Outline

Today:
- CFL condition
- Numerical examples using Clawpack
- Numerical dissipation of upwind
- Lax-Wendroff method (second order)
- Numerical dispersion, modified equations

Next:
- High resolution methods

Reading: Chapters 5 and 6

Godunov’s method

$Q_i^n$ 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.

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

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

For linear system \( q_t + Aq_x = 0 \), the Riemann solution consists of waves \( W_p \) propagating at constant speed \( \lambda_p \).

\[
\begin{align*}
W_{i+1/2} & \quad \lambda_1 \Delta t \\
W_{i-1/2} & \quad \lambda_2 \Delta t \\
W_{i+3/2} & \quad \lambda_3 \Delta t \\
\end{align*}
\]

\[
Q_i - Q_{i-1} = \sum_{p=1}^{m} \alpha_{i-1/2}^p r_p \equiv \sum_{p=1}^{m} W_{i-1/2}^p
\]

\[
Q_i^{n+1} = Q_i^n - \frac{\Delta t}{\Delta x} \left[ \lambda_2 W_{i-1/2} + \lambda_3 W_{i+1/2} + \lambda_1 W_{i+1/2} \right].
\]

Matrix splitting for upwind method

For \( q_t + Aq_x = 0 \), the upwind method (Godunov) is:

\[
Q_i^{n+1} = Q_i^n + \frac{\Delta t}{\Delta x} \left[ \frac{1}{m} \sum_{p=1}^{m} (\lambda_p^+ + \alpha_{i-1/2}^p r_p + \sum_{p=1}^{m} (\lambda_p^- - \alpha_{i+1/2}^p r_p) \left[ A^+ \Delta Q_{i-1/2} + A^- \Delta Q_{i+1/2} \right] \\
= Q_i^n + \frac{\Delta t}{\Delta x} \left[ A^+ (Q_i^n - Q_{i-1}^n) + A^- (Q_{i+1}^n - Q_i^n) \right]
\]

Natural generalization of upwind to a system.

If all eigenvalues are positive, then \( A^+ = A \) and \( A^- = 0 \).
If all eigenvalues are negative, then \( A^+ = 0 \) and \( A^- = A \).

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

<table>
<thead>
<tr>
<th>Stencil</th>
<th>CFL Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Stencil" /></td>
<td>$0 \leq \frac{\lambda_p \Delta t}{\Delta x} \leq 1$, \forall p</td>
</tr>
<tr>
<td><img src="image" alt="Stencil" /></td>
<td>$-1 \leq \frac{\lambda_p \Delta t}{\Delta x} \leq 0$, \forall p</td>
</tr>
<tr>
<td><img src="image" alt="Stencil" /></td>
<td>$-1 \leq \frac{\lambda_p \Delta t}{\Delta x} \leq 1$, \forall p</td>
</tr>
<tr>
<td><img src="image" alt="Stencil" /></td>
<td>$0 \leq \frac{\lambda_p \Delta t}{\Delta x} \leq 2$, \forall p</td>
</tr>
<tr>
<td><img src="image" alt="Stencil" /></td>
<td>$-\infty &lt; \frac{\lambda_p \Delta t}{\Delta x} &lt; \infty$, \forall p</td>
</tr>
</tbody>
</table>

R.J. LeVeque, University of Washington AMath 574, January 24, 2011 [FVMHP Sec. 4.4]

Notes:

**Numerical Experiments**

Experiment with the code in
$\$CLAW/apps/advection/1d/example1

Make the following changes in setrun.py:

- Upwind method (clawdata.order = 1)
- Finer grid (clawdata.mx = 100)
- Periodic boundary conditions
  
  clawdata.mthbc_xlower = 2
  clawdata.mthbc_xupper = 2
- Narrower pulse (beta = 300 or 3000)
- Courant number greater than 1.
  
  clawdata.cfl_desired = 1.1
  clawdata.cfl_max = 1.1

R.J. LeVeque, University of Washington AMath 574, January 24, 2011

Notes:

**Upwind for a linear system**

The one-sided method

$$Q^{n+1}_i = Q^n_i - \frac{\Delta t}{\Delta x} A(Q^n_i - Q^n_{i-1})$$

is stable only if $0 \leq \Delta t \lambda_p / \Delta x \leq 1$ for all $p$.

**Upwind method based on sign of each $\lambda^p$:**

Let

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

$\Lambda^+ = \text{diag}(\lambda^p)^+$, $\Lambda^- = \text{diag}(\lambda^p)^-$,

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

Then

$$Q^{n+1}_i = Q^n_i - \frac{\Delta t}{\Delta x} A^+(Q^n_i - Q^n_{i-1}) - \frac{\Delta t}{\Delta x} A^-(Q^{n+1}_{i+1} - Q^n_i).$$

R.J. LeVeque, University of Washington AMath 574, January 24, 2011 [FVMHP Sec. 4.10]
Symmetric methods

Centered in space, forward in time:

\[ Q_{i}^{n+1} = Q_{i}^{n} - \frac{\Delta t}{\Delta x} \left( \frac{1}{2} A \right) (Q_{i}^{n} - Q_{i-1}^{n}) - \frac{\Delta t}{\Delta x} \left( \frac{1}{2} A \right) (Q_{i+1}^{n} - Q_{i}^{n}) \]

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

Centered approximation to \( q_x \), but **unstable** for any fixed \( \Delta t/\Delta x \).

Lax-Friedrichs:

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

This is stable if \( \frac{\Delta t}{\Delta x} \leq 1 \) for all \( p \).

Numerical dissipation

Lax-Friedrichs:

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

This can be rewritten as

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

\[ = Q_{i}^{n} - \Delta t A \left( \frac{Q_{i+1}^{n} - Q_{i}^{n}}{2\Delta x} \right) + \Delta t \left( \frac{\Delta x^2}{2\Delta t} \right) \left( \frac{Q_{i+1}^{n} - 2Q_{i}^{n} + Q_{i-1}^{n}}{\Delta x^2} \right) \]

The unstable method with the addition of **artificial viscosity**.

Approximates \( q_t + A q_x = \epsilon q_{xx} \) (modified equation)

with \( \epsilon = \frac{\Delta x^2}{2\Delta t} = O(\Delta x) \) if \( \Delta t/\Delta x \) is fixed as \( \Delta x \to 0 \).

Modified Equations

The upwind method

\[ Q_{i}^{n+1} = Q_{i}^{n} - \frac{\Delta t}{\Delta x} u(Q_{i}^{n} - Q_{i-1}^{n}) \]

gives a first-order accurate approximation to \( q_t + uq_x = 0 \).

But it gives a **second-order** approximation to

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

This is an advection-diffusion equation.

Indicates that the numerical solution will diffuse.

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

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

Second-order accuracy?
Taylor series:
\[ q(x, t + \Delta t) = q(x, t) + \Delta t q_t(x, t) + \frac{1}{2} \Delta t^2 q_{tt}(x, t) + \ldots \]
From \( q_t = -A q_x \) we find \( q_{tt} = A^2 q_{xx} \).

\[ q(x, t + \Delta t) = q(x, t) - \Delta t A q_x(x, t) + \frac{1}{2} \Delta t^2 A^2 q_{xx}(x, t) + \ldots \]
Replace \( q_x \) and \( q_{xx} \) by centered differences:
\[ Q^{n+1}_i = Q^n_i - \frac{\Delta t}{2\Delta x} A(Q^n_{i+1} - Q^n_{i-1}) + \frac{1}{2} \frac{\Delta t^2}{\Delta x^2} A^2(Q^n_{i+1} - 2Q^n_i + Q^n_{i-1}) \]

Modified Equation for Lax-Wendroff
The Lax-Wendroff method
\[ Q^{n+1}_i = Q^n_i - \frac{\Delta t}{2\Delta x} A(Q^n_{i+1} - Q^n_{i-1}) + \frac{1}{2} \frac{\Delta t^2}{\Delta x^2} A^2(Q^n_{i+1} - 2Q^n_i + Q^n_{i-1}) \]
gives a second-order accurate approximation to \( q_t + u q_x = 0 \).
But it gives a third-order approximation to
\[ q_t + u q_x = -\frac{u \Delta t^2}{6} \left( 1 - \left( \frac{u \Delta t}{\Delta x} \right)^2 \right) q_{xxx}. \]
This has a dispersive term with \( O(\Delta x^2) \) coefficient.
Indicates that the numerical solution will become oscillatory.

Dispersion relation
Consider a single Fourier mode:
\[ q(x, 0) = e^{i\xi x} \implies q(x, t) = e^{i(\xi x - \omega t)} \]
Determine \( \omega(\xi) \) based on the PDE.
This is the dispersion relation.
\[ q_t = -i \omega q, \quad q_x = i \xi q, \quad q_{xx} = -\xi^2 q, \quad q_{xxx} = -i \xi^3 q, \ldots \]

\[ q_t + u q_x = 0 \implies \omega(\xi) = u \xi, \quad q(x, t) = e^{i \xi (x - ut)} \]
(translates at speed \( u \) for all \( \xi \))

\[ q_t + u q_x = \xi q_{xx} \implies q(x, t) = e^{-\xi^2 t} e^{i \xi (x - ut)} \] (decays)

\[ q_t + u q_x = \beta q_{xxx} \implies q(x, t) = e^{\xi (x - (u + \beta \xi^2) t)} \]
(translates at speed \( u + \beta \xi^2 \) that depends on wave number!)