Computational Fluid Dynamics

Lecture 10
February 27, 2017

Grétar Tryggvason

The Euler Equations

The Euler equations for 1D flow:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho u(E+p/\rho)\end{pmatrix}=0 \qquad \begin{aligned}p&=(\gamma-1)\rho e\\ c^2&=\gamma p/\rho\end{aligned}$$

where $E=e+u^2/2$ and we define $H=h+u^2/2;\ h=e+p/\rho$

Ideal Gas: $p=\rho RT;\quad e=e(T);\quad c_v=de/dT$

$$\boxed{\begin{aligned}&h=h(T);\quad c_p=dh/dT;\\ &h-e=RT\\ &R=h/T-e/T=c_p-c_v;\\ &\gamma=c_p/c_v;\end{aligned}}$$

Pressure and energy are related by

$$p=\rho(h-e)=(h/e-1)\rho e=(\gamma-1)\rho e$$

Speed of sound

$$c^2=\left(\frac{\partial p}{\partial \rho}\right)_s=\gamma RT=\gamma\frac{p}{\rho}$$

The Euler Equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho u(E+p/\rho)\end{pmatrix}=0 \qquad \begin{aligned}p&=(\gamma-1)\rho e\\ c^2&=\gamma p/\rho\end{aligned}$$

Expanding the derivative and rearranging the equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ u\\ p\end{pmatrix}+\begin{pmatrix}u & \rho & 0\\ 0 & u & 1/\rho\\ 0 & \rho c^2 & u\end{pmatrix}\frac{\partial}{\partial x}\begin{pmatrix}\rho\\ u\\ p\end{pmatrix}=0 \qquad \text{or} \qquad \frac{\partial \mathbf{f}}{\partial t}+\mathbf{A}\frac{\partial \mathbf{f}}{\partial x}=0$$

The Characteristics for the Euler Equations are found by finding the eigenvalues for A^T

$$\det(\mathbf{A}^T-\lambda\mathbf{I})=0$$

Find

$$\det(\mathbf{A}^T-\lambda\mathbf{I})=0 \qquad \text{or} \qquad \det\begin{pmatrix}u-\lambda & 0 & 0\\ \rho & u-\lambda & \rho c^2\\ 0 & 1/\rho & u-\lambda\end{pmatrix}=0$$

or:

$$(u-\lambda)\left((u-\lambda)^2-c^2\right)=0$$

$$(u-\lambda)=0\Rightarrow\lambda=u$$

$$\left((u-\lambda)^2-c^2\right)=0$$

$$\Rightarrow u-\lambda=\pm c$$

$$\Rightarrow \lambda=u\pm c$$

Therefore

$$\lambda_1=u-c;$$
$$\lambda_2=u+c;$$
$$\lambda_3=u$$

The Rankine-Hugoniot conditions

For a hyperbolic system

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

The speed of a discontinuity (shock) is found by moving to a frame where a shock moving with speed s is stationary

$$\cancelto{0}{\frac{\partial \mathbf{f}}{\partial t}}-s\frac{\partial \mathbf{f}}{\partial x}+\frac{\partial \mathbf{F}}{\partial x}=0$$

Integrating across the shock yields

$$s[\mathbf{f}]=[\mathbf{F}]$$

Computational Fluid Dynamics
Exact Solution:

For the Euler equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho u \\ \rho E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u \\ \rho u^2 + p \\ \rho u(E + p/\rho)\end{pmatrix} = 0$$

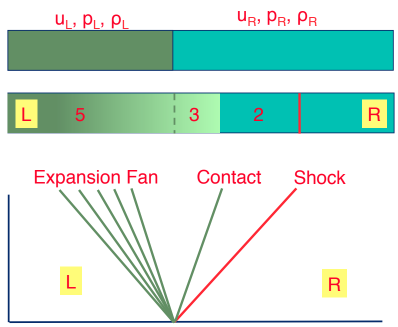
The Rankine-Hugoniot conditions are:

$$s(\rho_L - \rho_R) = \left((\rho u)_L - (\rho u)_R\right)$$
$$s\left((\rho u)_L - (\rho u)_R\right) = \left((\rho u^2 + p)_L - (\rho u^2 + p)_R\right)$$
$$s\left((\rho E)_L - (\rho E)_R\right) = \left((\rho u(E + p/\rho))_L - (\rho u(E + p/\rho))_R\right)$$

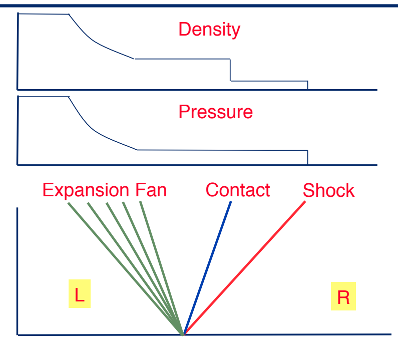
Computational Fluid Dynamics

The Shock-Tube Problem Exact Solution

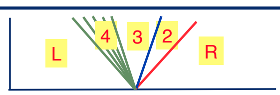
Computational Fluid Dynamics
The shock tube problem



Computational Fluid Dynamics



Computational Fluid Dynamics
Exact Solution:



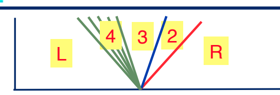
given: $p_L, \rho_L, u_L,$ $\quad p_R, \rho_R, u_R$

$$\alpha = \frac{\gamma+1}{\gamma-1}$$

$$c_L = \sqrt{\gamma p_L/\rho_L} \qquad c_R = \sqrt{\gamma p_R/\rho_R}$$

Consider the case $p_L > p_R$:
- Shock separates R and 2
- Contact discontinuity separates 2 and 3
- Expansion fan separates 3 and L

Computational Fluid Dynamics
Exact Solution:



given: $p_L, \rho_L, u_L,$ $\quad p_R, \rho_R, u_R$

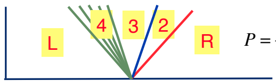
The Rankine-Hugoniot conditions give a nonlinear relation for the pressure jump across the shock $P = \frac{p_2}{p_R}$

$$P = \frac{p_L}{p_R}\left[1 - \frac{(\gamma-1)(c_R/c_L)(P-1)}{\sqrt{2\gamma\left(2\gamma + (\gamma+1)(P-1)\right)}}\right]^{2\gamma/(\gamma-1)}$$

which can be solved by iteration

Computational Fluid Dynamics
Exact Solution:

The speed of the shock and velocity behind the shock are found using RH conditions:



$$P = \frac{p_2}{p_R}$$

$$s_{shock} = u_R + c_R\left(\frac{\gamma-1+(\gamma+1)P}{2\gamma}\right)^{1/2} \qquad x_{shock} = x_0 + s_{shock}t$$

The speed of the contact is

$$s_{contact} = u_3 = u_2 = u_L + \frac{2c_L}{\gamma-1}\left(1-\left(P\frac{p_R}{p_L}\right)^{\frac{\gamma-1}{2\gamma}}\right) \qquad x_{contact} = x_0 + s_{contact}t$$

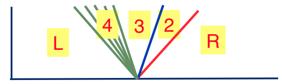
The left hand side of the fan moves with speed

$$s_{fL} = -c_L \qquad x_{fL} = x_0 + s_{fL}t$$

The right hand side of the fan moves with speed

$$s_{fR} = u_2 - c_3 \qquad x_{fR} = x_0 + s_{fR}t$$

Computational Fluid Dynamics
Exact Solution:



$$P = \frac{p_2}{p_R}$$

Left uniform state

given:

$p_L, \rho_L, u_L,$

$$c_L = \sqrt{\gamma p_L/\rho_L}$$

In the expansion fan (4)

$$\rho_4 = \rho_4\left(\frac{1+(s-x)}{c_L\alpha t}\right)^{\frac{2\gamma}{\gamma-1}}$$

$$u_4 = u_L + \frac{2}{\gamma+1}\left(\frac{s-x}{t}\right)$$

$$p_4 = p_4\left(\frac{1+(s-x)}{c_L\alpha t}\right)^{\frac{2\gamma}{\gamma-1}}$$

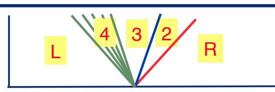
Behind the contact (3)

$$p_3 = p_2$$

$$u_3 = u_2$$

$$\rho_3 = \rho_L\left(P\frac{p_R}{p_L}\right)^{1/\gamma}$$

Computational Fluid Dynamics
Exact Solution:



$$\alpha = \frac{\gamma+1}{\gamma-1} \qquad P = \frac{p_2}{p_R}$$

Behind the contact (3)

$$p_3 = p_2$$

$$u_3 = u_2$$

$$\rho_3 = \rho_L\left(P\frac{p_R}{p_L}\right)^{1/\gamma}$$

Behind the shock (2)

$$\rho_2 = \left(\frac{1+\alpha P}{\alpha + P}\right)\rho_R$$

$$p_2 = P\,p_R$$

$$u_2 = u_L + \frac{2c_L}{\gamma-1}\left(1-\left(P\frac{p_R}{p_L}\right)^{\frac{\gamma-1}{2\gamma}}\right)$$

Right uniform state

given:

p_R, ρ_R, u_R

$$c_R = \sqrt{\gamma p_R/\rho_R}$$

Computational Fluid Dynamics
Exact Solution:

The velocity relative to the speed of sound defines the flow regime

$$c = \sqrt{\frac{\gamma p}{\rho}}$$

The Mach number is defined as the ratio of the local velocity over the speed of sound

$$Ma = \frac{u}{c}$$

$Ma < 1$ subsonic

$Ma > 1$ supersonic

Computational Fluid Dynamics
Exact Solution:

Test case:

Shocktube problem of G.A. Sod, JCP 27:1, 1978

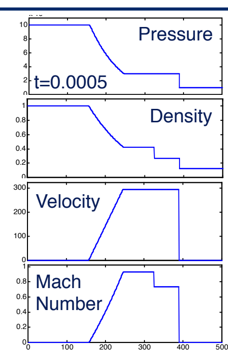
$p_L = 10^5;\quad \rho_L = 1.0;\quad u_L = 0$

$p_R = 10^4;\quad \rho_R = 0.125;\quad u_R = 0$

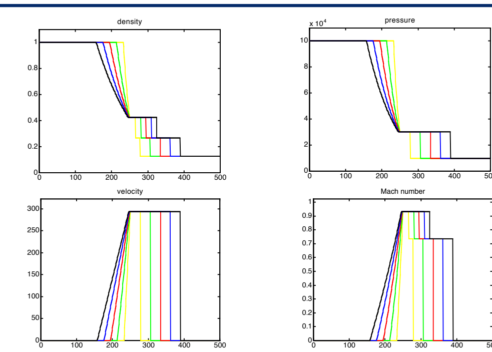
$t_{final} = 0.005$

$x_0 = 0.5$

Subsonic case



Computational Fluid Dynamics
Exact Solution:



Computational Fluid Dynamics
Exact Solution:

Test case:

Shocktube problem of G.A. Sod, JCP 27:1, 1978

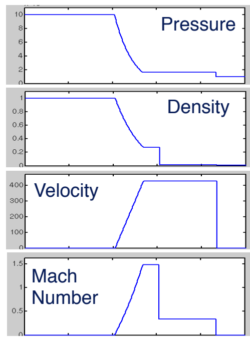
$p_L = 10^5; \quad \rho_L = 1.0; \quad u_L = 0$

$p_R = 10^4; \quad \rho_R = 0.01; \quad u_R = 0$

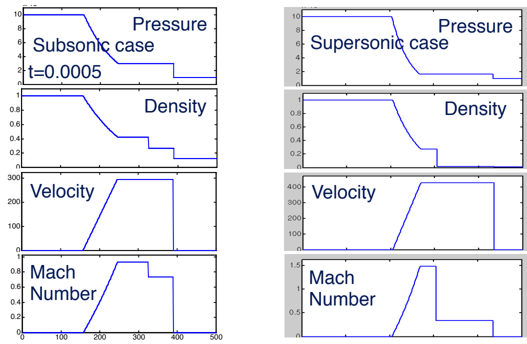
$t_{final} = 0.0025$

$x_0 = 0.5$

Supersonic case



Computational Fluid Dynamics



Computational Fluid Dynamics

Centered Schemes

Computational Fluid Dynamics
Flux Splitting

For a system of equations the characteristics often have different slopes and we therefore have different "upwind" directions

The simplest approach to solve the Euler equations is by using centered schemes where it is not necessary to take into account the characteristic direction

We will examine two such schemes:

The first order Lax-Friedrich scheme and

The second order Lax-Wendroff scheme

Computational Fluid Dynamics

First order Lax-Friedrich scheme

Computational Fluid Dynamics
Exact Solution:

Write the Euler equations as

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

Update the solution by

$$\mathbf{f}_j^{n+1} = \frac{1}{2}\left(\mathbf{f}_{j+1}^n + \mathbf{f}_{j-1}^n\right) - \frac{1}{2}\frac{\Delta t}{h}\left(\mathbf{F}_{j+1}^n - \mathbf{F}_{j-1}^n\right)$$

The error term contains an term including $h^2/\Delta t$ but by taking $h \sim \Delta t$ we have a first order scheme

The scheme is often written as

$$\mathbf{f}_j^{n+1} = \mathbf{f}_j^n - \frac{\Delta t}{h}\left(\mathbf{F}_{j+1/2}^n - \mathbf{F}_{j-1/2}^n\right)$$

where $\mathbf{F}_{j+1/2} = \frac{1}{2}\left(\mathbf{F}_{j+1} + \mathbf{F}_j\right) - \frac{1}{2}\frac{h}{\Delta t}\left(\mathbf{f}_{j+1}^n - \mathbf{f}_j^n\right)$

Computational Fluid Dynamics

```
%Euler equation--one-dimensional Lax-Friedrich
nx=256; maxstep=150;
gg=1.4;p_left=100000;p_right=10000;r_left=1;r_right=0.125;u_left=0;
xl=10.0;h=xl/(nx-1);time=0; for i=1:nx,x(i)=h*(i-1);end

r=zeros(1,nx);ru=zeros(1,nx);rE=zeros(1,nx);
rn=zeros(1,nx);run=zeros(1,nx);rEn=zeros(1,nx);
c=zeros(1,nx);u=zeros(1,nx);m=zeros(1,nx);

for i=1:nx,r(i)=r_right;ru(i)=0.0;rE(i)=p_right/(gg-1);end
for i=1:nx/2; r(i)=r_left;
        rE(i)=p_left/(gg-1)+0.5*r_left*u_left^2;
        ru(i)=r_left*u_left;end

rh=r;ruh=ru;rEh=rE;

cmax=sqrt( max(gg*p_right/r_right,gg*p_left/r_left) ); dt=0.45*h/cmax;

for istep=1:maxstep
  for i=1:nx,p(i)=(gg-1)*(rE(i)-0.5*(ru(i)*ru(i)/r(i)));end
  for i=2:nx-1
    rh(i)=0.5*(r(i+1)+r(i-1))-(0.5*dt/h)*(ru(i+1)-ru(i-1));
    ruh(i)=0.5*(ru(i+1)+ru(i-1))-...
                (0.5*dt/h)*((ru(i+1)^2/r(i+1))+p(i+1)-(ru(i-1)^2/r(i-1))-p(i-1));
    rEh(i)=0.5*(rE(i+1)+rE(i-1))-...
      (0.5*dt/h)*((ru(i+1)/r(i+1))*(rE(i+1)+p(i+1))...
                          -(ru(i-1)/r(i-1))*(rE(i-1)+p(i-1)) );
end

r=rh;ru=ruh;rE=rEh;
%plot(r,'r','linewidth',2); pause
time=time+dt,istep
end
```

Computational Fluid Dynamics

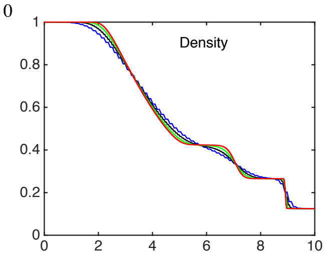
Shocktube problem of
G.A. Sod, JCP 27:1, 1978

Results for LF

$$p_L = 10^5; \quad \rho_L = 1.0; \quad u_L = 0$$
$$p_R = 10^4; \quad \rho_R = 0.125; \quad u_R = 0$$

Grid size
128
256
512
1024

Final time
0.0071



Computational Fluid Dynamics

Second order Lax-Wendroff scheme

Computational Fluid Dynamics
L-W with artificial viscosity

The Euler equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho \\ \rho u \\ \rho E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u \\ \rho u^2 + p \\ \rho u(E + p/\rho)\end{pmatrix} = 0$$

where $E = e + u^2/2; \quad p = (\gamma - 1)\rho e$

Add the artificial viscosity to RHS:

$$= \frac{\partial}{\partial x}\left(\alpha h^2 \rho \begin{vmatrix}0 \\ 1 \\ u\end{vmatrix} \left|\frac{\partial u}{\partial x}\right| \frac{\partial u}{\partial x} \right)$$

Computational Fluid Dynamics
L-W with artificial viscosity

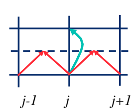
Solutions of the 1D Euler equation using Lax-Wendroff

$$\mathbf{f}^*_{j+1/2} = 0.5\left(\mathbf{f}^n_j + \mathbf{f}^n_{j+1}\right) - 0.5\frac{\Delta t}{h}\left(\mathbf{F}^n_{j+1} - \mathbf{F}^n_j\right) \qquad \text{Lax Fredrich}$$

$$\mathbf{f}^{n+1}_j = \mathbf{f}^n_j - \frac{\Delta t}{h}\left(\mathbf{F}^*_{j+1/2} - \mathbf{F}^*_{j-1/2}\right) \qquad \text{Leap Frog}$$

With an artificial viscosity term added to the corrector step

$$\mathbf{F}' = \mathbf{F} - \alpha h^2 \rho \begin{bmatrix}0 \\ 1 \\ u\end{bmatrix} \left|\frac{\partial u}{\partial x}\right| \frac{\partial u}{\partial x}$$



Computational Fluid Dynamics
L-W with artificial viscosity

Outline of L-W program

```
for istep=1:2000
    for i=1:nx,p(i)=.....; end

    for i=2:nx-1                        %prediction step
       rh(i)= .....
       ruh(i)= .....
       rEh(i)= .....
    end

    for i=1:nx,ph(i)=.....; end

    for i=2:nx-1                        %correction step
      r(i)= .....
      ru(i)= .....
      rE(i)= .....
    end

    for i=1:nx,u(i)=ru(i)/r(i);end

    for i=2:nx-1                        %artificial viscosity
      ru(i)= .....
      rE(i)= .....
    end
    time=time+dt,istep
end
```

Computational Fluid Dynamics
L-W with artificial viscosity

```
nx=129; artvisc=0.5;
hold off
%gg=1.4;p_left=100000;p_right=1000;r_left=1;r_right=0.01;
gg=1.4;p_left=100000;p_right=10000;r_left=1;r_right=0.125;
xl=10.0;h=xl/(nx-1);time=0;

r=zeros(1,nx);ru=zeros(1,nx);rE=zeros(1,nx);p=zeros(1,nx);
rh=zeros(1,nx);ruh=zeros(1,nx);rEh=zeros(1,nx);ph=zeros(1,nx);

for i=1:nx,r(i)=r_right;ru(i)=0.0;rE(i)=p_right/(gg-1);end
for i=1:nx/2; r(i)=r_left; rE(i)=p_left/(gg-1); end

rh=r;ruh=ru;rEh=rE;ph=p;

dt=0.25*h/sqrt(1.4*max([700+p_right/r_right,700+p_left/r_left]) )
```

```
for istep=1:2000

for i=1:nx,p(i)=(gg-1)*(rE(i)-0.5*(ru(i)*ru(i)/r(i)));end

for i=2:nx-1                                %prediction step
  rh(i)=0.5*(r(i)+r(i+1))-(0.5*dt/h)*(ru(i+1)-ru(i));
  ruh(i)=0.5*(ru(i)+ru(i+1))-(0.5*dt/h)*((ru(i+1)^2/r(i+1))+p(i+1)-(ru(i)^2/r(i))-p(i));
  rEh(i)=0.5*(rE(i)+rE(i+1))-...
              (0.5*dt/h)*((rE(i+1)*ru(i+1)/r(i+1))+(ru(i+1)*p(i+1)/r(i+1))...
                          -(rE(i)*  ru(i)  /r(i))-  (ru(i)  *p(i)  /r(i)));
end

for i=1:nx,ph(i)=(gg-1)*(rEh(i)-0.5*(ruh(i)*ruh(i)/rh(i)));end

for i=2:nx-1                                %correction step
 r(i)=r(i)-(dt/h)*(ruh(i)-ruh(i-1));
 ru(i)=ru(i)-(dt/h)*((ruh(i)^2/rh(i))-(ruh(i-1)^2/rh(i-1))+ph(i)-ph(i-1));
 rE(i)=rE(i)-(dt/h)*((rEh(i)*ruh(i)/rh(i))-(rEh(i-1)*ruh(i-1)/rh(i-1))+...
              (ruh(i)*ph(i)/rh(i))-(ruh(i-1)*ph(i-1)/rh(i-1)));
end

for i=1:nx,u(i)=ru(i)/r(i);end

for i=2:nx-1                                %artificial viscosity
 ru(i)=ru(i)+artvisc*(0.5*dt/h)*( (r(i)+r(i+1))*(u(i+1)-u(i))*abs(u(i+1)-u(i))...
                -(r(i)+r(i-1))*(u(i)-u(i-1))*abs(u(i)-u(i-1)) );

 rE(i)=rE(i)+artvisc*(0.25*dt/h)*...
            ( (u(i+1)+u(i))*(r(i)+r(i+1))*(u(i+1)-u(i))*abs(u(i+1)-u(i))...
             -(u(i)+u(i-1))*(r(i)+r(i-1))*(u(i)-u(i-1))*abs(u(i)-u(i-1)) );
end
time=time+dt,istep
plot(p,'b','linewidth',2),title('pressure'); pause
if(time > 0.005)break,end
end
```

Computational Fluid Dynamics
L-W with artificial viscosity

Test case:

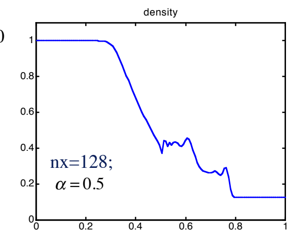
Results for LW

Shocktube problem of
G.A. Sod, JCP 27:1, 1978

$p_L = 10^5; \quad \rho_L = 1.0; \quad u_L = 0$

$p_R = 10^4; \quad \rho_R = 0.125; \quad u_R = 0$

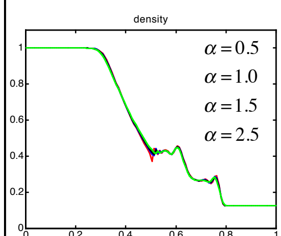
Final time: 0.005

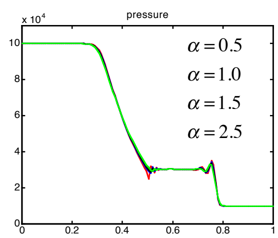


Computational Fluid Dynamics
L-W with artificial viscosity

Effect of α

nx=128;



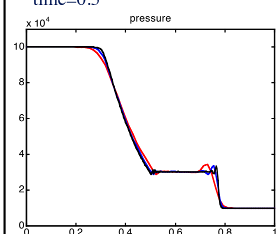


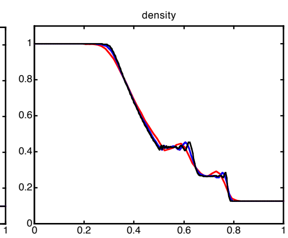
Computational Fluid Dynamics
L-W with artificial viscosity

Effect of resolution

nx=64;
nx=128;
nx=256;
time=0.5

$\alpha = 1.5$





Computational Fluid Dynamics

First order upwinding with flux splitting

Computational Fluid Dynamics
Flux Splitting

For conservation law

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

Which can be written as

$$\frac{\partial \mathbf{f}}{\partial t} + [A]\frac{\partial \mathbf{f}}{\partial x} = 0; \qquad [A] = \frac{\partial \mathbf{F}}{\partial \mathbf{f}}$$

Define $\mathbf{F} = \mathbf{F}^+ + \mathbf{F}^-$ where $[\lambda] = [\lambda^+] + [\lambda^-]$

So that

$$\frac{\partial \mathbf{f}}{\partial t} + \frac{\partial \mathbf{F}^+}{\partial x} + \frac{\partial \mathbf{F}^-}{\partial x} = 0$$

Are the positive and negative eigenvalues of A

Computational Fluid Dynamics
Flux Splitting

For nonlinear equation the splitting is not unique, different matrices can have the same eigenvalues

The history of flux splitting is interesting in that complex splitting were discovered first and the simpler one later

The simplest one is that Zha-Bilgen

Other examples are due to van Leer and Steger-Warming

Computational Fluid Dynamics
The Euler Equations

Zha-Bilgen flux splitting

$$\mathbf{F}^{+}=\max(u,0)\begin{bmatrix}\rho\\ \rho u\\ \rho e\end{bmatrix}+\begin{bmatrix}0\\ p^{+}\\ (pu)^{+}\end{bmatrix}$$

$$\mathbf{F}^{-}=\min(u,0)\begin{bmatrix}\rho\\ \rho u\\ \rho e\end{bmatrix}+\begin{bmatrix}0\\ p^{-}\\ (pu)^{-}\end{bmatrix},$$

Separate the advective fluxes and linearly interpolate the pressure terms between M=1 and M=-1.

$$p^{+}=p\begin{cases}0, & M\le -1\\ \frac{1}{2}(1+M), & -1<M<1,\\ 1, & M\ge 1\end{cases}\qquad p^{-}=p\begin{cases}1 & M\le -1\\ \frac{1}{2}(1-M), & -1<M<1,\\ 0, & M\ge 1\end{cases}$$

$$(pu)^{+}=p\begin{cases}0, & M\le -1\\ \frac{1}{2}(u+c), & -1<M<1\\ u, & M\ge 1\end{cases}\qquad (pu)^{-}=p\begin{cases}u & M\le -1\\ \frac{1}{2}(u-c), & -1<M<1.\\ 0, & M\ge 1\end{cases}$$

Computational Fluid Dynamics
The Euler Equations

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}^{+}}{\partial x}+\frac{\partial \mathbf{F}^{-}}{\partial x}=0.$$

$$\mathbf{f}_j^{n+1}=\mathbf{f}_j^{n}-\frac{\Delta t}{\Delta x}\Big(\mathbf{F}^{+}(\mathbf{f}_j^{n})-\mathbf{F}^{+}(\mathbf{f}_{j-1}^{n})+\mathbf{F}^{-}(\mathbf{f}_{j+1}^{n})-\mathbf{F}^{-}(\mathbf{f}_j^{n})\Big)$$

$$\mathbf{f}_j^{n+1}=\mathbf{f}_j^{n}-\frac{\Delta t}{\Delta x}\Big(\hat{\mathbf{F}}_{j+1/2}^{n}-\hat{\mathbf{F}}_{j-1/2}^{n}\Big)$$

$$\hat{\mathbf{F}}_{j+1/2}^{n}=\mathbf{F}^{+}(\mathbf{f}_j^{n})+\mathbf{F}^{-}(\mathbf{f}_{j+1}^{n}).$$

Using that max(f,0)=0.5*(f+|f|) and min(f,0)=0.5*(f-|f|)

$$\hat{\mathbf{F}}_{j+1/2}^{n}=\mathbf{F}(\mathbf{f}_{j+1}^{n})+\mathbf{F}(\mathbf{f}_j^{n})-\frac{1}{2}\left(|u_{j+1}^{n}|\begin{bmatrix}\rho_{j+1}^{n}\\ \rho_{j+1}^{n}u_{j+1}^{n}\\ \rho_{j+1}^{n}e_{j+1}^{n}\end{bmatrix}-|u_j^{n}|\begin{bmatrix}\rho_j^{n}\\ \rho_j^{n}u_j^{n}\\ \rho_j^{n}e_j^{n}\end{bmatrix}\right)$$
$$-\frac{1}{2}\left(\begin{bmatrix}0\\ p_{j+1}^{n}M_{j+1}^{n}\\ p_{j+1}^{n}c_{j+1}^{n}\end{bmatrix}-\begin{bmatrix}0\\ p_j^{n}M_j^{n}\\ p_j^{n}c_j^{n}\end{bmatrix}\right)$$

Computational Fluid Dynamics
Summary

```
%upwind-Zha-Bilgen flux splitting

nx=40*256; tfinal=0.005; xl=10.0; time=0; gg=1.4;
h=xl/(nx-1);for i=1:nx,x(i)=h*(i-1);end
p_left=100000;p_right=10000;r_left=1;r_right=0.125;u_left=0;

r=zeros(1,nx);ru=zeros(1,nx);rE=zeros(1,nx);
p=zeros(1,nx);c=zeros(1,nx);u=zeros(1,nx);m=zeros(1,nx);
F1=zeros(1,nx);F2=zeros(1,nx);F3=zeros(1,nx);

for i=1:nx,   r(i)=r_right;ru(i)=0.0;rE(i)=p_right/(gg-1);end
for i=1:nx/2; r(i)=r_left; rE(i)=p_left/(gg-1); end

cmax=sqrt( max(gg*p_right/r_right,gg*p_left/r_left) );
dt=0.45*h/cmax; maxstep=tfinal/dt;
```

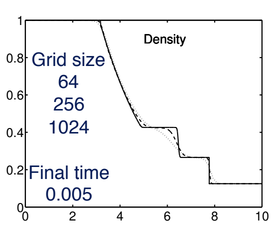
```
for istep=1:maxstep
  for i=1:nx,p(i)=(gg-1)*(rE(i)-0.5*(ru(i)*ru(i)/r(i)));end
  for i=1:nx,c(i)=sqrt( gg*p(i)/r(i) );end
  for i=1:nx,u(i)=ru(i)/r(i);end; for i=1:nx,m(i)=u(i)/c(i);end

  for i=1:nx-1
                      % Find fluxes
    F1(i)=0.5*(ru(i+1)+ru(i))-0.5*(abs(ru(i+1))-abs(ru(i)));

    F2(i)=0.5*(u(i+1)*ru(i+1)+p(i+1)+u(i)*ru(i)+p(i))...
        -0.5*(abs(u(i+1))*ru(i+1)-abs(u(i))*ru(i))...
        -0.5*(p(i+1)*m(i+1)-p(i)*m(i));

    F3(i)=0.5*(u(i+1)*(rE(i+1)+p(i+1))+u(i)*(rE(i)+p(i)))...
        -0.5*(abs(u(i+1))*rE(i+1)-abs(u(i))*rE(i))...
        -0.5*(p(i+1)*c(i+1)-p(i)*c(i));

    if m(i) > 1,  F2(i)=ru(i)*u(i)+p(i);
                       F3(i)=(rE(i)+p(i))*u(i);end
    if m(i) < -1, F2(i)=ru(i+1)*u(i+1)+p(i+1);
                    F3(i)=(rE(i+1)+p(i+1))*u(i+1);end
  end

  for i=2:nx-2
                      % Update solution
    r(i)=r(i)-(dt/h)*(F1(i)-F1(i-1));
    ru(i)=ru(i)-(dt/h)*(F2(i)-F2(i-1));
    rE(i)=rE(i)-(dt/h)*(F3(i)-F3(i-1));
  end

  time=time+dt,istep
end
```

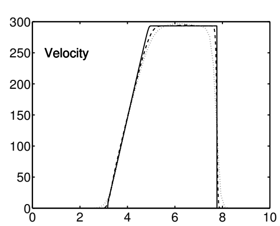

Computational Fluid Dynamics
The Euler Equations

Shock tube problem of G.A. Sod, JCP 27:1, 1978

$p_L=10^5;\quad \rho_L=1.0;\quad u_L=0$

$p_R=10^4;\quad \rho_R=0.125;\quad u_R=0$



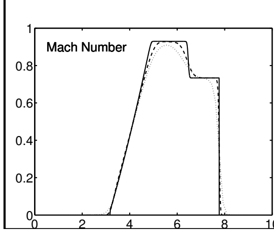


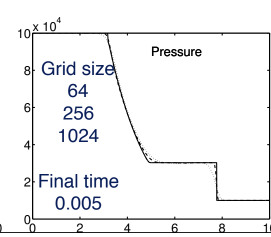
Computational Fluid Dynamics
The Euler Equations

Shock tube problem of G.A. Sod, JCP 27:1, 1978

$p_L=10^5;\quad \rho_L=1.0;\quad u_L=0$

$p_R=10^4;\quad \rho_R=0.125;\quad u_R=0$





Another example: one-dimensional Euler equation using the van Leer vector flux splitting

Van Leer

$$\mathbf{F}^{+}=\frac{\rho}{4c}(u+c)^2\begin{bmatrix}1\\ \dfrac{(\gamma-1)u+2c}{\gamma}\\ \dfrac{\left[(\gamma-1)u+2c\right]^2}{2(\gamma^2-1)}\end{bmatrix}\qquad \mathbf{F}^{-}=-\frac{\rho}{4c}(u-c)^2\begin{bmatrix}1\\ \dfrac{(\gamma-1)u-2c}{\gamma}\\ \dfrac{\left[2c-(\gamma-1)u\right]^2}{2(\gamma^2-1)}\end{bmatrix}$$

van Leer vector flux splitting

$$\begin{bmatrix}\rho u\\ \rho u^2+p\\ \rho u(E+\dfrac{p}{\rho})\end{bmatrix}=\frac{\rho}{4c}(u+c)^2\begin{bmatrix}1\\ \dfrac{(\gamma-1)u+2c}{\gamma}\\ \dfrac{\left[(\gamma-1)u+2c\right]^2}{2(\gamma^2-1)}\end{bmatrix}-\frac{\rho}{4c}(u-c)^2\begin{bmatrix}1\\ \dfrac{(\gamma-1)u-2c}{\gamma}\\ \dfrac{\left[2c-(\gamma-1)u\right]^2}{2(\gamma^2-1)}\end{bmatrix}$$

For example, the mass flux:

$$\frac{\rho}{4c}(u+c)^2-\frac{\rho}{4c}(u-c)^2$$

$$=\frac{\rho}{4c}\left(u^2+2uc+c^2-u^2+2uc-c^2\right)=\frac{\rho}{4c}4uc=\rho u$$

For 1D flow the fluxes are

$$\mathbf{F}^{\pm}=\pm\frac{\rho}{4c}(u\pm c)^2\begin{bmatrix}1\\ \dfrac{(\gamma-1)u\pm 2c}{\gamma}\\ \dfrac{\left[2c\pm(\gamma-1)u\right]^2}{2(\gamma^2-1)}\end{bmatrix}=\pm\frac{\rho c}{4}(M\pm 1)^2\begin{bmatrix}1\\ \dfrac{2c}{\gamma}\left(\dfrac{\gamma-1}{2}M\pm 1\right)\\ \dfrac{2c^2}{\gamma^2-1}\left(1\pm\dfrac{\gamma-1}{2}M\right)^2\end{bmatrix}$$

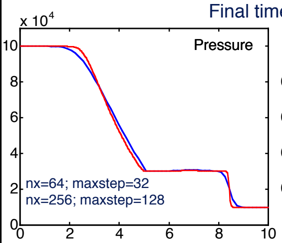

```
for istep=1:maxstep
  for i=1:nx,c(i)=sqrt( gg*(gg-1)*(rE(i)-0.5*(ru(i)^2/r(i)))/r(i) );end
 for i=1:nx,u(i)=ru(i)/r(i);end; for i=1:nx,m(i)=u(i)/c(i);end

for i=2:nx-1                                                    %upwind
 rn(i)=r(i)-(dt/h)*(...
 (0.25*r(i)*c(i)*(m(i)+1)^2) - (0.25*r(i-1)*c(i-1)*(m(i-1)+1)^2)+...
 (-0.25*r(i+1)*c(i+1)*(m(i+1)-1)^2) - (-0.25*r(i)*c(i)*(m(i)-1)^2)   );

 run(i)=ru(i)-(dt/h)*(...
 ( 0.25*r(i)*c(i)   *(m(i)+1)^2)  *(( 1+0.5*(gg-1)*m(i))  *2*c(i)  /gg) - ...
 ( 0.25*r(i-1)*c(i-1)*(m(i-1)+1)^2)*(( 1+0.5*(gg-1)*m(i-1))*2*c(i-1)/gg) + ...
 (-0.25*r(i+1)*c(i+1)*(m(i+1)-1)^2)*((-1+0.5*(gg-1)*m(i+1))*2*c(i+1)/gg) - ...
 (-0.25*r(i)*c(i)   *(m(i)-1)^2)  *((-1+0.5*(gg-1)*m(i))  *2*c(i)  /gg)   );

 rEn(i)=rE(i)-(dt/h)*(...
 ( 0.25*r(i)*c(i)   *(m(i)+1)^2)  *((1+0.5*(gg-1)*m(i))^2  *2*c(i)^2  /(gg^2-1)) - ...
 ( 0.25*r(i-1)*c(i-1)*(m(i-1)+1)^2)*((1+0.5*(gg-1)*m(i-1))^2*2*c(i-1)^2/(gg^2-1)) + ...
 (-0.25*r(i+1)*c(i+1)*(m(i+1)-1)^2)*((1-0.5*(gg-1)*m(i+1))^2*2*c(i+1)^2/(gg^2-1)) - ...
 (-0.25*r(i)*c(i)   *(m(i)-1)^2)  *((1-0.5*(gg-1)*m(i))^2  *2*c(i)^2  /(gg^2-1))    );
end
```

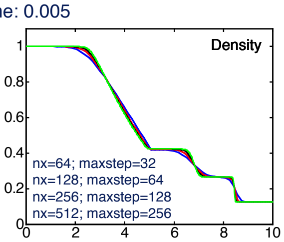

Effect of resolution

Shocktube problem of G.A. Sod, JCP 27:1, 1978

$$p_L=10^5;\quad \rho_L=1.0;\quad u_L=0$$
$$p_R=10^4;\quad \rho_R=0.125;\quad u_R=0$$

Final time: 0.005





For 2D flow the van Leer fluxes are

$$\mathbf{F}^{\pm}=\pm\frac{\rho}{4c}(u\pm c)^2\begin{bmatrix}1\\ \dfrac{(\gamma-1)u\pm 2c}{\gamma}\\ v\\ \dfrac{v^2}{2}+\dfrac{\left[(\gamma-1)u\pm 2c\right]^2}{2(\gamma^2-1)}\end{bmatrix};\quad \mathbf{G}^{\pm}=\pm\frac{\rho}{4c}(v\pm c)^2\begin{bmatrix}1\\ u\\ \dfrac{(\gamma-1)v\pm 2c}{\gamma}\\ \dfrac{u^2}{2}+\dfrac{\left[(\gamma-1)v\pm 2c\right]^2}{2(\gamma^2-1)}\end{bmatrix}$$

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}^{+}}{\partial x}+\frac{\partial \mathbf{F}^{-}}{\partial x}+\frac{\partial \mathbf{G}^{+}}{\partial y}+\frac{\partial \mathbf{G}^{-}}{\partial y}=0$$

Several other splitting schemes are possible, such as:

Steger-Warming

$$\mathbf{F}^{+}=\frac{\rho}{2\gamma}\begin{bmatrix}(2\gamma-1)u+c\\ 2(\gamma-1)u^{2}+(u+c)^{2}\\ (\gamma-1)u^{3}+\frac{1}{2}(u+c)^{3}+\frac{3-\gamma}{2(\gamma-1)}(u+c)c^{2}\end{bmatrix}$$

$$\mathbf{F}^{-}=\frac{1}{2}\frac{\rho}{\gamma}(u-c)\begin{bmatrix}1\\ u-c\\ \frac{1}{2}(u-c)^{2}+\frac{1}{2}c^{2}\left(\frac{3-\gamma}{\gamma-1}\right)\end{bmatrix}$$

Rewrite the flux terms in terms of Mach number:

$$\mathbf{F}^{+}=\left(\frac{\rho}{4c}(u+c)^{2}\begin{pmatrix}1\\ \frac{(\gamma-1)u+2c}{\gamma}\\ \frac{\left[(\gamma-1)u+2c\right]^{2}}{2(\gamma^{2}-1)}\end{pmatrix}\right)=\left(\frac{\rho c}{4}\left(\frac{u}{c}+1\right)^{2}\begin{pmatrix}1\\ \frac{2c}{\gamma}\left(1+\frac{\gamma-1}{2}\frac{u}{c}\right)\\ \frac{2c^{2}}{\gamma^{2}-1}\left(1+\frac{\gamma-1}{2}\frac{u}{c}\right)^{2}\end{pmatrix}\right)$$

$$=\left(\frac{\rho c}{4}(M+1)^{2}\begin{pmatrix}1\\ \frac{2c}{\gamma}\left(1+\frac{\gamma-1}{2}M\right)\\ \frac{2c^{2}}{\gamma^{2}-1}\left(1+\frac{\gamma-1}{2}M\right)^{2}\end{pmatrix}\right)$$

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}^{+}}{\partial x}+\frac{\partial \mathbf{F}^{-}}{\partial x}=0$$

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho E\end{pmatrix}+\frac{\partial}{\partial x}\left(\frac{\rho c}{4}(M+1)^{2}\begin{pmatrix}1\\ \frac{2c}{\gamma}\left(1+\frac{\gamma-1}{2}M\right)\\ \frac{2c^{2}}{\gamma^{2}-1}\left(1+\frac{\gamma-1}{2}M\right)^{2}\end{pmatrix}\right)+\frac{\partial}{\partial x}\left(-\frac{\rho c}{4}(M-1)^{2}\begin{pmatrix}1\\ \frac{2c}{\gamma}\left(-1+\frac{\gamma-1}{2}M\right)\\ \frac{2c^{2}}{\gamma^{2}-1}\left(1-\frac{\gamma-1}{2}M\right)^{2}\end{pmatrix}\right)=0$$

$$\frac{\partial \mathbf{f}}{\partial t}+\frac{\partial \mathbf{F}^{+}}{\partial x}+\frac{\partial \mathbf{F}^{-}}{\partial x}=0$$

Solve by

$$\mathbf{f}_{j}^{n+1}=\mathbf{f}_{j}^{n}-\left(\frac{\Delta t}{h}\right)\left(\mathbf{F}_{j}^{+}-\mathbf{F}_{j-1}^{+}\right)^{n}-\left(\frac{\Delta t}{h}\right)\left(\mathbf{F}_{j+1}^{-}-\mathbf{F}_{j}^{-}\right)^{n}$$

Where

$$\mathbf{F}^{+}=\frac{\rho c}{4}(M+1)^{2}\begin{pmatrix}1\\ \frac{2c}{\gamma}\left(1+\frac{\gamma-1}{2}M\right)\\ \frac{2c^{2}}{\gamma^{2}-1}\left(1+\frac{\gamma-1}{2}M\right)^{2}\end{pmatrix};\quad \mathbf{F}^{-}=-\frac{\rho c}{4}(M-1)^{2}\begin{pmatrix}1\\ \frac{2c}{\gamma}\left(-1+\frac{\gamma-1}{2}M\right)\\ \frac{2c^{2}}{\gamma^{2}-1}\left(1-\frac{\gamma-1}{2}M\right)^{2}\end{pmatrix}$$

WENO-3

Third order WENO

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

The semi-discrete equation is

$$\frac{d\mathbf{f}_{j}}{dt}+\frac{1}{\Delta x}\left(\mathbf{F}_{j+1/2}-\mathbf{F}_{j-1/2}\right)=0$$

The fluxes are the weighted sum

$$\mathbf{F}_{j+1/2}=\omega_{1}\mathbf{F}_{j+1/2}^{(1)}+\omega_{2}\mathbf{F}_{j+1/2}^{(2)}$$

where

$$\mathbf{F}_{j+1/2}^{(1)}=-\frac{1}{2}\mathbf{F}_{j-1}+\frac{3}{2}\mathbf{F}_{j}$$ 2nd order upwind

$$\mathbf{F}_{j+1/2}^{(2)}=\frac{1}{2}\mathbf{F}_{j}+\frac{1}{2}\mathbf{F}_{j+1}$$ 2nd order central

The weights are

$$\tilde{\omega}_{l}=\frac{\gamma_{l}}{\left(\varepsilon+\beta_{l}\right)^{2}}$$

$$\omega_{m}=\frac{\tilde{\omega}_{m}}{\sum_{l=1}^{2}\tilde{\omega}_{l}}$$

$$\beta_{1}=\left(f_{j}-f_{j-1}\right)^{2}$$

$$\beta_{2}=\left(f_{j+1}-f_{j}\right)^{2}$$

$$\gamma_{1}=\frac{1}{3};\quad \gamma_{2}=\frac{2}{3}$$

$$\varepsilon=10^{-6}$$

Computational Fluid Dynamics

$$\frac{d\mathbf{f}_j}{dt} + \frac{1}{\Delta x}\left(\mathbf{F}_{j+1/2} - \mathbf{F}_{j-1/2}\right) = 0$$

The time integration is done by a third order Runga-Kutta

$$f_j^{(1)} = f_j^n + \Delta t L\left(f^n, t^n\right)$$

$$f_j^{(2)} = \frac{3}{4} f_j^n + \frac{1}{4} f_j^{(1)} + \frac{1}{4}\Delta t L\left(f^{(1)}, t^n + \Delta t\right)$$

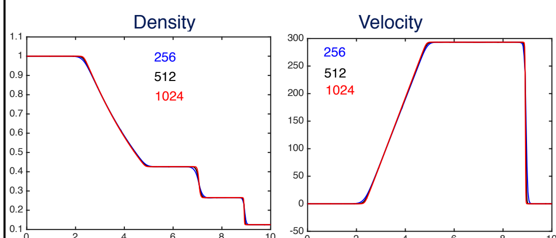
$$f_j^{n+1} = \frac{1}{3} f_j^n + \frac{2}{3} f_j^{(2)} + \frac{2}{3}\Delta t L\left(f^{(2)}, t^n + \frac{1}{2}\Delta t\right)$$

where $L(f,t) = -\dfrac{\partial F}{\partial x}$

Computational Fluid Dynamics
http://www.nd.edu/~gtryggva/CFD-Course/

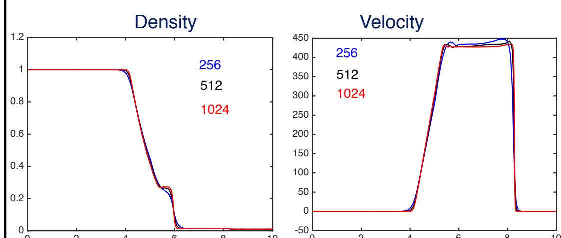
```
% one-dimensional Euler by WENO3-RK
% Third order scheme--Lax-Friedrich fluxes---compact scheme-3 times Euler
% Nothing is done to account for boundaries
%----------------------------------------------------------

nx=256; maxstep=150;

gg=1.4;p_left=100000;p_right=10000;r_left=1;r_right=0.125;u_left=0;
xl=10.0;h=xl/(nx-1);time=0; for i=1:nx,x(i)=h*(i-1);end

gam1=1/3;gam2=2/3; eps=0.000001;

r=zeros(1,nx);ru=zeros(1,nx);rE=zeros(1,nx);
rL=zeros(1,nx);ruL=zeros(1,nx);rEL=zeros(1,nx);
rR=zeros(1,nx);ruR=zeros(1,nx);rER=zeros(1,nx);
F1=zeros(1,nx); F2=zeros(1,nx); F3=zeros(1,nx);

for i=1:nx,r(i)=r_right;ru(i)=0.0;rE(i)=p_right/(gg-1);end
for i=1:nx/2; r(i)=r_left;
         rE(i)=p_left/(gg-1)+0.5*r_left*u_left^2;
         ru(i)=r_left*u_left;end

% These are set here to provide default values for the endpoints
for j=1:nx, rL(j)=r(j);ruL(j)=ru(j);rEL(j)=rE(j);
        rR(j)=r(j);ruR(j)=ru(j);rER(j)=rE(j); end

cmax=sqrt( max(gg*p_right/r_right,gg*p_left/r_left) ); dt=0.45*h/cmax;

plot(x,r,'r','linewidth',2); pause, % Plot initial conditions

for istep=1:maxstep

 rn=r; run=ru;rEn=rE;
 for  substep=1:3                    % third order

 % ****** WENO3 ******
 for j=3:nx-2,
  % LEFT values
  beta1=(r(j)-r(j-1))^2;  beta2=(r(j+1)-r(j))^2;
  omt1=(gam1)/(eps+beta1)^2; omt2=(gam2)/(eps+beta2)^2;
  omtsum=omt1+omt2; om1=omt1/omtsum; om2=omt2/omtsum;
  rL(j)=om1*(-0.5*r(j-1)+1.5*r(j))+om2*(0.5*r(j)+0.5*r(j+1));

  beta1=(ru(j)-ru(j-1))^2;  beta2=(ru(j+1)-ru(j))^2;
  omt1=(gam1)/(eps+beta1)^2; omt2=(gam2)/(eps+beta2)^2;
  omtsum=omt1+omt2; om1=omt1/omtsum; om2=omt2/omtsum;
  ruL(j)=om1*(-0.5*ru(j-1)+1.5*ru(j))+om2*(0.5*ru(j)+0.5*ru(j+1));

  beta1=(rE(j)-rE(j-1))^2;  beta2=(rE(j+1)-rE(j))^2;
  omt1=(gam1)/(eps+beta1)^2; omt2=(gam2)/(eps+beta2)^2;
  omtsum=omt1+omt2; om1=omt1/omtsum; om2=omt2/omtsum;
  rEL(j)=om1*(-0.5*rE(j-1)+1.5*rE(j))+om2*(0.5*rE(j)+0.5*rE(j+1));

 % RIGHT values
  beta1=(r(j+1)-r(j+2))^2;  beta2=(r(j)-r(j+1))^2;
  omt1=(gam1)/(eps+beta1)^2; omt2=(gam2)/(eps+beta2)^2;
  omtsum=omt1+omt2; om1=omt1/omtsum; om2=omt2/omtsum;
  rR(j)=om1*(-0.5*r(j+2)+1.5*r(j+1))+om2*(0.5*r(j+1)+0.5*r(j));

  beta1=(ru(j+1)-ru(j+2))^2;  beta2=(ru(j)-ru(j+1))^2;
  omt1=(gam1)/(eps+beta1)^2; omt2=(gam2)/(eps+beta2)^2;
  omtsum=omt1+omt2; om1=omt1/omtsum; om2=omt2/omtsum;
  ruR(j)=om1*(-0.5*ru(j+2)+1.5*ru(j+1))+om2*(0.5*ru(j+1)+0.5*ru(j));

  beta1=(rE(j+1)-rE(j+2))^2;  beta2=(rE(j)-rE(j+1))^2;
  omt1=(gam1)/(eps+beta1)^2; omt2=(gam2)/(eps+beta2)^2;
  omtsum=omt1+omt2; om1=omt1/omtsum; om2=omt2/omtsum;
  rER(j)=om1*(-0.5*rE(j+2)+1.5*rE(j+1))+om2*(0.5*rE(j+1)+0.5*rE(j));
 end

 for j=1:nx-1,                          % find the fluxes
  pL=(gg-1)*(rEL(j)-0.5*(ruL(j)*ruL(j)/rL(j)));
  pR=(gg-1)*(rER(j)-0.5*(ruR(j)*ruR(j)/rR(j)));
  F1(j)=0.5*(ruR(j)+ruL(j))-0.5*(h/dt)*(rR(j)-rL(j));
  F2(j)=0.5*((ruR(j)^2/rR(j))+pR+(ruL(j)^2/rL(j))+pL)-...
                        0.5*(h/dt)*(ruR(j)-ruL(j));
  F3(j)=0.5*((ruR(j)/rR(j))*(rER(j)+pR)+(ruL(j)/rL(j))*(rEL(j)+pL))-...
                        0.5*(h/dt)*(rER(j)-rEL(j));
 end;
 % **** End WENO3 ****

 for j=3:nx-2,
  r(j)=r(j)-(dt/h)*(F1(j)-F1(j-1));
  ru(j)=ru(j)-(dt/h)*(F2(j)-F2(j-1));
  rE(j)=rE(j)-(dt/h)*(F3(j)-F3(j-1));
 end;

  if substep==2,
   r=0.75*rn+0.25*r; ru=0.75*run+0.25*ru; rE=0.75*rEn+0.25*rE;          % RK
  elseif substep==3
   r=0.33334*rn+0.66666*r; ru=0.33334*run+0.66666*ru; rE=0.33334*rEn+0.66666*rE;    % RK
  end

 end % End of sub-iteration for RK-3 time integration

 plot(x,r,'r','linewidth',2); pause(0.001)
 time=time+dt; %istep
end

% plot(x,r,'k','linewidth',2);
```

Computational Fluid Dynamics
http://www.nd.edu/~gtryggva/CFD-Course/



Subsonic Case

Computational Fluid Dynamics
http://www.nd.edu/~gtryggva/CFD-Course/



Supersonic Case

Computational Fluid Dynamics
The Euler Equations

In general we solve:

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

by

$$\mathbf{f}_j^{n+1} = \mathbf{f}_j^n - \frac{\Delta t}{h}\left(\mathbf{F}_{j+1/2}^n - \mathbf{F}_{j-1/2}^n\right)$$

Where the fluxes are found in a variety of ways

$$F_{j+1/2} = F\left(\left(f^L\right)_{j+1/2}, \left(f^R\right)_{j+1/2}\right)$$

In principle we can solve this problem, the Riemann problem, exactly by assuming constant states and then integrate the fluxes over the time step.

Computational Fluid Dynamics
The Euler Equations

Predictor-corrector method: Limiting the variables

Predictor step $\quad f_j^{n+1/2} = f_j^n - \dfrac{\Delta t}{2h}\left(F_{j+1/2}^n - F_{j-1/2}^n\right)$

Variables $\quad f_{j+1/2}^L = f_j^{n+1/2} + \dfrac{1}{2}\Psi^L\left(f_j^{n+1/2} - f_{j-1}^{n+1/2}\right)$

$$f_{j+1/2}^R = f_{j+1}^{n+1/2} - \frac{1}{2}\Psi^R\left(f_{j+1}^{n+1/2} - f_j^{n+1/2}\right)$$

Find: $\quad F_{j+1/2}^{n+1/2} = F\left(\left(f^L\right)_{j+1/2}^{n+1/2}, \left(f^R\right)_{j+1/2}^{n+1/2}\right)$

Final step $\quad f_j^{n+1} = f_j^n - \dfrac{\Delta t}{h}\left(F_{j+1/2}^{n+1/2} - F_{j-1/2}^{n+1/2}\right)$

The Riemann problems for a linear hyperbolic system

The Riemann problem for a system of linear equations with constant coefficients. We start with:

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

The eigenvalues and eigenvectors of A are given by

$$\mathbf{A}\mathbf{e}_p = \lambda_p \mathbf{e}_p$$

If we normalize the eigenvectors to unity $|\mathbf{e}_p| = 1$ then the eigenvectors form an orthonormal basis and we can write

$$\mathbf{f} = \sum_{p=1}^{3} w_p \mathbf{e}_p \qquad\qquad w_p = \mathbf{f}\cdot\mathbf{e}_p$$

Substituting for $\mathbf{f}$ in the original equation, we get

$$\sum_{p=1}^{3}\frac{\partial w_p}{\partial t}\mathbf{e}_p + \sum_{p=1}^{3}\frac{\partial w_p}{\partial x}\mathbf{A}\mathbf{e}_p = 0$$

Or, using the definition of the eigenvalue and the fact that the eigenvectors are orthonormal, we have three sets of independent equations

$$\mathbf{A}\mathbf{e}_p = \lambda_p \mathbf{e}_p$$

$$\frac{\partial w_p}{\partial t} + \lambda_p \frac{\partial w_p}{\partial x} = 0$$

Whose solutions are given by

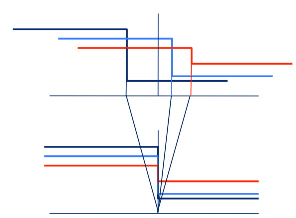
$$w_p(x,t) = w_{p0}(x - \lambda_p t)$$

Representing a simple translation of the initial conditions

For discontinuous initial conditions, where

$$\mathbf{w} = \mathbf{w}^L, \quad x < 0 \quad \& \quad \mathbf{w} = \mathbf{w}^R, \quad x > 0; \quad \mathbf{w} = \left(w_1, w_2, w_3\right)$$

The different components of the solution remain piecewise constant but the discontinuity moves left or right, depending on the sign of the eigenvalues.



Since the solution is piecewise constant the fluxes are easily found. In particular, for each of the linear equations the fluxes are:

$$w_p\lambda_p = \begin{cases} w_p^L \lambda_p \text{ if } \lambda_p > 0 \\ w_p^R \lambda_p \text{ if } \lambda_p < 0 \end{cases} = \frac{1}{2}\left\{\left(\lambda_p + |\lambda_p|\right)w_p^L + \left(\lambda_p - |\lambda_p|\right)w_p^R\right\}$$

$$= \frac{1}{2}\left\{\lambda_p\left(w_p^L + w_p^R\right) - |\lambda_p|\left(w_p^R - w_p^L\right)\right\}$$

Or, for the system

$$\mathbf{F}_{j+1/2} = \sum_p \frac{1}{2}\left\{\lambda_p\left(w_p^L + w_p^R\right) - |\lambda_p|\left(w_p^R - w_p^L\right)\right\}\mathbf{e}_p$$

For the Euler equations the situation is more complex. Start with the primitive form

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{C}\frac{\partial \mathbf{w}}{\partial x} = 0$$

$$\mathbf{C} = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho c^2 & u \end{pmatrix} \qquad\qquad \mathbf{w} = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}$$

We can find the eigenvectors in standard ways and use them to make the matric diagonal

Computational Fluid Dynamics
The Euler Equations

The eigenvector of

$$\mathbf{C} = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho c^2 & u \end{pmatrix}$$

are (using arbitrary scaling

Right eigenvectors

$$\mathbf{C}\mathbf{r}_p = \lambda_p \mathbf{r}_p$$

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{r}_2 = \begin{pmatrix} \frac{\rho}{2c} \\ \frac{1}{2} \\ \frac{\rho c}{2} \end{pmatrix}; \quad \mathbf{r}_3 = \begin{pmatrix} -\frac{\rho}{2c} \\ \frac{1}{2} \\ -\frac{\rho c}{2} \end{pmatrix}$$

Left eigenvectors

$$\mathbf{C}^T \mathbf{l}_p = \lambda_p \mathbf{l}_p$$

$$\mathbf{l}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}; \quad \mathbf{l}_2 = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}; \quad \mathbf{l}_3 = \begin{pmatrix} -\frac{1}{c^2} \\ \frac{1}{\rho c} \\ -\frac{1}{\rho c} \end{pmatrix}$$

Computational Fluid Dynamics
The Euler Equations

Form the matrices

$$Q_C = \begin{pmatrix} 1 & \frac{\rho}{2c} & -\frac{\rho}{2c} \\ 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{\rho c}{2} & -\frac{\rho c}{2} \end{pmatrix} \qquad Q_C^{-1} = \begin{pmatrix} 1 & 0 & -\frac{1}{c^2} \\ 0 & 1 & \frac{1}{\rho c} \\ 0 & 1 & -\frac{1}{\rho c} \end{pmatrix} \qquad \mathbf{w} = \begin{pmatrix} \rho \\ u \\ p \end{pmatrix}$$

Then

$$\frac{\partial \mathbf{w}}{\partial t} + \mathbf{C}\frac{\partial \mathbf{w}}{\partial x} = 0$$

$$Q_C^{-1}\mathbf{C}Q_C = \begin{pmatrix} u & 0 & 0 \\ 0 & u+c & 0 \\ 0 & 0 & u-c \end{pmatrix} \qquad \mathbf{C} = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 1/\rho \\ 0 & \rho c^2 & u \end{pmatrix}$$

Computational Fluid Dynamics
The Euler Equations

Multiply $\dfrac{\partial \mathbf{w}}{\partial t} + \mathbf{C}\dfrac{\partial \mathbf{w}}{\partial x} = 0$

To get the characteristic form

$$Q_C^{-1}\frac{\partial \mathbf{w}}{\partial t} + Q_C^{-1}\mathbf{C}\frac{\partial \mathbf{w}}{\partial x} = 0$$

Which can be written as

$$\frac{\partial \mathbf{v}}{\partial t} + \Lambda\frac{\partial \mathbf{v}}{\partial x} = 0$$

$$\Lambda = Q_C^{-1}\mathbf{C}Q_C = \begin{pmatrix} u & 0 & 0 \\ 0 & u+c & 0 \\ 0 & 0 & u-c \end{pmatrix}$$

or

$$\frac{\partial v_0}{\partial t} + u\frac{\partial v_0}{\partial x} = 0$$

$$\frac{\partial v_+}{\partial t} + (u+c)\frac{\partial v_+}{\partial x} = 0$$

$$\frac{\partial v_-}{\partial t} + (u-c)\frac{\partial v_-}{\partial x} = 0$$

Computational Fluid Dynamics
The Euler Equations

Although the Riemann problem can be solved for the general case when the fluids are not stationary, the solution is expensive (involving solving a nonlinear equation for the pressure ratio across the shock) and the only information that we need from the solution is the flux across the cell boundary.

Therefore, usually we use approximate Riemann solvers. Different approximations are possible but generally the result is a complex process. Here we will only outline it briefly