AE504 - Compressible Flow

Vittorio Baraldi

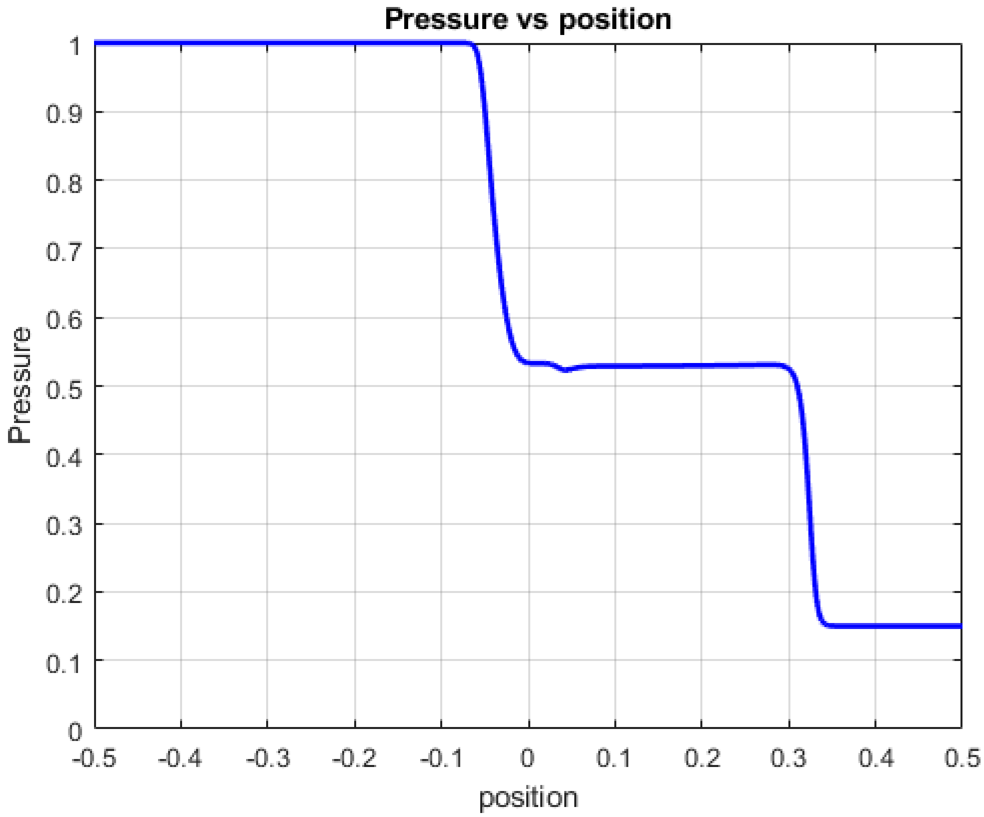
```
clc;clear all;close all;
```

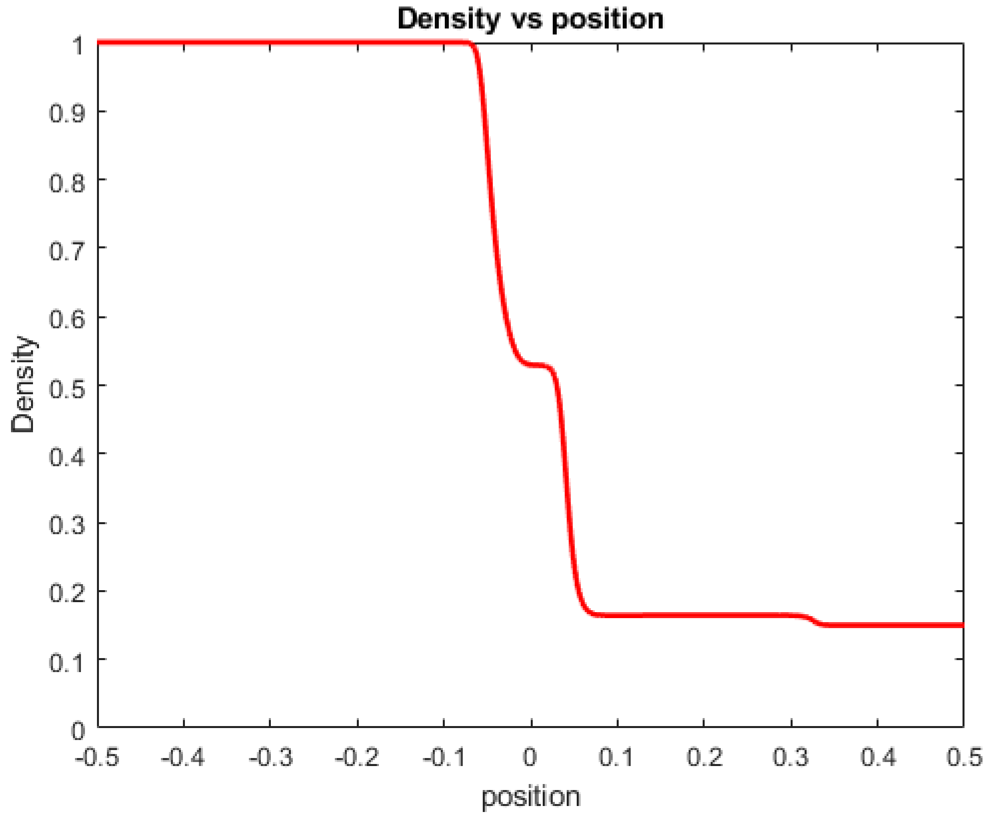
Ex. 2

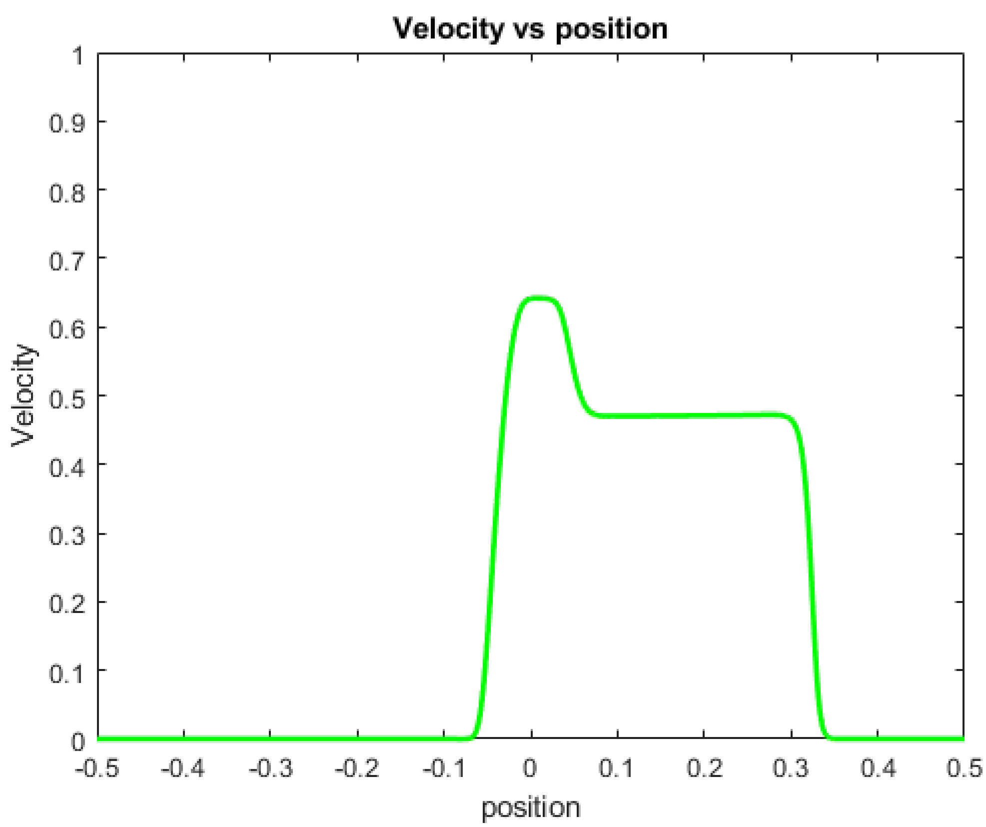
```
%Shock tube problem using flux vector splitting (Steger-Warming)
g=0;
gamma=1.4;
x_min=-0.5;
x_max=0.5;
N=500;
dx=(x_max-x_min)/N;
xplot=linspace(x_min,x_max,N);
for i =  1:N                      % Initialization at t(time)=0
    if i<=N/2
        rho(i) = 1.0;    %density
        P(i) = 1.0;      %pressure
        u(i) = 0.0;      %velocity
    else
        rho(i) = 0.15;
        P(i) = 0.15;
        u(i) = 0.0;

    end
end
 E = P/(gamma-1)+0.5*rho.*u.^2; %energy
 U2 = rho.*u;
 U3 = rho.*E;
t_0=0;
t = 30;
dt  = 0.001;
while t_0<t

        a = (gamma*P./rho).^0.5;   % speed of sound

        lambda1 = u;                 %first eigen value
        lambda2 = u + a;                     %second eigen value
        lambda3 = u - a;                 %third eigen value

        lambda1pos = 0.5*(lambda1+abs(lambda1));
        lambda2pos = 0.5*(lambda2+abs(lambda2));
        lambda3pos = 0.5*(lambda3+abs(lambda3));
        lambda1neg = 0.5*(lambda1-abs(lambda1));
        lambda2neg = 0.5*(lambda2-abs(lambda2));
        lambda3neg = 0.5*(lambda3-abs(lambda3));

        H = .5*u.^2+a.^2/(gamma-1);         % H = (energy+pressure)/
density --> entalpy
```

```
        FP1 = rho*0.5/gamma.*(2*gamma.*u+a-u);  %first element of flux
 matrix (positive)
        FP2 = rho.*0.5/gamma.*(2*(gamma-1).*u.^2+(u+a).^2);    %Second
 element of flux matrix
        FP3 = rho.*0.5/gamma.*((gamma-1).*u.^3+((u+a).^3)/2+((3-
gamma).*(u+a).*a.^2)./(2*(gamma-1))); %third element of flux matrix

        FN1 = rho.*0.5/gamma.*(u-a); %first element of flux matrix
 (negative)
        FN2 = rho.*0.5/gamma.*((u-a).^2); %second element
        FN3 = rho.*0.5/gamma.*(((u-a).^3)/2+((3-gamma).*(u-a).*a.^2)./
(2*(gamma-1))); %third element

        for i = 1:N-1                     %intercell numerical flux
            Fhp1(i) = FP1(i)+FN1(i+1);
            Fhn1(i+1) = FP1(i)+FN1(i+1);
            Fhp2(i) = FP2(i)+FN2(i+1);
            Fhn2(i+1) = FP2(i)+FN2(i+1);
            Fhp3(i) = FP3(i)+FN3(i+1);
            Fhn3(i+1) = FP3(i)+FN3(i+1);
        end

       for i=2:N-1
        rhon(i) = rho(i)-dt*(Fhp1(i)-Fhn1(i));       % Density at t =
 t+dt
        U2(i) = rho(i).*u(i)-dt*(Fhp2(i)-Fhn2(i));    % U2 at t=t+dt
        U3(i) = rho(i).*E(i)-dt*(Fhp3(i)-Fhn3(i));    % U3 at t = t+dt
       end

      % Boundary Conditions
       rhon(N) = rhon(N-1);
       U2(N) = U2(N-1);
       U3(N) = U3(N-1);
       rhon(1) = rhon(2);
       U2(1) = U2(2);
       U3(1) = U3(2);

       u = U2./rhon;                        % velocity at t+dt
       E  = U3./rhon;                       % energy at t+dt
       P = (gamma-1)*(E-0.5*rhon.*u.^2);   %pressure at t+dt

       rho = rhon; % new density
       t_0 = t_0+dt;          % time increment
end
%outputs
figure(1)
plot(xplot,P,'-b','LineWidth',2)
title('Pressure vs position')
xlabel('position')
ylabel('Pressure')
grid on
xlim([-0.5 0.5])
ylim([0 1])
```

```
figure(2)
plot(xplot,rho,'-r','LineWidth',2)
title('Density vs position')
xlabel('position')
ylabel('Density')
xlim([-0.5 0.5])
ylim([0 1])
figure(3)
plot(xplot,u,'-g','LineWidth',2)
title('Velocity vs position')
xlabel('position')
ylabel('Velocity')
xlim([-0.5 0.5])
ylim([0 1])
```




Density vs position
Density
position
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5


Velocity vs position
Velocity
position
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5

Published with MATLAB® R2020a