form, including closure relations (equations of state, for example). You should also prepare an entropy equation that will be calculable should transonic conditions (or centred rarefactions) arise.

Find the Jacobian of the corresponding quasilinear system, and calculate its eigenvalues and eigenvectors.

For an arbitrary pair of right and left states, solve the Riemann problem, either exactly or approximately. Then write down general formulas for the waves, $\mathcal{W}^p_{i-1/2}$, the wave speeds $s^{\pm}$, the fluctuations $\mathcal{A}^{\pm}\Delta Q_{i-1/2}$ and the entropy fix for the transonic case.

Project recipe, continued

Under $CLAW/myclaw, prepare a new directory for the routines you must write. Write the Riemann solver (**rp1.f** or **rp2n.f** and **rp2t.f**) in Fortran in this directory. Copy over $CLAW/util/(**testrp1.f** or **testrp2n.f**), modify it and use it to test your Riemann solver.

Figure out what special work space you need, what boundary conditions, source terms, and other things that you want, and what special variables you need to input or initialise. Then write (or copy and modify) the appropriate routines (**driver**, **setprob**, setaux, **qinit**, bcN, b4stepN, srcN) in the same directory. Write (or copy and modify) a **Makefile** that points to these files, and construct **setrun.py** and **setplot.py** to fill the data files and make the plots.

Finally: compile, run, and check your results.

Examples of potential projects using Clawpack

Euler equations:
- Explosive volcanic eruptions
- High-energy meteor impacts

Shallow-water equations:
- Tsunami in a fjord system or in a basin of varied bathymetry

Dusty gas equations:
- Fluidisation and hydrothermal venting
- Geysers
- Volcanic jets
- Pyroclastic flows
- DeLaval nozzle in a dusty gas

Airy-wave equations:
- Normal (deep or intermediate) water waves
- Pockmarks
- Atmospheric dispersal of contaminants
- Climate patterns

Elastic equations:
- Seismic waves and deformations following impacts or severe earthquakes

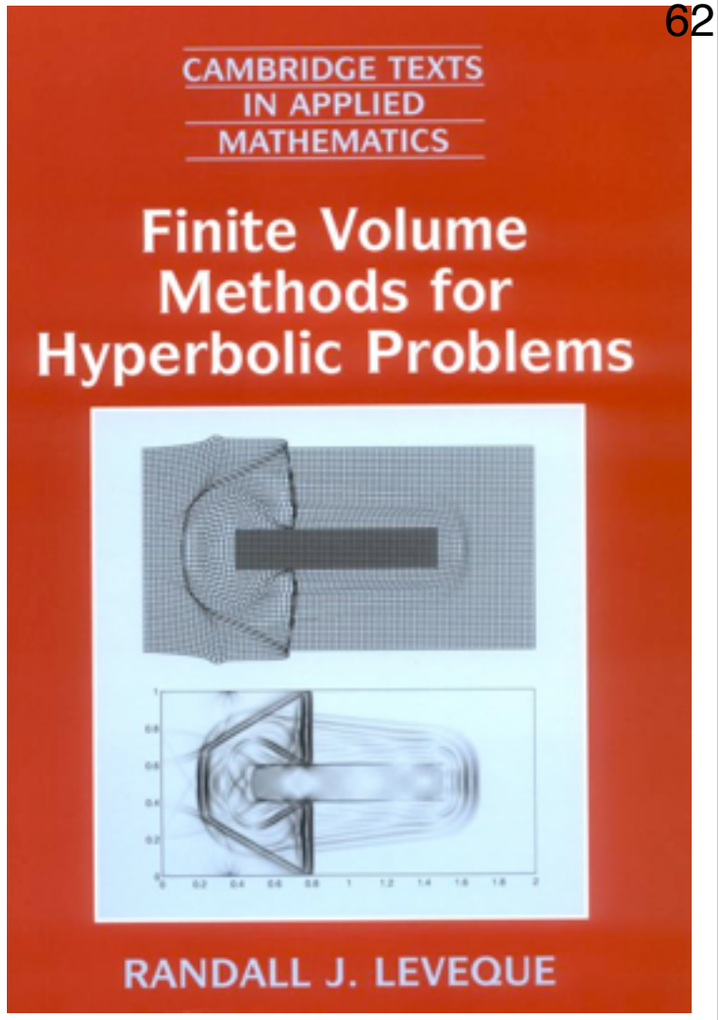
Assignment for next time

Skim Chapter 16, read Chapter 17 through 17.5 (more if you're interested) and read all of Chapters 18 and 19 (they are short!).

Write a one-paragraph draft description of a project you would like to do with Clawpack, including a description of the physical circumstances, and how you might implement it in code. Indicate the dimensionality of the problem, the equations you would like to solve (naming the equation set will be sufficient for this draft), whether there are source terms, what boundary conditions, etc.

Based on the project drafts, I will prepare some practical advice that I can give you.

Come talk to me regarding anything you would specifically like me to cover (or review) at the next lecture; particularly if you have any doubts about how to proceed with your project.



Next:
Waves in Elastic Media (Ch 22)
and project particulars