```
% Celestial Mechanics:  Two-Body Problem Plot (Earth-Sun)
% Inputs (Function Arguments):
%   1.  Time Interval
%   2.  Initial Conditions
%       ic = [ x1 y1 vx1 vy1 x2 y2 vx2 vy2]
%              x position, y position, x velocity, y velocty,   Body #1
%              x position, y position, x velocity, y velocty    Body #2
%   3.  Number of Steps: n
%   4.  Steps per point plotted: p
%   5.  Differential Equation "Solver"
%           a.  1 = RK4
%           b.  2 = Trapezoid
%           c.  3 = Euler
% Calls:  Trapezoid Method (trapstep.m:   one-step)
% Invocation: "Command Line" -
%   1.  Clear workspace variables:
%           clear all; clearvars; close all; clc;
%   2.  Example Usage:
% Prob.        Rearrange            |---Earth---||----Sun-----|
%  #           Function Arguments:  x1 y1 vx1 vy1 x2 y2 vx2 vy2  # Steps
% #9(a)        orbit2body([0 100], [2  2 .2 -.2 0  0  -.02 .02], 10000, 5,1)
% #9(b)        orbit2body([0 100], [2  2 .2 -.2 0  0  -.02 .02], 10000, 10,1)
%                   *** Orbit2Body_fig01.fig
% #10a(1)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 10000, 10,1)
%                   *** Orbit2Body_fig02.fig
% #10a(2)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 10000, 10,2)
%                   *** Orbit2Body_fig03.fig
% #10a(3)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 100000, 100,2)
%                   *** Orbit2Body_fig04.fig
% #10b(1)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 10000, 10,2)
%                   *** Orbit2Body_fig05.fig
% #10b(2)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 100000, 100,2)
%                   *** Orbit2Body_fig06.fig
% #10c(1)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 10000, 10,2)
%                   *** Orbit2Body_fig07.fig
% #10c(2)      orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 100000, 100,2)
%                   *** Orbit2Body_fig08.fig
% #10c(3)      orbit2body([0 500], [0  1 .1 -.1 -2 -1 -.01 .01], 100000, 100, 2)
%                   *** Orbit2Body_Trap_500.mp4
% #10c(4)      orbit2body([0 500], [0  1 .1 -.1 -2 -1 -.01 .01], 100000, 100, 1)
%                   *** Orbit2Body_RK4_500.mp4
% #11a(1)      orbit2body([0 100], [0  1 .2 -.2 -2 -1 -.2 .2], 100000, 100,1)
% #11a(2)      orbit2body([0 100], [0  1 .2 -.2 -2 -1 -.2 .2], 100000, 100,2)
% #11b(1)      orbit2body([0 100], [0  1 .2 -.2 -2 -1 -.2 .2], 100000, 100,1)
% #11b(2)      orbit2body([0 100], [0  1 .2 -.2 -2 -1 -.2 .2], 100000, 100,2)
% #11c(1)      orbit2body([0 100], [0  1 .2 -.2 -2 -1 -.2 .2], 100000, 100,1)

% #11c(2)      orbit2body([0 100], [0  1 .2 -.2 -2 -1 -.2 .2], 100000, 100,2)
```

```
% #10  (c)  orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 10000, 10,1)
% #10  (c)  orbit2body([0 100], [0  1 .1 -.1 -2 -1 -.01 .01], 10000, 10,2)
%
%      (d)  orbit2btest03([0 100], [2  2 .25 -.25 0  0  -.02 .02], 10000, 10)
%      (e)  orbit2btest03([0 100], [2.1 2.1 .25 -.25 0  0  -.02 .02], 10000, 5)
%      (f)  orbit2btest03([0 100], [2 .2  2 -.2 0 -.02  0  .02], 10000, 5)
%                                x1 vx1 y1 vy1 x2 vx2 y2 vy2
%                                |-------- Original -------|
% Prospective Upgrades:
%   1.  Add dynamic "Center-of-Mass" (COM) "X" position
%
function z=orbit2body(inter,ic,n,p,solver)

% plot n points over interval
h=(inter(2)-inter(1))/n;

% Incorporate Initial Conditons
x1=ic(1);y1=ic(2);vx1=ic(3);vy1=ic(4);
x2=ic(5);y2=ic(6);vx2=ic(7);vy2=ic(8);

vector_position_velocity = [x1 y1 vx1 vy1 x2 y2 vx2 vy2]
t(1)=inter(1);

% Construct Animation Objects for Graph
set(gca,'XLim',[-5 5],'YLim',[-5 5],...
    'XTick',[-5 0 5],'YTick',[-5 0 5]);
sun=animatedline('color','y','Marker','.','markersize',50);
suntail=animatedline('color', 'b', 'LineStyle', '-');
earth=animatedline('color','r','Marker','.','markersize',35);
earthtail=animatedline('color','g','LineStyle','-');
grid on;

%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 9 (RK4):  m1=0.03, m2=0.3    (fig01.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10a (RK4):  m1=0.03, m2=0.3    (fig02.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10a (Trapezoid:  h=0.01):  m1=0.03, m2=0.3    (fig03.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10a (Trapezoid:  h=0.001):  m1=0.03, m2=0.3    (fig04.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10b (Trapezoid:  h=0.01):  m1=0.05, m2=0.5    (fig05.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10b (Trapezoid:  h=0.001):  m1=0.05, m2=0.5    (fig06.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10c (Trapezoid:  h=0.01):  m1=0.08, m2=0.8    (fig07.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10c (Trapezoid:  h=0.001):  m1=0.08, m2=0.8    (fig08.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
```

```
%       'Problem 11b(2) (Trap:  h=0.001):  m1=m2=1    (fig12.fig)'});
%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 11c(1) (Trap:  h=0.001):  m1=m2=0.5    (fig13.fig)'});
title({'Celestial Mechanics:  Two (2) Body Problem';
       'Problem 11c(2) (Trap:  h=0.001):  m1=m2=0.5    (fig14.fig)'});

%[px,py]=ginput(1);             % Include these three lines
%[px1,py1]=ginput(1);           %   to enable mouse support
% y(1,:)=[px px1-px py py1-py]; %   2 clicks set direction

v=VideoWriter('Orbit2Body_Trap_m1=m2=0.5','MPEG-4');
open(v)

for k=1:n/p
    for i=1:p
        t(i+1)=t(i)+h;

        switch(solver)
            case 1
                vector_position_velocity(i+1,:)=rk4step(t(i),vector_position_velocity ↙
(i,:),h);
                fprintf('===> RK4 <===\n' );
            case 2
                vector_position_velocity(i+1,:)=trapstep(t(i), ↙
vector_position_velocity(i,:),h);
                fprintf('===> Trapezoid Method <===\n' );
            otherwise
                fprintf('*** "Scope" Parameter Invalid!!!\n' );
        end
    end
    %y(1,:)=y(p+1,:);
    vector_position_velocity(1,:)=vector_position_velocity(p+1,:);
    t(1)=t(p+1);
    clearpoints(sun);
    clearpoints(earth);

    temp_x1  = vector_position_velocity(1,1)
    temp_y1  = vector_position_velocity(1,2)
    temp_vx1 = vector_position_velocity(1,3)
    temp_vy1 = vector_position_velocity(1,4)
    temp_x2  = vector_position_velocity(1,5)
    temp_y2  = vector_position_velocity(1,6)
    temp_vx2 = vector_position_velocity(1,7)
    temp_vy2 = vector_position_velocity(1,8)

    addpoints(earth,vector_position_velocity(1,1),vector_position_velocity(1,2))
    addpoints(earthtail,vector_position_velocity(1,1),vector_position_velocity(1,2))
    addpoints(sun,vector_position_velocity(1,5),vector_position_velocity(1,6))
    addpoints(suntail,vector_position_velocity(1,5),vector_position_velocity(1,6))
```

```
    drawnow;
    frame=getframe;
    writeVideo(v,frame);
end
close(v)

%function y = trapstep(t,x,h) % one step of the Trapezoid Method
function y = trapstep(t,vec_pv,h) % one step of the Trapezoid Method
    % Evaluating each endpoint; taking average y value

%{
    z1=ydot(t,x);
    g=x+h*z1;
    z2=ydot(t+h,g);
    y=x+h*(z1+z2)/2;
%}
    z1=ydot(t,vec_pv);
    g=vec_pv+h*z1;
    z2=ydot(t+h,g);
    y=vec_pv+h*(z1+z2)/2;

function y=rk4step(t,vpv,h)
    s1 = ydot(t,vpv);
    s2 = ydot(t+h/2,vpv+h*s1/2);
    s3 = ydot(t+h/2,vpv+h*s2/2);
    s4 = ydot(t+h,vpv+h*s3);
    y=vpv+h*(s1+2*s2+2*s3+s4)/6;

% Differential Equations
function z = ydot(t,x)
% Problems 9, 10a:
%m1=0.03;
%m2=0.3;
%Problem 10b:
%m1=0.05;
%m2=0.5;
%Problem 10c:
%m1=0.08;
%m2=0.8;
%Problem 11a(1,2)
%m1=2;
%m2=2;
%Problem 11b(1,2)
%m1=1;
%m2=1;
%Problem 11b(1,2)
m1=0.5;
m2=0.5;
```

```
%Problem 11a:
%m1=2;
%m2=2;
%Problem 11b:
%m1=1;
%m2=1;
%Problem 11b:
%m1=0.5;
%m2=0.5;

g=1;
mg1=m1*g;
mg2=m2*g;
px1=x(1);py1=x(2);vx1=x(3);vy1=x(4);
px2=x(5);py2=x(6);vx2=x(7);vy2=x(8);
dist=sqrt((px2-px1)^2+(py2-py1)^2);
z=zeros(1,8);
z(1)=vx1;
z(2)=vy1;
z(3)=(mg2*(px2-px1))/(dist^3);
z(4)=(mg2*(py2-py1))/(dist^3);
z(5)=vx2;
z(6)=vy2;
z(7)=(mg1*(px1-px2))/(dist^3);
z(8)=(mg1*(py1-py2))/(dist^3);

%******************** BELOW THE LINE ***********************************
%y(1,:)=[x1 y1 vx1 vy1 x2 y2 vx2 vy2]

%title({'Celestial Mechanics:  Two (2) Body Problem';
%       'Problem 10b (Trapezoid):  m1=0.05, m2=0.5    (fig04.fig)'});

%title({'First line';'Second line'})
%xlabel({'First line';'Second line'})

%title({'Celestial Mechanics:  Two(2) Body Problem';'IC:  x1 = ',num2str(x(1))});

%head=animatedline('color','r','Marker','.','markersize',35);

%v=VideoWriter('num9','MPEG-4');

%{
    addpoints(earth,y(1,1),y(1,2))
    addpoints(earthtail,y(1,1),y(1,2))
    addpoints(sun,y(1,5),y(1,6))
    addpoints(suntail,y(1,5),y(1,6))
%}

% build y vector
```

```
        %y(i+1,:)=trapstep(t(i),y(i,:),h);

%{
figure
h = animatedline;
axis([0 4*pi -1 1])

for x = linspace(0,4*pi,10000)
    y = sin(x);
    addpoints(h,x,y)
    drawnow limitrate
end
%}
%{
    addpoints(head,y(1,1),y(1,3))
    addpoints(sun,y(1,5),y(1,7))
    addpoints(tail,y(1,1),y(1,3))
    addpoints(suntail,y(1,5),y(1,7))
%}
%{
    temp00 = vector_position_velocity(1,1)
    temp01 = vector_position_velocity(1,2)
    temp02 = vector_position_velocity(1,3)
    temp03 = vector_position_velocity(1,4)
    temp04 = vector_position_velocity(1,5)
    temp05 = vector_position_velocity(1,6)
    temp06 = vector_position_velocity(1,7)
    temp07 = vector_position_velocity(1,8)
%}
%{
    addpoints(earth,y(1,1),y(1,2))
    addpoints(earthtail,y(1,1),y(1,2))
    addpoints(sun,y(1,5),y(1,6))
    addpoints(suntail,y(1,5),y(1,6))
%}
```