



**Wydział Matematyki  
i Nauk Informacyjnych**

POLITECHNIKA WARSZAWSKA

# Estymacja parametrów modelu Lotki-Volterra

Modelowanie Matematyczne 2023/2024 semestr  
zimowy

Zadanie Projektowe nr 2

Autor: Adam Grącikowski

Prowadzący: dr inż. Jakub Wagner

Oświadczam, że niniejsza praca, stanowiąca podstawę do uznania osiągnięcia efektów uczenia się z przedmiotu Modelowanie Matematyczne została wykonana przeze mnie samodzielnie.

Warszawa, 14 grudnia 2023

# Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>3</b>
<b>2</b>	<b>Metodyka i wyniki doświadczeń</b>	<b>3</b>
2.1	Estymacja parametrów pierwszego równania układu . . . . .	4
2.1.1	Jawna metoda Eulera . . . . .	5
2.1.2	Niejawna metoda Eulera . . . . .	6
2.1.3	Metoda Adamsa-Bashfortha trzeciego rzędu . . . . .	7
2.2	Estymacja parametrów drugiego równania układu . . . . .	8
2.2.1	Jawna metoda Eulera . . . . .	8
2.2.2	Niejawna metoda Eulera . . . . .	9
2.2.3	Metoda Adamsa-Bashfortha trzeciego rzędu . . . . .	10
2.3	Estymacja parametrów obu równań układu . . . . .	11
2.3.1	Jawna metoda Eulera . . . . .	11
2.3.2	Ode45 . . . . .	12
2.4	Stan równowagi w układzie Lotki-Volterra . . . . .	14
<b>3</b>	<b>Testy na danych rzeczywistych oraz interpretacja wyników</b>	<b>15</b>
<b>4</b>	<b>Wnioski</b>	<b>19</b>
<b>5</b>	<b>Listing</b>	<b>20</b>

## Przyjęte oznaczenia

$[a, b]$	przedział obustronnie domknięty o końcach $a$ i $b$
$\frac{dy}{dt}$	pochodna funkcji $y$ po zmiennej $t$
<b>A</b>	wielkość macierzowa (pogrubiona czcionka)
<b>b</b>	wielkość wektorowa (pogrubiona czcionka)
$h$	krok całkowania

# 1 Wprowadzenie

Przedmiotem pracy jest analiza i estymacja parametrów modelu Lotki-Volterry, opisującego liczebność populacji dwóch gatunków, między którymi występuje zależność typu drapieżnik-ofiara.

Równanie Lotki-Volterry, nazwane od nazwisk dwóch matematyków - Alfreda Lotki'ego oraz Vito Volterry, jest systemem równań różniczkowych, którego zastosowanie wykracza poza czysto matematyczną sferę, obejmując obszary takie jak biologia, ekologia, czy ekonomia. Układ ten pozwala na lepsze zrozumienie i prognozowanie dynamiki populacji, co ma kluczowe znaczenie w podejmowaniu decyzji dotyczących różnego rodzaju zasobów naturalnych, czy ochrony środowiska.

Rozważany przez nas wariant równań Lotki-Volterry przedstawia układ (1):

$$\begin{cases} \frac{dx}{dt} = r_x x(t) + r_{xy} x(t)y(t) + r_{xx} x^2(t) \\ \frac{dy}{dt} = r_y y(t) + r_{yx} x(t)y(t) + r_{yy} y^2(t) \end{cases} \quad (1)$$

gdzie  $x$  to liczebność populacji gatunku ofiary,  $y$  to liczebność populacji gatunku drapieżnika,  $t$  oznacza czas, a  $r_x, r_{xy}, r_{xx}, r_y, r_{yx}, r_{yy} \in \mathbb{R}$  są parametrami modelu.

Poszczególne elementy układu (1) można zinterpretować następująco:

- $r_x$  - współczynnik wzrostu populacji  $x$ . Im większa jest wartość tego współczynnika, tym szybszy wzrost liczebności populacji.
- $r_y$  - współczynnik wzrostu populacji  $y$ . Im większa jest wartość tego współczynnika, tym szybszy wzrost liczebności populacji.
- $r_{xy}$  - współczynnik wpływu populacji  $y$  na wzrost populacji  $x$ . Określa sposób, w jaki zmiana liczebności populacji  $y$  wpływa na tempo wzrostu populacji  $x$ .
- $r_{yx}$  - współczynnik wpływu populacji  $x$  na wzrost populacji  $y$ . Określa sposób, w jaki zmiana liczebności populacji  $x$  wpływa na tempo wzrostu populacji  $y$ .
- $r_{xx}$  - współczynnik samoregulacji populacji  $x$ . Wartość ta określa w jaki sposób populacja  $x$  reguluje swój własny wzrost (konkurencja wewnątrzgatunkowa).
- $r_{yy}$  - współczynnik samoregulacji populacji  $y$ . Wartość ta określa w jaki sposób populacja  $y$  reguluje swój własny wzrost (konkurencja wewnątrzgatunkowa).

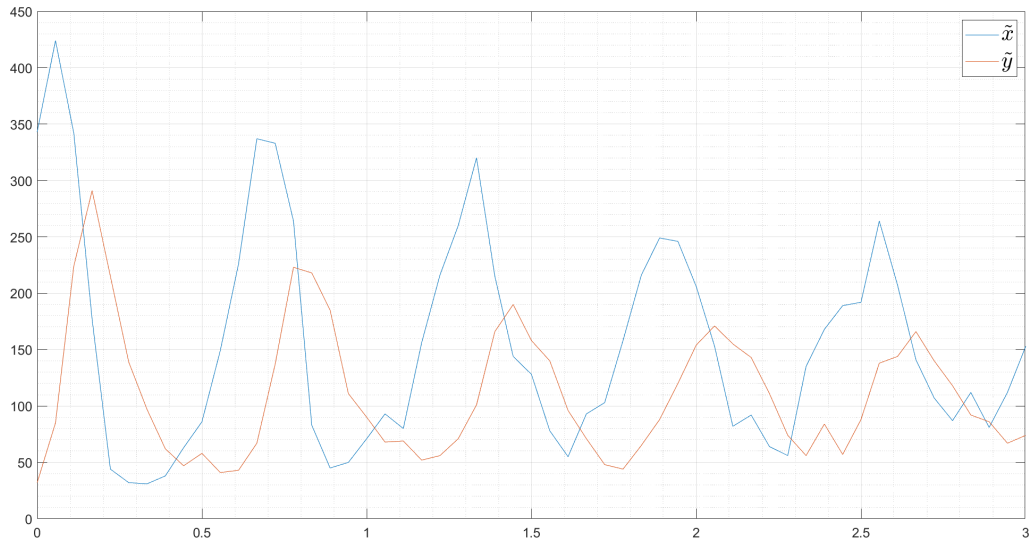
## 2 Metodyka i wyniki doświadczeń

W tej części pracy zajmiemy się estymacją parametrów  $r_x, r_{xy}, r_{xx}, r_y, r_{yx}, r_{yy}, \hat{x}_1, \hat{y}_1$  charakteryzujących przyjęty przez nas model (1).

W rozważanym przez nas scenariuszu dysponujemy pomiarami liczebności populacji dwóch gatunków. Niech  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N$  oraz  $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N$  oznaczają wyniki pomiarów wartości funkcji liczebności  $x$  i  $y$  w chwilach  $t_1, t_2, \dots, t_N$ . Na podstawie zebranych w wyniku pomiarów wartości chcielibyśmy dopasować współczynniki  $r_x, r_{xy}, r_{xx}, r_y, r_{yx}, r_{yy}$  oraz liczebności  $\hat{x}_1, \hat{y}_1$  obu gatunków w chwili początkowej  $t_1$ . Znajomość tych współczynników jest kluczowa dla pełniejszego zrozumienia zależności pomiędzy gatunkami  $x$  i  $y$  oraz potencjalnej predykcji ich wzajemnego rozwoju.

Przez  $\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N$  oraz  $\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N$  będziemy oznaczać estymaty wartości funkcji liczebności  $x$  i  $y$  w chwilach  $t_1, t_2, \dots, t_N$ , otrzymane przy pomocy wybranych, numerycznych metod rozwiązywania równań różniczkowych.

Wykres 1 przedstawia liczebność fikcyjnych gatunków  $x$  i  $y$  w zależności od czasu  $t$ . Dane te zostały wygenerowane komputerowo i będą stanowiły podstawę naszych rozważań w dalszych podrozdziałach.



Rysunek 1: Liczebność gatunków  $x$  i  $y$  w chwilach pomiarowych  $t_1, t_2, \dots, t_N$

Listing 1 zawiera implementacje ogólnego schematu postępowania, który będzie wykorzystywany w podrozdziałach 2.1 oraz 2.2 w celu estymacji parametrów modelu (1).

## 2.1 Estymacja parametrów pierwszego równania układu

Naszym pierwszym podejściem do problemu estymacji parametrów przyjętego modelu (1) będzie próba wyznaczenia wartości  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$ ,  $\hat{x}_1$ , minimalizujących wskaźnik dopasowania modelu do danych. Wskaźnik ten definiujemy wzorem (2):

$$J_x = \sum_{n=1}^N (\hat{x}_n - \tilde{x}_n)^2 \quad (2)$$

Wskaźnik  $J_x$  to suma kwadratów różnic między wartościami rzeczywistymi, a modelowanymi. Jest to popularny wskaźnik, który mierzy odchylenie prognozowanego modelu od danych rzeczywistych.

Do minimalizacji wartości wskaźnika  $J_x$  wykorzystamy funkcję `fminsearch`, czyli funkcję ze środowiska MATLAB, służącą do minimalizacji funkcji nieliniowych jednej lub wielu zmiennych z ograniczeniami. Algorytm tej funkcji oparty jest na iteracyjnym obliczaniu wartości funkcji celu, ażeby znaleźć jej lokalne minimum. Parametrem wejściowym `fminsearch` jest punkt startowy, w którym rozpoczyna ona swoje poszukiwania, natomiast jednym z parametrów wyjściowych jest wektor optymalnych wartości zmiennych, w którym funkcja celu osiąga szukane minimum.

Tym samym, problem, który staramy się rozwiązać sprowadziliśmy do problemu wyznaczenia optymalnego punktu startowego dla funkcji `fminsearch`. Ten z kolei, rozwiążemy wyznaczając wartość  $J_x$  dla niewielkiej liczby zestawów wartości  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$  i  $\hat{x}_1$ , a następnie przyjmując za optymalny punkt startowy zestaw dający najlepszy (najmniejszy) wynik.

Wartości parametrów  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$  i  $\hat{x}_1$  sprawdzimy z następujących zakresów:

$$r_x \in [0, 40], \quad r_{xy} \in [-1, 0], \quad r_{xx} \in [-0.1, 0], \quad \hat{x}_1 \in [100, 1000]$$

### 2.1.1 Jawną metodą Eulera

Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskamy za pomocą jawnej metody Eulera (3):

$$\begin{cases} \hat{x}_n = \hat{x}_{n-1} + f_x(\hat{x}_{n-1}, \tilde{y}_{n-1}) \cdot \Delta t_n, & n = 2, \dots, N \\ \Delta t_n \equiv t_n - t_{n-1} \\ f_x(x, y) \equiv r_x x + r_{xy} xy + r_{xx} x^2 \end{cases} \quad (3)$$

po przyjęciu wartości początkowej  $\hat{x}_1$  oraz podstawieniu wartości  $\tilde{y}_1, \tilde{y}_2, \dots, \tilde{y}_N$ .

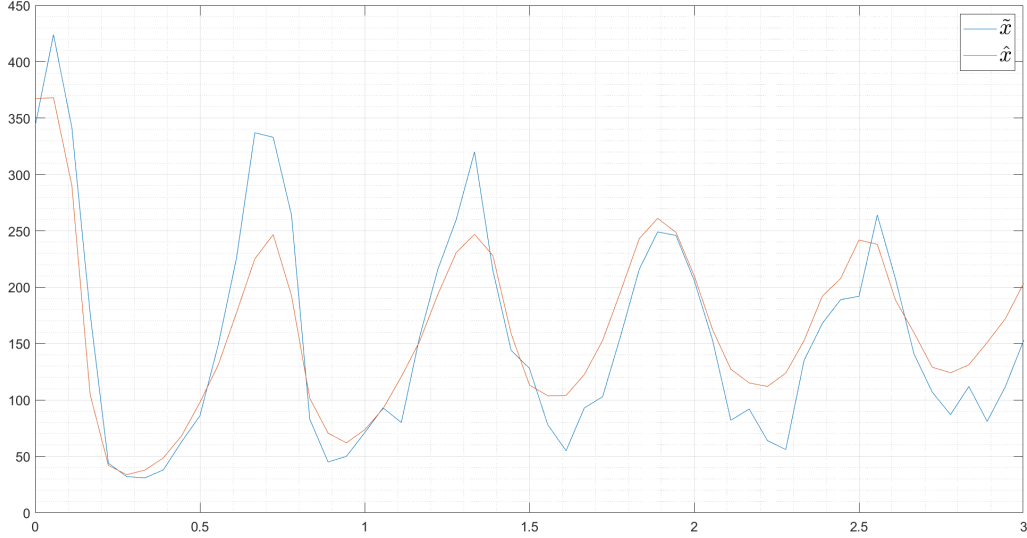
Jawna metoda Eulera jest jednym z najprostszych i najbardziej intuicyjnych sposobów numerycznego rozwiązania równań różniczkowych zwyczajnych. Metoda ta jest rzędu pierwszego, co oznacza, że jej błąd jest proporcjonalny do kwadratu wielkości kroku czasowego. Metoda ta jest jawna ponieważ wartość  $\hat{x}_n$  jest wyrażona bezpośrednio za pomocą wartości obliczonych w poprzedniej iteracji, czyli w chwili  $t_{n-1}$ .

Listing 2 zawiera implementację jawnej metody Eulera, dostosowaną do naszego modelu (1). W wyniku próbkowania przedziałów i wyznaczania wskaźnika  $J_x$  (2) dla każdej kombinacji współczynników  $r_x, r_{xy}, r_{xx}$  i  $\hat{x}_1$  otrzymujemy punkt startowy dla funkcji `fminsearch` przedstawiony w pierwszym wierszu tabeli 1. Drugi wiersz tabeli 1 zawiera natomiast zestaw optymalnych współczynników zwrócony przez funkcję `fminsearch`.

$r_x$	$r_{xy}$	$r_{xx}$	$\hat{x}_1$
12.41	-0.07	-0.02	357.14
13.18	-0.07	-0.03	367.48

Tabela 1: Parametry modelu otrzymane w wyniku estymacji z zastosowaniem jawnej metody Eulera

Uzyskane przy pomocy jawnej metody Eulera estymaty  $\hat{x}_2, \dots, \hat{x}_N$  zostały przedstawione na wykresie 2 na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$ .



Rysunek 2: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$

Na podstawie wykresu 2 możemy stwierdzić, że estymaty  $\hat{x}_2, \dots, \hat{x}_N$  dość znacząco odbiegają od wartości  $\tilde{x}_2, \dots, \tilde{x}_N$ . Ten wynik nie powinien być jednak dla nas szczególnie zaskakujący, ponieważ estymowaliśmy parametry pierwszego równania modelu (1), zapominając niejako o drugim równaniu.

### 2.1.2 Niejawna metoda Eulera

Postępując zgodnie z tą samą metodyką, co w podrozdziale 2.1.1, wyznaczmy ponownie parametry  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$  i  $\hat{x}_1$ . Tym razem skorzystamy z niejawnej metody Eulera, która dana jest wzorem (4):

$$\begin{cases} \hat{x}_n = \hat{x}_{n-1} + f_x(\hat{x}_n, \tilde{y}_n) \cdot \Delta t_n, & n = 2, \dots, N \\ \Delta t_n \equiv t_n - t_{n-1} \\ f_x(x, y) \equiv r_x x + r_{xy} x y + r_{xx} x^2 \end{cases} \quad (4)$$

W każdej iteracji będziemy zmuszeni rozwiązywać równanie kwadratowe (6), ponieważ  $f_x$  jest wielomianem drugiego stopnia zmiennej  $\hat{x}_n$ . Za szukaną wartość  $\hat{x}_n$  będziemy przyjmować większy z pierwiastków równania (6).

$$\hat{x}_n = \hat{x}_{n-1} + f_x(\hat{x}_n, \tilde{y}_n) \cdot \Delta t_n \quad (5)$$

$$(r_{xx} \Delta t_n) \hat{x}_n^2 + (r_x \Delta t_n + r_{xy} \tilde{y}_n \Delta t_n - 1) \hat{x}_n + \hat{x}_{n-1} = 0 \quad (6)$$

Listing 4 zawiera implementację niejawnej metody Eulera, dostosowaną do naszego modelu (1).

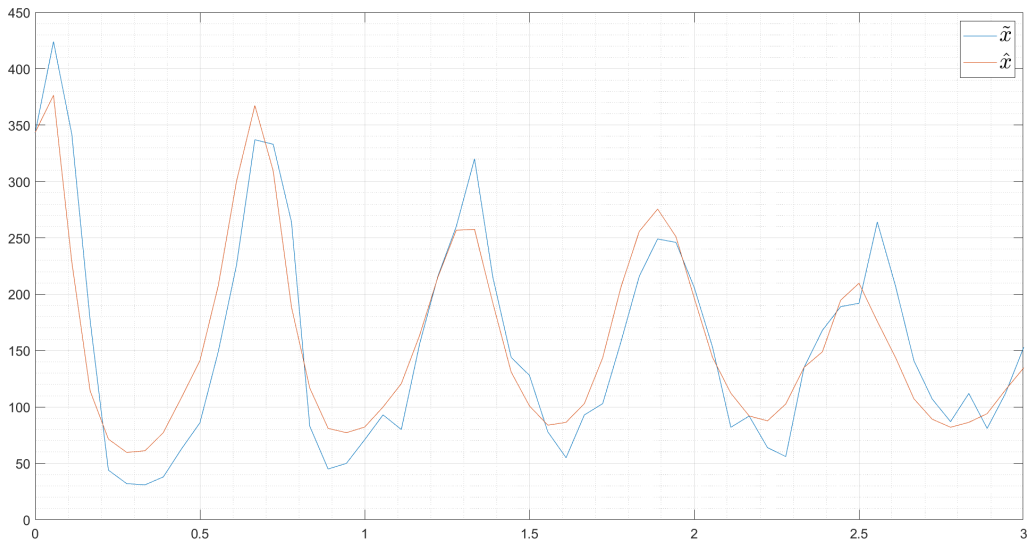
W wyniku próbkowania przedziałów i wyznaczania wskaźnika  $J_x$  (2) dla każdej kombinacji współczynników  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$  i  $\hat{x}_1$  otrzymujemy punkt startowy dla funkcji `fminsearch` przedstawiony w pierwszym wierszu tabeli 2. Drugi wiersz tabeli 2 zawiera natomiast zestaw optymalnych współczynników zwrócony przez funkcję `fminsearch`.

$r_x$	$r_{xy}$	$r_{xx}$	$\hat{x}_1$
17.78	-0.11	-0.04	410.34
9.65	-0.09	0.00	343.48

Tabela 2: Parametry modelu otrzymane w wyniku estymacji z zastosowaniem niejawnej metody Eulera

Porównując wartości w tabeli 2 z wartościami w tabeli 1 zauważamy, że wraz ze zmianą iteracyjnej metody rozwiązywania równań różniczkowych zwyczajnych, otrzymaliśmy istotnie różne wartości współczynników  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$  i  $\hat{x}_1$ .

Uzyskane przy pomocy niejawnej metody Eulera estymaty  $\hat{x}_2, \dots, \hat{x}_N$  zostały przedstawione na wykresie 3 na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$ .



Rysunek 3: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy niejawnej metody Eulera na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$

Ponownie, otrzymane przez nas estymaty  $\hat{x}_2, \dots, \hat{x}_N$  dość znacząco odbiegają od wartości  $\tilde{x}_2, \dots, \tilde{x}_N$ . Wynik ten poprawimy jednak w podrozdziale 2.3.

### 2.1.3 Metoda Adamsa-Bashfortha trzeciego rzędu

Tym razem skorzystamy z metody Adamsa-Bashfortha trzeciego rzędu, która dana jest wzorem (7):

$$\begin{cases} \hat{x}_2 = \hat{x}_1 + f_x(\hat{x}_1, \tilde{y}_1) \cdot \Delta t_2 \\ \hat{x}_3 = \hat{x}_2 + \left( \frac{3}{2}f_x(\hat{x}_2, \tilde{y}_2) - \frac{1}{2}f_x(\hat{x}_1, \tilde{y}_1) \right) \cdot \Delta t_3 \\ \hat{x}_n = \hat{x}_{n-1} + \frac{1}{12} \left( 23f_x(\hat{x}_{n-1}, \tilde{y}_{n-1}) - 16f_x(\hat{x}_{n-2}, \tilde{y}_{n-2}) + 5f_x(\hat{x}_{n-3}, \tilde{y}_{n-3}) \right) \cdot \Delta t_n \\ n = 4, \dots, N \end{cases} \quad (7)$$

Metoda Adamsa-Bashfortha trzeciego rzędu jest jawną metodą wielokrokową, ponieważ w każdej iteracji korzystamy z wartości w chwilach  $t_{n-1}$ ,  $t_{n-2}$  oraz  $t_{n-3}$ . Metoda ta jest trzeciego rzędu, co oznacza, że jej błąd jest proporcjonalny do trzeciej potęgi wielkości kroku czasowego.

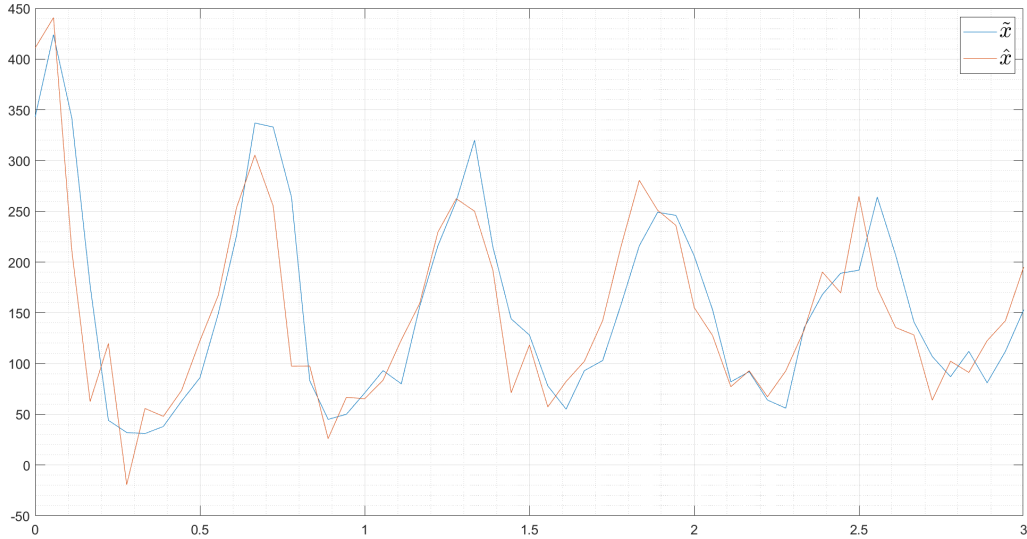
Listing 6 zawiera implementację metody Adamsa-Bashfortha trzeciego rzędu, dostosowaną do naszego modelu (1).

W wyniku próbkowania przedziałów i wyznaczania wskaźnika  $J_x$  (2) dla każdej kombinacji współczynników  $r_x$ ,  $r_{xy}$ ,  $r_{xx}$  i  $\hat{x}_1$  otrzymujemy punkt startowy dla funkcji `fminsearch` przedstawiony w pierwszym wierszu tabeli 2. Drugi wiersz tabeli 2 zawiera natomiast zestaw optymalnych współczynników zwrócony przez funkcję `fminsearch`.

$r_x$	$r_{xy}$	$r_{xx}$	$\hat{x}_1$
16.84	-0.11	-0.04	320.41
17.07	-0.12	-0.03	410.98

Tabela 3: Parametry modelu otrzymane w wyniku estymacji z zastosowaniem metody Adamsa-Bashfortha trzeciego rzędu

Uzyskane przy pomocy metody Adamsa-Bashfortha trzeciego rzędu estymaty  $\hat{x}_2, \dots, \hat{x}_N$  zostały przedstawione na wykresie 4 na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$ .



Rysunek 4: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy metody Adamsa-Bashfortha trzeciego rzędu na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$



## 2.2 Estymacja parametrów drugiego równania układu

Naszym drugim podejściem do problemu estymacji parametrów przyjętego modelu (1) będzie próba wyznaczenia wartości  $r_y$ ,  $r_{yx}$ ,  $r_{yy}$ ,  $\hat{y}_1$ , minimalizujących wskaźnik dopasowania modelu do danych. Wskaźnik ten definiujemy wzorem (8):

$$J_y = \sum_{n=1}^N (\hat{y}_n - \tilde{y}_n)^2 \quad (8)$$

Wskaźnik  $J_y$  jest analogiczny do wskaźnika  $J_x$  wprowadzonego w podrozdziale 2.1. Do minimalizacji wartości wskaźnika  $J_y$  wykorzystamy ponownie funkcję `fminsearch`.

Wartości parametrów  $r_y$ ,  $r_{yx}$ ,  $r_{yy}$  i  $\hat{y}_1$  sprawdzimy z następujących zakresów:

$$r_y \in [-40, 0], \quad r_{yx} \in [0, 1], \quad r_{yy} \in [-0.1, 0], \quad \hat{y}_1 \in [10, 200]$$

### 2.2.1 Jawna metoda Eulera

Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskamy za pomocą jawnej metody Eulera (9), po przyjęciu wartości początkowej  $\hat{y}_1$  oraz podstawieniu wartości  $\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_N$ .

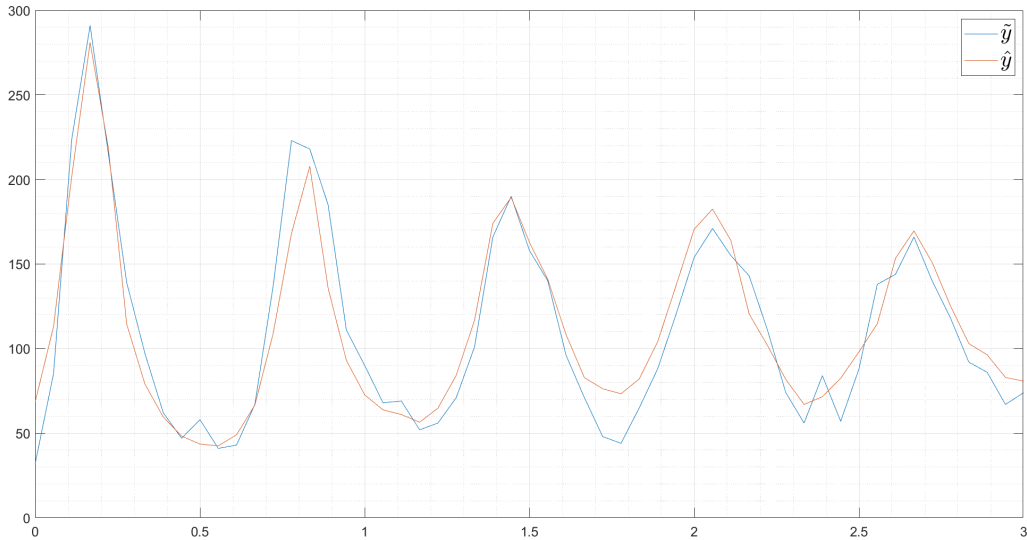
$$\begin{cases} \hat{y}_n = \hat{y}_{n-1} + f_y(\tilde{x}_{n-1}, \hat{y}_{n-1}) \cdot \Delta t_n, & n = 2, \dots, N \\ \Delta t_n \equiv t_n - t_{n-1} \\ f_y(x, y) \equiv r_y y + r_{yx} x y + r_{yy} y^2 \end{cases} \quad (9)$$

W wyniku próbkowania przedziałów i wyznaczania wskaźnika  $J_y$  (8) dla każdej kombinacji współczynników  $r_y$ ,  $r_{yx}$ ,  $r_{yy}$  i  $\hat{y}_1$  otrzymujemy punkt startowy dla funkcji `fminsearch` przedstawiony w pierwszym wierszu tabeli 5. Drugi wiersz tabeli 5 zawiera natomiast zestaw optymalnych współczynników zwrócony przez funkcję `fminsearch`.

$r_y$	$r_{yx}$	$r_{yy}$	$\hat{y}_1$
-1.38	0.03	-0.03	92.82
-3.27	0.05	-0.03	68.49

Tabela 4: Parametry modelu otrzymane w wyniku estymacji z zastosowaniem jawnej metody Eulera

Uzyskane przy pomocy jawnej metody Eulera estymaty  $\hat{y}_2, \dots, \hat{y}_N$  zostały przedstawione na wykresie 5 na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$ .



Rysunek 5: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

### 2.2.2 Niejawna metoda Eulera

Postępując zgodnie z tą samą metodyką, co w podrozdziale 2.2.1, wyznaczmy ponownie parametry  $r_y$ ,  $r_{yx}$ ,  $r_{yy}$  i  $\hat{y}_1$ . Skorzystamy z niejawnej metody Eulera, która dana jest wzorem (10):

$$\begin{cases} \hat{y}_n = \hat{y}_{n-1} + f_y(\tilde{x}_n, \hat{y}_n) \cdot \Delta t_n, & n = 2, \dots, N \\ \Delta t_n \equiv t_n - t_{n-1} \\ f_y(x, y) \equiv r_y y + r_{yx} x y + r_{yy} y^2 \end{cases} \quad (10)$$

W każdej iteracji będziemy zmuszeni rozwiązywać równanie kwadratowe (12), ponieważ  $f_y$  jest wielomianem drugiego stopnia zmiennej  $\hat{y}_n$ . Za szukaną wartość  $\hat{y}_n$  będziemy przyjmować większy z pierwiastków równania (12).

$$\hat{y}_n = \hat{y}_{n-1} + f_y(\tilde{y}_n, \hat{y}_n) \cdot \Delta t_n \quad (11)$$

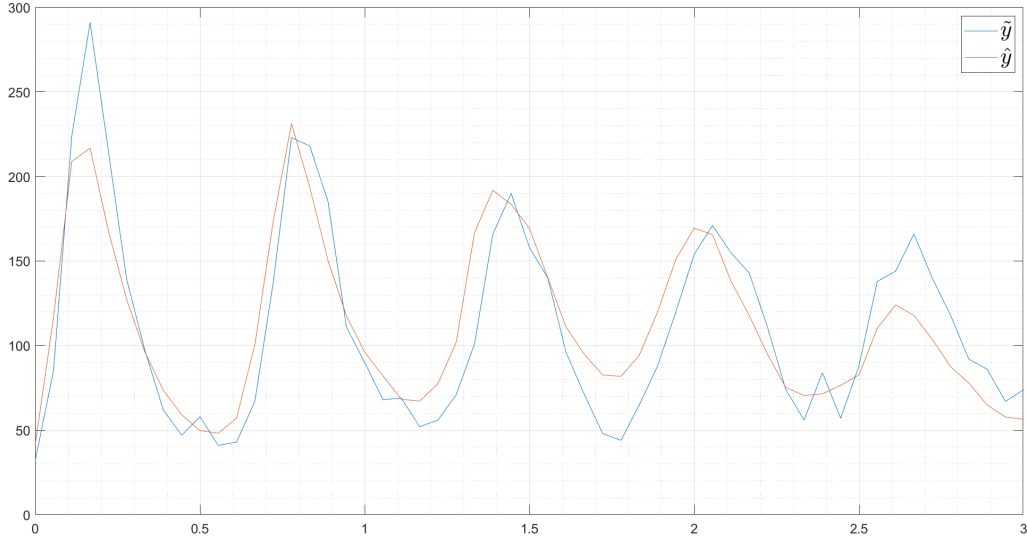
$$(r_{yy} \Delta t_n) \hat{y}_n^2 + (r_y \Delta t_n + r_{yx} \tilde{x}_n \Delta t_n - 1) \hat{y}_n + \hat{y}_{n-1} = 0 \quad (12)$$

W wyniku próbkowania przedziałów i wyznaczania wskaźnika  $J_y$  (8) dla każdej kombinacji współczynników  $r_y$ ,  $r_{yx}$ ,  $r_{yy}$  i  $\hat{y}_1$  otrzymujemy punkt startowy dla funkcji `fminsearch` przedstawiony w pierwszym wierszu tabeli 5. Drugi wiersz tabeli 5 zawiera natomiast zestaw optymalnych współczynników zwrócony przez funkcję `fminsearch`.

$r_y$	$r_{yx}$	$r_{yy}$	$\hat{y}_1$
-4.44	0.03	-0.01	82.07
-7.19	0.04	0.00	41.32

Tabela 5: Parametry modelu otrzymane w wyniku estymacji z zastosowaniem niejawnej metody Eulera

Uzyskane przy pomocy niejawnej metody Eulera estymaty  $\hat{y}_2, \dots, \hat{y}_N$  zostały przedstawione na wykresie 6 na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$ .



Rysunek 6: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy niejawnej metody Eulera na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

### 2.2.3 Metoda Adamsa-Bashfortha trzeciego rzędu

Tym razem skorzystamy z metody Adamsa-Bashfortha trzeciego rzędu, która dana jest wzorem (13):

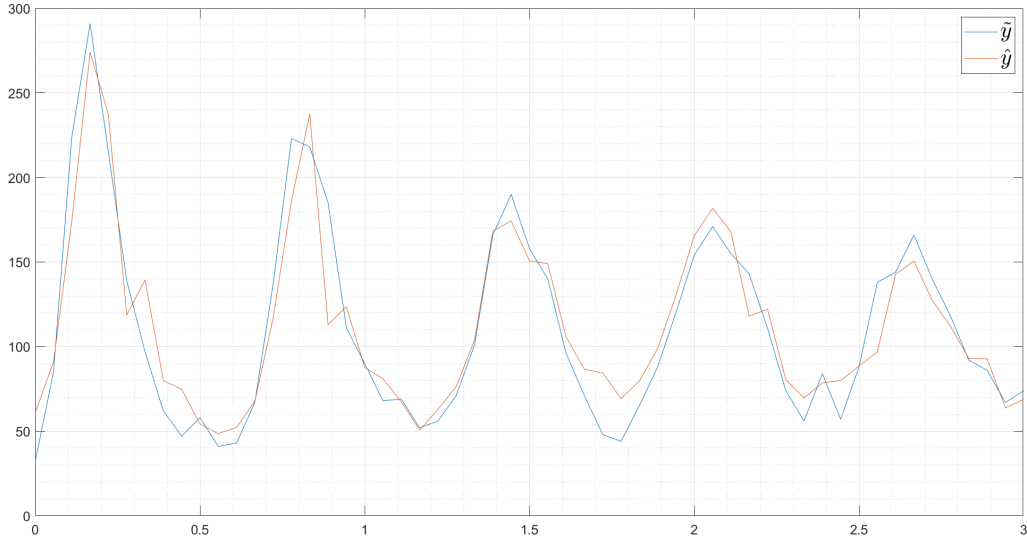
$$\begin{cases} \hat{y}_2 = \hat{y}_1 + f_y(\tilde{x}_1, \hat{y}_1) \cdot \Delta t_2 \\ \hat{y}_3 = \hat{y}_2 + \left( \frac{3}{2} f_y(\tilde{x}_2, \hat{y}_2) - \frac{1}{2} f_y(\tilde{x}_1, \hat{y}_1) \right) \cdot \Delta t_3 \\ \hat{y}_n = \hat{y}_{n-1} + \frac{1}{12} \left( 23 f_y(\tilde{x}_{n-1}, \hat{y}_{n-1}) - 16 f_y(\tilde{x}_{n-2}, \hat{y}_{n-2}) + 5 f_y(\tilde{x}_{n-3}, \hat{y}_{n-3}) \right) \cdot \Delta t_n \\ n = 4, \dots, N \end{cases} \quad (13)$$

W wyniku próbkowania przedziałów i wyznaczania wskaźnika  $J_y$  (8) dla każdej kombinacji współczynników  $r_y$ ,  $r_{yx}$ ,  $r_{yy}$  i  $\hat{y}_1$  otrzymujemy punkt startowy dla funkcji `fminsearch` przedstawiony w pierwszym wierszu tabeli 6. Drugi wiersz tabeli 6 zawiera natomiast zestaw optymalnych współczynników zwrócony przez funkcję `fminsearch`.

$r_y$	$r_{yx}$	$r_{yy}$	$\hat{y}_1$
-6.32	0.05	-0.02	60.00
-7.87	0.05	0.01	60.79

Tabela 6: Parametry modelu otrzymane w wyniku estymacji z zastosowaniem metody Adamsa-Bashfortha trzeciego rzędu

Uzyskane przy pomocy metody Adamsa-Bashfortha trzeciego rzędu estymaty  $\hat{y}_2, \dots, \hat{y}_N$  zostały przedstawione na wykresie 7 na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$ .



Rysunek 7: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy metody Adamsa-Bashfortha trzeciego rzędu na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

## 2.3 Estymacja parametrów obu równań układu

Naszym trzecim i ostatecznym podejściem do problemu estymacji parametrów  $r_x, r_{xy}, r_{xx}, r_y, r_{yx}, r_{yy}, \hat{x}_1, \hat{y}_1$  przyjętego modelu (1) będzie próba minimalizacji wskaźnika  $J$  dopasowania modelu do danych. Wskaźnik ten definiujemy wzorem (14):

$$J \equiv J_x + J_y = \sum_{n=1}^N (\hat{x}_n - \tilde{x}_n)^2 + \sum_{n=1}^N (\hat{y}_n - \tilde{y}_n)^2 \quad (14)$$

Wskaźnik  $J$  jest sumą wskaźników  $J_x$  oraz  $J_y$  zdefiniowanych w 2.1 oraz 2.2 odpowiednio.

Na tym etapie za optymalny punkt startowy dla funkcji `fminsearch` przyjmujemy jeden z wariantów zestawów parametrów modelu (1) uzyskanych w 2.1 oraz 2.2.

### 2.3.1 Jawna metoda Eulera

Wykorzystamy znane z 2.1.1 oraz 2.2.1 wzory iteracyjne (15):

$$\begin{cases} \hat{x}_n = \hat{x}_{n-1} + f_x(\hat{x}_{n-1}, \hat{y}_{n-1}) \cdot \Delta t_n, & n = 2, \dots, N \\ \hat{y}_n = \hat{y}_{n-1} + f_y(\hat{x}_{n-1}, \hat{y}_{n-1}) \cdot \Delta t_n, & n = 2, \dots, N \\ f_x(x, y) \equiv r_x x + r_{xy} xy + r_{xx} x^2 \\ f_y(x, y) \equiv r_y y + r_{yx} xy + r_{yy} y^2 \\ \Delta t_n \equiv t_n - t_{n-1} \end{cases} \quad (15)$$

Aby uzyskać satysfakcjonującą nas dokładność przybliżenia rozwiązania układu (1) musimy dobrać odpowiednio mały krok całkowania  $h$ . W naszym przypadku będzie to  $h = 0.001$ .

W celu obliczenia wartości wskaźnika  $J$  (14) zinterpolujemy uzyskane rozwiązanie w punktach  $t_1, t_2, \dots, t_N$  przy pomocy funkcji `interp1` ze środowiska MATLAB.

Funkcja `interp1` służy do interpolacji jednowymiarowej, czyli procesu szacowania wartości danych w punktach innych niż te, które są nam z góry znane. Jako parametry wejściowe `interp1` przyjmuje wektor węzłów interpolacji, wektor wartości funkcji interpolowanej w węzłach, a także wektor argumentów, w których chcemy uzyskać interpolowane wartości oraz opcjonalny argument określający metodę interpolacji.

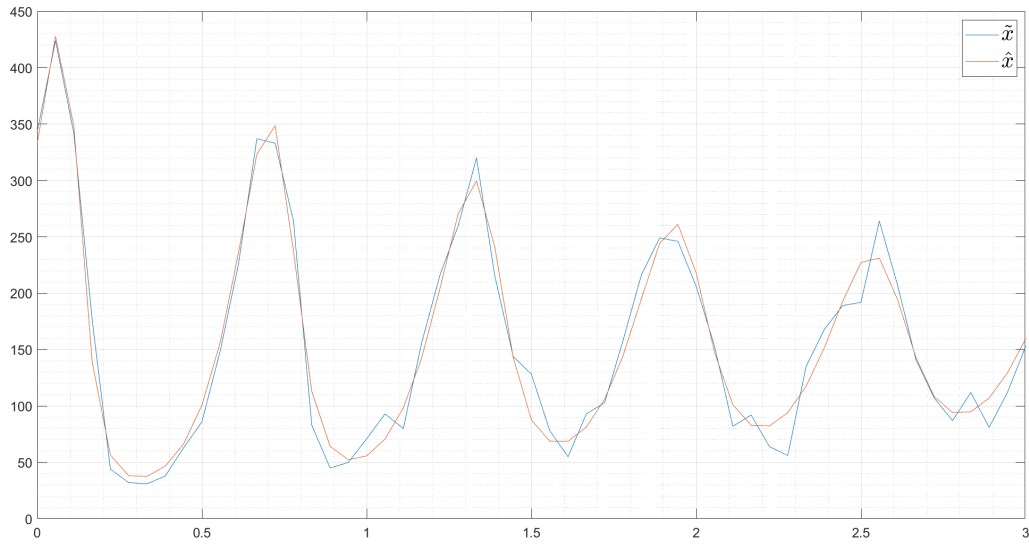
Za punkt startowy przyjmujemy kombinację parametrów uzyskanych w podrozdziałach 2.1.1 oraz 2.2.1, ponieważ one również bazowały na jawnej metodzie Eulera. W ten sposób mamy nadzieję zobaczyć poprawę dokładności estymacji danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$  oraz  $\tilde{y}_2, \dots, \tilde{y}_N$ .

Tabela 7 przedstawia zestaw optymalnych parametrów modelu (1) zwróconych przez funkcję `fminsearch`.

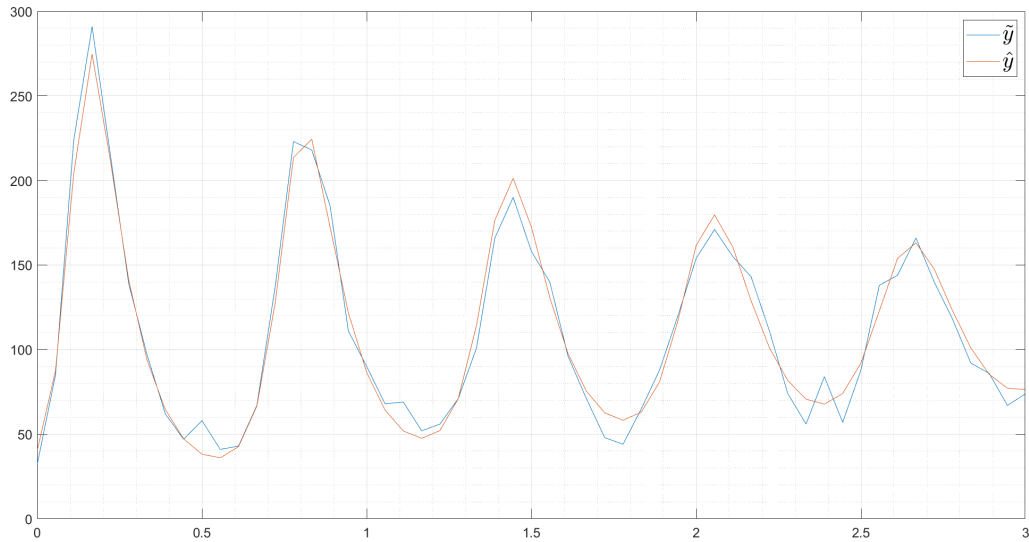
$r_x$	$r_{xy}$	$r_{xx}$	$\hat{x}_1$
17.17	-0.11	-0.026	357.20
$r_y$	$r_{yx}$	$r_{yy}$	$\hat{y}_1$
-4.29	0.05	-0.03	56.16

Tabela 7: Optymalne parametry modelu otrzymane w wyniku estymacji z zastosowaniem metody Eulera

Listing 8 zawiera implementację jawnej metody Eulera zastosowanej dla obu równań układu (1). Wykresy 8 i 9 przedstawiają estymaty  $\hat{x}_2, \dots, \hat{x}_N$  oraz  $\hat{y}_2, \dots, \hat{y}_N$  na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$  oraz  $\tilde{y}_2, \dots, \tilde{y}_N$  odpowiednio.



Rysunek 8: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$



Rysunek 9: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

Na wykresach 8 i 9 możemy zaobserwować poprawę dokładności wyników procesu estymacji w porównaniu do wyników otrzymanych w podrozdziałach 2.1.1 oraz 2.2.1.

### 2.3.2 Ode45

Do szacowania parametrów rozważanego modelu (1) możemy również posłużyć się wbudowaną w środowisko MATLAB funkcją `ode45`, która wykorzystuje parę metod Rungego-Kutty rzędu czwartego oraz piątego.

Wartością zwracaną funkcji `ode45` jest macierz zawierająca umieszczone wierszowo wektory reprezentujące wartości rozwiązania w punktach określonych odpowiednimi elementami wektora kolumnowego  $t$ , który jest jedną z wartości funkcji `ode45`.

Funkcja `ode45` sama dobiera sobie krok całkowania, dlatego ponownie, czyli tak jak w podrozdziale 2.3.1 do obliczenia wartości wskaźnika  $J$  (14) zinterpolujemy uzyskane rozwiązanie w punktach  $t_1, t_2, \dots, t_N$  przy pomocy funkcji `interp1`.

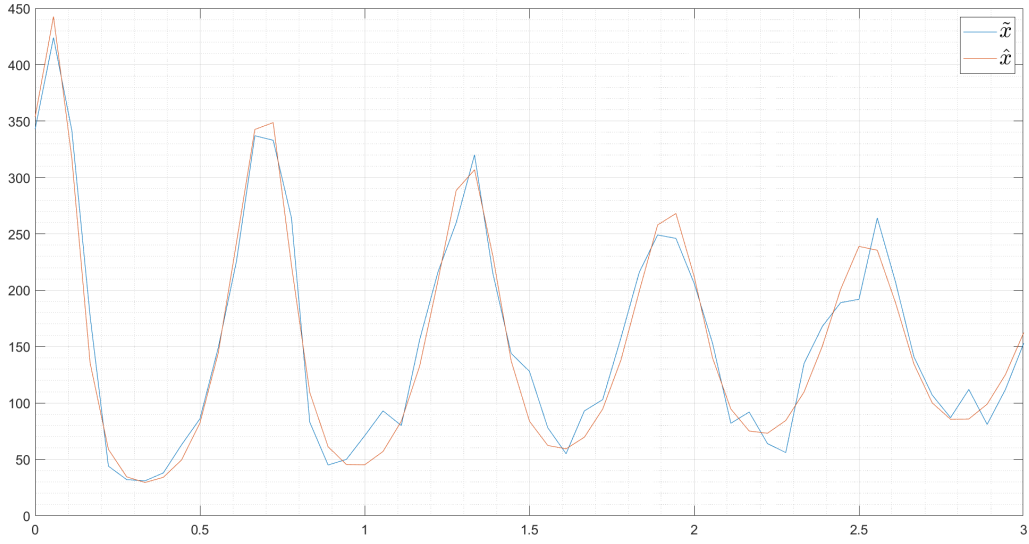
Tym razem za punkt startowy przyjmijmy kombinacje parametrów uzyskanych w podrozdziałach 2.1.3 oraz 2.2.3, czyli bazujące na metodzie Adamsa-Bashforth trzeciego rzędu.

Tabela 8 przedstawia zestaw optymalnych parametrów modelu (1) zwróconych przez funkcję `fminsearch`.

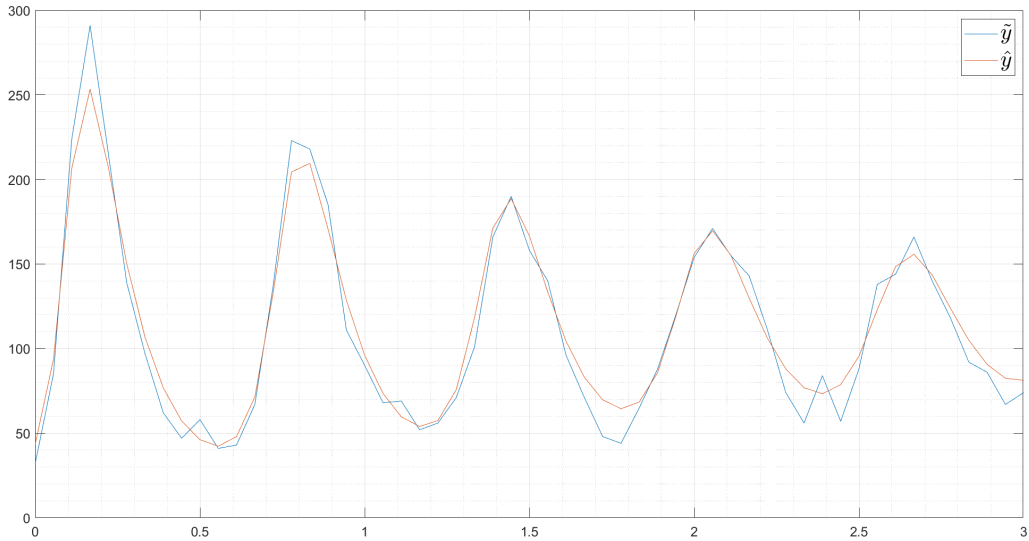
$r_x$	$r_{xy}$	$r_{xx}$	$\hat{x}_1$
18.57	-0.15	-0.01	353.86
$r_y$	$r_{yx}$	$r_{yy}$	$\hat{y}_1$
-7.84	0.05	0.01	46.60

Tabela 8: Optymalne parametry modelu otrzymane w wyniku estymacji z zastosowaniem funkcji `ode45`

Listing 9 zawiera implementację funkcji `ode45` zastosowanej dla obu równań układu (1). Wykresy 10 i 11 przedstawiają estymaty  $\hat{x}_2, \dots, \hat{x}_N$  oraz  $\hat{y}_2, \dots, \hat{y}_N$  na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$  oraz  $\tilde{y}_2, \dots, \tilde{y}_N$  odpowiednio.



Rysunek 10: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy funkcji `ode45` na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$



Rysunek 11: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy funkcji `ode45` na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

Na wykresach 10 i 11 możemy zaobserwować poprawę dokładności wyników procesu estymacji w porównaniu do wyników otrzymanych w podrozdziałach 2.1.3 oraz 2.2.3.

## 2.4 Stan równowagi w układzie Lotki-Volterry

W tym rozdziale zajmujemy się zagadnieniem znajdowania stanu równowagi pomiędzy gatunkami, których liczebność w modelu (1) opisują funkcje  $x$  i  $y$ .

Stan równowagi charakteryzuje się brakiem zmian wartości funkcji  $x$  i  $y$  w czasie. Warunek ten jest równoważny zerowaniu się pochodnych funkcji  $x$  i  $y$  tej samej chwili  $t$ . Stan równowagi opisują zatem warunki (16):

$$\begin{cases} \frac{dx}{dt} = 0 \\ \frac{dy}{dt} = 0 \end{cases} \quad (16)$$

Po podstawieniu znanych wartości do układu równań Lotki-Volterry (1) otrzymujemy nieliniowy układ równań (17):

$$\begin{cases} 0 = r_x x(t) + r_{xy} x(t)y(t) + r_{xx} x^2(t) \\ 0 = r_y y(t) + r_{yx} x(t)y(t) + r_{yy} y^2(t) \end{cases} \quad (17)$$

Rozwiązanie tego układu dostarcza nam szukanych wartości funkcji  $x$  i  $y$ , dla których układ Lotki-Volterry (1) pozostaje w stanie chwilowej równowagi.

Do rozwiązania układu (17) możemy skorzystać z funkcjonalności, jaką oferuje MATLAB Symbolic Toolbox, czyli funkcję `solve`, służącą do rozwiązywania równań i układów równań symbolicznych oraz funkcję `subs`, służącą podstawieniu wartości liczbowych pod zmienne symboliczne.

Układ równań (17) rozwiązaliśmy dla zestawów parametrów wyznaczonych w podrozdziałach 2.3.1 oraz 2.3.2. Listing 10 zawiera implementację funkcji rozwiązującej układ (17) z wykorzystaniem MATLAB Symbolic Toolbox.

Tabela 9 zawiera rozwiązania układu (17) dla zestawu parametrów wyznaczonych w podrozdziale 2.3.1, natomiast tabela 10 dla zestawu parametrów wyznaczonych w podrozdziale 2.3.2. Wartości w obu tabelach zostały zaokrąglone do liczb całkowitych w celu przypomnienia znaczenia wyznaczanych wartości  $x$  oraz  $y$ , czyli liczebności poszczególnych gatunków modelu Lotki-Volterry.

$x$	0	149	4749	0
$y$	0	112	0	-2256

Tabela 9: Rozwiązania układu (17) dla zestawu parametrów wyznaczonych w podrozdziale 2.3.1

$x$	0	148	2104	0
$y$	0	117	0	-918

Tabela 10: Rozwiązania układu (17) dla zestawu parametrów wyznaczonych w podrozdziale 2.3.2

Dodatkowo, przyjmujemy, że interesującymi dla nas rozwiązaniami, są rozwiązania, w których liczebności obu populacji  $x$  i  $y$  są niezerowe (i oczywiście dodatnie).

Takie pary wartości  $x$  i  $y$  to odpowiednio (146, 124) dla zestawu parametrów z 2.3.1 oraz (147, 117) dla zestawu parametrów z 2.3.2. Z zadowoleniem stwierdzamy, że pary te są do siebie "zbliżone".

### 3 Testy na danych rzeczywistych oraz interpretacja wyników

W ostatniej części pracy zbadamy, w jakim stopniu nasz sposób postępowania sprawdza się na przykładzie rzeczywistych pomiarów liczebności populacji. W tym celu postąpimy zgodnie z tą samą metodyką, co w podrozdziałach 2.1, 2.2 oraz 2.3.

Dane, które w tym rozdziale rozważamy, pochodzą z klasycznego przykładu obrazującego zastosowanie układu równań Lotki-Volterry, czyli badań przeprowadzonych na populacji zajęcy amerykańskich i rysia kanadyjskich z Zatoki Hudsona w Ameryce Północnej. Badania te miały miejsce od 1900 do 1920 roku.



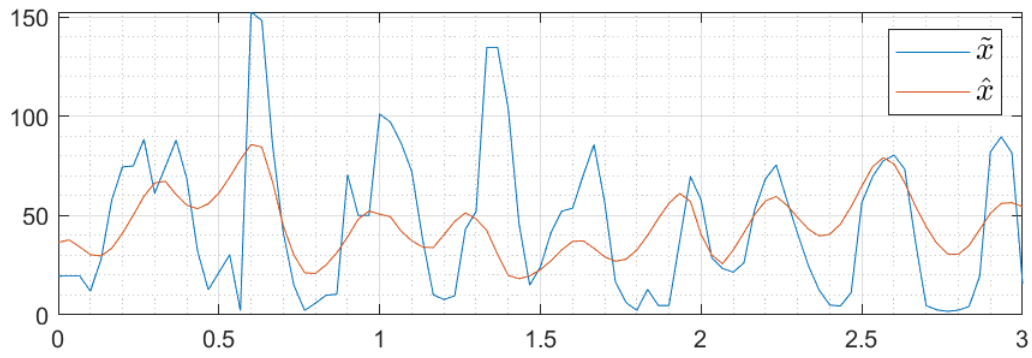
Rysunek 12: Przedstawiciel gatunku rysia kanadyjskiego



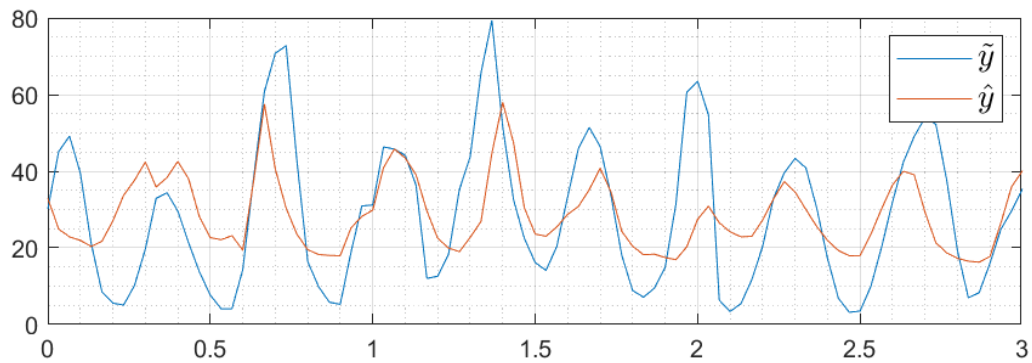
Rysunek 13: Przedstawiciel gatunku zajęcia amerykańskiego

Wykresy 14 i 15 stanowią odpowiedniki wykresów 2 oraz 5 odpowiednio. Już w rozdziałach 2.1.1 i 2.2.1, mogliśmy zaobserwować, że zastosowane przez nas metody estymacji parametrów przyjętego modelu (1) mają tendencję do wygładzania nagłych zmian funkcji liczebności  $x$  i  $y$ . Jest to szczególnie widoczne na przykładzie gatunków zajęcia amerykańskiego i rysia kanadyjskiego, których to liczebności zmieniają się bardziej gwałtownie, niż wartości danych syntetycznych, na których pracowaliśmy w rozdziale 2.



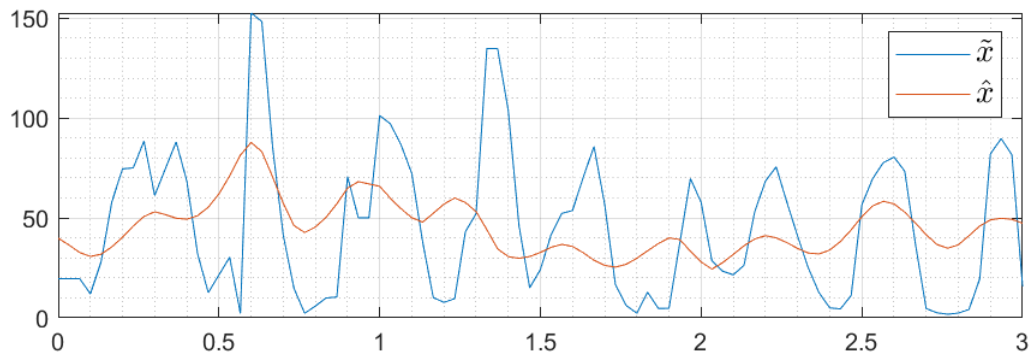


Rysunek 14: Estymaty  $\tilde{x}_2, \dots, \tilde{x}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\hat{x}_2, \dots, \hat{x}_N$

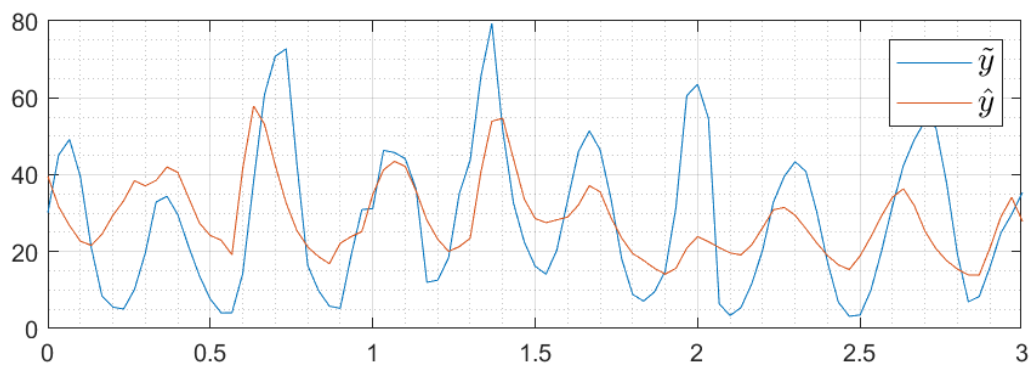


Rysunek 15: Estymaty  $\tilde{y}_2, \dots, \tilde{y}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\hat{y}_2, \dots, \hat{y}_N$

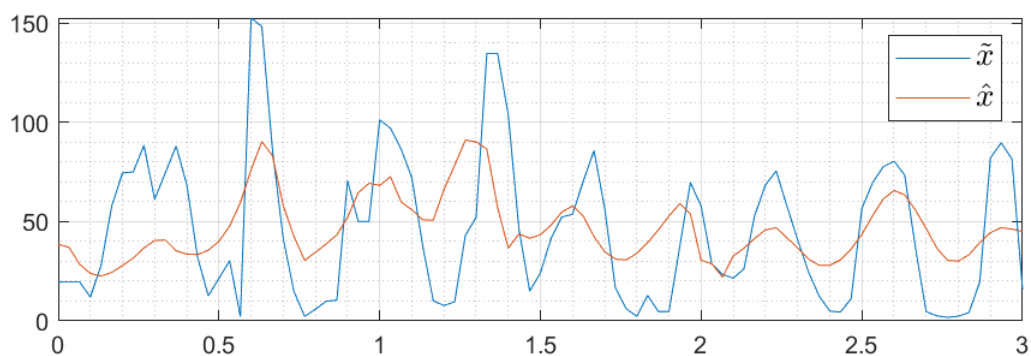
Wykresy 16 i 17 stanowią odpowiedniki wykresów 3 oraz 6, natomiast wykresy 18 i 19, odpowiedniki wykresów 4 i 7. Pojedyncza estymacja parametrów pierwszego oraz drugiego równania modelu (1) z wykorzystaniem niejawnej metody Eulera oraz metody Adamsa-Bashfortha trzeciego rzędu również daje wyniki mniej dokładne niż te, które otrzymywaliśmy w rozdziale 2 dla tych samych metod iteracyjnych.



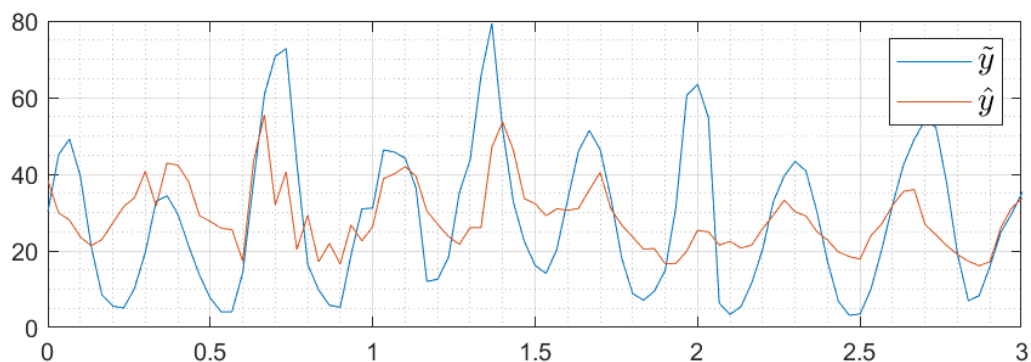
Rysunek 16: Estymaty  $\tilde{x}_2, \dots, \tilde{x}_N$  uzyskane przy pomocy niejawnej metody Eulera na tle danych pomiarowych  $\hat{x}_2, \dots, \hat{x}_N$



Rysunek 17: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy niejawnej metody Eulera na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

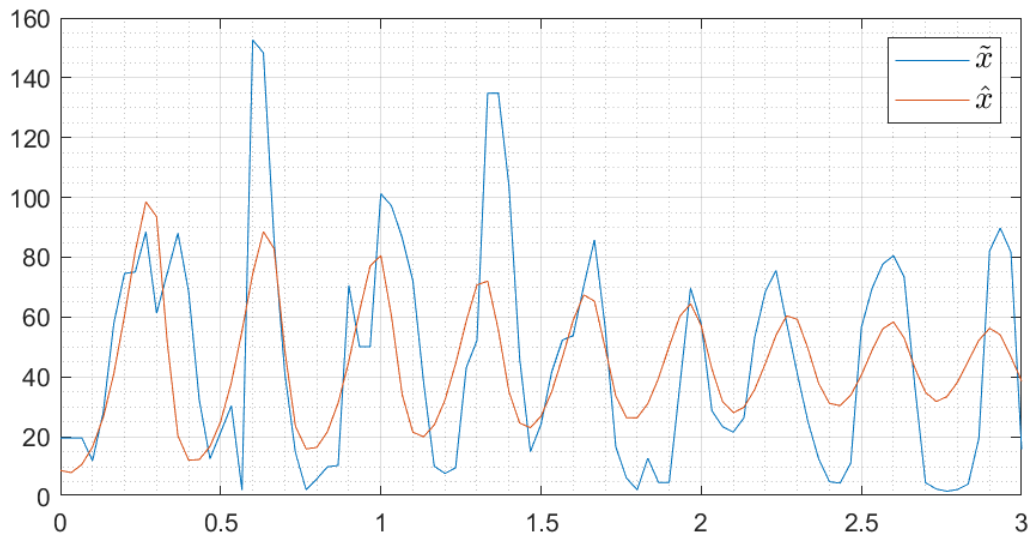


Rysunek 18: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy metody Adamsa-Bashfortha trzeciego rzędu na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$

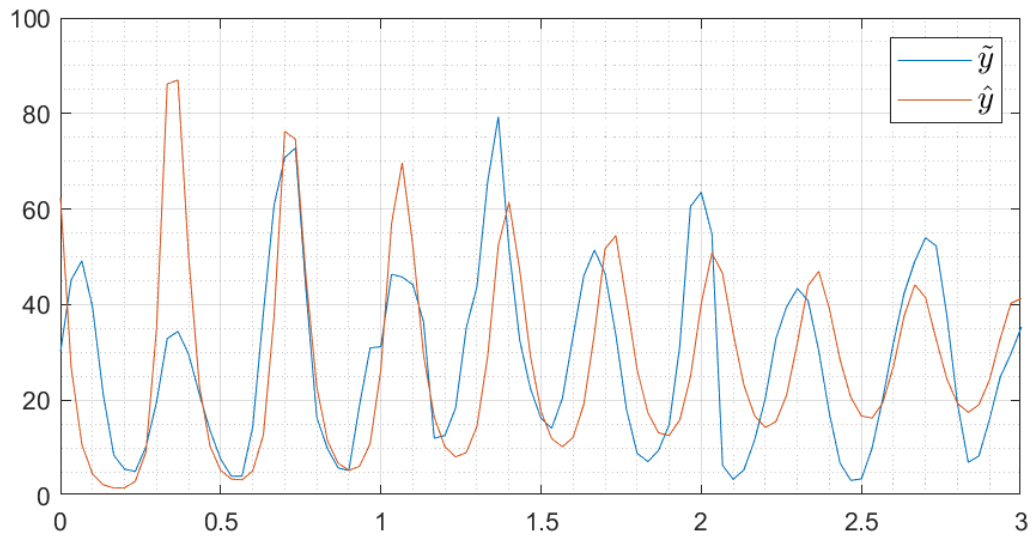


Rysunek 19: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy metody Adamsa-Bashfortha trzeciego rzędu na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

Zastanawiając się nad przyczyną takiego stanu rzeczy, dochodzimy do wniosku, że dane rzeczywiste są podatne na różnego rodzaju zaburzenia. Pamiętajmy, że przyjęty przez nas model (1) zakłada występowanie w ekosystemie tylko i wyłącznie dwóch gatunków, jednak środowisko, w którym żyją zajmujące amerykańskie (13) i rysie kanadyjskie (12) jest znacznie bardziej złożone. Na liczebność rzeczywistych gatunków w dużej mierze ma wpływ obecność organizmów spoza rozważanego przez nas, zamkniętego układu.

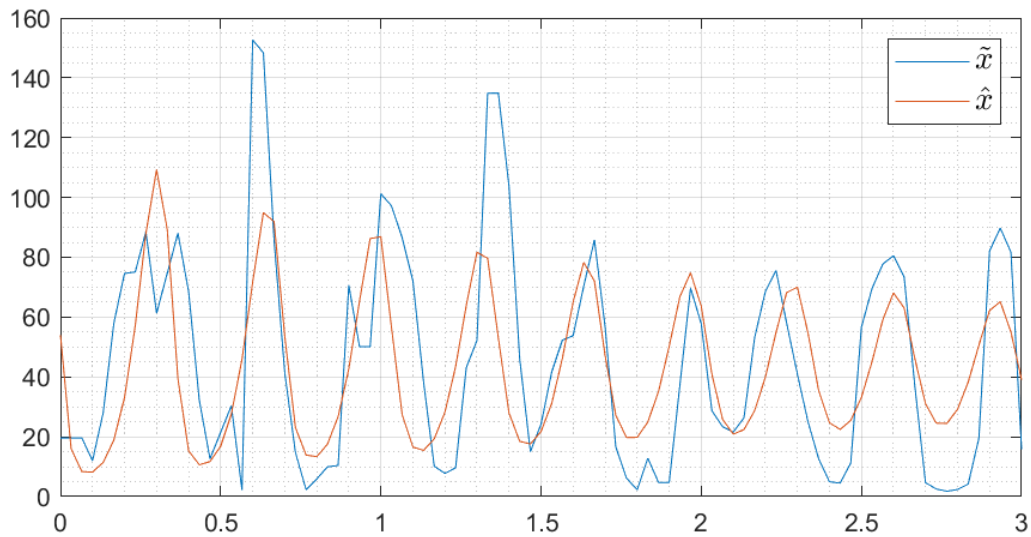


Rysunek 20: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy funkcji ode45 na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$

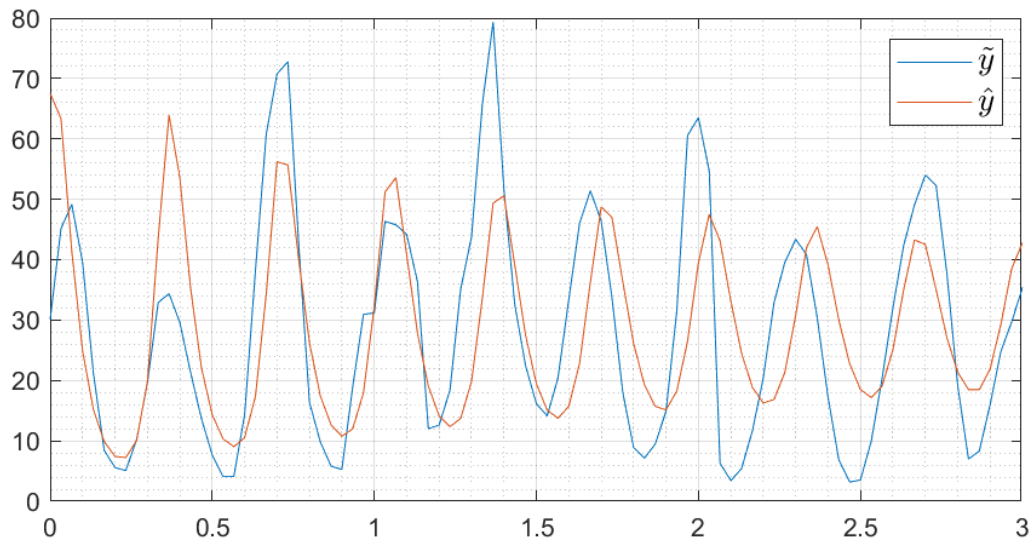


Rysunek 21: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy funkcji ode45 na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

Wykresy 20 i 21 stanowią odpowiedniki wykresów 10 oraz 11, natomiast wykresy 22 i 23, odpowiedniki wykresów 14 i 15. W przypadku jednoczesnej estymacji parametrów obu równań modelu (1) tak jak w rozdziale 2 otrzymujemy poprawę dopasowania do danych pomiarowych, jednak wciąż estymaty  $\hat{x}_2, \dots, \hat{x}_N$  oraz  $\hat{y}_2, \dots, \hat{y}_N$  odbiegają znacząco od  $\tilde{x}_2, \dots, \tilde{x}_N$  i  $\tilde{y}_2, \dots, \tilde{y}_N$ .



Rysunek 22: Estymaty  $\hat{x}_2, \dots, \hat{x}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\tilde{x}_2, \dots, \tilde{x}_N$



Rysunek 23: Estymaty  $\hat{y}_2, \dots, \hat{y}_N$  uzyskane przy pomocy jawnej metody Eulera na tle danych pomiarowych  $\tilde{y}_2, \dots, \tilde{y}_N$

## 4 Wnioski

Równania Lotki-Volterry pozwalają nam lepiej zrozumieć dynamikę populacji w ekosystemach drapieżnik-ofiara. Z wartości współczynników rozważanego modelu (1) możemy pozyskać istotne informacje na temat wzajemnego wpływu obu gatunków.

W rozdziale 2 przekonaliśmy się, że nawet dla syntetycznych danych pomiarowych problem estymacji parametrów przyjętego modelu (1) jest zagadnieniem złożonym - wymagającym wielu kroków i podejść.

W rozdziale 3 zobaczyliśmy, że to, co zdaje się dobrze działać dla syntetycznych danych pomiarowych, nie do końca sprawdza się w przypadku danych rzeczywistych. Środowisko, w którym żyją rozważane przez nas zające amerykańskie (13) i rysie kanadyjskie (12) jest znacznie bardziej złożone, a na liczebności tych gatunków może wpływać wiele czynników, których nasz prosty model matematyczny nie uwzględnia.

## 5 Listing

```
function [pxo, pyo] = estimation_manager(filename, nx, ny, f, options)
% Project 2
% Adam Gracikowski, 327350
%
% Funkcja realizująca część wspólna rozwiązywanego w zadaniach 1. i 2.
% problemu estymacji parametrów modelu
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   nx - wektor określający gęstość próbkowania przedziałów dla parametrów
%       pierwszego równania różniczkowego
%   ny - wektor określający gęstość próbkowania przedziałów dla parametrów
%       drugiego równania różniczkowego
%   f - iteracyjna metoda do rozwiązywania pojedynczego równania
%       różniczkowego
%   options - argument typu optimset modyfikujący wywołanie funkcji
%       fminsearch wewnątrz estimation_manager
%   ft - uchwyt do funkcji skalującej wektor czasu
% OUTPUT:
%   pxo - optymalne parametry dla pierwszego równania różniczkowego
%   pyo - optymalne parametry dla drugiego równania różniczkowego

if nargin < 6; ft = @(t) t; end
if nargin < 5; options = []; end
[tp, xp, yp, Px, Py] = getData(filename, nx, ny, ft);
tp = ft(tp);
px_min = 0*nx; py_min = 0*ny;
Jx_min = Inf; Jy_min = Inf;
% liczba wszystkich kombinacji parametrów pierwszego równania
nC = size(Px, 1);
% szukanie najmniejszej wartości Jx
for i = 1:nC
    xe = f(Px(i,:), tp, yp);
    Jx = sum((xe - xp).^2);
    if Jx < Jx_min
        Jx_min = Jx;
        px_min = Px(i,:);
    end % if
end % for
% liczba wszystkich kombinacji parametrów pierwszego równania
nC = size(Py, 1);
% szukanie najmniejszej wartości Jy
for i = 1:nC
    ye = f(Py(i,:), tp, xp);
    Jy = sum((ye - yp).^2);
    if Jy < Jy_min
        Jy_min = Jy;
        py_min = Py(i,:);
    end % if
end % for
% wywołania fminsearch
matchingIndicatorX = @(p) sum((f(p, tp, yp) - xp).^2);
pxo = fminsearch(matchingIndicatorX, px_min, options);
matchingIndicatorY = @(p) sum((f(p, tp, xp) - yp).^2);
pyo = fminsearch(matchingIndicatorY, py_min, options);
end % function
```

Listing 1: Implementacja metodyki estymacji parametrów modelu (1)

```

function ve = explicit_euler(p, tp, up)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   p - parametry dla rownania rozniczkowego, czyli zestaw:
%       (r_x, r_xy, r_xx, x_1) lub (r_y, r_yx, r_yy, y_1)
%   tp - wektor wartosci czasu z pomiarow
%   up - wartosci liczebnosci populacji z pomiarow
% OUTPUT:
%   ve - wektor obliczonych estymat liczebnosci populacji

f = @(v, u) p(1)*v + p(2)*v*u + p(3)*v*v;
ve = 0*up;

% wartosc poczatkowa
ve(1) = p(end);

for i = 2:length(ve)
    dt = tp(i) - tp(i-1);
    ve(i) = ve(i-1) + f(ve(i-1), up(i-1))*dt;
end % for

end % function

```

Listing 2: Implementacja jawnej metody Eulera dostosowana do przyjętego modelu (1)

```

function [pxo, pyo] = zad1ab(filename, options, ft)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebnosci dwoch gatunkow
%   options - argument typu optimset modyfikujacy wywołanie funkcji
%             fminsearch wewnątrz estimation_manager, domyslnie []
%   ft - uchwyt do funkcji skalujacej wektor czasu, domyslnie ft(t) = t
% OUTPUT:
%   pxo - optymalne parametry dla pierwszego rownania rozniczkowego
%   pyo - optymalne parametry dla drugiego rownania rozniczkowego

if nargin < 3; ft = @(t) t; end
if nargin < 2; options = []; end

% gestosc probkowania wskazanych przedzialow
nx = [20, 10, 20, 50];
ny = [20, 10, 20, 40];

[pxo, pyo] = estimation_manager(filename, nx, ny, ...
    @explicit_euler, options, ft);

end % function

```

Listing 3: Estymacja parametrów pierwszego i drugiego równania przy pomocy jawnej metody Eulera

```

function ve = implicit_euler(p, tp, up)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   p - parametry dla rownania rozniczkowego, czyli zestaw:
%       (r_x, r_xy, r_xx, x_1) lub (r_y, r_yx, r_yy, y_1)
%   tp - wektor wartosci czasu z pomiarow
%   up - wartosci liczebnosci populacji z pomiarow
% OUTPUT:
%   ve - wektor obliczonych estymat liczebnosci populacji

ve = 0*up;
% wartosc poczatkowa
ve(1) = p(4);

for i = 2:length(ve)
    dt = tp(i) - tp(i-1);
    % w kazdej iteracji rozwiazuujemy rownanie kwadratowe
    [x1, x2] = solve_quadratic([p(3)*dt, p(1)*dt + ...
        p(2)*up(i)*dt - 1, ve(i-1)]);
    % wybieramy wieksze rozwiazanie
    ve(i) = max(x1, x2);
end % for
function [x1, x2] = solve_quadratic(p)
    % funkcja rozwiazuujaca rownanie kwadratowe
    a = p(1); b = p(2); c = p(3);
    delta = b^2 - 4*a*c;
    if delta < 0
        x1 = NaN;
        x2 = NaN;
        return;
    end % if
    % unikamy odejmowania liczb bliskich
    % i zwiazanych z nim bkedow numerycznych
    x1 = (-b - sign(b)*sqrt(delta)) / (2*a);
    x2 = c / (a * x1);
end
end % function

```

Listing 4: Implementacja niejawnej metody Eulera dostosowana do przyjętego modelu (1)

```

function [pxo, pyo] = zad2a(filename, options, ft)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   options - argument typu optimset modyfikujący wywołanie funkcji
%             fminsearch wewnątrz estimation_manager, domyślnie []
%   ft - uchwyt do funkcji skalującej wektor czasu, domyślnie ft(t) = t
% OUTPUT:
%   pxo - optymalne parametry dla pierwszego równania różniczkowego
%   pyo - optymalne parametry dla drugiego równania różniczkowego

if nargin < 3; ft = @(t) t; end
if nargin < 2; options = []; end

% gestosc probkowania wskazanych przedzialow
nx = [10, 20, 20, 30];
ny = [10, 30, 20, 30];

[pxo, pyo] = estimation_manager(filename, nx, ny, ...
    @implicit_euler, options, ft);
end % function

```

Listing 5: Estymacja parametrów pierwszego i drugiego równania przy pomocy niejawnej metody Eulera

```

function ve = adams_bashforth(p, tp, up)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   p - parametry dla równania różniczkowego, czyli zestaw:
%       (r_x, r_xy, r_xx, x_1) lub (r_y, r_yx, r_yy, y_1)
%   tp - wektor wartości czasu z pomiarów
%   up - wartości liczebności populacji z pomiarów
% OUTPUT:
%   ve - wektor obliczonych estymat liczebności populacji

f = @(v, u) p(1)*v + p(2)*v*u + p(3)*v*v;

ve = 0*up;

% wartosci poczatkowe
ve(1) = p(end);
ve(2) = ve(1) + f(ve(1), up(1)) * ...
    (tp(2) - tp(1));
ve(3) = ve(2) + (3/2 * f(ve(2), up(2)) - 1/2 * f(ve(1), up(1))) * ...
    (tp(3) - tp(2));

for i = 4:length(ve)
    dt = tp(i) - tp(i-1);
    ve(i) = ve(i-1) + 1/12 * (23*f(ve(i-1), up(i-1)) - ...
        16*f(ve(i-2), up(i-2)) + ...
        5*f(ve(i-3), up(i-3)))*dt;
end % for

end % function

```

Listing 6: Implementacja metody Adamsa-Bashforth trzeciego rzędu dostosowana do przyjętego modelu (1)



```

function [pxo, pyo] = zad2b(filename, options, ft)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   options - argument typu optimset modyfikujący wywołanie funkcji
%             fminsearch wewnątrz estimation_manager, domyślnie []
%   ft - uchwyt do funkcji skalującej wektor czasu, domyślnie ft(t) = t
% OUTPUT:
%   pxo - optymalne parametry dla pierwszego równania różniczkowego
%   pyo - optymalne parametry dla drugiego równania różniczkowego

if nargin < 3; ft = @(t) t; end
if nargin < 2; options = []; end

% gestosc probkowania wskazanych przedzialow
nx = [20, 20, 20, 50];
ny = [20, 20, 20, 20];

[pxo, pyo] = estimation_manager(filename, nx, ny, ...
    @adams_bashforth, options, ft);
end % function

```

Listing 7: Estymacja parametrów pierwszego i drugiego równania przy pomocy metody Adamsa-Bashforth trzeciego rzędu

```

function [pxo, pyo, xe, ye] = zad3b(filename, px, py, options, ft, h)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   px - parametry dla pierwszego równania różniczkowego
%   py - parametry dla drugiego równania różniczkowego
%   options - argument typu optimset modyfikujący wywołanie funkcji
%             fminsearch, domyślnie []
%   ft - uchwyt do funkcji skalującej wektor czasu, domyślnie ft(t) = t
%   h - długość kroku całkowania, domyślnie h = 0.01
% OUTPUT:
%   pxo - optymalne parametry dla pierwszego równania różniczkowego
%   pyo - optymalne parametry dla drugiego równania różniczkowego
%   xe - estymaty wartości funkcji x uzyskane przy pomocy
%        jawnej metody Eulera
%   ye - estymaty wartości funkcji y uzyskane przy pomocy
%        jawnej metody Eulera

if nargin < 6; h = 0.01; end
if nargin < 5; ft = @(t) t; end
if nargin < 4; options = []; end

% wczytanie danych syntetycznych
dataTable = readtable(filename);
dataTable.Properties.VariableNames = {'t', 'x', 'y'};
xp = dataTable.x;
yp = dataTable.y;
tp = ft(dataTable.t);
t = min(tp):h:max(tp);

% wywołanie fminsearch
p_opt = fminsearch(@matchingIndicator, [px, py], options);
pxo = p_opt(1:4);
pyo = p_opt(5:end);

[xe, ye] = explicit_euler3(tp, p_opt);

function J = matchingIndicator(p)
    [xe, ye] = explicit_euler3(tp, p);
    J = sum((xe - xp).^2) + sum((ye - yp).^2);
end
function [xe, ye] = explicit_euler3(tp, p)
    px = p(1:4); py = p(5:end);
    fx = @(x, y) px(1)*x + px(2)*x*y + px(3)*x*x;
    fy = @(x, y) py(1)*y + py(2)*y*x + py(3)*y*y;
    xe = 0*tp; xe(1) = px(4);
    ye = 0*tp; ye(1) = py(4);
    for i = 2:length(t)
        dt = t(i) - t(i-1);
        xe(i) = xe(i-1) + fx(xe(i-1), ye(i-1))*dt;
        ye(i) = ye(i-1) + fy(xe(i-1), ye(i-1))*dt;
    end % for
    xe = interp1(t, xe, tp);
    ye = interp1(t, ye, tp);
end
end % function

```

Listing 8: Jednoczesna estymacja parametrów obu równań modelu (1) przy pomocy jawnej metody Eulera

```

function [pxo, pyo, xe, ye] = zad3a(filename, px, py, options, ft)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   px - parametry dla pierwszego równania różniczkowego
%   py - parametry dla drugiego równania różniczkowego
%   options - argument typu optimset modyfikujący wywołanie funkcji
%             fminsearch, domyślnie []
%   ft - uchwyt do funkcji skalującej wektor czasu, domyślnie ft(t) = t
% OUTPUT:
%   pxo - optymalne parametry dla pierwszego równania różniczkowego
%   pyo - optymalne parametry dla drugiego równania różniczkowego
%   xe - estymaty wartości funkcji x uzyskane przy pomocy
%        jawnej metody Eulera
%   ye - estymaty wartości funkcji y uzyskane przy pomocy
%        jawnej metody Eulera

if nargin < 5; ft = @(t) t; end
if nargin < 4; options = []; end

% wczytanie danych syntetycznych
dataTable = readtable(filename);
dataTable.Properties.VariableNames = {'t', 'x', 'y'};
xp = dataTable.x;
yp = dataTable.y;
tp = ft(dataTable.t);

% wywołanie fminsearch
p_opt = fminsearch(@matchingIndicator, [px, py], options);
pxo = p_opt(1:4);
pyo = p_opt(5:end);

[xe, ye] = ode_45(tp, p_opt);

function J = matchingIndicator(p)
    [xe, ye] = ode_45(tp, p);
    J = sum((xe - xp).^2) + sum((ye - yp).^2);
end
function [xe, ye] = ode_45(tp, p)
    px = p(1:4); py = p(5:end);
    [t, v] = ode45(@f, [min(tp) max(tp)], [px(4) py(4)]);
    xe = interp1(t, v(:,1), tp);
    ye = interp1(t, v(:,2), tp);
    function dvdt = f(~, v)
        x = v(1);
        y = v(2);
        dvdt = [px(1)*x + px(2)*x.*y + px(3)*x.*x; ...
                py(1)*y + py(2)*x.*y + py(3)*y.*y];
    end
end
end % function

```

Listing 9: Jednoczesna estymacja parametrów obu równań modelu (1) przy pomocy funkcji ode45

```

function sol = zad4(px, py)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   px - parametry dla pierwszego rownania rozniczkowego
%   py - parametry dla drugiego rownania rozniczkowego
% OUTPUT:
%   sol - macierz zawierajaca w swoich wierszach pary wartosci (x, y)
%         bedace rozwiazaniem rozwazanego ukladu rownan nieliniowych

% deklaracja zmiennych symbolicznych
syms x y r_x r_xy r_xx r_y r_yx r_yy real

% zdefiniowanie rownan ukladu
eq1 = r_x*x + r_xy *x* y + r_xx*x^2 == 0;
eq2 = r_y*y + r_yx*x*y + r_yy*y^2 == 0;

sol = solve([eq1, eq2], [x, y]);

% podstawienie wartosci pod zmienne symboliczne
sol = subs(sol, [r_x r_xy r_xx r_y r_yx r_yy], ...
    [px(1:end-1) py(1:end-1)]);
sol = [double(sol.x), double(sol.y)];

end % function

```

Listing 10: Implementacja rozwiązania problemu stanu równowagi w układzie Lotki-Volterra

```

function result = combvec(varargin)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   varargin - dowolna liczba wektorow poziomych zawierajacych kolejne
%              wartosci, ktorzych zestawy roznych kombinacji chcemy znalezc
% OUTPUT:
%   result - macierz zawierajaca w swoich wierszach wszystkie kombinacje
%            wartosci zapisanych w wektorach wejsciowych

lengths = cellfun(@length, varargin);

% sprawdzenie poprawnosci parametrow wejsciowych
if nargin == 0 || any(lengths == 0)
    result = []; return;
end % if

% liczba wszystkich kombinacji
nC = prod(lengths);
result = zeros(nC, nargin);

for i = 1:nargin
    rep = nC / lengths(i);
    result(:, i) = repmat(reshape(repmat(varargin{i}, rep, 1), [], 1), ...
        size(result, 1)/(rep*lengths(i)), 1);
    nC = nC / lengths(i);
end % for

end % function

```

Listing 11: Funkcja generująca macierz kombinacji parametrów równań dla modelu 1

```

function [tp, xp, yp, Px, Py] = getData(filename, nx, ny, ft)
% Project 2
% Adam Gracikowski, 327350
%
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   nx - wektor określający gęstość próbkowania przedziałów dla parametrów
%        pierwszego równania różniczkowego
%   ny - wektor określający gęstość próbkowania przedziałów dla parametrów
%        drugiego równania różniczkowego
%   ft - uchwyt do funkcji skalującej wektor czasu, domyślnie ft(t) = t
% OUTPUT:
%   tp - wektor wartości czasu z pomiarów zawartych w pliku filename
%   xp - wektor wartości funkcji x z pomiarów zawartych w pliku filename
%   yp - wektor wartości funkcji y z pomiarów zawartych w pliku filename
%   Px - macierz, zawierająca wszystkie kombinacje parametrów pierwszego
%        równania, (w jej wierszach kolejno: r_x, r_xy, r_xx, x_1)
%   Py - macierz, zawierająca wszystkie kombinacje parametrów drugiego
%        równania, (w jej wierszach kolejno: r_y, r_yx, r_yy, y_1)

if nargin < 4; ft = @(t) t; end

% wczytanie danych syntetycznych
dataTable = readtable(filename);

dataTable.Properties.VariableNames = {'t', 'x', 'y'};
xp = dataTable.x;
yp = dataTable.y;
tp = ft(dataTable.t);

% próbkowanie wskazanych przedziałów parametrów
px1 = linspace(0, 40, nx(1)); py1 = linspace(-40, 0, ny(1));
px2 = linspace(-1, 0, nx(2)); py2 = linspace(0, 1, ny(2));
px3 = linspace(-0.1, 1, nx(3)); py3 = linspace(-0.1, 0, ny(3));
px4 = linspace(100, 1000, nx(4)); py4 = linspace(10, 200, ny(4));

% tworzenie macierzy kombinacji parametrów
Px = combvec(px1, px2, px3, px4);
Py = combvec(py1, py2, py3, py4);

end % function

```

Listing 12: Funkcja odpowiedzialna za wczytywanie danych z pliku

```

function [] = simulation(filename, options)
% Project 2
% Adam Gracikowski, 327350
%
% Funkcja wywołująca funkcje z zadaniami od 1 do 4 oraz
% tworząca odpowiednie wykresy
% INPUT:
%   filename - nazwa pliku z danymi na temat liczebności dwóch gatunków
%   options - struktura zawierająca parametry symulacji

name = strcat("Estymacja dla pliku: ", filename);

tp = options.ft(options.tp);
xp = options.xp;
yp = options.yp;

% zadania 1:
[pxe, pye] = zad1ab(filename, options.set1, options.ft);
xe = explicit_euler(pxe, tp, yp);
ye = explicit_euler(pye, tp, xp);

fig = figure;
set(fig, 'Name', name);

addPlot([3 2 1], tp, xp, xe, ["$\tilde{x}$", "$\hat{x}$"], ...
    "Estymacja $\tilde{x}$ przy pomocy jawnej metody Eulera");
addPlot([3 2 2], tp, yp, ye, ["$\tilde{y}$", "$\hat{y}$"], ...
    "Estymacja $\tilde{y}$ przy pomocy jawnej metody Eulera");

% zadanie 2:

% a)
[pxi, pyi] = zad2a(filename, options.set2a, options.ft);
xi = implicit_euler(pxi, tp, yp);
yi = implicit_euler(pyi, tp, xp);

addPlot([3 2 3], tp, xp, xi, ["$\tilde{x}$", "$\hat{x}$"], ...
    "Estymacja $\tilde{x}$ przy pomocy niejawnej metody Eulera");
addPlot([3 2 4], tp, yp, yi, ["$\tilde{y}$", "$\hat{y}$"], ...
    "Estymacja $\tilde{y}$ przy pomocy niejawnej metody Eulera");

% b)
[pxa, pya] = zad2b(filename, options.set2b, options.ft);
xa = adams_bashforth(pxa, tp, yp);
ya = adams_bashforth(pya, tp, xp);

addPlot([3 2 5], tp, xp, xa, ["$\tilde{x}$", "$\hat{x}$"], ...
    "Estymacja $\tilde{x}$ przy pomocy metody Adamsa-Bashforth");
addPlot([3 2 6], tp, yp, ya, ["$\tilde{y}$", "$\hat{y}$"], ...
    "Estymacja $\tilde{y}$ przy pomocy metody Adamsa-Bashforth");

% zadanie 3:

% a)
[pxoa, pyoa, xo, yo] = zad3a(filename, pxe, pye, ...
    options.set3a, options.ft);

fig = figure;
set(fig, 'Name', name);
addPlot([2 2 1], tp, xp, xo, ["$\tilde{x}$", "$\hat{x}$"], ...
    "Estymacja $\tilde{x}$ przy pomocy ode45, punkt startowy z 1a");

```

```

addToPlot([2 2 2], tp, yp, yo, ["$\tilde{y}$", "$\hat{y}$"], ...
    "Estymacja $\tilde{y}$ przy pomocy ode45, punkt startowy z 1b");

% b)
[pxob, pyob, xo, yo] = zad3b(filename, pxa, pya, ...
    options.set3b, options.ft, options.h);

addToPlot([2 2 3], tp, xp, xo, ["$\tilde{x}$", "$\hat{x}$"], ...
    "Estymacja $\tilde{x}$ przy pomocy jawnej metody Eulera, punkt startowy z 2b");
addToPlot([2 2 4], tp, yp, yo, ["$\tilde{y}$", "$\hat{y}$"], ...
    "Estymacja $\tilde{y}$ przy pomocy jawnej metody Eulera, punkt startowy z 2b");

% zadanie 4:

% obliczamy rozwiazania ukkladu okreslajacego stan rownowagi dla parametrow
% startowych z zadania 3a:
sol3a = zad4(pxo, pyo);

% obliczamy rozwiazania ukkladu okreslajacego stan rownowagi dla parametrow
% startowych z zadania 3b:
sol3b = zad4(pxob, pyob);

disp("zadanie 4: " + filename);
disp("sol3a"); disp(sol3a);
disp("sol3b"); disp(sol3b);

function [] = addToPlot(subplotIdx, tp, vp, ve, ...
    subplotLegend, subplotTitle)
    subplot(subplotIdx(1), subplotIdx(2), subplotIdx(3));
    plot(tp, vp, tp, ve);
    adjustAxes(gca);
    legend(subplotLegend(1), subplotLegend(2), ...
        'Interpreter', 'latex', 'FontSize', 13);
    title(subplotTitle, 'Interpreter', 'latex');
end

function [] = adjustAxes(ax)
    ax.XGrid = "on";
    ax.XMinorGrid = "on";
    ax.YGrid = "on";
    ax.YMinorGrid = "on";
end
end

```

Listing 13: Funkcja odpowiedzialna za tworzenie wykresów w skrypcie głównym programu

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Modelowanie Matematyczne 2023/2024 - semestr zimowy
% Projekt 2: Estymacja parametrow modelu Lotki-Volterra
% Autor: Adam Gracikowski, 327350
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% symulacja dla pliku: dane15.csv

filename = "dane15.csv";
dataTable = readtable(filename);
options = struct('xp', dataTable.x, ...
    'yp', dataTable.y, ...
    'tp', dataTable.t, ...
    'ft', @(t) t, ...
    'set1', [], ...
    'set2a', optimset('MaxFunEvals', 4000), ...
    'set2b', [], ...
    'set3a', optimset('MaxIter', 4000, 'MaxFunEvals', 5000), ...
    'set3b', optimset('MaxIter', 3000, 'MaxFunEvals', 3000), ...
    'h', 0.01);

symulation(filename, options);

% symulacja dla pliku: HudsonBay.csv

filename = "HudsonBay.csv";
dataTable = readtable(filename);
dataTable.Properties.VariableNames = {'t', 'x', 'y'};
options = struct('xp', dataTable.x, ...
    'yp', dataTable.y, ...
    'tp', dataTable.t, ...
    'ft', @(t) rescale(t, 0, 3), ...
    'set1', [], ...
    'set2a', [], ...
    'set2b', optimset('MaxIter', 4000, 'MaxFunEvals', 4000), ...
    'set3a', optimset('MaxIter', 4000, 'MaxFunEvals', 4000), ...
    'set3b', optimset('MaxIter', 4000, 'MaxFunEvals', 4000), ...
    'h', 0.01);

symulation(filename, options);

```

Listing 14: Główny skrypt projektu

## Źródła

- [1] [https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra\\_equations](https://en.wikipedia.org/wiki/Lotka%E2%80%93Volterra_equations)
- [2] [https://en.wikipedia.org/wiki/Ordinary\\_differential\\_equation](https://en.wikipedia.org/wiki/Ordinary_differential_equation)
- [3] [https://en.wikipedia.org/wiki/Linear\\_multistep\\_method](https://en.wikipedia.org/wiki/Linear_multistep_method)
- [4] [https://en.wikipedia.org/wiki/Euler\\_method](https://en.wikipedia.org/wiki/Euler_method)
- [5] <https://www.mathworks.com/help/matlab/ref/fminsearch.html>
- [6] <https://www.mathworks.com/help/matlab/ref/interp1.html>
- [7] <https://www.mathworks.com/help/matlab/ref/ode45.html>
- [8] <https://www.animalsaroundtheglobe.com/how-the-hare-evades-the-lynx/>