

Wizualizacja ruchu planet

Konrad Kapłański, Filip Jurczak, Karolina Krupińska

Politechnika Krakowska im. Tadeusza Kościuszki

Wydział Fizyki Matematyki i Informatyki

Fizyka techniczna VI semestr

22 czerwca 2018

Zdefiniowanie problemu

Pierwszym, rozpatrywanym przez nas problemem, będzie przedstawienie orbity planety, krążącej wokół masywnego Słońca.
Kolejnym problemem będzie symulacja ruchu dwóch planet, działających na siebie nawzajem.

Siła grawitacji

Siła działająca na ciała w obu układach, to siła grawitacji, opisana wzorem:

$$F=GMmr^{-2}$$

Zadajemy położenia i prędkości początkowe planet, a program symuluje ich ruch

Prawa Keplera

- Każda planeta krąży po orbicie eliptycznej, a Słońce znajduje się w jednym z dwóch ognisk elipsy

Prawa Keplera

- Każda planeta krąży po orbicie eliptycznej, a Słońce znajduje się w jednym z dwóch ognisk elipsy
- Promień wodzący poprowadzony ze środka Słońca do środka planety zakreśla równe pola powierzchni w równych odstępach czasu

Prawa Keplera

- Każda planeta krąży po orbicie eliptycznej, a Słońce znajduje się w jednym z dwóch ognisk elipsy
- Promień wodzący poprowadzony ze środka Słońca do środka planety zakreśla równe pola powierzchni w równych odstępach czasu
- Sześciiany wielkich półosi orbit jakichkolwiek dwóch planet mają się tak do siebie, jak kwadraty ich okresów obiegu

Pierwszy problem

Aby zobrazować problem, wykorzystaliśmy równania Netona, a następnie numerycznie wygenerowane dane przedstawiliśmy na wykresie

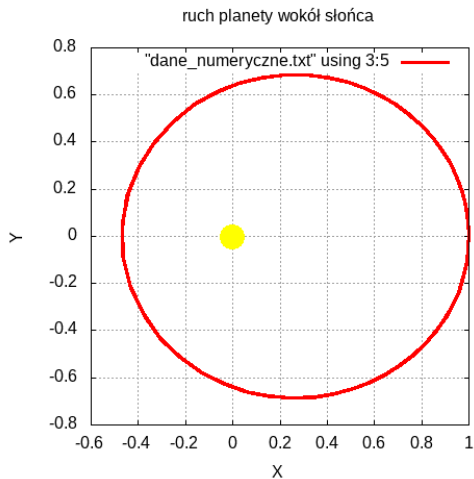
Równania Newtona opisujące ruch planety wokół słońca

$$r = \sqrt{x(t)^2 + y(t)^2}$$

$$x''(t) = -xr^{-3}$$

$$y''(t) = -yr^{-3}$$

Orbita



Rysunek: Tor ruchu planety

Algorytm Verleta

Algorytm Verleta – metoda numeryczna służąca do całkowania równań ruchu, czyli do obliczania położeń i prędkości układu oddziałujących ciał w funkcji czasu. Rutynowo wykorzystywany w symulacjach fizycznych (głównie w dynamice molekularnej), rzadziej w grafice komputerowej. Znany jest także pod innymi nazwami, np. jako jawna metoda różnic centralnych

Algorytm Verleta

algorytm prędkościowy:

$$r(t+\Delta t) = r(t) + \Delta t v(t + \Delta t/2)$$

$$v(t + \Delta t/2) = v(t - \Delta t/2) + \Delta t f(t)/m$$

gdzie $f(t)$ - siła w chwili t

Algorytm w naszym programie

```
for(int i=0;i<1000;i++)
{
//Metoda Verleta

    x1 += (vx1 + (aPoprzX1/2)*dt)*dt;
    y1 += (vy1 + (aPoprzY1/2)*dt)*dt;
    x2 += (vx2 + (aPoprzX2/2)*dt)*dt;
    y2 += (vy2 + (aPoprzY2/2)*dt)*dt;

    R = sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));

    aAktX1 = g*m2*(x2-x1)/R/R/R;
    aAktY1 = g*m2*(y2-y1)/R/R/R;
    aAktX2 = -g*m1*(x2-x1)/R/R/R;
    aAktY2 = -g*m1*(y2-y1)/R/R/R;

    vx1+=( (aPoprzX1 + aAktX1)/2)*dt;
    vy1+=( (aPoprzY1 + aAktY1)/2)*dt;
    vx2+=( (aPoprzX2 + aAktX2)/2)*dt;
    vy2+=( (aPoprzY2 + aAktY2)/2)*dt;

    aPoprzX1 = aAktX1;
    aPoprzY1 = aAktY1;
    aPoprzX2 = aAktX2;
    aPoprzY2 = aAktY2;
```

Rysunek: stosowanie metody Verleta

Zapis danych

```
#include <fstream>
#include <cmath>
#include <iostream>

using namespace std;

int main()
{
    double g = 100;
    double m1 = 5000, m2 = 5000;
    double x1 = -550, y1 = 700, x2 = -700, y2 = -200;
    double R = sqrt((x1-x2)*(x1-x2) + (y1-y2)*(y1-y2));
    double vx1 = -9, vy1 = -10, vx2 = -vx1*m1/m2, vy2 = -vy1*m1/m2;
    double aAktX1, aAktY1, aAktX2, aAktY2, aPoprzX1 = g*m2*(x2-x1)/R/R/R,
           aPoprzY1 = g*m2*(y2-y1)/R/R/R,
           aPoprzX2 = -g*m1*(x2-x1)/R/R/R,
           aPoprzY2 = -g*m1*(y2-y1)/R/R/R;

    double dt = 1;

    fstream file;
    file.open("data.txt",ios::out);
    file << x1 << "\t" << y1 << "\t" << x2 << "\t" << y2 << "\t" << vx1 << "\t" << aPoprzX1 << "\n";
}
```

Rysunek: Dzięki zapisowi danych do pliku, można później ich użyć

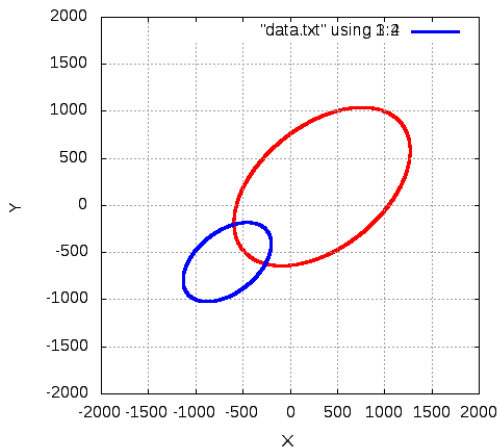
Użycie Gnuplota

```
# odczyt.txt

set term png
set output "orbita.png"
set xrange [-2000:2000]
set yrange [-2000:2000]
set xlabel "X"
set ylabel "Y"
set title "ruch dwóch planet"
set size ratio -1
set grid
set multi
set style data lines
set style fill solid
plot "data.txt" using 1:2 lw 2 linecolor rgb "red"
plot "data.txt" using 3:4 lw 2 linecolor rgb "blue"
quit
```

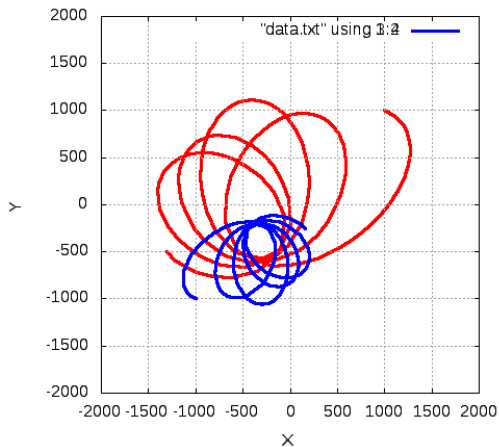
Rysunek: program wyświetlający za pomocą gnuplota, wyliczone orbity

Co otrzymaliśmy



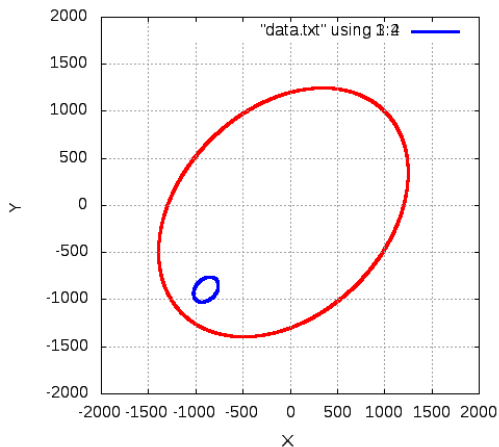
Rysunek: wizualizacja obliczonych orbit

Co otrzymaliśmy



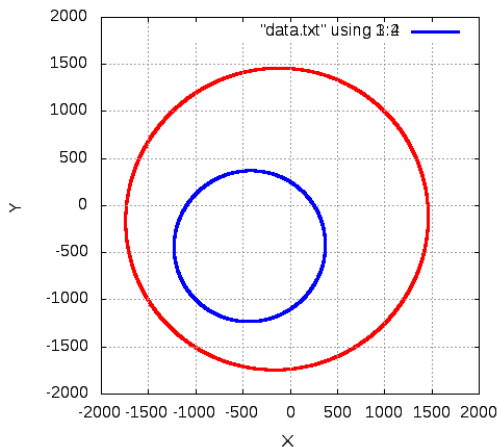
Rysunek: wizualizacja obliczonych orbit

Co otrzymaliśmy



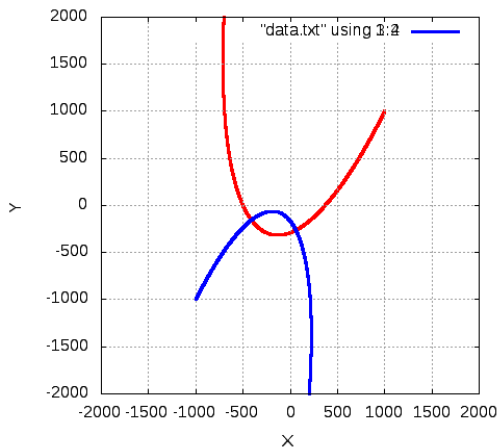
Rysunek: wizualizacja obliczonych orbit

beginframeCo otrzymaliśmy



Rysunek: wizualizacja obliczonych orbit

Co otrzymaliśmy



Rysunek: wizualizacja obliczonych orbit

Bibliografia

- ilf.fizyka.pw.edu.pl
- gnuplot.sourceforge.net
- wikipedia.org