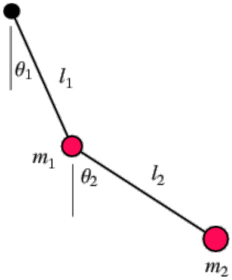
Wahadło podwójne. Chaos.

ab

28 listopada 2016

Tematyka

- Równania podwójnego wahadła
- Algorytmy całkowania
- Chaos



Równania ruchu wahadła możemy otrzymać z równań Eulera-Lagrange'a. Lagranżjan $L = T - V$ (Zadanie).

$$(m_1 + m_2) l_1 \ddot{\theta}_1 + m2 l_2 \ddot{\theta}_2 \cos(\delta) + m_2 l_2 \dot{\theta}_2^2 \sin(\delta) + g(m_1 + m_2) \sin\theta_1 = 0\,,$$
$$m_2 l_2 \ddot{\theta}_1 + m_2 l_1 \ddot{\theta}_1 \cos(\delta) - m_2 l_1 \dot{\theta}_1^2 \sin(\delta) + m_2 g \sin\theta_2 = 0\,,$$

gdzie $\delta = \theta_1 - \theta_2\,.$

Stąd

$$
\begin{aligned}
\ddot{\theta}_1 &= -[gm\sin\theta_1 - gm_2\cos(\delta)\sin\theta_2 + l_1 m_2\cos(\delta)\sin(\delta)\dot{\theta}_1^2] + l_2 m_2\sin(\delta)\dot{\theta}_2^2]/D_1 \\
\ddot{\theta}_2 &= [gm\cos(\delta)\sin\theta_1 - gm\sin\theta_2 + l_1 m\sin(\delta)\dot{\theta}_1^2 + l_2 m_2\cos(\delta)\sin(\delta)\dot{\theta}_2^2]/D_2
\end{aligned}
$$

gdzie $D_1 = l_1(m - m_2\cos^2\delta)$, $D_2 = l_2/l_1 D_1$.

Schemat i równania III

Kładąc $\theta_1 = y_0, \dot{\theta}_1 = y_1$, $\theta_2 = y_3, \dot{\theta}_2 = y_4$, równania te zapiszemy w postaci

$$
\begin{aligned}
\dot{y}_0 &= y_1 \\
\dot{y}_1 &= -[gm \sin y_0 - gm_2 \cos(\delta) \sin y_2 + l_1 m_2 \cos(\delta) \sin(\delta) y_1^2] + l_2 m_2 \sin(\delta) y_4]/D_1 \\
\dot{y}_2 &= y_3 \\
\dot{y}_3 &= [gm \cos(\delta) \sin y_0 - gm \sin y_3 + l_1 m \sin(\delta) y_1^2 + l_2 m_2 \cos(\delta) \sin(\delta) y_4^2]/D_2
\end{aligned}
$$

gdzie $\delta = y_0 - y_3$, a D_1 i D_2 są dane jak poprzednio. Postać ta jest odpowiednia dla całkowania numerycznego.

Używamy dowolnej metody Rungego-Kutty; 4-go lub wyższego rzędu.
Jeden krok całkowania równania 1-go rzędu

$$\dot{y} = f(t, y), \qquad y(t_0) = y_0.$$

z krokiem h, metodą, np. RK4:

$$\begin{aligned}
k_1 &= hf(t, y_t) \\
k_2 &= hf(t + h/2, y_t + k_1/2) \\
k_3 &= hf(t + h/2, y_t + k_2/2) \\
k_4 &= hf(t + h, y_t + k_3) \\
y_{t+h} &= y_t + (k_1 + 2k_2 + 2k_3 + k_4)/6
\end{aligned}$$

Tutaj y_t jest numeryczną wartością $y(t)$ (przybliżoną).

Metoda `derivatives(...)` jest metodą abstrakcyjną. Klasa, ktora rozszerza tę metodę musi ją zaimplementować. W naszym wypadku, podwójnego wahadła będzie ona zwracać tablicę zawierającą wyniki obliczeń prawych stron równań pierwszego rzędu dla wahadła podwójnego opisanych wcześniej.

Metody tej używają całkujące metody klasy `RungeKutta` (`forthOrder`, `cashKarp`, `fehlberg`).

Java: równanie; Pen2Eq I

```
class Pend2Eq extends Equation {

    // double pendulum
    static double G  = 9.81; // acceleration due to gravity, in m/s^2
    static double L1 = 0.6;  // length of pendulum 1 in m
    static double L2 = 1.0;  // length of pendulum 2 in m
    static double M1 = 1.0;  // mass of pendulum 1 in kg
    static double M2 = 2.0;  // mass of pendulum 2 in kg

    double[] derivatives(double t, double[] state){
        double[] dydx = new double[state.length];

        double delta = state[2] - state[0];
        double cd = Math.cos(delta);
        double sd = Math.sin(delta);
        double s0 = Math.sin(state[0]);
        double s2 = Math.sin(state[2]);
        double M = M1 + M2;
        double den = L1*(M - M2*cd*cd);

        dydx[0] = state[1];
```

Java: równanie; Pen2Eq II

```
        dydx[2] = state[3];
        dydx[1] = -(G*M*s0 - cd*G*M2*s2 + cd*L1*M2*sd*state[1]*state[1] +
                    L2*M2*sd*state[3]*state[3])/(L1*(M - cd*cd*M2));
        dydx[3] = (cd*G*M*s0 - G*M*s2 + L1*M*sd*state[1]*state[1] +
                   cd*L2*M2*sd*state[3]*state[3])/(L2*(M - cd*cd*M2));
        return dydx;
    }
}
```