INDUSTRIAL MATHEMATICS (MATH 6514)
INSTRUCTOR: DR. MARTIN SHORT

# Analyzing Shocks in a Standard Traffic Flow Problem

December 12, 2023

Marie Bertsche
Johnathan Corbin
Emmanuel Lyngberg

# Introduction

When we consider cars on the road, we think of them as discrete bodies that interact with one another because we think of them as autonomous and independent in their decision making. However, when we think of traffic as a concept, we think of a flow with a certain density because the tight, slow moving of the cars takes away that feeling of independence. However, regardless of how busy the road is, we can model the flow of traffic in this continuous density manner. For simplicity, we consider two dimensions: a singular axis for movement $x$ and a time dimension $t$. We begin by defining the density of traffic at a particular $x$ and time $t$ as $\rho(x,t)$. Additionally, we consider the velocity of the traffic at $x$ with time $t$ as $u(x,t)$. However, in our case, again to simplify, we will consider the velocity as a function of the density. This is not entirely unreasonable as we know that the speed of cars is relative to how many there are around them, and thus for denser areas of a road we would expect the average velocity of the cars to be lower. Thus, we define the velocity as $u(\rho) = u_{max}(1 - \frac{\rho}{\rho_{max}})$. This definition requires that we define some maximum velocity and density, which is again not unreasonable as we set speed limits on roads for safety, and the density of cars can only grow to a point where all cars are touching one another back to back. For computations we choose $u_{max} = 1$ and $\rho_{max} = 10$, but these can be amended to fit whatever particular road if interest. Together, we use the previous two definitions to define the flux as $q = \rho \cdot u(\rho)$, which is the product of the density and velocity. We can now define the fundamental partial differential equation that is common amongst these standard traffic flow problems, and that is

$$\frac{\partial \rho}{\partial t} + \frac{\partial q}{\partial x} = 0$$

This is derived from the idea of car conservation. We require that no cars can randomly appear or disappear on any one piece of road, and so

$$\frac{d}{dt}N(t) = cars\ in - cars\ out$$

If we then integrate over some boundaries $a$ and $b$, we get that

$$\frac{dN}{dt} = \frac{d}{dt}\int_a^b \rho(x,t)dx = q(a,t) - q(b,t)$$

If we now begin from the other direction where we consider the conservation law on $[a,x]$, we get

$$q(a,t) - q(x,t) = \frac{d}{dt}\int_a^x \rho(s,t)ds = \int_a^x \frac{\partial \rho}{\partial t}(s,t)ds$$

Differentiating with respect to $x$ we see that

$$\frac{\partial \rho}{\partial t} = \frac{\partial q}{\partial x}$$

which yields our PDE.

We must now consider some initial density function where $t = 0$. This does not have to be unique, but for the purposes of this analysis and exploration, we will choose

$$\rho(x,0) = \rho_0 = \rho_L + \frac{\rho_M - \rho_L}{(1+e^{-(x+L)/\ell})^a} + \frac{\rho_R - \rho_M}{(1+e^{-(x-L)/\ell})^a}$$

This formula provides leniency in its ability to assign different parameter values for the various parameters that exist. For our purposes in simulation, we will consider the following

$$\begin{aligned} \rho_L &= 1 \\ \rho_M &= 3 \\ \rho_R &= 8 \\ \ell &= 2 \\ L &= 50 \\ a &= \frac{1}{4} \end{aligned}$$

Note that this yields an initial density that is increasing, and thus we are confident that shocks will eventually form.

## Determining Initial Shock Formation

In order to determine where and when the initial shock forms, we note that at the instantaneous moment of shock formation, the two characteristics that meet to generate this shock will be infinitesimally close to one another on the x-axis where they originate. Hence, if we define the location that the characteristic that we use to determine our shock at $t = 0$ as $x_{s0}$, then we define the other characteristic that creates this shock as having location $x_{s0} + dx$ for some very small $dx$ value. Thus, since they are characteristics, their paths are defined as

$$x = x_{s0} + c(\rho_0(x_{s0})t$$
$$x = x_{s0} + dx + c(\rho_0(x_{s0} + dx)t$$

Setting those two equations equal to one another (since that creates the shock) and doing some algebra, we find

$$\frac{-dx}{dx} = \frac{t[c(\rho_0(x_{s0} + dx)) - c(\rho_0(x_{s0}))]}{dx}$$

Rearranging, we get that

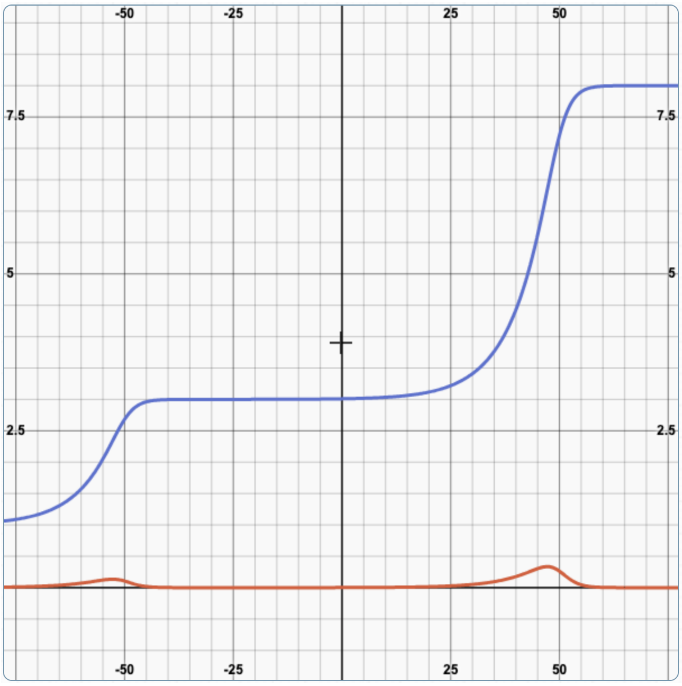
$$t = -\frac{1}{\frac{d}{dx}[c(\rho_0(x))]|_{x_{s0}}}$$

Since we want to find the first time that this happens, we minimize this equation, which is equivalent to maximizing the denominator (where we include the negative). If we note that $\frac{d}{dx}[c(\rho_0(x))] = \frac{dc}{d\rho}\frac{d\rho_0}{dx}$ and $c = u_{max}(1 - \frac{2\rho}{\rho_{max}})$, we can substitute the derivative of $c$ with respect to $\rho$ in to find

$$t = \frac{\rho_{max}}{2u_{max}} \frac{1}{\frac{d\rho_0}{dx}|_{x_{s0}}}$$

Ignoring the positive constant out front, we see that all we need to do to determine the location of this first shock characteristic is to maximize the derivative of $\rho_0$ with respect to $x$. Thus, if we subsitute our working values of $\rho_L, \rho_R, \rho_M, \ell, L$, and $a$, and then differentiate twice, we can solve for the roots of this function which will indicate the possible maxima or minima. If we inspect the function visually (or by differentiating again), we see that these roots are indeed maxima. A plot is given below for reference.

**Figure 1:** Plot of $\rho_0(x)$ in blue and $\frac{d\rho_0}{dx}$ in red




As we can see, the sudden increase in the density function indicates the formation of shocks in those areas, and we can see that while there are two local maxima for our red curve, the greater maximum occurs at $x = 47.2274$. The other local maximum occurs at $x = -52.7726$, which we will discuss in more detail further along.

Now that we have discovered the originating location of the characteristic that determines our shock, we can find the time and location of this shock quite easily. For the time, we simply use the formula we derived previously:

$$\tau_1 = \frac{\rho_{max}}{2u_{max}} \frac{1}{\frac{\partial \rho_0}{\partial x}|_{xs0}} = 14.953$$

In order to find the location, we just follow the path of the characteristic until it reaches $\tau_1$. Note that we implicitly determine the initial speed of the shock in determining the location, which is

$$c_1 = c(\rho_0(x_{s0})) = 1(1 - \frac{2\rho_0(x_{s0})}{10}) = -0.2687$$

$$x_{\tau_1} = x_{s0} + c(\rho_0(x_{s0})\tau_1 = 47.2274 + (-0.2687)(14.953) = 43.2089$$

Therefore, our first shock forms at location $x = 43.2089$ at time $t = 14.953$ with initial speed $c_1 = -0.2687$.

It would now make sense to investigate the other local maximum for another possible shock formation, however we must first be careful to check that this first shock would not reach the location of the second shock before it could form. We verify this by using a MATLAB program to numerically solve for the location of the first shock in the time that it would have before the second shock would theoretically form. We find that the first shock is not very fast, and does not even pass $x = 0$ before the second shock forms. Further in the discussion there will be more visualizations to display this fact, but for now take it to be true. We now find $\tau_2$ the time of this shock formation the same way we did for the first shock, given that we know that $x_{s0}$ for this shock is $-52.7726$.

$$\tau_2 = \frac{\rho_{max}}{2u_{max}} \frac{1}{\frac{\partial \rho_0}{\partial x}|_{x_{s0}}} = 37.383$$

With this time, we now find $x_{\tau_2}$ as well as $c_2$ (the initial speed) again using the same approach as for the first shock:

$$c_2 = 1(1 - \frac{2\rho_0(x_{s0})}{10}) = 0.5325$$

$$x_{\tau_2} = x_{s0} + c(\rho_0(x_{s0}))\tau_2 = -52.7726 + (0.5325)(37.383) = -32.8862$$

Therefore, the other shock that would form in our system does so at location $x = -32.8862$ at time $t = 37.383$ with initial speed $c_2 = 0.5325$.

## SHOCK BEHAVIOR AS $t \to \infty$

In determining the long term behavior of our shocks, whether or not they meet, and the possible resulting shock that may or may not form, we develop a program in MATLAB to numerically solve for the shock trajectories by using very small updates in time.

In our space time plot we observe three unique shocks. Following the upward trajectory of the time axis we see that the first shock to form is the blue shock. A reasonable time after, the second red shock forms to the left. As we derived previously, the speed of the first and second shock are such that they will eventually meet as we see in the plot. At this point of

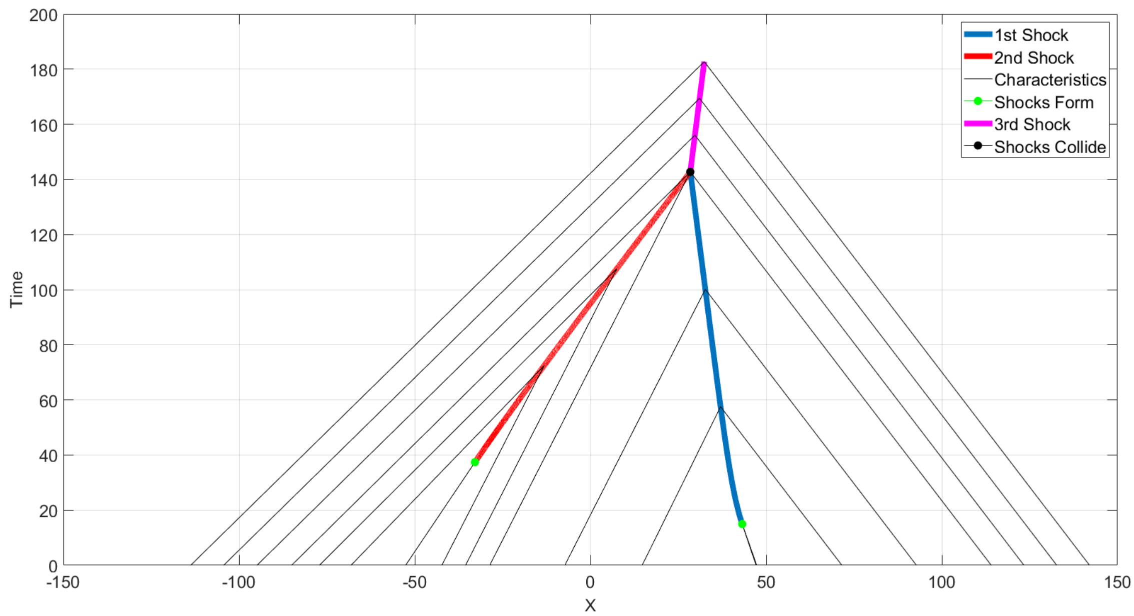



**Figure 2:** Space-time plot of characteristics and the three shocks that form

meeting we can follow the same process to calculate the third shock using the $\rho-$ of the left shock and the $\rho+$ of the right shock. As time goes to infinity, we see that we are just left with the third pink shock that will continue on its trajectory for all time, as there are no other shocks to perturb its movement. Our plot only shows this trajectory for a small amount of time to show its path. For all shocks we plot we also include black lines that are some chosen characteristic curves that are used in calculating the shock trajectories.

The code used to calculate the trajectory of the shocks is available in the appendix. To start, the locations of the characteristics that eventually form the shocks at $t = 0$ is determined. This is done by finding where the second spatial derivative of density is equal to zero. From here, the speed of the characteristic that creates each shock is determined and used to find the locations where the shocks form. The previously discussed equation for time is then utilized to find the time when each shock forms.

Knowing when and where both shocks form allows us to numerically simulate the trajectory of the shock over time. To do so, we define a very small timestep, $\Delta t$, which is on the order of $10^{-5}$. The main portion of the shock trajectory script is a loop that continuously increments time by this timestep. At each iteration of the loop, the position of the shock is updated based on its velocity. We know that by definition, two characteristics are meeting at this new location. The simulation then numerically solves for the originating location of these characteristics at this point in time. When the location of the characteristic is known, we can then calculate the density to the left and right of the shock using the defined formula for the initial density distribution. Lastly, the velocity of the shock is recalculated using the new left and right densities and then the loop repeats.

The script was used to model each of the three shocks in the same manner. For the first two shocks, their trajectories were simulated until the point where they contact each other to form a third shock. At this point, they combine to form a third shock. For this third shock, the initial speed is still determined by the jump in density across the shock. However, the density on the left comes from the leftmost density of the red shock and then right density comes from the rightmost density of the blue shock. The 3rd shock was only simulated for a small portion of time, but would continue on as depicted.

# APPENDIX

## Script to Calculate 1st and 2nd Shock Propagation

```
clear, clc
close all

% Constants for our scenario
pR = 8;
pL = 1;
pM = 3;
l = 2;
L = 50;
a = 1/4;
pmax = 10;
umax = 1;

% Defining various symbolic functions
syms x x01 x02
p = p0(x, pR, pL, pM, L, l, a);
dp_dx = diff(p);
d2p_dx2 = diff(dp_dx);
tau = pmax ./ (2*umax) * (1./dp_dx);
dtau_dx = diff(tau);
```

```
% Solving for the positions of the two shocks at t=0
xs1_t_0 = vpasolve(d2p_dx2==0, x, 47);
xs2_t_0 = vpasolve(d2p_dx2==0, x, -53);

% Calculating the times when two shocks form
x = xs1_t_0;
dp_dx_1 = subs(dp_dx);
tau1 = (pmax/(2*umax))*(1/double(dp_dx_1));
x = xs2_t_0;
dp_dx_2 = subs(dp_dx);
tau2 = (pmax/(2*umax))*(1/double(dp_dx_2));

%Calculating the shock locations at these new tau values
xs_tau1 = xs1_t_0 + (1-2*p0(xs1_t_0, pR, pL, pM, L, l, a)/pmax)*tau1;
xs_tau2 = xs2_t_0 + (1-2*p0(xs2_t_0, pR, pL, pM, L, l ,a)/pmax)*tau2;

% Setting initial conditions of the first shock that forms
xs0 = xs_tau1;
ts0 = tau1;
ps0 = p0(xs1_t_0, pR, pL, pM, L, l, a);
vs0 = umax*(1-(2*ps0)/pmax);

%Setting initial conditions of second shock that forms

xs02 = xs_tau2;
ts02 = tau2;
ps02 = p0(xs2_t_0, pR, pL, pM, L, l, a);
vs02 = umax*(1-(2*ps02)/pmax);

% Set some initial conditions for the iteration
dt = 10^(-5);
x1 = xs0;
vs1 = vs0;
x2=xs02;
vs2 = vs02;
```

```
t = ts0;
counter = 1;
t_intersect = 160;
iter = t_intersect/dt;
iter = 1; % Only set for exporting the code to latex
x_s1_pos = zeros(1, iter);
x_s2_pos = zeros(1,iter);
t_all = zeros(1, iter);

x_minus1 = xs0 - 5;
x_plus1 = xs0 + 5;
x_minus2 = xs02 - 5;
x_plus2 = xs02 + 5;

x_char1 = zeros(3, iter);
x_char2 = zeros(3, iter);

while (abs(x1 - x2) > 0.01) && (counter < iter)
    % Add current shock position to tracker
    x_s1_pos(counter) = x1;
    % Update current timestep and add it to tracker
    t = t + dt;
    t_all(counter) = t;
    if t>=tau2
        x_s2_pos(counter) = x2;
        x2 = x2 + vs2*dt;
        c2 = umax*(1 - 2*p0(x02, pR, pL, pM, L, l, a)/pmax);
        eqn2 = x2 - x02 - c2 * t;

        x_minus2 = vpasolve(eqn2, x02, x_minus2 - 1);
        x_plus2 = vpasolve(eqn2, x02, x_plus2 + 1);
        try
            x_char2(:, counter) = [x_minus2, x_plus2, x2];
        catch
            x_minus2
```

```
        x_plus2
        x2
    end
    p_minus2 = p0(x_minus2, pR, pL, pM, L, l, a);
    p_plus2 = p0(x_plus2, pR, pL, pM, L, l, a);
    vs2 = umax*(1-(p_plus2 + p_minus2)/pmax);
end
% Calculate new position of shock
x1 = x1 + vs1*dt;


% Calculate speed of characteristics
c1 = umax*(1 - 2*p0(x01, pR, pL, pM, L, l, a)/pmax);

% Define equation to be solved
eqn1 = x1 - x01 - c1 * t;

% Solve for x+ and x-
x_minus1 = vpasolve(eqn1, x01, x_minus1 - 1);
x_plus1 = vpasolve(eqn1, x01, x_plus1 + 1);
x_char1(:, counter) = [x_minus1, x_plus1, x1];

% Solve for p+ and p-
p_minus1 = p0(x_minus1, pR, pL, pM, L, l, a);
p_plus1 = p0(x_plus1, pR, pL, pM, L, l, a);

% Solve for new shock speed
vs1 = umax*(1 - (p_plus1 + p_minus1)/pmax);

% Display percent complete. Comment out to run faster
if mod(counter, 1000) == 0
    percent = counter / iter * 100;
    disp(['Percentage: ', num2str(percent)])
end
```

```
    % Update counter
    counter = counter + 1;
end

% save('results4.mat')
```

## Script to Calculate 3rd Shock Propagation

```
clear, clc
close all

% Constants for our scenario
pR = 8;
pL = 1;
pM = 3;
l = 2;
L = 50;
a = 1/4;
pmax = 10;
umax = 1;

% Defining various symbolic functions
syms x x03
p = p0(x, pR, pL, pM, L, l, a);

x3 = 28.422;
t3 = 142.723;

x_l = -85.0352;
x_r = 114.06;

p_l = p0(x_l, pR, pL, pM, L, l, a);
p_r = p0(x_r, pR, pL, pM, L, l, a);

vs3 = umax*(1-(p_r + p_l)/pmax);
```

```
% Set some initial conditions for the iteration
dt = 10^(-3);
t = t3;
counter = 1;
t_intersect = 160;
iter = t_intersect/dt;
iter = 1; % Set to export code to latex

x_s3_pos = zeros(1, iter);
t_all = zeros(1, iter);

x_minus3 = x_l;
x_plus3 = x_r;

x_char3 = zeros(3, iter);

while (counter < iter)
    % Add current shock position to tracker
    x_s3_pos(counter) = x3;
    % Update current timestep and add it to tracker
    t = t + dt;
    t_all(counter) = t;

    % Calculate new position of shock
    x3 = x3 + vs3*dt;

    % Calculate speed of characteristics
    c3 = umax*(1 - 2*p0(x03, pR, pL, pM, L, l, a)/pmax);

    % Define equation to be solved
    eqn1 = x3 - x03 - c3 * t;

    % Solve for x+ and x-
    x_minus3 = vpasolve(eqn1, x03, [x_minus3-100, x_minus3+0.1]);
```

```
    x_plus3 = vpasolve(eqn1, x03, [x_plus3-0.1, x_plus3+100]);
    try
        x_char3(:, counter) = [x_minus3, x_plus3, x3];
    catch
        x_minus3
        x_plus3
    end

    % Solve for p+ and p-
    p_minus3 = p0(x_minus3, pR, pL, pM, L, l, a);
    p_plus3 = p0(x_plus3, pR, pL, pM, L, l, a);

    % Solve for new shock speed
    vs3 = umax*(1 - (p_plus3 + p_minus3)/pmax);

    % Display percent complete. Comment out to run faster
    if mod(counter, 1000) == 0
        percent = counter / iter * 100;
        disp(['Percentage: ', num2str(percent)])
    end


    % Update counter
    counter = counter + 1;
end

% save('3rd_shock.mat')
```

## Script to Plot Results

```
clear, clc
close all

load("results3.mat", 'x_s1_pos', 'x_char1')
load('2nd_shock.mat')
```

```
% Constants for our scenario
pR = 8;
pL = 1;
pM = 3;
l = 2;
L = 50;
a = 1/4;
pmax = 10;
umax = 1;

% Defining various symbolic functions
syms x
p = p0(x, pR, pL, pM, L, l, a);
dp_dx = diff(p);
d2p_dx2 = diff(dp_dx);
tau = pmax ./ (2*umax) * (1./dp_dx);
dtau_dx = diff(tau);

x = -100:10:100;

p = subs(p);
v = umax*(1-(2*p)/pmax);

t = 0:10:200;

xlabel('Position (x)')
ylabel('Time (s)')

id1 = find(x_s1_pos > 28.4279);
% id11 = find(x_s1_pos > )
id2 = find(x_s2_pos ~= 0);
id22 = find(x_s2_pos < 28.4279);
id2 = intersect(id2, id22);
```

```
x_s1_pos_valid = x_s1_pos(id1);
x_s2_pos_valid = x_s2_pos(id2);

x_char1 = x_char1(:, id1);
x_char2 = x_char2(:, id2);
t2 = t_all(id2);

load('results3.mat', 't_all')
t1 = t_all(id1);

% Plot first shock
plot(x_s1_pos_valid, t1, 'LineWidth', 6, 'DisplayName', '1st Shock')
hold on
% Plot second shock
plot(x_s2_pos_valid, t2, 'LineWidth', 6, 'Color', 'r',...
    'DisplayName', '2nd Shock')
hold on

% Plot first characteristics of first shock
plot([x_char1(1,1), x_char1(3, 1)], [0, t1(1)], 'Color', 'k',...
    'DisplayName', 'Characteristics')
plot([x_char1(2,1), x_char1(3, 1)], [0, t1(1)], 'Color', 'k',...
    'HandleVisibility', 'off')

% Plot first characteristics of second shock
plot([x_char2(1,1), x_char2(3, 1)], [0, t2(1)], 'Color', 'k',...
    'HandleVisibility', 'off')
plot([x_char2(2,1), x_char2(3, 1)], [0, t2(1)], 'Color', 'k',...
    'HandleVisibility', 'off')

% Plot 1/3 characteristics of first shock
plot([x_char1(1, 42589), x_char1(3, 42589)], [0, t1(42589)], 'Color',...
    'k', 'HandleVisibility', 'off')
plot([x_char1(2, 42589), x_char1(3, 42589)], [0, t1(42589)], 'Color',...
    'k', 'HandleVisibility', 'off')
```

```
% Plot 1/3 characteristics of second shock
plot([x_char2(1, 3511), x_char2(3, 3511)], [0, t2(3511)], 'Color', 'k',...
    'HandleVisibility', 'off')
plot([x_char2(2, 3511), x_char2(3, 3511)], [0, t2(3511)], 'Color', 'k',...
    'HandleVisibility', 'off')

% Plot 2/3 characteristics of first shock
plot([x_char1(1, 85178), x_char1(3, 85178)], [0, t1(85178)], 'Color',...
    'k', 'HandleVisibility', 'off')
plot([x_char1(2, 85178), x_char1(3, 85178)], [0, t1(85178)], 'Color',...
    'k', 'HandleVisibility', 'off')

% Plot 2/3 characteristics of second shock
plot([x_char2(1, 7023), x_char2(3, 7023)], [0, t2(7023)], 'Color', 'k',...
    'HandleVisibility', 'off')
plot([x_char2(2, 7023), x_char2(3, 7023)], [0, t2(7023)], 'Color', 'k',...
    'HandleVisibility', 'off')

% Plot final characteristics of first shock
plot([x_char1(1,end), x_char1(3, end)], [0, t1(end)], 'Color', 'k',...
    'HandleVisibility', 'off')
plot([x_char1(2,end), x_char1(3, end)], [0, t1(end)], 'Color', 'k',...
    'HandleVisibility', 'off')

% Plot final characteristics of second shock
plot([x_char2(1,end), x_char2(3, end)], [0, t2(end)], 'Color', 'k',...
    'HandleVisibility', 'off')
plot([x_char2(2,end), x_char2(3, end)], [0, t2(end)], 'Color', 'k',...
    'HandleVisibility', 'off')

% Plot a bunch of characteristics
% for i = 1:length(x)
%     characteristic = x(i) + v(i).*t;
%     plot(characteristic, t, 'k')
```

```
%     hold on
% end

plot(min(x_s2_pos_valid), min(t2), 'Marker', '.', 'MarkerSize', 30,...
    'Color', 'g', 'DisplayName', 'Shocks Form')
plot(max(x_s1_pos_valid), min(t1), 'Marker', '.', 'MarkerSize', 30,...
    'Color', 'g', 'HandleVisibility', 'off')

clear t_all
load('3rd_shock.mat')
plot(x_s3_pos(1:end-1), t_all(1:end-1), 'LineWidth', 6, 'Color',...
    'magenta', 'DisplayName', '3rd Shock')
plot(max(x_s2_pos_valid), max(t2), 'Marker', '.', 'MarkerSize', 30,...
    'Color', 'k', 'DisplayName', 'Shocks Collide')

% Plot 1/3 characteristics of 3rd shock
plot([x_char3(1, 13333), x_char3(3, 13333)], [0, t_all(13333)],...
    'Color', 'k', 'HandleVisibility', 'off')
plot([x_char3(2, 13333), x_char3(3, 13333)], [0, t_all(13333)],...
    'Color', 'k', 'HandleVisibility', 'off')

% Plot 2/3 characteristics of 3rd shock
plot([x_char3(1, 26666), x_char3(3, 26666)], [0, t_all(26666)],...
    'Color', 'k', 'HandleVisibility', 'off')
plot([x_char3(2, 26666), x_char3(3, 26666)], [0, t_all(26666)],...
    'Color', 'k', 'HandleVisibility', 'off')

% Plot final characteristics of 3rd shock
plot([x_char3(1,end-1), x_char3(3, end-1)], [0, t_all(end-1)],...
    'Color', 'k', 'HandleVisibility', 'off')
plot([x_char3(2,end-1), x_char3(3, end-1)], [0, t_all(end-1)],...
    'Color', 'k', 'HandleVisibility', 'off')

grid on
xlabel('X')
```

```
ylabel('Time')
legend
fontsize(18, 'points')
```