

```

% ### EXprojectileM2.m ###          [08.09.16 CB]

% simple code to demonstrate path of 2-D projectile under influence of
% gravity
% w/o drag; this code also uses Euler's method to explicitly solve the
% associated ODE (red x's) in addition to the exact analytic solution
% (blue circles)

% the relevant ODEs (given no drag) are:
% [ones stemming from force law]
%  $d(v_x)/dt = 0$       [SOL] -->  $v_x = \text{constA.}$ 
%  $d(v_y)/dt = -g$      [SOL] -->  $v_y = \text{constB.} - g*t$ 
% [ones to solve below]
%  $dx/dt (=v_x) = \text{constA.}$       [SOL] -->  $x(t) = \text{constC.} + t* \text{constA.}$ 
%  $dy/dt (=v_y) = \text{constB.} - (g/m)*t$       [SOL] -->  $y(t) = \text{constD.} + t* \text{constB.} - (g/2)*t^2$ 

% Questions for students:
% o How does Newton's 2nd law fit in here?
% o What are the various constants (e.g., constA., constB., ...)? How
do
% you determine such?
% o What is "Euler's method"?
% o Why/how are the red and blue curves different? How does such
depend
% upon the step size (h)?
% o Where does the mass of the object factor in?

clear
% -----
% User params.
v0= 10;      % Initial velocity [m/s]
theta= 40;   % Angle of projection [degrees]

h= 0.01; % step-size in time [s]
g= 9.8; % acceleration due to gravity [m/s^2]
m= 1.0; % particle mass
x0= 0;   y0= 1;      % initial coords. [m]

% -----

theta= theta* pi/180; % convert angle to radians

% [derived quantities from analytic solution]
xmax=x0+ v0^2*sin(2*theta)/g;      % max. horizontal dist.
ymax=y0+ v0^2*sin(theta)^2/(2*g); % max. vertical dist.,.
td=2*v0*sin(theta)/g;             %total time before landing

figure(1); clf; grid on; hold on;

```

```

xlabel('Distance [m]'); ylabel('Height [m]');
title('Projectile path (no damping)');
xlim([0 1.1*xmax]); ylim([0 1.1*yymax]);

x1= x0; y1= y0;      % initialize loop variables for Euler solution

for t=0:h:td+h

    % SOL1: analytic solution
    x=x0+ v0* t* cos(theta);          %analytic solution for x
    y=y0+ v0* t* sin(theta)- g* t^2/2; %analytic solution for y

    plot(x,y,'r*',x1,y1,'bo') % plot
    getframe;

    % SOL2: Euler's method to solve relevant ODE
    x1= x1+ h* v0* cos(theta);
    y1= y1+ h* (v0* sin(theta)- g*t);
end

% modified from original source:
% http://numericalcomputation.blogspot.ca/2012/06/matlab-projectile-motion-by-eulers.html

```