

---

## Table of Contents

.....	1
inputs .....	1
calculations .....	1
plot .....	2
case b .....	3
case b 2 .....	5
case c .....	7

```
% 2d non dimensional 3 body problem
% Arjan Reyes, 25.11.09
```

```
clc;
clear all;
```

```
% we are given positions and velocities
% ode45 can only solve first order diff eqs
% write a function to go from pos and vel to vel and accel
% ode takes this function at each time step to calc position and velocity
which are then used in the next timestep
% ok i get it now
```

## inputs

```
obj_i=[mass,position x, position y, velocity x, velocity y]
```

```
%case a
```

```
obj_1=[1,0.7216878365, 0.08333333333,-0.2041241452, 1.7677669530];
obj_2=[2,-0.1443375673, 0.58333333333,-1.4288690166, -0.3535533906];
obj_3=[3,-0.1443375673, -0.4166666667,1.0206207262, -0.3535533906];
```

```
obj_matrix=[obj_1;obj_2;obj_3];
```

```
%translate to state vector initial
```

```
% Phi =[r1x,r1y,v1x,v1y,r2x,r2y,v2x,v2y,r3x,r3y,v3x,v3y];
```

```
Phi_0=[];
```

```
for i=1:3
```

```
    for j=2:5
```

```
        Phi_0(end+1)=obj_matrix(i,j);
```

```
    end
```

```
end
```

```
Phi_0=Phi_0(:);
```

## calculations

```
function ddt_phi = derivFunc(t, Phi)
```

```
    % Constants
```

---

```

G = 1;
m1 = 1; m2 = 2; m3 = 3;
r1 = Phi(1:2);
r2 = Phi(5:6);
r3 = Phi(9:10);
v1 = Phi(3:4);
v2 = Phi(7:8);
v3 = Phi(11:12);
r12 = r2 - r1; % vector from 1 to 2
r13 = r3 - r1; % vector from 1 to 3
r23 = r3 - r2; % vector from 2 to 3

a1 = G*m2*(r12)/norm(r12)^3 + G*m3*(r13)/norm(r13)^3;
a2 = G*m1*(-r12)/norm(r12)^3 + G*m3*(r23)/norm(r23)^3;
a3 = G*m1*(-r13)/norm(r13)^3 + G*m2*(-r23)/norm(r23)^3;

ddt_phi = [v1(:); a1(:); v2(:); a2(:); v3(:); a3(:)];
end

% [t,Phi] = ode45(derivative vector, time span, initial condition)
timeSpan=[0,10]; % time start, time end | it picks time step automatically

[t,Phi_calc]=ode45(@derivFunc,timeSpan,Phi_0);

```

## plot

```

figure;

subplot(2,2,1)
hold on;
grid on;
xlabel('x'); ylabel('y');
title('3-Body Trajectories Case A');

% making a 2d position plot, where time is not an axis
% object 1 full trajectory

plot(Phi_calc(:,1),Phi_calc(:,2),'r','DisplayName','Object 1');

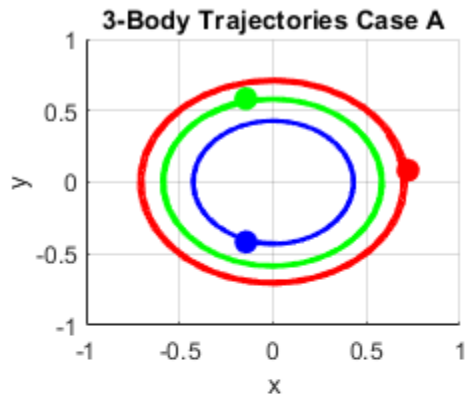
% object 2 full trajectory
plot(Phi_calc(:,5),Phi_calc(:,6),'g','DisplayName','Object 2');

% object 3 full trajectory
plot(Phi_calc(:,9),Phi_calc(:,10),'b','DisplayName','Object 3');

% starting positions
h1=plot(Phi_0(1),Phi_0(2),'ro','MarkerFaceColor','r','MarkerSize',8);
h2=plot(Phi_0(5),Phi_0(6),'go','MarkerFaceColor','g','MarkerSize',8);
h3=plot(Phi_0(9),Phi_0(10),'bo','MarkerFaceColor','b','MarkerSize',8);

hold off;

```



## case b

```
%case b 1
obj_1=[1,0.5773502692, 0,0.1732050808, 0];
obj_2=[1,-0.2886751346, 0.5,-0.0866025404, 0.15];
obj_3=[1,-0.2886751346, -.5,-0.0866025404, -0.15];

obj_matrix=[obj_1;obj_2;obj_3];

%translate to state vector initial
% Phi =[r1x,r1y,v1x,v1y,r2x,r2y,v2x,v2y,r3x,r3y,v3x,v3y];
Phi_0=[];
for i=1:3
    for j=2:5
        Phi_0(end+1)=obj_matrix(i,j);
    end
end

Phi_0=Phi_0(:);

function ddt_phi = derivFunc1(t, Phi)
    % Constants
    G = 1;
    m1 = 1; m2 = 1; m3 = 1;
    r1 = Phi(1:2);
```

---

```

r2 = Phi(5:6);
r3 = Phi(9:10);
v1 = Phi(3:4);
v2 = Phi(7:8);
v3 = Phi(11:12);
r12 = r2 - r1; % vector from 1 to 2
r13 = r3 - r1; % vector from 1 to 3
r23 = r3 - r2; % vector from 2 to 3

a1 = G*m2*(r12)/norm(r12)^3 + G*m3*(r13)/norm(r13)^3;
a2 = G*m1*(-r12)/norm(r12)^3 + G*m3*(r23)/norm(r23)^3;
a3 = G*m1*(-r13)/norm(r13)^3 + G*m2*(-r23)/norm(r23)^3;

ddt_phi = [v1(:); a1(:); v2(:); a2(:); v3(:); a3(:)];
end

[t,Phi_calc]=ode45(@derivFunc1,timeSpan,Phi_0);

% plot

subplot(2,2,2)
hold on;
grid on;
xlabel('x'); ylabel('y');
title('3-Body Trajectories case b1');

% making a 2d position plot, where time is not an axis
% object 1 full trajectory

plot(Phi_calc(:,1),Phi_calc(:,2),'r','DisplayName','Object 1');

% object 2 full trajectory
plot(Phi_calc(:,5),Phi_calc(:,6),'g','DisplayName','Object 2');

% object 3 full trajectory
plot(Phi_calc(:,9),Phi_calc(:,10),'b','DisplayName','Object 3');

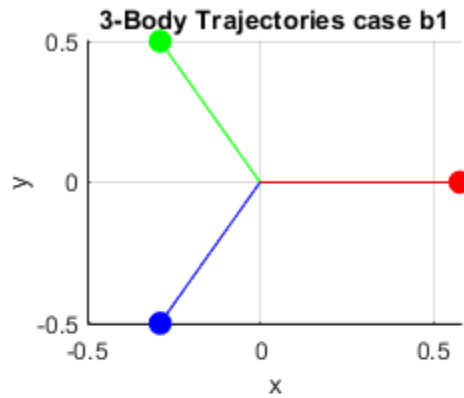
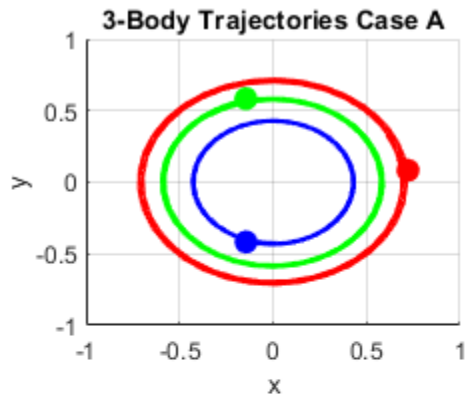
% starting positions
h1=plot(Phi_0(1),Phi_0(2), 'ro', 'MarkerFaceColor','r', 'MarkerSize',8);
h2=plot(Phi_0(5),Phi_0(6), 'go', 'MarkerFaceColor','g', 'MarkerSize',8);
h3=plot(Phi_0(9),Phi_0(10), 'bo', 'MarkerFaceColor','b', 'MarkerSize',8);

hold off;

Warning: Failure at t=7.580176e-01. Unable to meet integration tolerances
without reducing the step size below the smallest value allowed
(1.776357e-15)
at time t.

```

---



## case b 2

```

obj_1=[1,0.5773502692, 0,0.0577350269, 0];
obj_2=[1,-0.2886751346, 0.5,-0.0288675135, 0.05];
obj_3=[1,-0.2886751346, -.5,-0.0288675135, -0.05];

obj_matrix=[obj_1;obj_2;obj_3];

%translate to state vector initial
% Phi =[r1x,r1y,v1x,v1y,r2x,r2y,v2x,v2y,r3x,r3y,v3x,v3y];
Phi_0=[];
for i=1:3
    for j=2:5
        Phi_0(end+1)=obj_matrix(i,j);
    end
end

Phi_0=Phi_0(:);

[t,Phi_calc]=ode45(@derivFunc1,timeSpan,Phi_0);

% plot

subplot(2,2,3)
hold on;

```

```

grid on;
xlabel('x'); ylabel('y');
title('3-Body Trajectories Case b2');

% making a 2d position plot, where time is not an axis
% object 1 full trajectory

plot(Phi_calc(:,1),Phi_calc(:,2),'r','DisplayName','Object 1');

% object 2 full trajectory
plot(Phi_calc(:,5),Phi_calc(:,6),'g','DisplayName','Object 2');

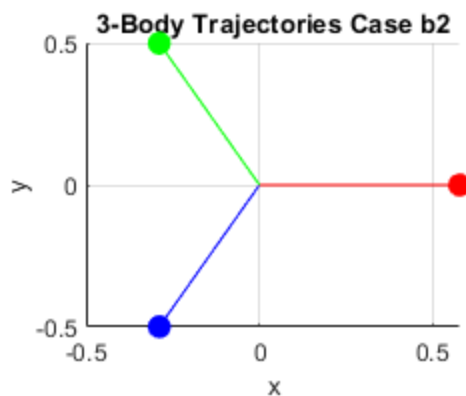
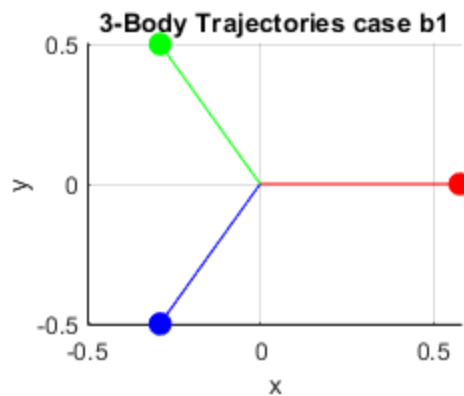
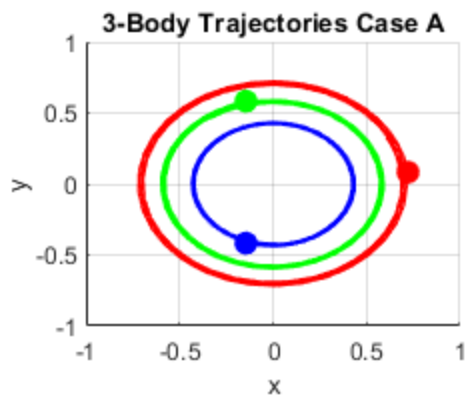
% object 3 full trajectory
plot(Phi_calc(:,9),Phi_calc(:,10),'b','DisplayName','Object 3');

% starting positions
h1=plot(Phi_0(1),Phi_0(2), 'ro', 'MarkerFaceColor','r', 'MarkerSize',8);
h2=plot(Phi_0(5),Phi_0(6), 'go', 'MarkerFaceColor','g', 'MarkerSize',8);
h3=plot(Phi_0(9),Phi_0(10), 'bo', 'MarkerFaceColor','b', 'MarkerSize',8);

hold off;

```

*Warning: Failure at t=6.762897e-01. Unable to meet integration tolerances without reducing the step size below the smallest value allowed (1.776357e-15) at time t.*



---

## case c

```
obj_1=[1,0.97000436, -0.24308753,0.4662036850, 0.43236573];
obj_2=[1,-0.97000436, 0.24308753,0.4662036850, 0.43236573];
obj_3=[1,0.0, 0.0,-0.93240737, -0.86473146];

obj_matrix=[obj_1;obj_2;obj_3];

%translate to state vector initial
% Phi =[r1x,r1y,v1x,v1y,r2x,r2y,v2x,v2y,r3x,r3y,v3x,v3y];
Phi_0=[];
for i=1:3
    for j=2:5
        Phi_0(end+1)=obj_matrix(i,j);
    end
end

Phi_0=Phi_0(:);

[t,Phi_calc]=ode45(@derivFunc1,timeSpan,Phi_0);

% plot

subplot(2,2,4)
hold on;
grid on;
xlabel('x'); ylabel('y');
title('3-Body Trajectories Case c');

% making a 2d position plot, where time is not an axis
% object 1 full trajectory

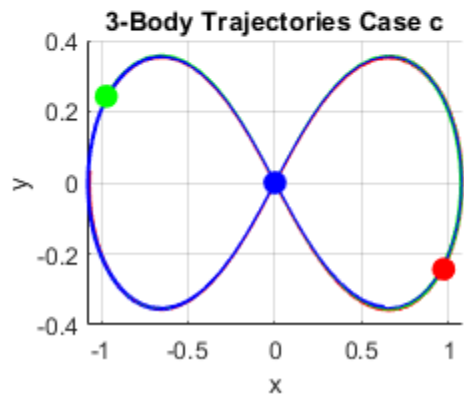
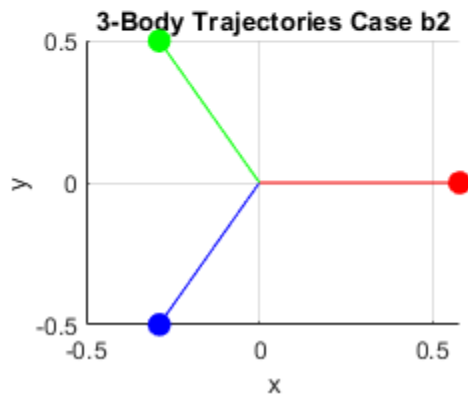
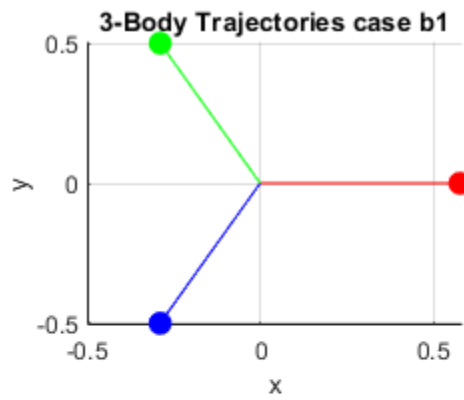
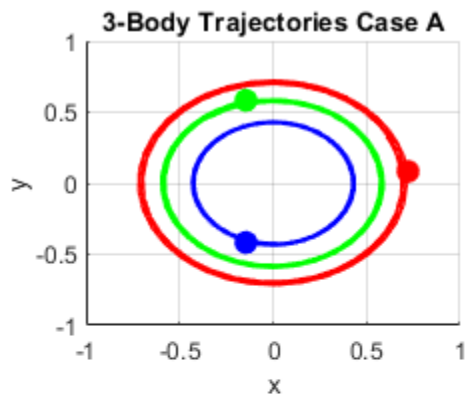
plot(Phi_calc(:,1),Phi_calc(:,2),'r','DisplayName','Object 1');

% object 2 full trajectory
plot(Phi_calc(:,5),Phi_calc(:,6),'g','DisplayName','Object 2');

% object 3 full trajectory
plot(Phi_calc(:,9),Phi_calc(:,10),'b','DisplayName','Object 3');

% starting positions
h1=plot(Phi_0(1),Phi_0(2), 'ro', 'MarkerFaceColor','r', 'MarkerSize',8);
h2=plot(Phi_0(5),Phi_0(6), 'go', 'MarkerFaceColor','g', 'MarkerSize',8);
h3=plot(Phi_0(9),Phi_0(10), 'bo', 'MarkerFaceColor','b', 'MarkerSize',8);

hold off;
```



*Published with MATLAB® R2024b*