```matlab
% Copied and modified code from ballode.m from Matlab.

% model parameters
m = 1; % mass
c = 0.01; % friction coefficient
g = 9.81; % acceleration of gravity
BallDynamics = @(t,y) vertcat(y(3:4),-[0;g] -c/m*norm(y(3:4),2)*y(3:4));

tstart = 0;
tfinal = 20;
y0 = [0;10;0;0];
refine = 4;
options = odeset('Events',@events,'OutputFcn',@odeplot,'OutputSel',2,...
   'Refine',refine);

fig = figure;
ax = axes;
ax.XLim = [0 20];
ax.YLim = [0 11];
box on
hold on;

tout = tstart;
yout = y0.';
for i = 1:10
   % Solve until the first terminal event.
   [t,y,te,ye,ie] = ode23(BallDynamics,[tstart tfinal],y0,options);
   fprintf('stopped at t = %G when y = %G and y_dot = %G\n', t(end), y(end,2), y(end,4));
   if ~ishold
      hold on
   end
   % Accumulate output. This could be passed out as output arguments.
   nt = length(t);
   tout = [tout; t(2:nt)];
   yout = [yout; y(2:nt,:)];

   ud = fig.UserData;
   if ud.stop
      break;
   end

   y0(1) = y(nt,1);
   y0(2) = y(nt,2);
   y0(3) = y(nt,3);
   y0(4) = -1*y(nt,4);

   % A good guess of a valid first timestep is the length of the last valid
   % timestep, so use it for faster computation.  'refine' is 4 by default.
   options = odeset(options,'InitialStep',t(nt)-t(nt-refine),...
      'MaxStep',t(nt)-t(1));

   tstart = t(nt);
end

plot(teout,yeout(:,1),'ro')
xlabel('time');
ylabel('height');
```

```
title('Ball trajectory and the events');
hold off
odeplot([],[],'done');


function [value,isterminal,direction] = events(t,y)
    % Locate the time when height passes through zero in a decreasing direction
    % and stop integration.
    value = y(2);     % detect height = 0
    isterminal = 1;   % stop the integration
    direction = -1;   % negative direction
end
```