

Lista 3

W liście 3 odchodzimy od symulacji w Matlabie / Simulinku i skupiamy się na Pythonie. Przedmiotem tej listy jest wprowadzenie do symulacji w Pythonie na przykładzie samodzielnej implementacji metody Eulera oraz wykorzystania pakietu `scipy.integrate`. Symulowane będą znane już układy Lotki-Volterry oraz Lorenza.

Gwoli przypomnienia układ Lotki-Volterry definiujemy jako

$$\begin{cases} \frac{dx}{dt} = (a - by)x \\ \frac{dy}{dt} = (cx - d)y, \end{cases} \quad (1)$$

gdzie x - populacja ofiar, y - populacja drapieżników, t - czas, a - częstość narodzin ofiar, b - częstość umierania ofiar, c - częstość narodzin drapieżników, d - częstość umierania drapieżników. Możemy przyjąć następujące wartości parametrów: $a = 1.2$, $b = 0.6$, $c = 0.3$, $d = 0.8$ oraz populacje początkowe $x_0 = 2$ i $y_0 = 1$.

Układ Lorenza zdefiniowany jest za pomocą układu trzech równań różniczkowych:

$$\begin{cases} \frac{dx}{dt} = \sigma(y - x) \\ \frac{dy}{dt} = x(\rho - z) - y \\ \frac{dz}{dt} = xy - \beta z, \end{cases} \quad (2)$$

Możemy przyjąć wartości parametrów $\sigma = 10$, $\beta = \frac{8}{3}$ i $\rho = 28$ oraz warunki początkowe $x(0) = y(0) = z(0) = 1$.

Należy zasymulować znane z poprzednich list układy Lotki-Volterry i Lorenza na dwa sposoby:

1. Przy użyciu metody Eulera - porównać wygenerowane symulacje dla co najmniej 3 różnych wartości kroku dt i przedstawić w sposób podobny do rysunków 1 i 2
2. Przy użyciu metody `integrate.odeint` z pakietu `scipy` i przedstawić w sposób podobny do rysunków 3, 4 i 5.

Dodatkowo, dla każdej zaproponowanej w metodzie Eulera wartości kroku dt należy policzyć średni błąd aproksymacji. Jako wartości prawdziwe można wykorzystać wartości funkcji wyliczone przez metodę `odeint`.

Uwagi i wskazówki

- Przy ustawieniu zbyt dużego kroku dt możemy otrzymać rozwiązanie rozbiegające do nieskończoności. Spośród 3 zaproponowanych kroków dt można przedstawić co najwyżej jedno takie rozwiązanie.
- Wykresy przedstawione w liście były uzyskane za pomocą biblioteki `matplotlib`. W celu stworzenia wykresu 3D należy stworzyć podwykres przy użyciu należącej do klasy `Figure` metody `add_subplot(projection='3d')`.
- Wykres 3D jest wymagany dla obu metod.

Układ Lotki-Volterra zamodelowany przy pomocy metody Eulera

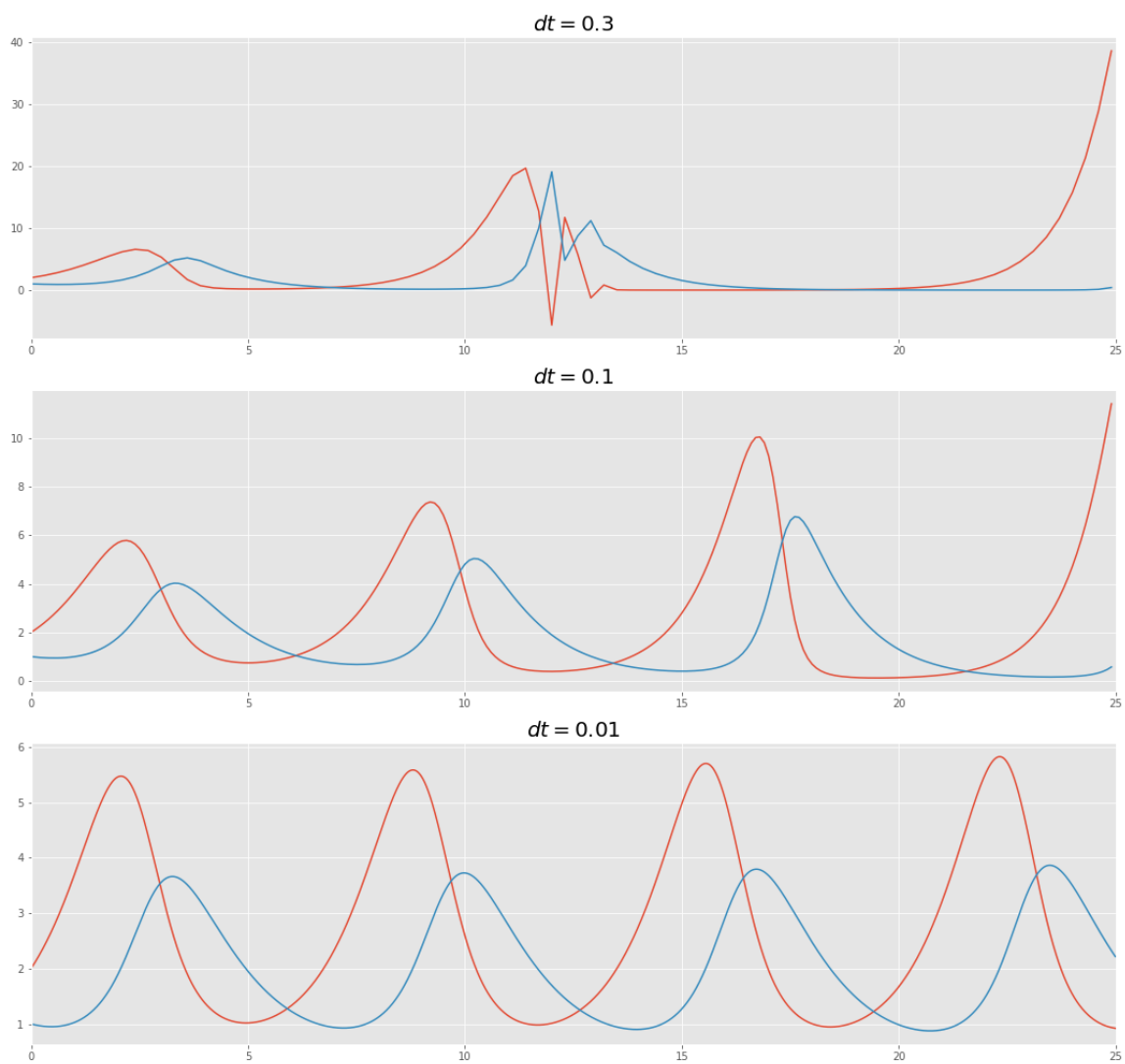


Figure 1: Przykładowy rezultat symulacji przedstawiający populacje drapieżników i ofiar w czasie $t \in [0, 25]$ dla podanych parametrów oraz kroków symulacji $dt \in \{0.3, 0.1, 0.01\}$

Układ Lorenza zamodelowany przy pomocy metody Eulera

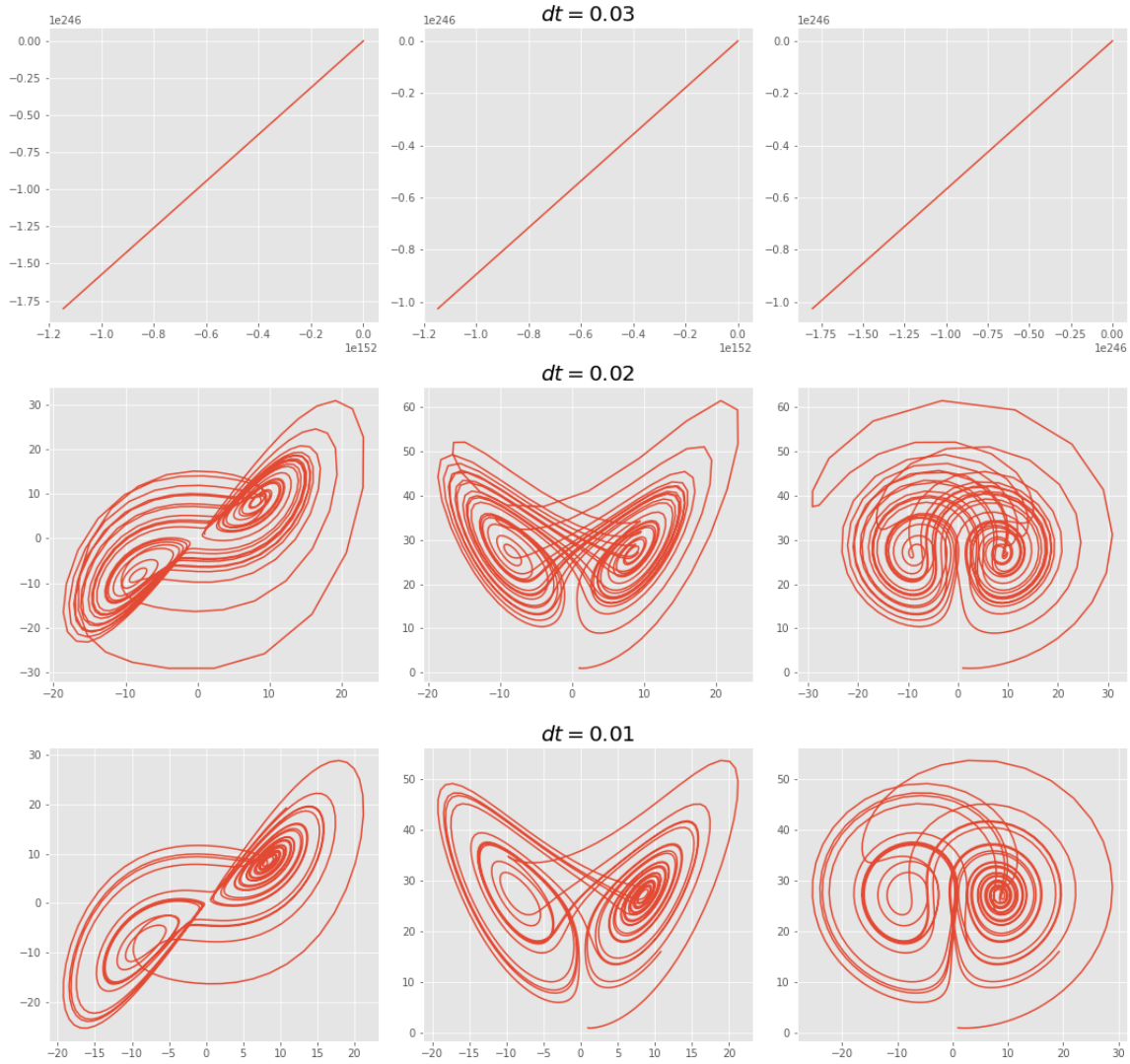


Figure 2: Przykładowy rezultat symulacji przedstawiający układ Lorenza w czasie $t \in [0, 25]$ dla podanych parametrów oraz kroków symulacji $dt \in \{0.03, 0.02, 0.01\}$

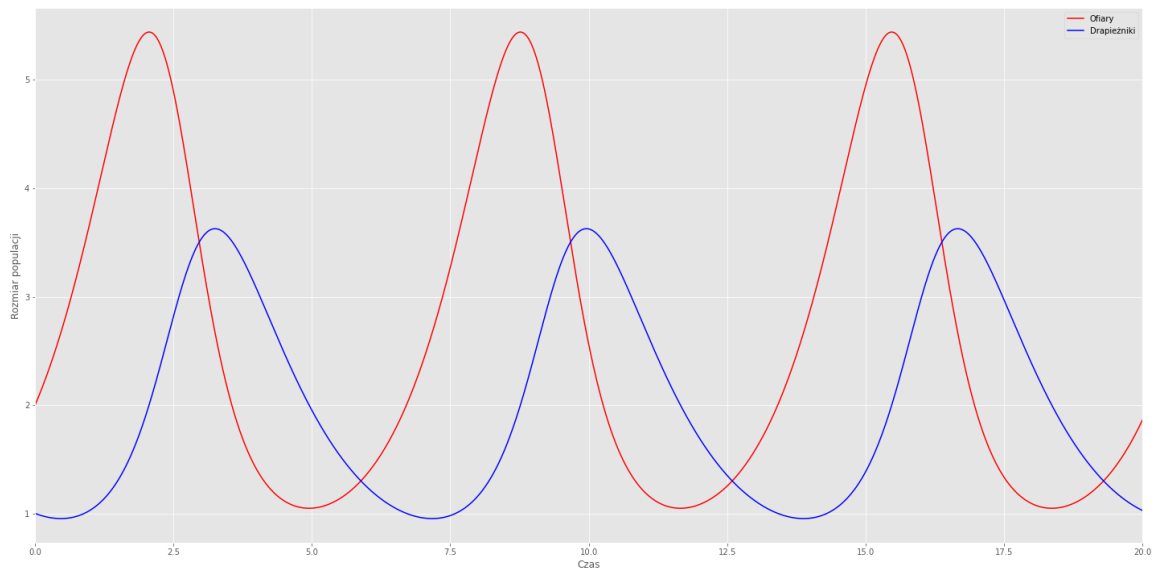


Figure 3: Przykładowy rezultat symulacji przedstawiający populacje drapieżników i ofiar w czasie $t \in [0, 25]$ dla podanych parametrów i $dt = 0.002$

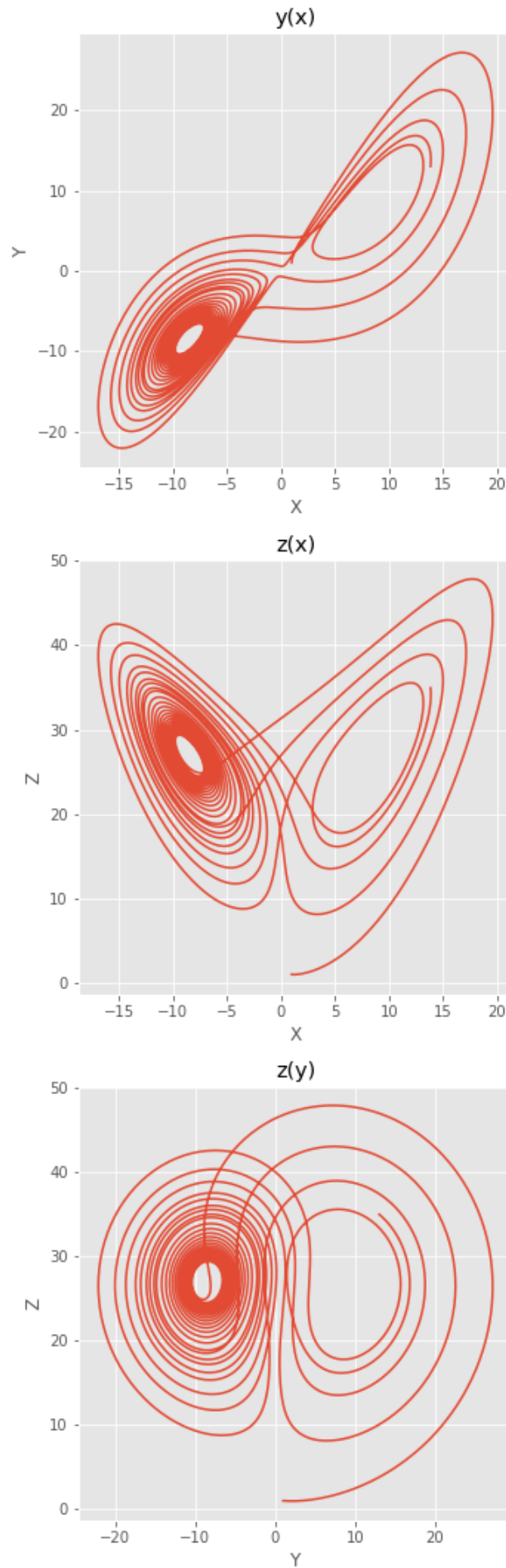


Figure 4: Przykładowy rezultat symulacji za pomocą pakietu `scipy` przedstawiający układ Lorenza w czasie $t \in [0, 25]$ dla podanych parametrów i $dt = 0.002$

Wizualizacja $z(x, y)$ układu Lorenza

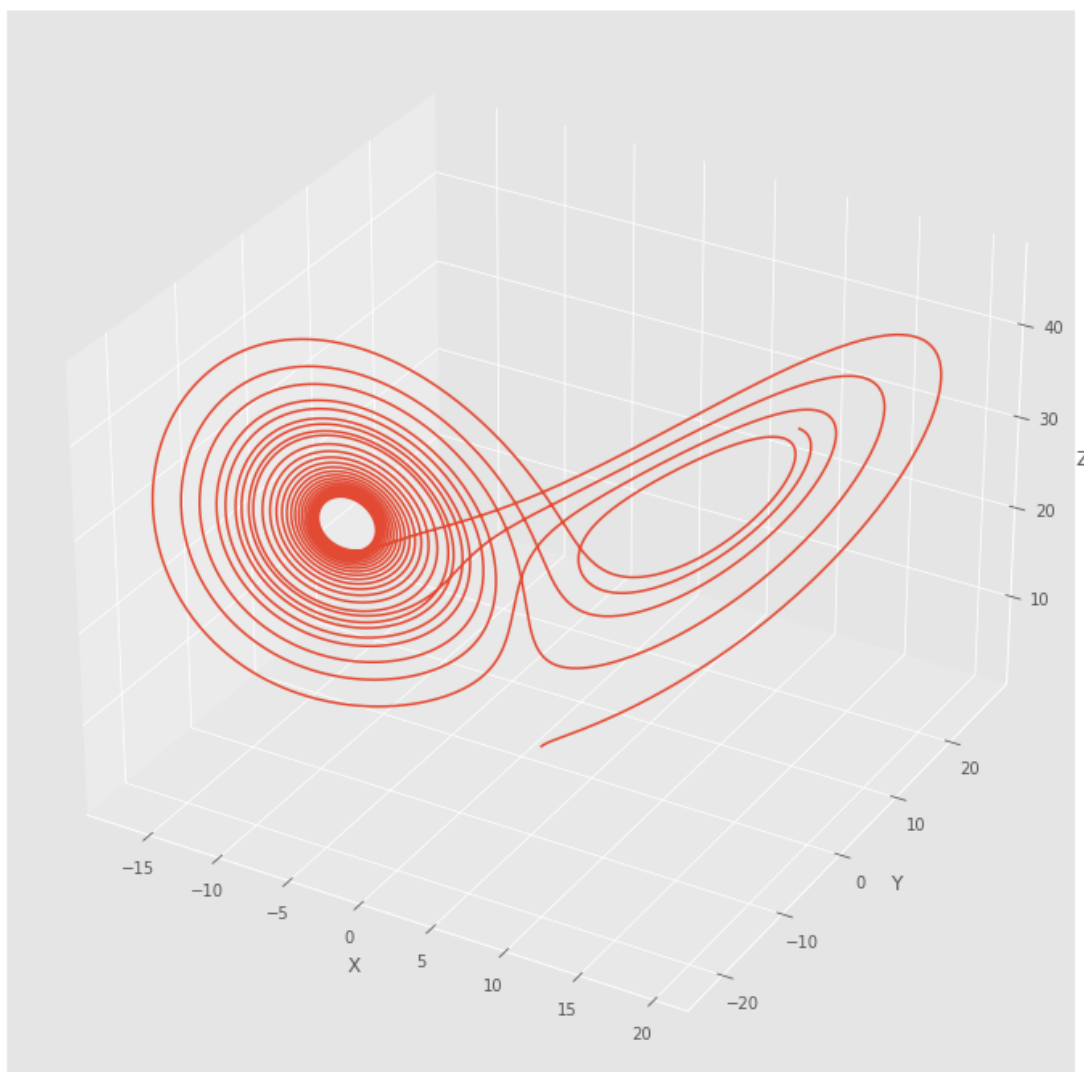


Figure 5: Przykładowy trójwymiarowy rezultat symulacji za pomocą pakietu `scipy` przedstawiający układ Lorenza w czasie $t \in [0, 25]$ dla podanych parametrów i $dt = 0.002$