```
function analytical = shock(dt)
  global rho_vector
  global p_vector
  global velocity_vector
  global dtnew

  dtnew = dt;
  % define variables
  p4 = 2.0; %pressure
  p1 = 1.0; %pressure
  rho4 = 2.0; %density
  rho1 = 1.0;%density
  gamma = 1.4;
  l = 2.0;
  a1 = sqrt(gamma*p1/rho1);
  a4 = sqrt(gamma*p4/rho4);

  A = (gamma-1)*(a1/a4);
  B = 2*gamma;
  C = gamma +1;
  D = -(2*gamma)/(gamma-1);


  tolerance = 1e-8;
  maxiter = 100000;
  error = 1;
  x = 0.5*(p4/p1); %ratio p2/p1
  n = 0;

  % Iterating to get P2/P1
  while error>=tolerance&n<=maxiter


      top = A*B*C*(x-1);
      bottom = 2*(B*(B+C*(x-1)))^(3/2);
      firstpart = top/bottom;
      secondpart = (A)/(sqrt(B*(B+(C*(x-1)))));
      thirdpart = (1-(A*(x-1))/(sqrt(B*(B+(C*(x-1)))))).^(D-1);
      forthpart = x*(1-(A*(x-1))/(sqrt(B*(B+(C*(x-1)))))).^(D);
      fprime = D*x*(firstpart-secondpart)*(thirdpart)+forthpart;
      f = x*(1-(A*(x-1))/(sqrt(B*(B+(C*(x-1)))))).^(D);
      g = p4/p1;
      y = x-(f-g)/(fprime);
      error = abs(y-x);
      x = y;
      n = n+1;
  end

  p2byp1 = x;

  %solve for P2
```

```
p2 = p1*x;
p3 = p2;
g = (gamma*p2)/(p2-p1)-(gamma-1)/(2);
w = sqrt(((p2-p1)*(g))/(rho1));
rho2 = ((g)/(g-1))*rho1;
u2 = w/g;
u3 = u2;
u4 = 0;
u1 = 0;
rho3 = rho4*(p3/p4)^(1./gamma);
a3 = sqrt(gamma*p3/rho3);
a2 = sqrt(gamma*p2/rho2);
x0 = 1;
t0 = 0;
maxTimeStep = 17;
dt = 0.1;
tmax = dt*maxTimeStep;

%SET UP ARRAYS
xmin = 0.0;
xmax = 2.0;
imax = 41;
dx = (xmax-xmin)/(imax-1);
x_vector_dumb = xmin:dx:xmax;
x_vector = x_vector_dumb.';
p_vector = zeros(imax,1);
rho_vector = zeros(imax,1);
v_expansion = zeros(imax,1);
velocity_vector = zeros(imax,1);

%INITAL CONDITION

p_vector(1:21) = p4;
p_vector(21:41) = p1;
rho_vector(1:21) = rho4;
rho_vector(21:41) = rho1;
t0 = 0.;
x0 = 1.;
for t = 0:dtnew:18*dtnew


    x1 = x0+w*(t-t0);
    x2 = x0+(u2)*(t-t0);
    x3 = x0+(u3-a3)*(t-t0);
    x4 = x0-(a4)*(t-t0);


    %PUTTING VALUES IN
    for index = 1: imax

        if x_vector(index) <= x4
            rho_vector(index) = rho4;
```

```
              p_vector(index) = p4;
              velocity_vector(index) = 0.0;
          elseif (x_vector(index)<=x3)
              v_expansion(index) = (2./(gamma+1.)).*(a4+(x_vector(index)-x0)/(…
              p_vector(index) = p4.*(1.- ((gamma-1.)./(2.)).*(abs(v_expansion(…
              rho_vector(index) = rho4.*(1.- ((gamma-1.)./(2.)).*(abs(v_expans…
%               rho_vector(index) = rho4.*(p_vector(index)/p4).^(1./gamma);
              velocity_vector(index) = v_expansion(index);

          elseif (x_vector(index)<=x2)
              rho_vector(index) = rho3;
              p_vector(index) = p2;
              velocity_vector(index) = u2;
          elseif (x_vector(index)<=x1)
              rho_vector(index) = rho2;
              p_vector(index) = p2;
              velocity_vector(index) = u2;
          else
              rho_vector(index) = rho1;
              p_vector(index) = p1;
              velocity_vector(index) = u1;
          end
      end
      t = t+dtnew;
% counter = counter +1;
%   end
end
```