Metody numeryczne

Lista 10

Katarzyna Korsak 229707

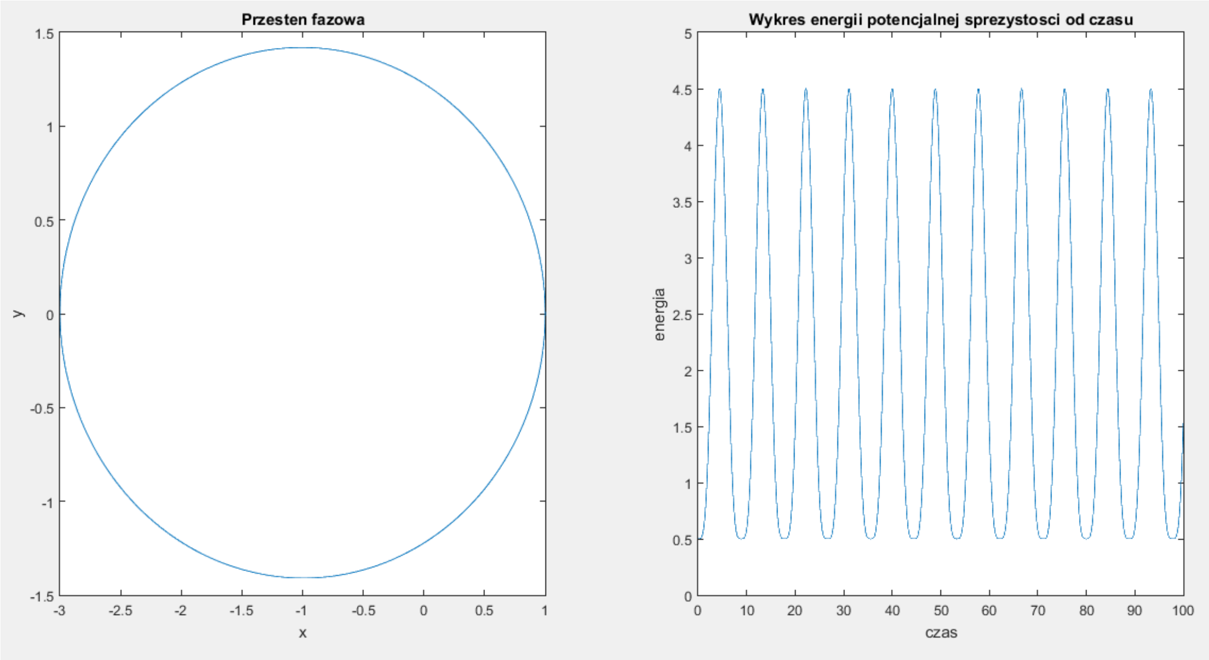
Piątek 7^{30}

Zadanie 1

1.1 Rozwiązanie zadania

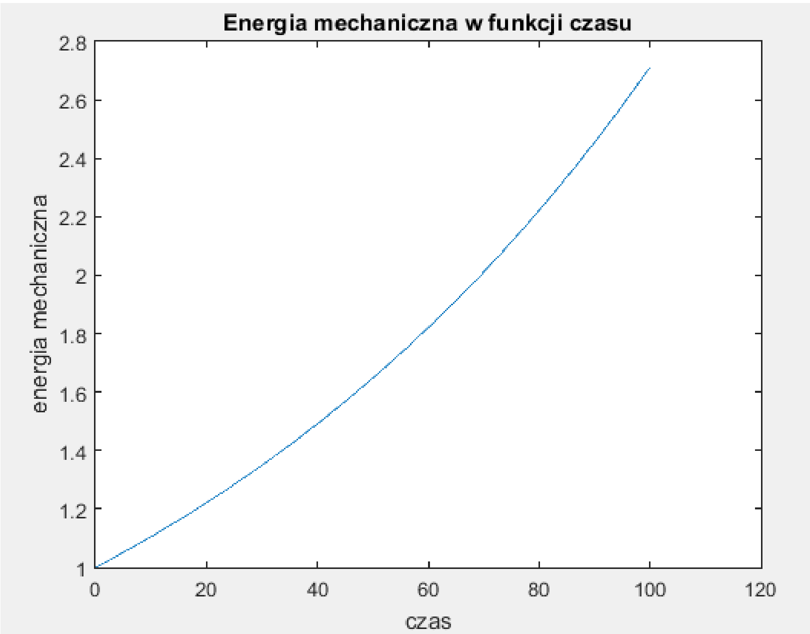
```
zad1.m
clc,clear;

h=0.01;
x=1;
t=0;
v=0;
k = 1; m = 1;
a=-x;

xVec=[];
vVec=[];
tVec=[];

for i=1:10000

    xVec=[xVec;x];
    vVec=[vVec;v];
    tVec=[tVec;t];

    vmid = v + h*a * 0.5;
    x = x + h*vmid;
    aNew = -x;
    v = v + h*0.5*(a + aNew);

    t = t + h;

end

figure;
plot(xVec,vVec);

E=(xVec.*xVec+vVec.*vVec)/2;
figure;
plot(tVec,E);
```

1.2 Wyniki



Rysunek 1. Wykresy przestrzeni fazowej i energii mechanicznej dla metody Verleta.

1.3 Wnioski



Rysunek 2. Wykres energii mechanicznej dla metody Eulera

Porównując wykresy energii obiektu dla metody Verleta i Eulera można stwierdzić, że pierwsza z nich dała większy rozrzut wartości, jednak jest to wykres oscylacji i prawidłowa energia jest osiągana. Dla drugiej z metod odchyłek jest mniejszy, jednak jest to zależność w funkcji rosnącej.

Zadanie 2

2.1 Rozwiązanie zadania

kepler.m

```
function [ tVec,yVec,xVec,vxVec,vyVec,rmin,rmax ] = kepler(
x0,y0,vx0,vy0,t0,tk,h )
%ax = -GM*x0/rmin^3;
%ay = -GM*y0/rmin^3;

xVec=[];
yVec=[];
vxVec=[];
vyVec=[];
tVec=[];

xVec(1)=x0;
yVec(1)=y0;
vxVec(1)=vx0;
vyVec(1)=vy0;
tVec(1)=t0;
t=0;
x=x0;
y=y0;
vx=vx0;
vy=vy0;

GM=4*pi^2;
rmin=sqrt(x*x+y*y);
n = floor((tk-t0)/h);

for i=1:n;
    r=sqrt(x*x+y*y);
    ax = -GM*x/(r^3);
   ay = -GM*y/(r^3);
   vxmid=vx+h*ax*0.5;
   vymid=vy + h*ay*0.5;
   x=x+h*vxmid;
   y=y+h*vymid;
    r=sqrt(x*x+y*y);
   ax = -GM*x/(r^3);
    ay = -GM*y/(r^3);
    vx=vxmid+h*0.5*ax;
    vy=vymid+h*0.5*ay;

    xVec=[xVec,x];
   yVec=[yVec,y];
   vxVec=[vxVec,vx];
   vyVec=[vyVec,vy];
   t=t+h;
   tVec=[tVec,t];

 if i==n
   rmax = sqrt(x^2+y^2);
 end
end

end
```

```
zad.2.m
clc,clear all;
x0 = 1;
y0 = 0;
vx0 = 0;
vy0 = 2*pi;
h=0.01;
t0=0;
tk=10;
[ tVec,yVec,xVec,vxVec,vyVec,rmin,rmax ] = kepler( x0,y0,vx0,vy0,t0,tk,h );
r = (rmin+rmax)/2;
T = tVec(find(xVec>=(max(xVec)-0.05*max(xVec))));
okres = T(2)-T(1);
kepler = okres^2/r^2;
a = max([max(xVec), max(yVec)]);
figure
subplot(3,1,1)
plot(xVec, yVec);
title('Przetrzen fazowa');
xlabel('x [au]')
ylabel('y [au]')
subplot(3,1,2)
plot(tVec, xVec);
title('Polozenie x');
xlabel('t [lata]')
ylabel('x [au]')
subplot(3,1,3)
plot(tVec, yVec);
title('Polozenie y');
xlabel('t [lata]')
ylabel('y [au]')
```

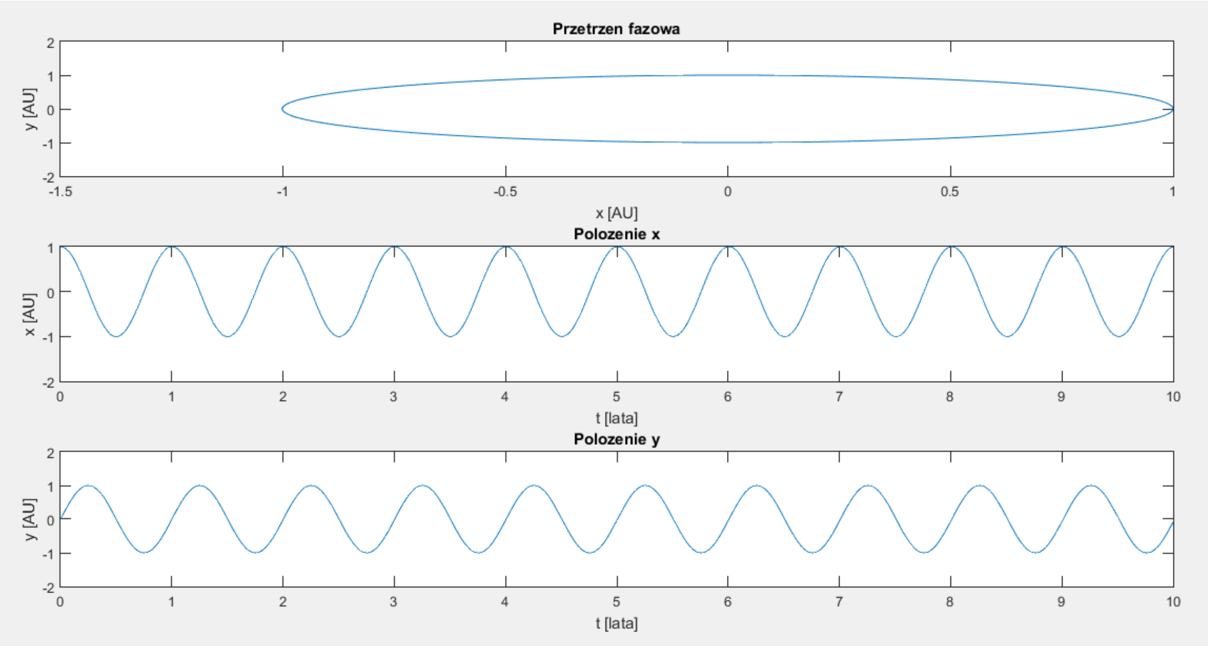
Dla założenia $v_x(0)=0$ jedna z półosi pokrywa się z osią x i jest większą półosią elipsy. By obliczyć okres znaleziono dwa punkty x o maksymalnej wartości, z możliwością błędu równą 5%. Do nich dopasowano odpowiednie czasy wektora tVec.

W celu sprawdzenia spełnienia III prawa Kepplera dla kilkunastu różnych wartości początkowych wykorzystano następujący kod:

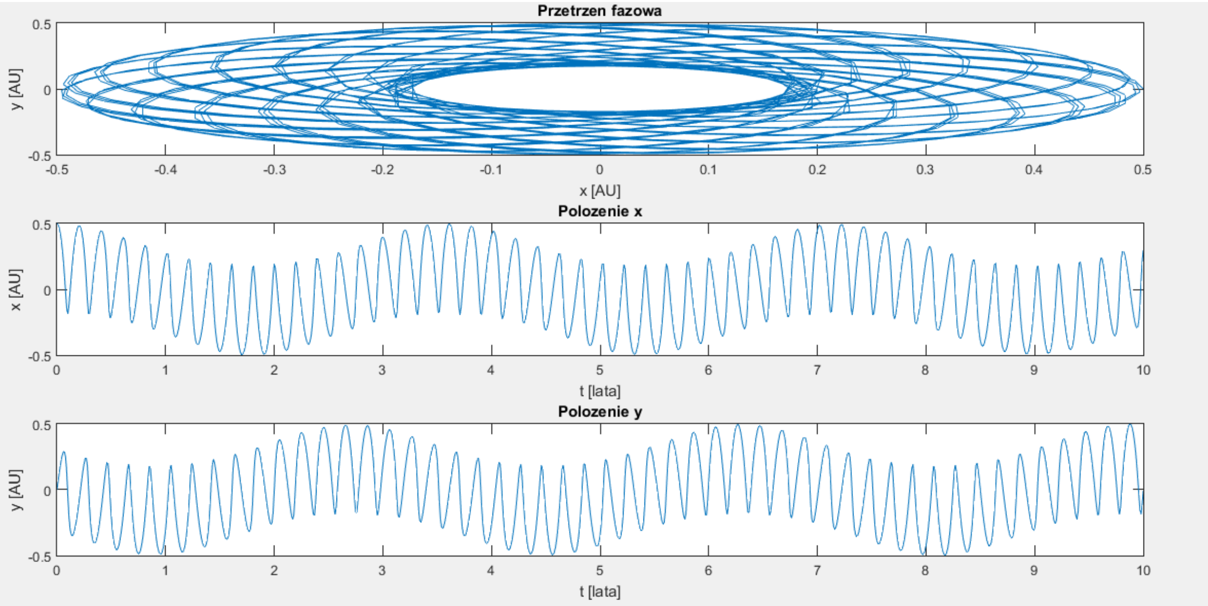
```
zad2b.m
clc,clear all;
vx0=0;
t0=0;
tk=10;
h=0.01;
x0=[1,0.5,3,0,2,1,1,5,2.5,3,0.5,6,0];
y0=[0,1,2,3,0.5,1.5,3,4,0.2,0.1,2.4,1.7,0];
vy0=[2*pi,pi,0,pi/2,2*pi,pi,3/4*pi,4*pi,0,pi,2*pi,3/2*pi,pi];
ak=[];
Tk=[];
for i=1:13
[ tVec,yVec,xVec,vxVec,vyVec ] = kepler( x0(i),y0(i),vx0,vy0(i),t0,tk,h );
[pks,loc,w,p] = findpeaks(yVec);
T=tVec(loc(2))-tVec(loc(1));
a=(max(xVec)-min(xVec))/2;
ak=[ak;a];
Tk=[Tk;T];
vy0=vy0+0.1;
end
tL=log(Tk);
```

```
aL=3/2*log(ak);
figure;
plot(tL,aL);
```

2.2 Wyniki



Rysunek 3. Wykresy dla danych początkowych: x(0)=1, y(0)=0, vx(0)=0, vy(0)=2Π.



Rysunek 4. Wykresy po zmianie parametru x(0) na wartość 0.5.

2.3 Wnioski

Wyznaczone wartości okresów i długości półosi powinny na wykresie logarytmicznym wyznaczyć zależność liniową, zgodnie z III prawem Kepplera, gdzie na jednej osi odkłada się wartość logarytmu okresu a na drugiej 3/2 wartości logarytmu z wartości wielkości półosi.

Zadanie 3

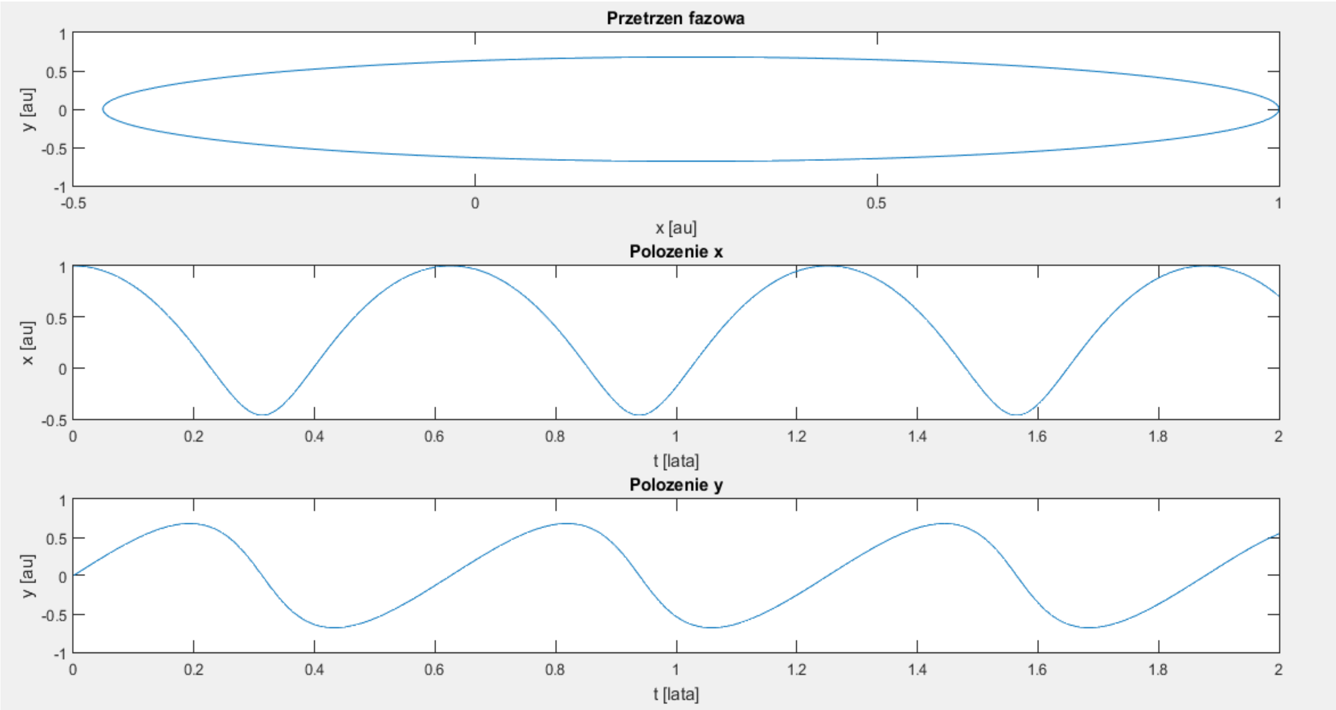
3.1 Rozwiązanie zadania

```
ode.m
function [dxdt] = ode(t, a)
GM = 4*pi^2;
dxdt = [a(2);(-GM/sqrt(a(1)^2+a(3)^2)^3)*a(1);a(4);(-
GM/sqrt(a(3)^2+a(1)^2)^3)*a(3)];
end
```

```
zad3.m
opts = odeset('RelTol', 1e-6, 'AbsTol', 1e-6);
[t, xODE] = ode45(@kepler, [0 2], [1 0 0 5], opts);
x = xODE(:,1);
y = xODE(:,3);
figure;
subplot(3,1,1);
plot(x,y);
xlabel('x [au]');
ylabel('y [au]');
title('Przetrzen fazowa');
subplot(3,1,2);
plot(t,x);
title('Polozenie x');
xlabel('t [lata]');
ylabel('x [au]');
subplot(3,1,3);
plot(t,y);
title('Polozenie y');
xlabel('t [lata]');
ylabel('y [au]');
```

3.2 Wyniki

Funkcja *kepler* zwraca wektor kolumnowy, w którym kolumny oznaczają kolejno odległość na osi x od Słońca, prędkość w kierunku x, odległość na osi y od Słońca, prędkość w kierunku y. Zastosowanie funkcji *odeset* umożliwia uwzględnienie tolerancji błędu.



Rysunek 5. Wykresy przestrzeni fazowej i położeń dla warunków początkowych: x(0)=1, y(0)=0, vx(0)=0, vy(0)=5.