Programowanie I R

Zadania – seria 11.

Obliczenia naukowe: równania różniczkowe zwyczajne.

Zadanie 1. `pendulum` – Wahadło matematyczne.

Równanie ruchu wahadła matematycznego ma postać

$$\frac{\mathrm{d}^2\theta}{\mathrm{d}t^2} + \frac{g}{l}\sin\theta = 0,$$

gdzie g jest przyspieszeniem grawitacyjnym, l – długością wahadła, zaś $\theta = \theta(t)$ – zmienną dynamiczną opisującą wychylenie wahadła z położenia równowagi.

Napisz program `pendulum` rozwiązujący równanie ruchu wahadła matematycznego numerycznie (bez stosowania przybliżenia małych drgań). Program powinien wyznaczać wychylenie θ i prędkość kątową $\omega = \mathrm{d}\theta/\mathrm{d}t$ wahadła o zadanej długości l w przedziale czasowym $t \in [0, t_{\max}]$ z krokiem δt dla zadanych warunków początkowych $\theta(0) = \theta_0$ i $\omega(0) = \omega_0$. Jako argumenty wywołania program powinien przyjmować pięć liczb zmiennoprzecinkowych reprezentujących kolejno: długość wahadła l, wychylenie początkowe θ_0, początkową prędkość kątową ω_0, czas trwania symulacji $t_{\max}$ oraz krok czasowy δt, a także opcjonalny przełącznik określający metodę numerycznego rozwiązywania równania:

- `--Euler` – metoda Eulera,
- `--midpoint` – metoda punktu środkowego,
- `--RK4` – metoda Rungego–Kutty czwartego rzędu.

W przypadku braku przełącznika należy zastosować metodę Rungego–Kutty czwartego rzędu.

Wynikiem działania programu powinno być wypisanie na standardowe wyjście listy czasów oraz odpowiadających im wartości wychylenia i prędkości kątowej oraz narysowanie wykresu przedstawiającego diagram fazowy $\omega = \omega(\theta)$.

Zadanie 2. `lv` – Model Lotki–Volterry.

Rozważmy ekosystem złożony z dwóch oddziałujących ze sobą populacji: drapieżników oraz ich ofiar. Niech $x = x(t)$ będzie liczebnością populacji ofiar, zaś $y = y(t)$ – liczebnością populacji drapieżników. Ewolucję czasową tych wielkości opisują równania Lotki–Volterry:

$$\begin{cases} \dfrac{\mathrm{d}x}{\mathrm{d}t} &= (a - by)\,x, \\ \dfrac{\mathrm{d}y}{\mathrm{d}t} &= (cx - d)\,y. \end{cases}$$

Stałe parametry a, b, c i d opisują odpowiednio: naturalny przyrost populacji ofiar, zmniejszanie się populacji ofiar wskutek drapieżnictwa, wzrost populacji drapieżników związany z dostępnością pożywienia oraz naturalne zmniejszanie się populacji drapieżników.

Napisz program `lv` rozwiązujący numerycznie równania Lotki–Volterry. Program powinien wyznaczać liczebność populacji ofiar x i liczebność populacji drapieżników y w przedziale czasowym $t \in [0, t_{\max}]$ z krokiem

δt dla zadanych warunków początkowych $x(0) = x_0$ i $y(0) = y_0$ oraz zadanych wartości parametrów a, b, c i d. Jako argumenty wywołania program powinien przyjmować osiem liczb zmiennoprzecinkowych reprezentujących kolejno: parametry a, b, c i d, wartości początkowe x_0 i y_0, czas trwania symulacji $t_{\max}$ oraz krok czasowy δt. Posłuż się metodą Rungego–Kutty czwartego rzędu.

Wynikiem działania programu powinno być wypisanie na standardowe wyjście listy czasów oraz odpowiadających im liczebności populacji ofiar i drapieżników oraz narysowanie wykresu przedstawiającego zależności $x = x(t)$ oraz $y = y(t)$.

Zadanie 3. `duffing` – Oscylator Duffinga.

Oscylatorem Duffinga nazywamy układ mechaniczny opisywany równaniem ruchu

$$\frac{\mathrm{d}^2x}{\mathrm{d}t^2} + \delta\frac{\mathrm{d}x}{\mathrm{d}t} + \beta x + \alpha x^3 = \gamma\cos(\omega t).$$

Wielkość $x = x(t)$ jest zmienną dynamiczną opisującą wychylenie oscylatora z położenia równowagi, zaś stałe α, β, γ i δ – parametrami. Układ ten wykazuje zachowanie chaotyczne – niewielka zmiana warunków początkowych znacząco wpływa na jego ewolucję.

Wykorzystując moduł `scipy.integrate.solve_ivp`, napisz program `duffing` rozwiązujący numerycznie równanie ruchu oscylatora Duffinga. Program powinien wyznaczać wychylenie x i prędkość $v = \mathrm{d}x/\mathrm{d}t$ oscylatora o zadanych wartościach parametrów

$$\alpha = \omega = 1, \quad \beta = -1, \quad \delta = 0,2, \quad \gamma = 0,3$$

w przedziale czasowym $t \in [0,\, t_{\max}]$ z krokiem δt dla trzech zestawów wartości początkowych:

$$\begin{cases} x_{01} = 0,99 \\ v_{01} = 0, \end{cases} \qquad \begin{cases} x_{02} = 1 \\ v_{02} = 0, \end{cases} \qquad \begin{cases} x_{03} = 1.01 \\ v_{03} = 0. \end{cases}$$

Jako argumenty wywołania program powinien przyjmować dwie liczby zmiennoprzecinkowe reprezentujące kolejno czas trwania symulacji $t_{\max}$ oraz krok czasowy δt. Program powinien rysować na jednym wykresie diagramy fazowe $v = v(t)$ dla wszystkich trzech zestawów wartości początkowych.

Opracowanie: Bartłomiej Zglinicki.