Wahadło matematyczne

Marek Łukasiewicz

czerwiec 2017

Techniki komputerowe II - zadanie domowe nr 2

Zadanie: Różniczkowe równania ruchu wahadła matematycznego o dużym wychyleniu tzn. $\sin(\alpha) \neq \alpha$

Wyprowadzenie

Opiszmy ruch punktu materialnego we współrzędnych biegunowych o początku w punkcie zamocowania nici. Ponieważ nić jest nieważka i nierozciągliwa, stale pozostaje prosta, a punkt porusza się po okręgu o promieniu l. Na potrzeby tego modelu załóżmy że odchylenie nici od pionu nie przekracza $\frac{\pi}{2}$.

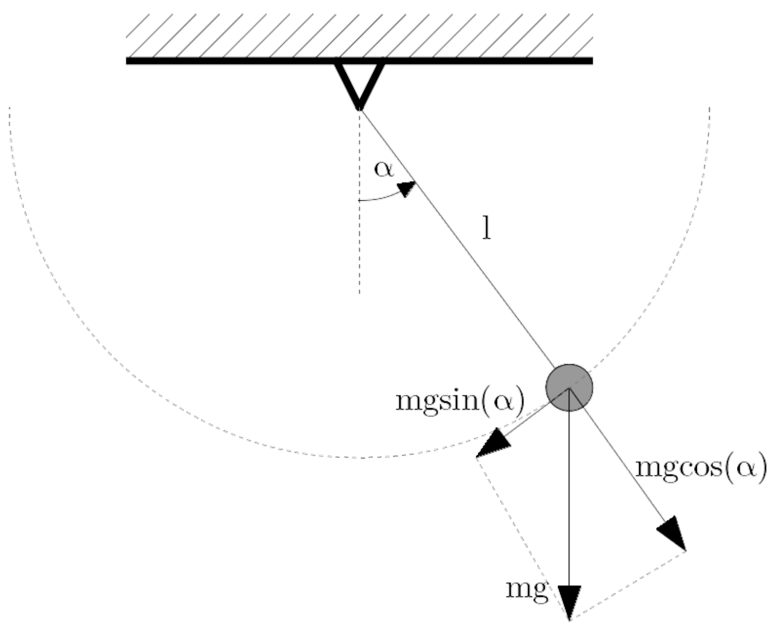


Figure 1:

Wtedy:

Równowaga sił na kierunku normalnym:

$$\sum F^n = 0$$
$$-N + m \cdot \cos(\alpha) = 0$$
$$N = m \cdot \cos(\alpha)$$

gdzie N to napięcie nici

Zasada zmienności krętu:

$$\overline{\varepsilon} \cdot I = \sum \overline{M}_0$$
$$I = m \cdot l^2$$
$$\varepsilon = \frac{d^2\alpha}{dt^2}$$
$$\sum M_0 = M_0(m \cdot \overline{g}) = -l \cdot m \cdot g \cdot \sin(\alpha)$$
$$\frac{d^2\alpha}{dt^2} = -\frac{g}{l}\sin(\alpha)$$

Jeśli dodatkowo wprowadzimy prędkość kątową $\omega = \frac{d\alpha}{dt}$, można przekształcić powyższe równanie różniczkowe II rzędu na układ dwóch równań I rzędu:

$$\begin{cases} \frac{d\omega}{dt} = -\frac{g}{l}\sin(\alpha) \\ \frac{d\alpha}{dt} = \omega \end{cases}$$

Energia mechaniczna

Energia mechaniczna jest sumą energii mechanicznej potencjalnej i kinetycznej:

$$E_m = E_p + E_k$$

Ponieważ jednorodne pole grawitacyjne które jest zadane w tym zadaniu jest polem potencjalnym, można liczyć energię potencjalną grawitacji jako różnicę potencjałów. Przyjmując za poziom odniesienia położenie równowagi, otrzymujemy:

$$E_p = m \cdot g \cdot h_{obecne} - m \cdot g \cdot h_{odniesienia} \implies E_p = m \cdot g \cdot \Delta h \implies E_p = m \cdot g \cdot l \cdot \Big(1 - cos(\alpha)\Big)$$

Energię kinetyczną obliczymy korzystając z podstawowego wzoru:

$$\begin{cases} E_k = \frac{m \cdot v^2}{2} \\ v = \omega \cdot l \end{cases} \implies E_k = \frac{1}{2} \cdot m \cdot \omega^2 \cdot l^2$$

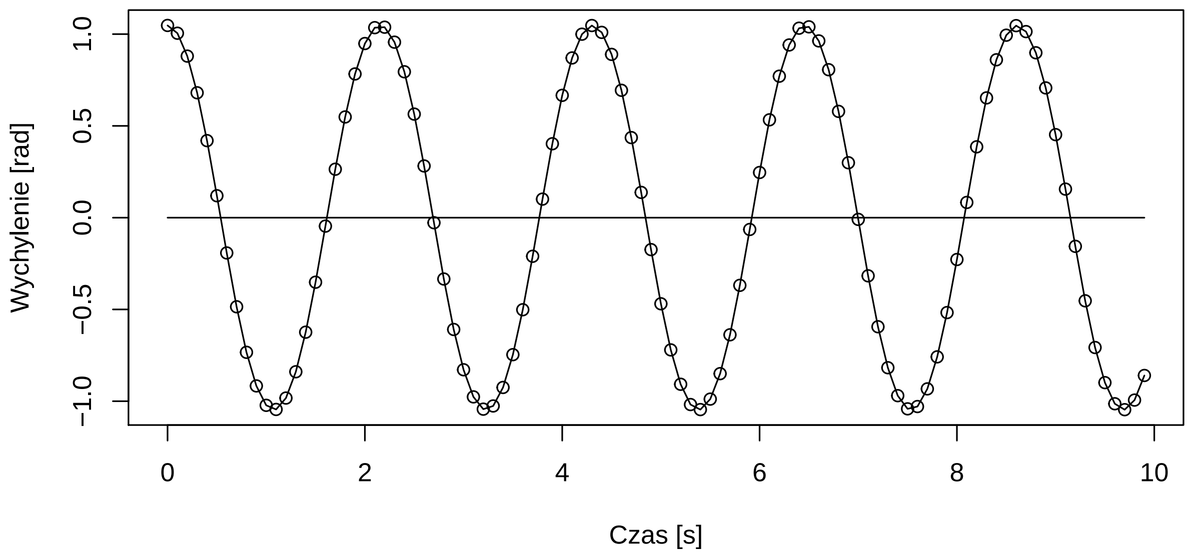
Ponieważ masa punktu nie wpływa na opis ruchu, jedynie na wartość energii mechanicznej, dla wygody przedstawimy energię po podzieleniu przez masę.

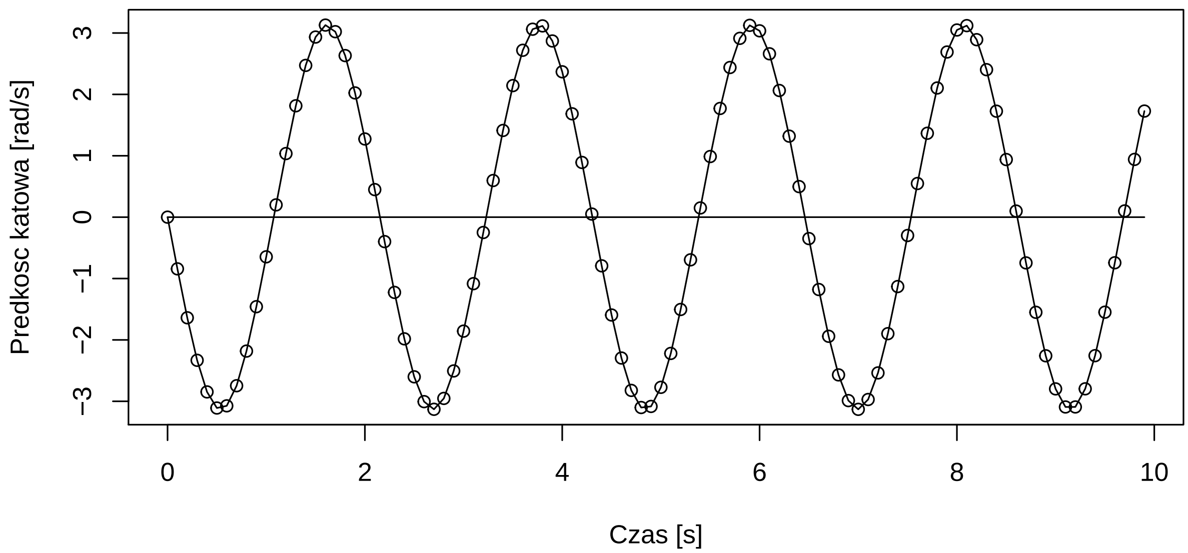
$$e_m = \frac{E_m}{m} = \frac{E_p + E_k}{m} = g \cdot l \cdot \Big(1 - cos(\alpha)\Big) + \frac{1}{2} \cdot \omega^2 \cdot l^2$$

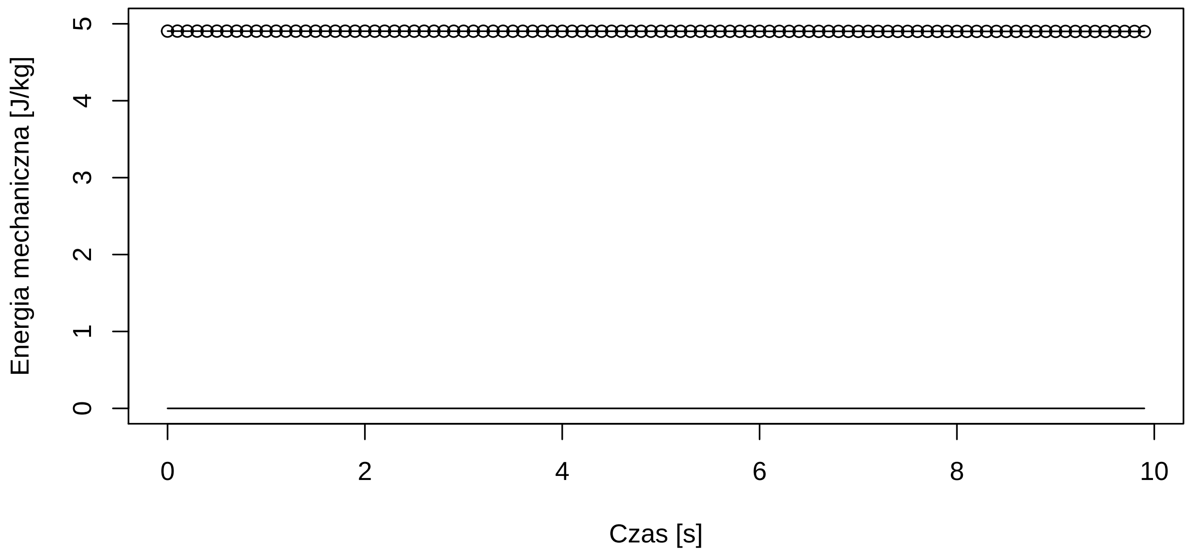
Przykładowe dane

Zestaw 1:

Przyspieszenie ziemskie $g = 9,81[\frac{m}{s^2}]$, wahadło długości $l = 1[m]$, o początkowym wychyleniu $\alpha_0 = \frac{\pi}{3} = 60°$, bez nadanej prędkości początkowej

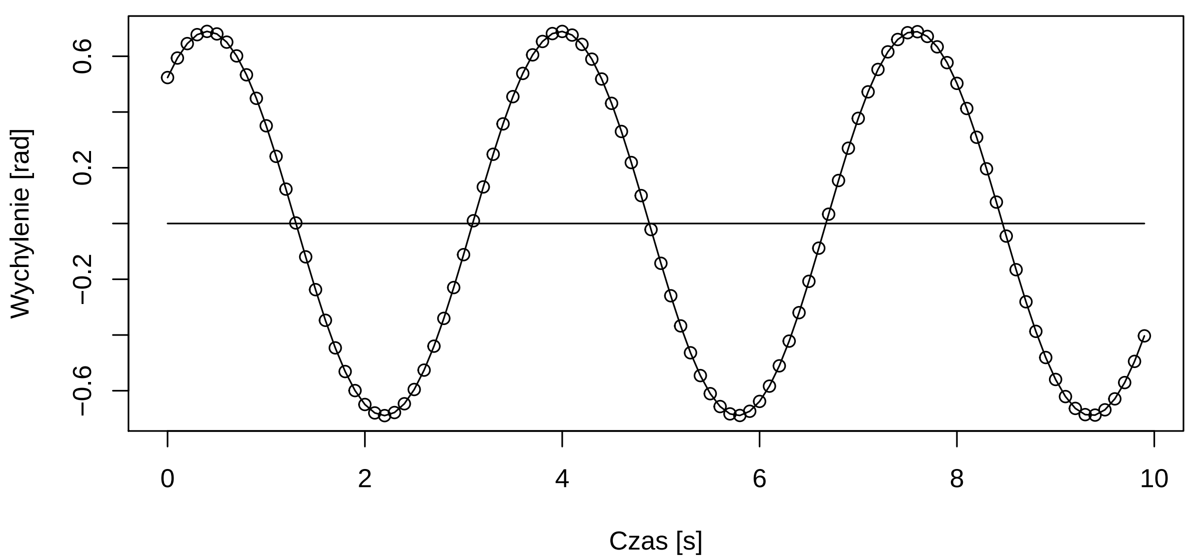


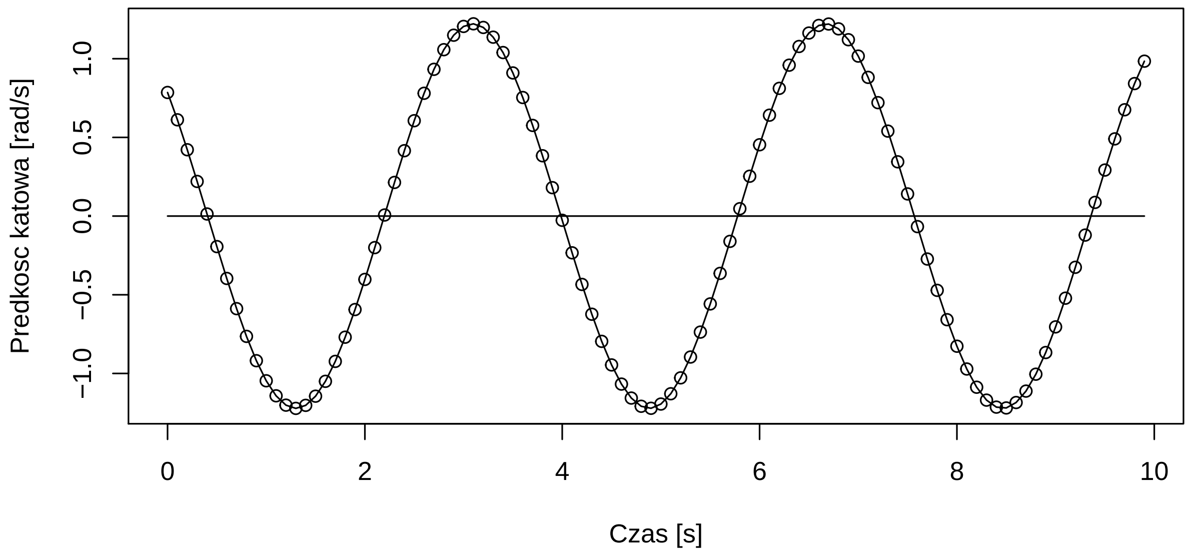


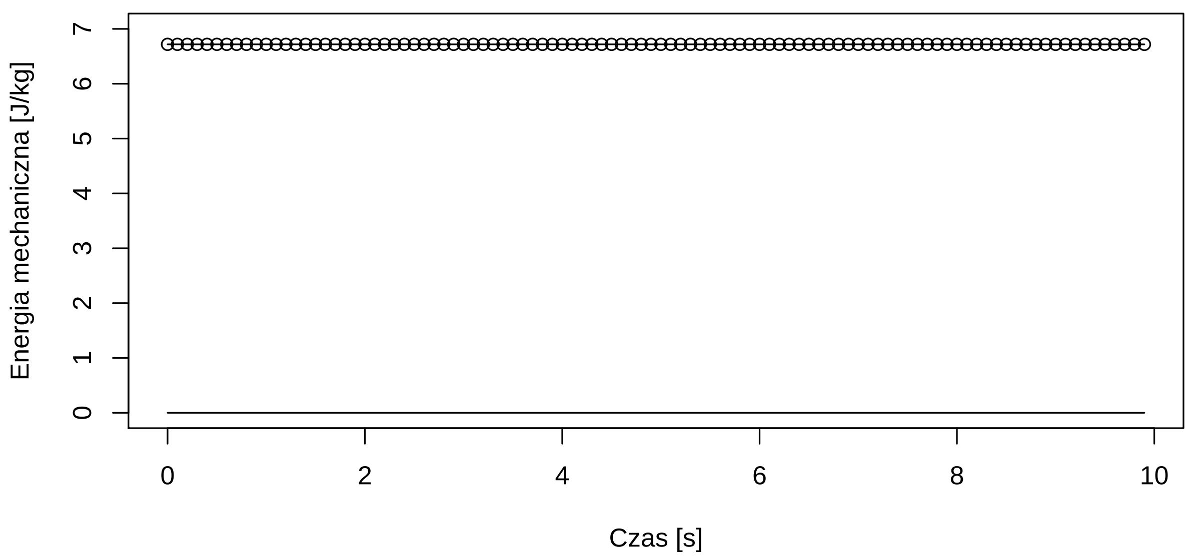


Zestaw 2:

Przyspieszenie ziemskie $g = 9,81[\frac{m}{s^2}]$, wahadło długości $l = 3[m]$, o początkowym wychyleniu $\alpha_0 = \frac{\pi}{6} = 30°$, z nadaną początkową prędkością $\omega_0 = \frac{\pi}{4}[\frac{1}{s}] = 45°[\frac{1}{s}]$




Predkosc katowa [rad/s]
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
Czas [s]


Energia mechaniczna [J/kg]
7
6
5
4
3
2
1
0
0
2
4
6
8
10
Czas [s]

Kod źródłowy rozwiązania

Kod źródłowy dostępny też pod adresem: https://github.com/Maarrk/wahadlo-projekt

Plik main.cpp:

```
#define _USE_MATH_DEFINES
#include <math.h>
#include <cstdlib>
#include <iostream>
#include <fstream>

// https://github.com/ccfd/courses/blob/master/code/info2/rk4.h
// https://github.com/ccfd/courses/blob/master/code/info2/rk4.cpp
#include "rk4.h"

using namespace std;

#define N 2 //ilość równań w układzie
#define ITER 110 //maksymalna ilość kroków całkowania

double g = 9.81;//przyspieszenie grawitacyjne
double l = 3;//długość wahadła

void rhs(double x, double y[], double out[]);//funkcja licząca prawe strony równań
double e_mech(double al, double om);//zwraca energię mechaniczną na kg masy

int main() {
    ofstream fout = ofstream("output.txt");

    double t0 = 0;//początkowy czas
    double tf = 10;//końcowy czas

    double al0 = M_PI / 6;//początkowe wychylenie
    double om0 = M_PI / 4;//początkowa prędkość kątowa

    cout << "Podaj dlugosc wahadla:\n";
    cin >> l;
    cout << "Podaj poczatkowe wychylenie:\n";
    cin >> al0;
    cout << "Podaj poczatkowa predkosc katowa:\n";
    cin >> om0;

    double h = 0.1;//krok całkowania

    double y[ITER][N] = { 0 };//tablica wartości wektorów [t][n]

    y[0][0] = al0;
    y[0][1] = om0;

    for (int i = 0; t0 + i*h < tf; i++) {
        vrk4(t0 + i*h, y[i], h, N, rhs, y[i + 1]);//krok całkowania
    }
```

main.cpp - ciąg dalszy

```
    cout << "t\talfa\tomega\tenergia\n";
    //lub fout żeby wypisać do pliku
    for (int i = 0; i < 100; i++) {
        cout << t0 + i*h << "\t" << y[i][0] << "\t" << y[i][1] << "\t" << e_mech(y[i][0], y[i][1]) << e
    }

    //system("notepad output.txt");
    system("pause");
    return 0;
}

void rhs(double x, double y[], double out[]) {
    out[0] = y[1];//alfa'
    out[1] = -g / l*sin(y[0]);//omega'
}

double e_mech(double al, double om) {
    double pot = g*(1 - cos(al))*l;//energia potencjalna grawitacji
    double kin = l*l*om*om / 2;//energia kinetyczna
    return pot + kin;
}
```