

# Homework3.3

Blue

20230224

## 1 Circular Orbit

Using the initial condition, we get the circular orbit. Although it's not a perfect circle, it looks close to it. Also, we have the energy of the planet. It has accuracy to 9 decimal places.

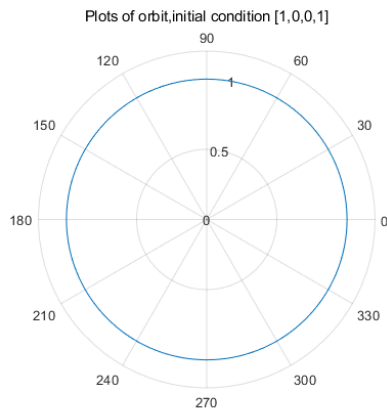


Figure 1: The circular orbit

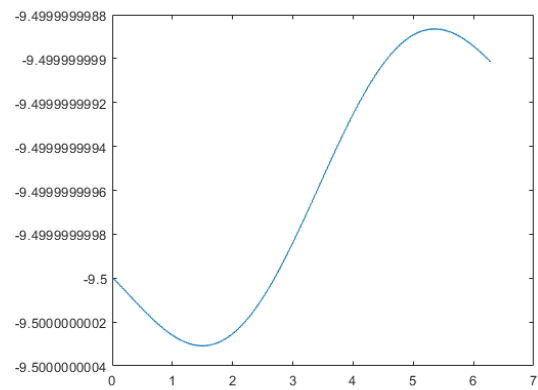


Figure 2: The energy of circular orbit planet

If we change the initial condition. We can get a ellipse orbit. The energy of the ellipse orbit planet changes periodically.

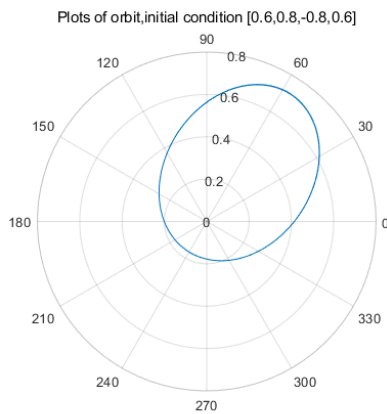


Figure 3: The Elliptical Orbit

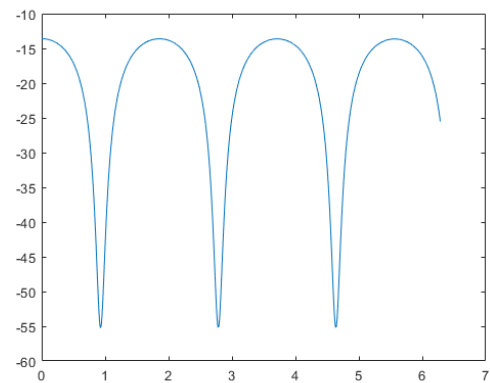


Figure 4: The energy of Elliptical Orbit planet

## 2 Elliptical Orbit

First we use forward Euler's method, it's not a good method as shown below. The orbit and the energy changes as the time grows. We set the initial condition as  $r = [0.36, 0.64], v = [-0.48, 0.48]$ .

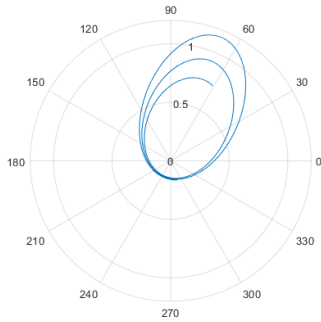


Figure 5: The Elliptical Orbit using forward Euler's method.

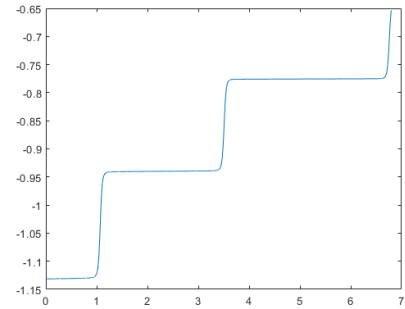


Figure 6: The energy of Elliptical Orbit planet

Then we try Verlet's method, whose accuracy is much better than Euler's method.

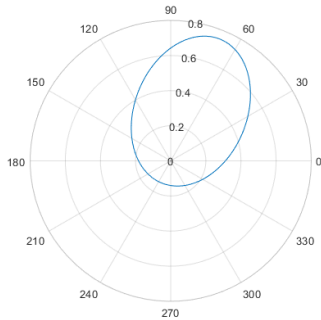


Figure 7: The Elliptical Orbit using Verlet's method

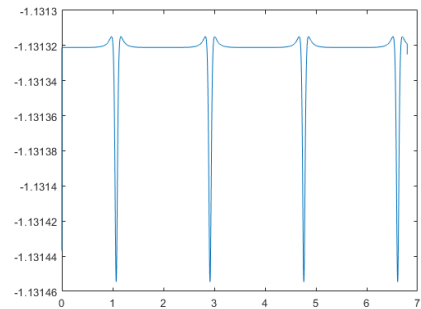


Figure 8: The energy of Elliptical Orbit planet

Finally, we try Ode45 which is the most accurate one.

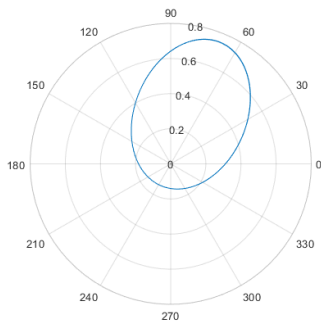


Figure 9: The Elliptical Orbit using ode45

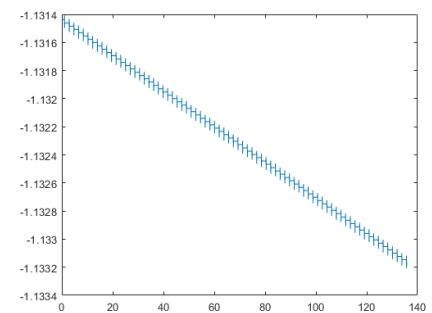


Figure 10: The energy of Elliptical Orbit planet

If we set the initial speed larger than 1. We get a bad result, showing that the planet is escaping. We choose condition as  $r = [0.72, 1.2], v = [-0.96, 0.9]$ . Even the most accurate one ode45 shows that it will escape. The other two methods shows similarly.

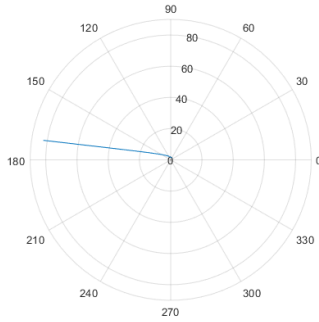


Figure 11: The Escaping planet using ode45

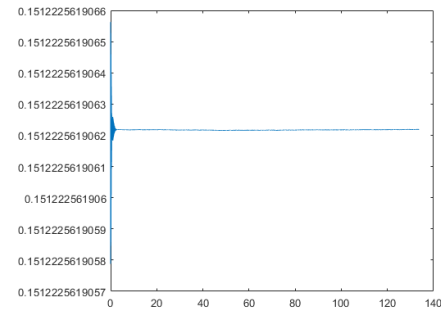


Figure 12: The energy of Escaping planet

If I need to choose between three methods, I will choose Ode45 since its a function built in MATLAB. Also it should be the most accurate one.

### 3 Multiple planets & Movies

I produce two movies. One is elliptical orbit, another is cicular orbit. I choose initial condition carefully. The scripts are attached to the end of the document.

## 4 code

### 4.1 Circular Orbit

```

1 G=1;
2 m0=10;
3 n0=2;
4 t0=2*pi;
5 r0=[1,0,0,1];
6 r1=[0.4,0.6,-0.6,0.4];
7 opt1=odeset('MaxStep',t0/200);
8 [T1,R1]=ode45(@(t,r) myode(t,r,n0),[0 t0],r0,opt1);
9 [T2,R2]=ode45(@(t,r) myode(t,r,n0),[0 t0],r1,opt1);
10 [theta1,rho1]=cart2pol(R1(:,1),R1(:,2));
11 figure(1)
12 polarplot(theta1,rho1);
13 [theta2,rho2]=cart2pol(R2(:,1),R2(:,2));
14 title('Plots of orbit, initial condition [1,0,0,1]');
15 figure(2)
16 polarplot(theta2,rho2);
17 title('Plots of orbit, initial condition [0.6,0.8,-0.8,0.6]');
18 figure(3)
19 E=zeros(size(T1));
20 [m,n]=size(T1);
21 for i=1:m
22     E(i,:)=G*m0/norm(R1(i,1:2))+(norm(R1(i,3:4))^2)/2;
23 end
24 plot(T1,E)
25 figure(4)

```

```

26 E=zeros ( size (T2) );
27 [m,n]=size (T2) ;
28 for i=1:m
29     E(i,:)=-G*m0/norm(R2(i,1:2))+(norm(R2(i,3:4))^2)/2;
30 end
31 plot (T2,E)
32 function drdt=myode(t,r,n)%in r and drdt 1-n is the original function ,n+1-2n is
    derivative
33 G=1;
34 m0=1.0;
35 drdt=zeros (2*n,1) ;
36 r2=norm (r (1:n)) ;
37 for i=1:n
38     drdt (i)=r (n+i) ;
39     drdt (n+i)=-G*m0*r (i) / (r2^3) ;
40 end
41 end

```

## 4.2 Elliptical Orbit

```

1 G=1;
2 m0=1;
3 n0=2;
4 N=10000;
5 %r and v are row vector each one
6 r=zeros (N+1,n0) ;
7 v=zeros (N+1,n0) ;
8 a=0.6;
9 b=0.8;
10 cos1=0.6;
11 sin1=0.8;
12 % a=1;
13 % b=1;
14 % cos1=sqrt (2)/2;
15 % sin1=sqrt (2)/2;
16 r (1,:)=[a*cos1,b*sin1] ;
17 v (1,:)=[-a*sin1,b*cos1] ;
18 T=2*pi*norm (r (1,:)) / norm (v (1,:)) ;
19 dt=T/N;
20 for i=2:N+1
21     r (i,:)=r (i-1,:)+dt*v (i-1,:);
22     v (i,:)=v (i-1,:)+dt*(-G*m0*r (i-1,:)/(norm (r (i-1,:))^3);
23 end
24 figure (1)
25 [theta1,rho1]=cart2pol (r (:,1),r (:,2)) ;
26 polarplot (theta1,rho1) ;
27 figure (2)
28 t1=linspace (0,T,N+1);
29 E=zeros (N+1,1);
30 for i=1:N+1
31     E(i,:)=-G*m0/norm (r (i,:)) +(norm (v (i,:))^2)/2;
32 end
33 plot (t1,E);
34
35 %Verlet 's method

```

```

36 r1=zeros(N+1,n0);
37 v1=zeros(N+1,n0);
38 r1(1:2,:)=r(1:2,:);
39 v1(1:2,:)=v(1:2,:);
40 for i=3:N+1
41     r1(i,:)=2*r1(i-1,:)-r1(i-2,:)+dt^2*(-G*m0*r1(i-1,:)/(norm(r1(i-1,:))^3);
42     v1(i-1,:)=(r1(i,:)-r1(i-2,:))/(2*dt);
43 end
44 v1(N+1,:)=v1(N,:)+dt*(-G*m0*r1(N,:)/(norm(r1(N,:))^3);
45 figure(3)
46 [theta2,rho2]=cart2pol(r1(:,1),r1(:,2));
47 polarplot(theta2,rho2);
48 figure(4)
49 t2=linspace(0,T,N+1);
50 E1=zeros(N+1,1);
51 for i=1:N+1
52     E1(i,:)=G*m0/norm(r1(i,:))+(norm(v1(i,:))^2)/2;
53 end
54 plot(t2,E1);
55
56
57 %ode45
58 r0=[a*cos1,b*sin1,-a*sin1,b*cos1];
59 opt1=odeset('MaxStep',T/500);
60 [T1,R1]=ode45(@(tk,rk) myode(tk,rk,n0),[0 20*T],r0,opt1);
61 figure(5)
62 [theta3,rho3]=cart2pol(R1(:,1),R1(:,2));
63 polarplot(theta3,rho3);
64 figure(6)
65 t2=linspace(0,T,N+1);
66 E2=zeros(size(T1));
67 [m,n]=size(T1);
68 for i=1:m
69     E2(i,:)=G*m0/norm(R1(i,1:2))+(norm(R1(i,3:4))^2)/2;
70 end
71 plot(T1,E2)
72 function drdt=myode(t,r,n)%in r and drdt 1-n is the original function,n+1-2n is
    derivative
73 G=1;
74 m0=1.0;
75 drdt=zeros(2*n,1);
76 r2=norm(r(1:n));
77 for i=1:n
78     drdt(i)=r(n+i);
79     drdt(n+i)=-G*m0*r(i)/(r2^3);
80 end
81 end

```

## 4.3 Multiple planets and Movies

### 4.3.1 Elliptical Orbit

```

1 n0=2;
2 t0=8*pi;
3 a=0.8;
4 b=1.4;

```

```

5  cos1=0.6;
6  sin1=0.8;
7  r0=[0.4*a*cos1,0.4*b*sin1,-a*sin1/sqrt(0.4),b*cos1/sqrt(0.4)];
8  r1=[0.45*a*cos1,0.45*b*sin1,-a*sin1/sqrt(0.45),b*cos1/sqrt(0.45)];
9  r2=[0.6*a*cos1,0.6*b*sin1,-a*sin1/sqrt(0.6),b*cos1/sqrt(0.6)];
10 r3=[0.8*a*cos1,0.8*b*sin1,-a*sin1/sqrt(0.8),b*cos1/sqrt(0.8)];
11 r4=[a*cos1,b*sin1,-a*sin1,b*cos1];
12 opt1=odeset('MaxStep',2*pi/50);
13 sol0=ode45(@(t,r) myode(t,r,n0),[0 t0],r0,opt1);
14 sol1=ode45(@(t,r) myode(t,r,n0),[0 t0],r1,opt1);
15 sol2=ode45(@(t,r) myode(t,r,n0),[0 t0],r2,opt1);
16 sol3=ode45(@(t,r) myode(t,r,n0),[0 t0],r3,opt1);
17 sol4=ode45(@(t,r) myode(t,r,n0),[0 t0],r4,opt1);
18 Nt=400;
19 t=linspace(0,t0,Nt);
20 R0=deval(sol0,t);
21 R1=deval(sol1,t);
22 R2=deval(sol2,t);
23 R3=deval(sol3,t);
24 R4=deval(sol4,t);
25
26 vidobj = VideoWriter('movie1.mp4','mpeg-4');
27 open(vidobj);
28 for k=1:Nt
29     scatter(0,0,50,'rd','markerfacecolor','r'); hold on;
30     scatter(R0(1,k),R0(2,k),20,'ok','markerfacecolor','k');
31     scatter(R1(1,k),R1(2,k),20,'ok','markerfacecolor','y');
32     scatter(R2(1,k),R2(2,k),40,'ok','markerfacecolor','b');
33     scatter(R3(1,k),R3(2,k),40,'ok','markerfacecolor','r');
34     scatter(R4(1,k),R4(2,k),80,'ok','markerfacecolor','m');
35     plot(R0(1,1:80),R0(2,1:80));
36     plot(R1(1,:),R1(2,:));
37     plot(R2(1,:),R2(2,:));
38     plot(R3(1,:),R3(2,:));
39     plot(R4(1,:),R4(2,:));
40     xlim([-3 1]);
41     ylim([-1.3 2.1]);
42     pbaspect([100 100 1]);
43     set(gca,'FontSize',20);
44     set(gcf,'color','w');
45     box on;
46     title('My solar system');
47     currFrame = getframe(gcf);
48     hold off;
49     writeVideo(vidobj,currFrame);
50 end
51 close(vidobj);
52
53
54 function drdt=myode(t,r,n)%in r and drdt 1-n is the original function,n+1-2n is
    derivative
55 G=1;
56 m0=1.0;
57 drdt=zeros(2*n,1);
58 r2=norm(r(1:n));
59 for i=1:n

```

```

60     drdt(i)=r(n+i);
61     drdt(n+i)=-G*m0*r(i)/(r2^3);
62 end
63 end

```

### 4.3.2 Circular Orbit

```

1  n0=2;
2  t0=2*pi;
3  r0=[0.2,0,0,sqrt(1/0.2)];
4  r1=[0.4,0,0,sqrt(1/0.4)];
5  r2=[0.6,0,0,sqrt(1/0.6)];
6  r3=[0.8,0,0,sqrt(1/0.8)];
7  r4=[1,0,0,1];
8  opt1=odeset('MaxStep',2*pi/50);
9  sol0=ode45(@(t,r) myode(t,r,n0),[0 t0],r0,opt1);
10 sol1=ode45(@(t,r) myode(t,r,n0),[0 t0],r1,opt1);
11 sol2=ode45(@(t,r) myode(t,r,n0),[0 t0],r2,opt1);
12 sol3=ode45(@(t,r) myode(t,r,n0),[0 t0],r3,opt1);
13 sol4=ode45(@(t,r) myode(t,r,n0),[0 t0],r4,opt1);
14 Nt=300;
15 t=linspace(0,t0,Nt);
16 R0=deval(sol0,t);
17 R1=deval(sol1,t);
18 R2=deval(sol2,t);
19 R3=deval(sol3,t);
20 R4=deval(sol4,t);
21
22 vidobj = VideoWriter('movie.mp4','mpeg-4');
23 open(vidobj);
24 for k=1:Nt
25     scatter(0,0,100,'rd','markerfacecolor','r'); hold on;
26     scatter(R0(1,k),R0(2,k),20,'ok','markerfacecolor','k');
27     scatter(R1(1,k),R1(2,k),20,'ok','markerfacecolor','y');
28     scatter(R2(1,k),R2(2,k),40,'ok','markerfacecolor','b');
29     scatter(R3(1,k),R3(2,k),40,'ok','markerfacecolor','r');
30     scatter(R4(1,k),R4(2,k),80,'ok','markerfacecolor','m');
31     plot(R0(1,1:80),R0(2,1:80));
32     plot(R1(1,:),R1(2,:));
33     plot(R2(1,:),R2(2,:));
34     plot(R3(1,:),R3(2,:));
35     plot(R4(1,:),R4(2,:));
36     xlim([-1 1]);
37     ylim([-1 1]);
38     pbaspect([100 100 1]);
39     set(gca,'FontSize',20);
40     set(gcf,'color','w');
41     box on;
42     title('My_solar_system');
43     currFrame = getframe(gcf);
44     hold off;
45     writeVideo(vidobj,currFrame);
46 end
47 close(vidobj);
48

```

```

49 function drdt=myode(t,r,n)%in r and drdt 1-n is the original function,n+1-2n is
    derivative
50 G=1;
51 m0=1.0;
52 drdt=zeros(2*n,1);
53 r2=norm(r(1:n));
54 for i=1:n
55     drdt(i)=r(n+i);
56     drdt(n+i)=-G*m0*r(i)/(r2^3);
57 end
58 end

```