Advection

Grétar Tryggvason
Spring 2011

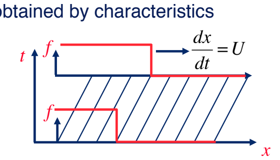
Discontinuous solutions—shocks

Consider the linear Advection Equation

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = 0 \qquad f(x,0) = \begin{cases} f_L & x < x_0 \\ f_R & x > x_0 \end{cases} \qquad (f_L > f_R)$$

The analytic solution is obtained by characteristics

$$\frac{dx}{dt} = U; \qquad \frac{df}{dt} = 0;$$



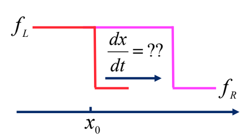
Discontinuity of solution is allowed

Inviscid Burgers' Equation

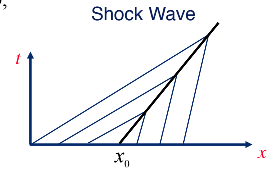
$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = 0 \qquad f(x,0) = \begin{cases} f_L & x < x_0 \\ f_R & x > x_0 \end{cases} \qquad (f_L > f_R)$$

Characteristics

$$\frac{dx}{dt} = f; \quad \frac{df}{dt} = 0;$$

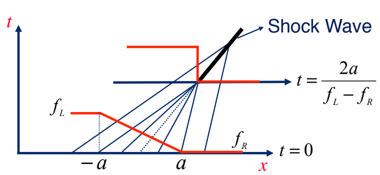


Shock Wave



A slight variation of the initial condition – formation of shock

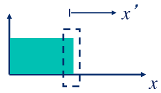
$$f(x,0) = \begin{cases} f_L & x < -a \\ \frac{1}{2}\left[(f_L + f_R) - (f_L - f_R)\frac{x}{a}\right] & -a < x < a \\ f_R & x > a \end{cases}$$



Shock Speed

Computational Fluid Dynamics
Discontinuous Solutions

The speed of the shock



Write: $x' = x - Ct$

$\rightarrow$ $$\frac{\partial f}{\partial t} = \frac{\partial f}{\partial t} + \frac{\partial f}{\partial x'}\frac{\partial x'}{\partial t}$$

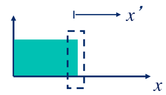
Substitute into: $\frac{\partial f}{\partial t} + \frac{\partial F}{\partial x} = 0$

$\rightarrow$ $$\frac{\partial f}{\partial t} - C\frac{\partial f}{\partial x'} + \frac{\partial F}{\partial x'} = 0 \quad \text{where} \quad \frac{\partial x'}{\partial t} = C$$

$\rightarrow$ $$\int_{\Delta\to 0}\left(\frac{\partial f}{\partial t} - C\frac{\partial f}{\partial x'} + \frac{\partial F}{\partial x'}\right)dx = 0$$

$$\int_{\Delta\to 0}\cancelto{0}{\left(\frac{\partial f}{\partial t}\right)}dx - \int_{\Delta\to 0}\left(C\frac{\partial f}{\partial x'}\right)dx + \int_{\Delta\to 0}\left(\frac{\partial F}{\partial x'}\right)dx = 0$$

Computational Fluid Dynamics
Discontinuous Solutions



$$-\int_{\Delta\to 0}\left(C\frac{\partial f}{\partial x'}\right)dx + \int_{\Delta\to 0}\left(\frac{\partial F}{\partial x'}\right)dx = 0$$

$\rightarrow$ $$-C(f_R - f_L) + (F_R - F_L) = 0$$ Rankine-Hugoniot relations

$\rightarrow$ $$C = \frac{F_R - F_L}{f_R - f_L}$$

Computational Fluid Dynamics
Discontinuous Solutions

Example

$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = 0 \qquad F = \frac{1}{2}f^2$$

$$C = \frac{F_R - F_L}{f_R - f_L} = \frac{1}{2}\frac{f_R^2 - f_L^2}{f_R - f_L} = \frac{1}{2}\frac{\cancel{(f_R - f_L)}(f_R + f_L)}{\cancel{f_R - f_L}}$$

$\rightarrow$ $$C = \frac{1}{2}(f_R + f_L)$$

Computational Fluid Dynamics
Discontinuous Solutions

Conservative Schemes are guarantied to give the correct Shock Speed since they correspond to a direct application of the conservation principles.

Non-conservative schemes may or may not do so.

Computational Fluid Dynamics
Conservative Method

Conservation and shock speed

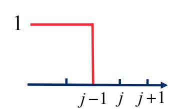
Example: inviscid Burgers equation:

$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = 0 \quad \text{or} \quad \frac{\partial f}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}f^2\right) = 0$$

In capturing the correct solution behavior for discontinuous Initial data, conservative methods are essential (Lax).

Computational Fluid Dynamics
Conservative Method

Example: for inviscid Burgers equation with discontinuous initial data



Consider upwind and forward Euler scheme:

Non-conservative form $f_t + ff_x = 0$

$$f_j^{n+1} = f_j^n - \frac{\Delta t}{h}f_j^n\left(f_j^n - f_{j-1}^n\right) = 0$$ Never moves!

Conservative form $f_t + \left(\frac{1}{2}f^2\right)_x = 0$

$$f_j^{n+1} = f_j^n - \frac{\Delta t}{2h}\left((f_j^n)^2 - (f_{j-1}^n)^2\right) = \frac{\Delta t}{2h}$$

Computational Fluid Dynamics

The Entropy Conditions

Computational Fluid Dynamics
The Entropy Conditions

Inviscid Burgers' Equation

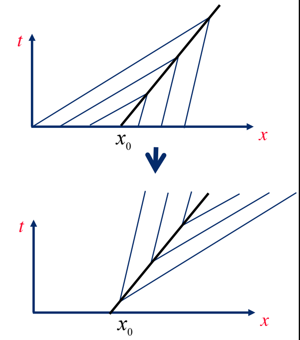
$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = 0$$

The transformation

$$x \to -x; \quad t \to -t$$

Leaves the equation unchanged but results in an unphysical solution.

The entropy condition is used to select the correct solution



Computational Fluid Dynamics
The Entropy Conditions

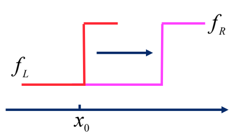
Reverse Shock (?)

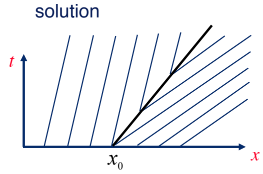
$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = 0 \qquad f(x,0) = \begin{cases} f_L & x < x_0 \\ f_R & x > x_0 \end{cases} \qquad (f_L < f_R)$$

Characteristics

$$\frac{dx}{dt} = f; \quad \frac{df}{dt} = 0;$$

Unstable, entropy-violating solution





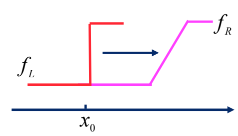
Computational Fluid Dynamics
The Entropy Conditions

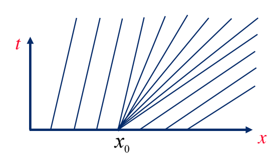
Rarefaction Wave (physically correct solution)

$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = 0 \qquad f(x,0) = \begin{cases} f_L & x < x_0 \\ f_R & x > x_0 \end{cases} \qquad (f_L < f_R)$$

Characteristics

$$\frac{dx}{dt} = f; \quad \frac{df}{dt} = 0;$$





Computational Fluid Dynamics
The Entropy Conditions

Weak solutions to hyperbolic equations may not be unique.

How can we find a physical solution out of many weak solutions?

In fluid mechanics, the actual physics always includes dissipation, i.e. in the form of viscous Burgers' equation:

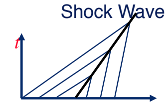
$$\frac{\partial f}{\partial t} + f\frac{\partial f}{\partial x} = \varepsilon\frac{\partial^2 f}{\partial x^2}$$

Therefore, what we are truly seeking is the solution to the viscous Burgers' equation in the limit of $\varepsilon \to 0$

Computational Fluid Dynamics
Entropy Condition

For a conservation equation

$$\frac{\partial f}{\partial t} + \frac{\partial F(f)}{\partial x} = 0$$



Entropy Condition:
A discontinuity propagating with speed C satisfies the entropy condition if

Version I: $F'(f_L) > C > F'(f_R)$

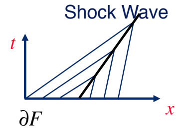
Version II: $\dfrac{F(f) - F(f_L)}{f - f_L} \geq C \geq \dfrac{F(f) - F(f_R)}{f - f_R}$ for $f_L \geq f \geq f_R$

And some others…

Computational Fluid Dynamics
Entropy Condition

Given a conservation equation

$$\frac{\partial f}{\partial t}+\frac{\partial F(f)}{\partial x}=0$$



Rewrite in “characteristic” form

$$\frac{\partial f}{\partial t}+\frac{\partial F}{\partial f}\frac{\partial f}{\partial x}=0 \quad \text{where:} \quad \frac{dt}{ds}=1; \quad \frac{dx}{ds}=\frac{\partial F}{\partial f}$$

or: $$\frac{dx}{dt}=\frac{\partial F}{\partial f}=F'(f)$$

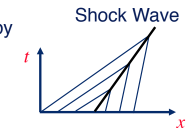
The Entropy Condition states that the characteristics must “enter” the discontinuity. Thus, its speed C satisfies must satisfy

$$F'(f_L)>C>F'(f_R)$$

Computational Fluid Dynamics
Entropy Condition

Similarly, the shock speed is given by

$$C=\frac{F_R-F_L}{f_R-f_L}$$



Thus

$$\frac{F(f)-F(f_L)}{f-f_L}\geq C\geq\frac{F(f)-F(f_R)}{f-f_R} \quad \text{for } f_L\geq f\geq f_R$$

Means that the hypothetical shock speed for values of f between the left and the right state must give shock speeds that are larger on the left and smaller on the right.

Computational Fluid Dynamics

Advecting a shock with several schemes

Computational Fluid Dynamics

Example Problem: Linear Wave Equation

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=0, \quad -\infty<x<\infty, \ t\geq 0$$

$$f(x,0)=\begin{cases}1 & x<0\\ 0 & x>0\end{cases}$$

Exact Solution: $f(x-Ut)$

Apply various numerical methods

Computational Fluid Dynamics

$$f_j^{n+1}=f_j^n-\frac{\Delta t}{h}U(f_j^n-f_{j-1}^n)$$ Upwind

$$f_j^{n+1}=f_j^{n-1}-\frac{U\Delta t}{h}\left(f_{j+1}^n-f_{j-1}^n\right)$$ Leap-frog

Lax-Wendroff

$$f_j^{n+1}=f_j^n-\frac{U\Delta t}{2h}\left(f_{j+1}^n-f_{j-1}^n\right)+\frac{U^2\Delta t^2}{2h^2}\left(f_{j+1}^n-2f_j^n+f_{j-1}^n\right)$$

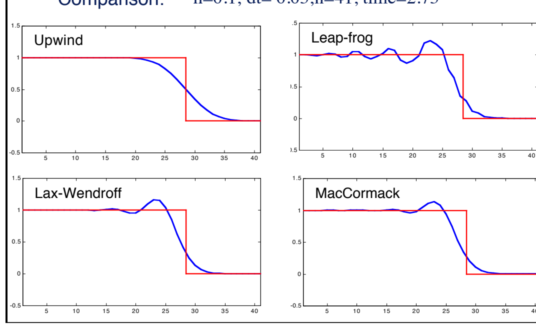
$$f_j^t=f_j^n-U\frac{\Delta t}{h}\left(f_{j+1}^n-f_j^n\right)$$

MacCormack

$$f_j^{n+1}=\frac{1}{2}\left[f_j^n+f_j^t-U\frac{\Delta t}{h}\left(f_j^t-f_{j-1}^t\right)\right]$$

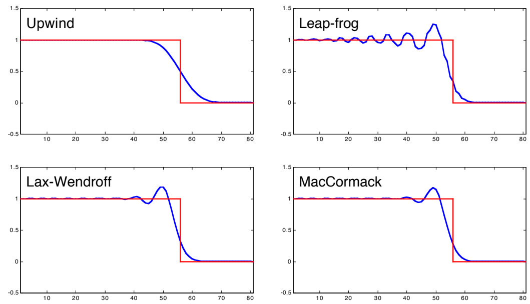
Computational Fluid Dynamics

Comparison: h=0.1; dt= 0.05;n=41, time=2.75



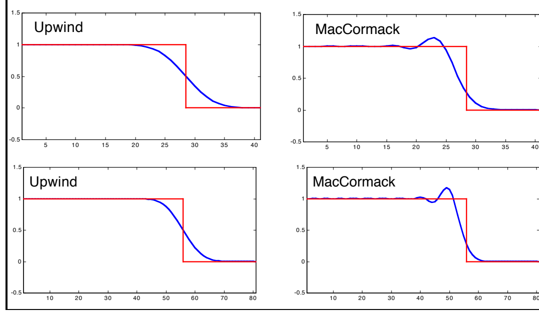
Computational Fluid Dynamics

Comparison: h=0.05; dt= 0.025;n=81, time=2.75



Computational Fluid Dynamics

Comparison: n=41 and n=81, time=2.75



Computational Fluid Dynamics

Observation 1:
Second-order methods tends to capture shaper solution (better accuracy), but they produce wiggly solutions.

Observation 2:
First-order methods are dissipative and less accurate, but the solution does not oscillate.
(preserves monotonicity).

Computational Fluid Dynamics

Modified Equation

Computational Fluid Dynamics

Looking at the structure of the error terms—by deriving the Modified Equation—can often lead to insight into how the approximate solution behaves

Computational Fluid Dynamics

Derive modified equation for upwind difference method:

$$\frac{f_j^{n+1}-f_j^n}{\Delta t}+\frac{U}{h}(f_j^n-f_{j-1}^n)=0$$

Using Taylor expansion:

$$f_j^{n+1}=f_j^n+\frac{\partial f}{\partial t}\Delta t+\frac{\partial^2 f}{\partial t^2}\frac{\Delta t^2}{2}+\frac{\partial^3 f}{\partial t^3}\frac{\Delta t^3}{6}+\cdots$$

$$f_{j-1}^n=f_j^n-\frac{\partial f}{\partial x}h+\frac{\partial^2 f}{\partial x^2}\frac{h^2}{2}-\frac{\partial^3 f}{\partial x^3}\frac{h^3}{6}+\cdots$$

Computational Fluid Dynamics

Substituting

$$\frac{1}{\Delta t}\left\{\left[\cancel{f_j^n}+\frac{\partial f}{\partial t}\Delta t+\frac{\partial^2 f}{\partial t^2}\frac{\Delta t^2}{2}+\frac{\partial^3 f}{\partial t^3}\frac{\Delta t^3}{6}+\cdots\right]-\cancel{f_j^n}\right\}$$

$$+\frac{U}{h}\left\{\cancel{f_j^n}-\left[\cancel{f_j^n}-\frac{\partial f}{\partial x}h+\frac{\partial^2 f}{\partial x^2}\frac{h^2}{2}-\frac{\partial^3 f}{\partial x^3}\frac{h^3}{6}+\cdots\right]\right\}=0$$

Therefore,

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=-\frac{\Delta t}{2}f_{tt}+\frac{Uh}{2}f_{xx}-\frac{\Delta t^2}{6}f_{ttt}-\frac{Uh^2}{6}f_{xxx}+\cdots$$

It helps the interpretation if all terms are written in f_{xx}, f_{xxx}

Computational Fluid Dynamics

Taking further derivatives (first in time, then in space):

$$f_{tt}+Uf_{xt}=-\frac{\Delta t}{2}f_{ttt}+\frac{Uh}{2}f_{xxt}-\frac{\Delta t^2}{6}f_{tttt}-\frac{Uh^2}{6}f_{xxxt}+\cdots$$

$$+\quad -Uf_{tx}-U^2f_{xx}=\frac{U\Delta t}{2}f_{ttx}-\frac{U^2h}{2}f_{xxx}+\frac{U\Delta t^2}{6}f_{tttx}+\frac{U^2h^2}{6}f_{xxxx}+\cdots$$

$$f_{tt}=U^2f_{xx}+\Delta t\left(\frac{-f_{ttt}}{2}+\frac{U}{2}f_{ttx}+O(\Delta t)\right)$$

$$+\Delta x\left(\frac{U}{2}u_{xxt}-\frac{U^2}{2}u_{xxx}+O(h)\right)$$

Computational Fluid Dynamics

Similarly, we get

$$f_{ttt}=-U^3f_{xxx}+O(\Delta t,h)$$

$$f_{ttx}=U^2f_{xxx}+O(\Delta t,h)$$

$$f_{xxt}=-Uf_{xxx}+O(\Delta t,h)$$

Final form of the modified equation: $\lambda=\frac{U\Delta t}{h}$

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=\frac{Uh}{2}(1-\lambda)f_{xx}-\frac{Uh^2}{6}(2\lambda^2-3\lambda+1)f_{xxx}$$

$$+O\left[h^3,h^2\Delta t,h\Delta t^2,\Delta t^3\right]$$

Computational Fluid Dynamics

By applying upwind differencing, we are effectively solving:

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=\frac{Uh}{2}(1-\lambda)f_{xx}-\frac{Uh^2}{6}(2\lambda^2-3\lambda+1)f_{xxx}+\cdots$$

Numerical dissipation (diffusion)

Also note that the CFL condition $\lambda=\frac{U\Delta t}{h}<1$

ensures a positive diffusion coefficient

$(\cdots)f_{xx}$ Dissipation

$(\cdots)f_{xxx}$ Dispersion

Computational Fluid Dynamics

Consider: $$\frac{\partial f}{\partial t}=\alpha\frac{\partial^n f}{\partial x^n}$$

Look for solutions of the form:

$$f(x,t)=a(t)e^{ikx}$$

Substitute to get

$$\frac{da(t)}{dt}=(ik)^n\alpha a(t)$$

solve

$$a(t)=a_o e^{(ik)^n \alpha t}$$

For even n we get diffusion, for odd n we get dispersion

Computational Fluid Dynamics

First-Order Methods and Diffusion $\lambda=\frac{U\Delta t}{h}$

Upwind:

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=\frac{Uh}{2}(1-\lambda)f_{xx}-\frac{Uh^2}{6}(2\lambda^2-3\lambda+1)f_{xxx}$$

Lax-Friedrichs:

$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=\frac{Uh}{2}\left(\frac{1}{\lambda}-\lambda\right)f_{xx}+\frac{Uh^2}{3}(1-\lambda^2)f_{xxx}$$

Dissipative = Smearing

Computational Fluid Dynamics

Second-Order Methods and Dispersion

$$\lambda = \frac{U\Delta t}{h}$$

Lax-Wendroff:

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = -\frac{Uh^2}{2}\left(1-\lambda^2\right) f_{xxx} - \frac{Uh^3}{8}\lambda\left(1-\lambda^2\right) f_{xxxx}$$

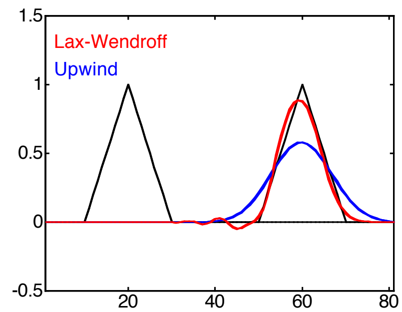
Beam-Warming:

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = \frac{Uh^2}{6}(1-\lambda)(2-\lambda) f_{xxx} - \frac{Uh^3}{8}(1-\lambda)^2(2-\lambda) f_{xxxx}$$

Dispersive = Wiggles

Computational Fluid Dynamics

dt=0.25*h



Computational Fluid Dynamics

First-Order Methods and Diffusion

$$\lambda = \frac{U\Delta t}{h}$$

Upwind:

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = \frac{Uh}{2}(1-\lambda) f_{xx} - \frac{Uh^2}{6}\left(2\lambda^2 - 3\lambda + 1\right) f_{xxx}$$

Dissipative = Smearing

Lax-Wendroff:

$$\frac{\partial f}{\partial t} + U\frac{\partial f}{\partial x} = -\frac{Uh^2}{2}\left(1-\lambda^2\right) f_{xxx} - \frac{Uh^3}{8}\lambda\left(1-\lambda^2\right) f_{xxxx}$$

Dispersive = Wiggles

Computational Fluid Dynamics

Artificial Viscosity

Computational Fluid Dynamics
Artificial Viscosity

Use centered difference method (e.g. Lax-Wendroff)
- Second order accuracy
- Oscillation near discontinuity

In order to "damp out" oscillation, we can either
- Use implicit numerical viscosity (upwind) or
- Add an explicit numerical viscosity

Computational Fluid Dynamics
Artificial Viscosity

Von Neumann and Richtmyer (1950)

Consider: $$\frac{\partial f}{\partial t} + \frac{\partial F}{\partial x} = 0$$

O(1) coefficient

Replace the flux by: $$F' = F - \alpha\frac{\partial f}{\partial x}; \quad \alpha = Dh^2\left|\frac{\partial f}{\partial x}\right|$$

Giving: $$\frac{\partial f}{\partial t} + \frac{\partial F}{\partial x} = -\frac{\partial}{\partial x}\left(-\alpha\frac{\partial f}{\partial x}\right) = \underbrace{\frac{\partial}{\partial x}\left(Dh^2\left|\frac{\partial f}{\partial x}\right|\frac{\partial f}{\partial x}\right)}_{\to 0 \text{ as } h \to 0}$$

- Simulates the effect of the physical viscosity on the grid scale, concentrated around discontinuity and negligible elsewhere.
- h^2 is necessary to keep the viscous term of higher order.

Computational Fluid Dynamics
Artificial Viscosity

Example: Linear Advection Equation:

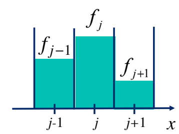
$$\frac{\partial f}{\partial t}+U\frac{\partial f}{\partial x}=\frac{\partial}{\partial x}\left(Dh^2\left|\frac{\partial f}{\partial x}\right|\frac{\partial f}{\partial x}\right)$$

Discretization by Lax-Wendroff

$$f_j^{n+1}=f_j^n-\frac{U\Delta t}{2h}\left(f_{j+1}^n-f_{j-1}^n\right)+\frac{1}{2}\left(\frac{U\Delta t}{h}\right)^2\left(f_{j+1}^n-2f_j^n+f_{j-1}^n\right)$$
$$+\frac{\Delta t}{h}\left[\left(Dh^2\left|\frac{\partial f}{\partial x}\right|\frac{\partial f}{\partial x}\right)_{j+1/2}-\left(Dh^2\left|\frac{\partial f}{\partial x}\right|\frac{\partial f}{\partial x}\right)_{j-1/2}\right]$$

Computational Fluid Dynamics
Artificial Viscosity

Approximate:



$$\left(Dh^2\left|\frac{\partial f}{\partial x}\right|\frac{\partial f}{\partial x}\right)_{j+1/2}=D\left|f_{j+1}^n-f_j^n\right|\left(f_{j+1}^n-f_j^n\right)$$

gives

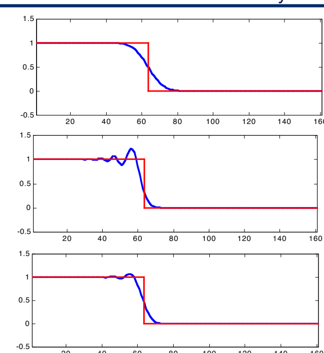
$$f_j^{n+1}=f_j^n-\frac{U\Delta t}{2h}\left(f_{j+1}^n-f_{j-1}^n\right)+\frac{1}{2}\left(\frac{U\Delta t}{h}\right)^2\left(f_{j+1}^n-2f_j^n+f_{j-1}^n\right)$$
$$+D\frac{\Delta t}{h}\left[\left|f_{j+1}^n-f_j^n\right|\left(f_{j+1}^n-f_j^n\right)-\left|f_j^n-f_{j-1}^n\right|\left(f_j^n-f_{j-1}^n\right)\right]$$

Computational Fluid Dynamics
Artificial Viscosity

```
% Artificial Viscosity by LaxWendroff
n=161; nstep=250; length=4.0;h=length/(n-1);dt=0.25*h;
y=zeros(n,1);f=zeros(n,1);f(1)=1;time=0;

for m=1:nstep,m
  hold off;plot(f,'linewidt',2);  axis([1 n -0.5, 1.5]);hold on;
  plot([1,dt*(m-1)/h+1.5,dt*(m-1)/h+1.5,n],[1,1,0,0],'r','linewidt',2);pause;
  y=f; time=time+dt;
  for i=2:n-1,
    f(i)=y(i)-(0.5*dt/h)*(y(i+1)-y(i-1))+...
       (0.5*dt*dt/h/h)*(y(i+1)-2.0*y(i)+y(i-1))+...
              +0.5*(dt/h)*(abs(y(i+1)-y(i))*(y(i+1)-y(i))-...
                       abs(y(i)-y(i-1))*(y(i)-y(i-1)) );
  end;
end;
```

Computational Fluid Dynamics
Artificial Viscosity



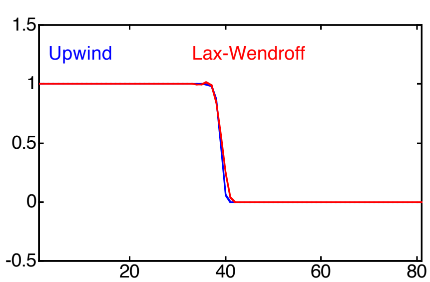
upwind

Lax-Wendroff
D=0

Lax-Wendroff
D=0.5

Computational Fluid Dynamics

Nonlinear advection equation



Computational Fluid Dynamics

Artificial viscosity can be used with most other centered difference schemes and was, for a while, THE way aeronautical computations were done. Other types, including higher order, have been used.