```
clear all;
close all;

% Set Parameters:
nt = 100; %Number of Temporal Grid Points
nx = 100; %Number of Spatial Grid Points
T = 2.8; %Final Time
c = 1; %Speed
nIntegrationTerms = 100; %Number of Nodes for the IntegrateTrapazoidal Routine


%f = @(x)
df_dx = @(x) exp(-((x-0.5).^2)./0.005)+3*exp(((-(x).^2))/0.005)+2*exp(-((x+0.5).^2)/0.↙
005);  % Initial condition.

% --- INITIALIZE THE GRIDS:
t = linspace(0,T,nt); %Initial Time, End Time, Spacing
x = linspace(-8,8,nx); %Left End point, Right End Point, Spacing

uStore=zeros(nx,nt); %Initialize Storage Matrix

for j=1:nt; %loop over time

    uTemp=zeros(nx,1); %initialize temporary variable to give the illusion of time↙
evolution
    for k=1:nx; %loop over space
        a = x(k) - c*t(j); %lower integration value
        b = x(k) + c*t(j); %upper integration value
         uStore(k,j) = integrateTrapezoidal(df_dx,a,b,nIntegrationTerms);
         uTemp(k)=uStore(k,j);

    end

        %plot(x, uTemp);
        %axis([-8, 8, -2, 2])
        %pause(0.1)   %How long of a pause between frames

end

LEP = -8; %left bound
REP = 8;  %right bound
h=(REP-LEP)/nx; %spacing

% --- Generate a first derivative matrix:
row = zeros(1,nx);  % Initialize n row vector of length
                   % n with zeros.
row(2) = 1;
row(nx) = -1;

col = zeros(nx,1);  % Initialize n column vector of length
                   % n with zeros.
```

```
col(2) = -1;
col(nx) = 1;

D1 = toeplitz(col,row)/(2*h);

D1x = D1*uStore;

for v=1:nt;
    uTemp=zeros(nx,1);
    for q=1:nx;
        uTemp(q)=D1x(q,v);
    end

    plot(x,uTemp)
    axis([-8, 8, -2, 2])
    pause(0.1)   %How long of a pause between frames

end
```