Igor Popławski

Rozwiązywanie równań różniczkowych

Metody numeryczne (MNUB), semestr 24L

Prowadzący: Paweł Mazurek

Politechnika Warszawska, Wydział Elektroniki i Technik Informacyjnych

Warszawa 23.05.2024r.

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

Spis treści

1.Lista symboli matematycznych i akronimów

x,y,z – zmienne skalarne

x,y,z – wektory zmiennych skalarnych

$\mathbf{Z}$, $\mathbf{X}$, $\mathbf{Y}$ – macierze zmiennych sklarnych

2.Wprowadzenie

Projekt polega na rozwiązywaniu układu równań Lotka-Volterry postaci:

$$\frac{dx(t)}{dt} = p_1 \cdot x(t) - p_2 \cdot x(t) \cdot y(t)$$
$$\frac{dy(t)}{dt} = p_3 \cdot x(t) \cdot y(t) - p_4 \cdot x(t) \quad (1)$$

Gdzie:

p_1, p_2, p_3, p_4 - odpowiednie współczynniki

$x(t)$ – funkcja populacji ofiar

$y(t)$ - funkcja populacji drapieżników

Rozwiązywania dokonujemy za pomocą dwóch metod rozwiązywania równań różniczkowych otwartych (otwarta metoda Eulera, metoda Heuna) oraz dwóch metod zamkniętych (zamknięta metoda Eulera, metoda Adamsa-Moultona). Naszym celem jest również zbadanie średnich zagregowanych błędów rozwiązań dla wybranych metod w zależności od kroku całkowania.

Oprócz tego mamy zamiar wyznaczyć wartość współczynników p_1, p_2, p_3, p_4 dla których zagregowane średnie błędy względne są najmniejsze dla metody używanej w funkcji „ode45" używanej w programie MATLAB.

Metody użyte w zadaniu projektowym:

Otwarta metoda Eulera

Metoda ta polega na wyznaczeniu wartości następnego punktu oddalonego o dany krok całkowania na podstawie wartości poprzedniego punktu. Wzór którym posługujemy się wyznaczając następne punkty ma postać:

$$y_n = y_{n-1} + h_{n-1} \cdot f(t_{n-1}, y_{n-1}) \quad (2)$$

Gdzie:

y_n – następna wartość którą wyznaczamy

y_{n-1} – poprzednia wartość którą wyznaczyliśmy

h_{n-1} –krok całkowania użyty przy wyznaczaniu poprzedniego punktu

$f()$ – wartość funkcji którą rozpatrujemy dla danych argumentów

Zamknięta metoda Eulera

Metoda ta polega na wyznaczeniu kolejnych wartości punktów za pomocą wartości poprzedniej oraz wartości punktu który próbujemy wyznaczyć. Metoda ta wymaga więc rozwiązania równania mającego postać:

$$y_n = y_{n-1} + h \cdot f(t_n, y_n) \quad (3)$$

Gdzie:

y_n – następna wartość którą wyznaczamy

y_{n-1} – poprzednia wartość którą wyznaczyliśmy

h – krok całkowania

$f()$ – wartość funkcji którą rozpatrujemy dla danych argumentów

Metoda Heuna

Otwarta metoda Heuna polega na obliczeniu następnych wartości rozwiązań oddalonych o krok całkowania h za pomocą wzoru:

$$y_n = y_{n-1} + \frac{1}{2} \cdot h \cdot [f(t_{n-1}, y_{n-1}) + f\big(t_{n-1} + h, y_{n-1} + h \cdot f(t_{n-1}, y_{n-1})\big)] \quad (4)$$

Gdzie:

y_n – następna wartość którą wyznaczamy

y_{n-1} – poprzednia wartość którą wyznaczyliśmy

h – krok całkowania

$f()$ – wartość funkcji którą rozpatrujemy dla danych argumentów

Metoda Adamsa-Moultona

Metoda Adamsa Moultona polega na wyznaczeniu obecnego rozwiązania na podstawie rozwiązania poprzedniego oraz obecnego za pomocą wzoru:

$$y_n = y_{n-1} + h \cdot \sum_{k=0}^{K} \beta_k \cdot f(t_{n-k}, y_{n-k}) \quad (5)$$

Gdzie:

y_n – następna wartość którą wyznaczamy

y_{n-1} – poprzednia wartość którą wyznaczyliśmy

h – krok całkowania

$f()$ – wartość funkcji którą rozpatrujemy dla danych argumentów

β_k- określone wartości współczynników dla danego K

3.Metodyka i wyniki doświadczeń

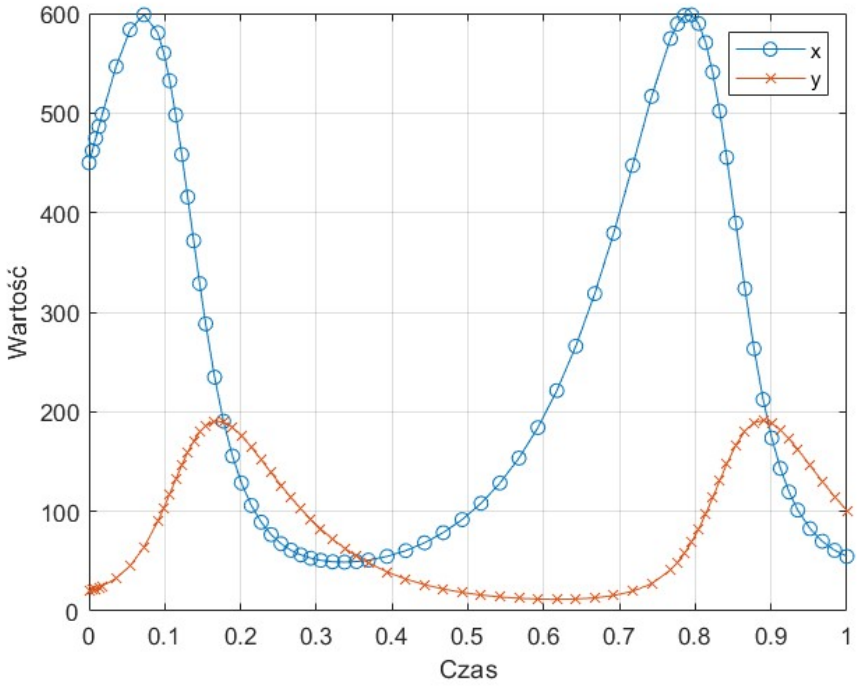
Zad.1

W zadaniu pierwszym rozwiązujemy układ równań (1) za pomocą wbudowanej w MATLAB funkcji „ode45”. Wartości współczynników p użytych w układzie równań przedstawiono w Tabeli 1.

Tabela 1. Wartości współczynników p użytych w równaniu Lotka-Volterry

Współczynnik	p_1	p_2	p_3	p_4
Wartość	9	0.14	0.05	11

Układ rozwiązujemy w przedziale czasu od 0 do 1 i przyjmujemy wartości początkowe jako x(0)=450 oraz y(0)=20. Kolejne rozwiązania zapisujemy w wektorze **xy** a punkty w czasie dla których w których wyznaczaliśmy rozwiązania w wektorze **t**. By wyznaczyć przebieg funkcji wykreślamy zależności naszych wektorów za pomocą funkcji „plot”. Rozwiązania równań w przedziale czasu przedstawiono na Rys.1.



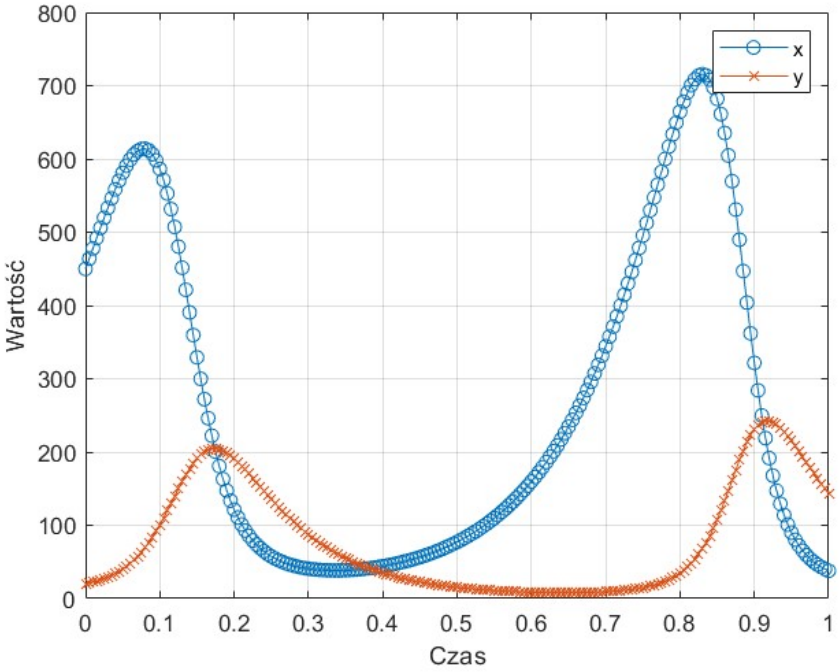
Rys.1 Rozwiązania układu równań Lotki-Volterry w czasie za pomocą funkcji ode45

Zad.2

W zadaniu drugim rozwiązujemy układ równań za pomocą otwartej metody Eulera, zamkniętej metody Eulera, metody Heuna oraz metody Adamsa-Moultona rzędu 3. Dla wszystkich metod używamy kroku całkowania h=0.005. Tworzymy więc wektor czasu z krokiem h w przedziale od 0 do 1. Jako wartości początkowe ponownie przyjmujemy x(0)=450 oraz y(0)=20.

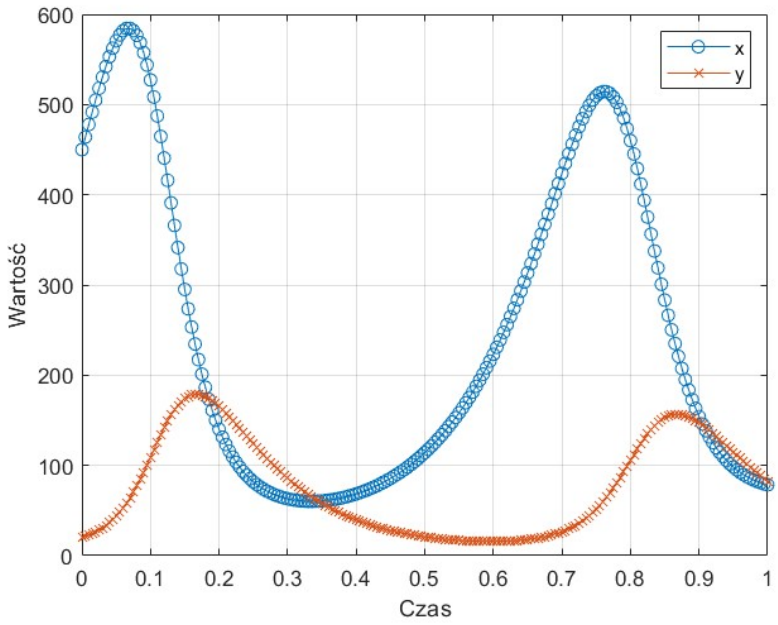
By użyć otwartej metody Eulera w programie MATLAB stosujemy pętle w której dla każdej chwili czasu obliczamy wartość *x* i *y* na podstawie wzoru (2). Wartości te zapisujemy do wektora **xy2** by

brać w każdym następnym kroku poprzednie rozwiązania. Otrzymane funkcje przedstawiono na Rys.2.



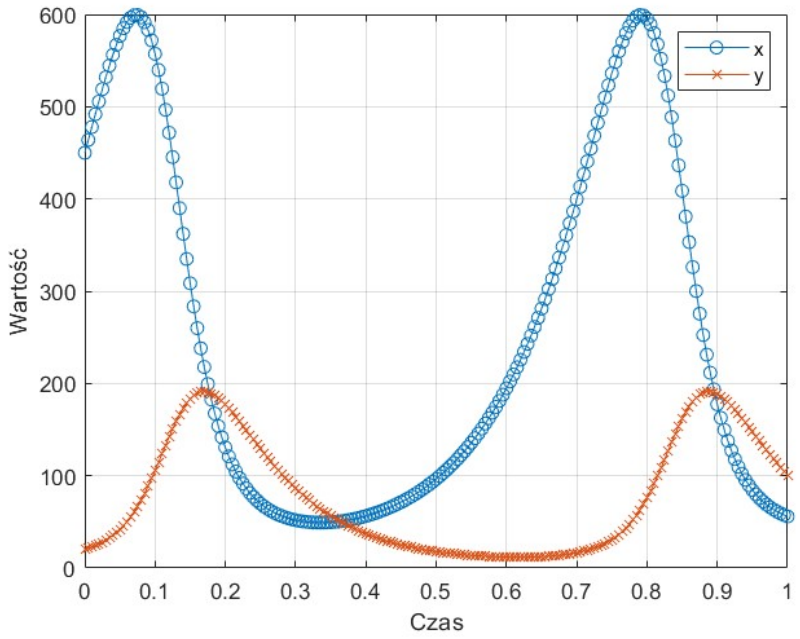
Rys.2 Rozwiązania układu równań Lotki-Volterry w czasie za pomocą otwartej metody Eulera

By użyć zamkniętej metody Eulera w programie MATLAB stosujemy pętle w której dla każdej chwili czasu obliczamy wartość *x* i *y* poprzez użycie funkcji fsolve do rozwiązania równania (3). Wartości te zapisujemy do wektora **xy3** by brać w każdym następnym kroku poprzednie rozwiązania. Otrzymane funkcje przedstawiono na Rys.3.



Rys.3 Rozwiązania układu równań Lotki-Volterry w czasie za pomocą zamkniętej metody Eulera

By użyć otwartej metody Heuna w programie MATLAB stosujemy pętle w której dla każdej chwili czasu obliczamy wartość *x* i *y* na podstawie wzoru (4). Wartości te zapisujemy do wektora **xy4** by brać w każdym następnym kroku poprzednie rozwiązania. Otrzymane funkcje przedstawiono na Rys.4.



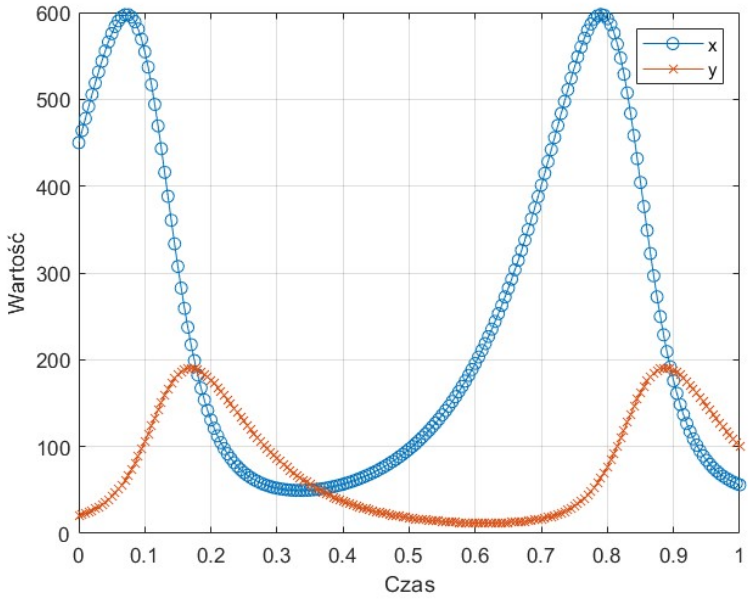
Rys.4 Rozwiązania układu równań Lotki-Volterry w czasie za pomocą metody Heuna

By użyć zamkniętej metody Adamsa-Moultona w programie MATLAB stosujemy pętle w której dla każdej chwili czasu obliczamy wartość *x* i *y* poprzez użycie funkcji „fsolve" do rozwiązania równania (5). Jako współczynniki β_k dla K=2 podstawiamy wartości zgodnie z Tabelą 2.

Tabela 2. Wartości współczynników β_k dal K=2

K	β_0	β_1	β_2
2	$\frac{5}{12}$	$\frac{8}{12}$	$\frac{-1}{12}$

Wartości rozwiązań zapisujemy do wektora **xy5** by brać w każdym następnym kroku poprzednie rozwiązania. Otrzymane funkcje przedstawiono na Rys.5.



Rys.5 Rozwiązania układu równań Lotki-Volterry w czasie za pomocą metody zamkniętej metody Adamsa-Moultona

Zad.3

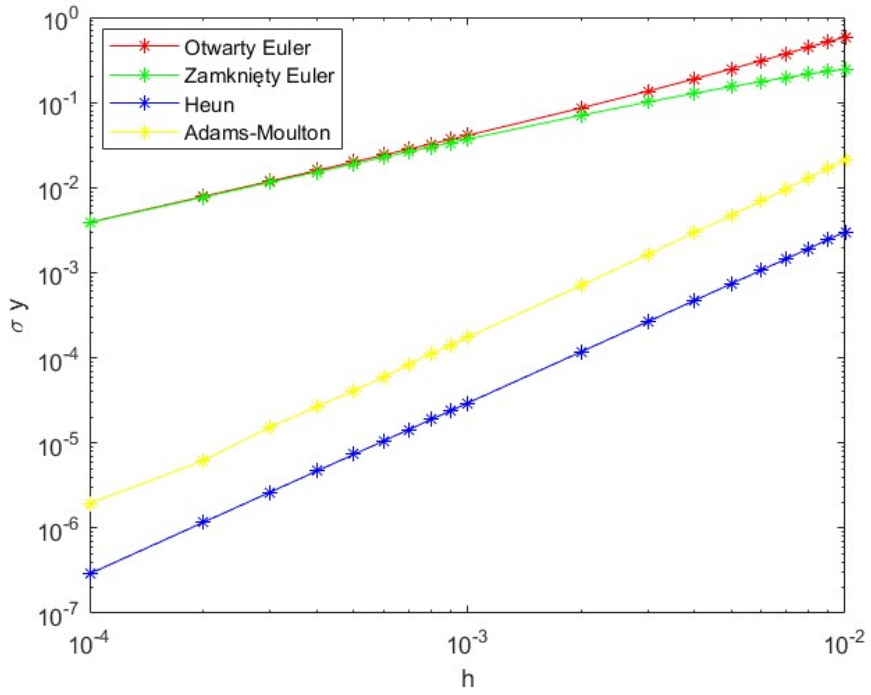
W zadaniu trzecim wyznaczamy zagregowany błąd względny dla rozwiązań zmiennej *y* otrzymany z każdej z metod. By to zrobić wyznaczamy wektor rozwiązań referencyjnych przy pomocy funkcji „ode45". Robimy to poprzez użycie funkcji „norm" na wektorze różnic rozwiązań referencyjnych i rozwiązań otrzymanych daną metodą. Następnie dzielimy otrzymaną wartość na wartość uzyskaną z użycia funkcji „norm" na wektorze rozwiązań otrzymanych daną metodą. Wykonujemy te czynności dla każdej z metod. Otrzymujemy więc cztery wartości przedstawione w Tabeli 3.

Tabela 3. Zagregowany błąd względny δ_y rozwiązań uzyskanych przy użyciu różnych metod

Metoda	Otwarta Eulera	Zamknięta Eulera	Heuna	Adamsa-Moultona 3 rzędu
Wartość zagregowanego błędu względnego δ_y	0.2480	0.1532	0.0007	0.0047

Zad.4

W zadaniu czwartym musimy policzyć zagregowany błąd względny dla różnych kroków całkowania w przedziale $[10^{-4},10^{-2}]$. Wybierzemy więc punkty postaci: 10^{-4} , $2 \cdot 10^{-4}$, $3 \cdot 10^{-4}$, … , 10^{-3} , $2 \cdot 10^{-3}$, $3 \cdot 10^{-3}$, … , 10^{-2}. Dla każdego z tych kroków tworzymy nowy wektor rozwiązań referencyjnych przy pomocy funkcji „ode45", a następnie obliczamy zagregowane błędy względne w ten sam sposób jak w zadaniu 3. Wartości obliczonych błędów wpisujemy do poszczególnych wektorów. Następnie wykreślamy zależności wektorów błędów od wartości kroku całkowania za pomocą funkcji „loglog".



Rys.6 Zależność wartości zagregowanego błędu względnego δ_y od długości kroku całkowania h

Zad.5

W zadaniu piątym naszym zadaniem jest znalezienie parametrów p dla których wartość opisana równaniem (6) jest najmniejsza. Kryterium przedstawiono równaniem poniżej:

$$J(p) = \sum_{n=1}^{N} (\hat{x}_n(p) - \tilde{x}_n)^2 + \sum_{n=1}^{N} (\hat{y}_n(p) - \tilde{y}_n)^2 \tag{6}$$

Gdzie:
$\hat{x}(p), \hat{y}(p)$ – estymaty x and y w chwilach określonych w pliku MNUB_24L_P3_dane32.csv, wyznaczone w wyniku rozwiązania układu równań (1) za pomocą funkcji ode45
$\tilde{x}_n, \tilde{y}_n$ – dane znajdujące się w wektorach **x** i **y**, w pliku MNUB_24L_P3_dane32.csv
N – ilość punktów w wektorze **t** z pliku

Wykonujemy to za pomocą funkcji „fminsearch" w której jako argumenty podajemy początkowe parametry p umieszczone w wektorze **p_0** oraz funkcje która dla każdych p oblicza estymatę funkcją „ode45" i zwraca wartość kryterium $J(p)$ wyznaczaną na podstawie otrzymanej estymaty i wartości z wektorów z pliku.
Wartości wyliczone za pomocą funkcji „fminsearch" wpisujemy do wektora **p_opt**.
Wartości wektora **p_opt** przedstawiono w Tabeli 4.

Tabela 2. Wartości współczynników p dla których warunek opisany równaniem (6) ma najmniejsza wartość

p_1	p_2	p_3	p_4
12.9167	0.1253	0.0592	15.6909

4.Dyskusja wyników eksperymentów numerycznych

Zad.1

Za pomocą funkcji „ode45" mogliśmy skutecznie rozwiązać układ równań Lotka-Volterry. Dla naszych współczynników p widzimy ze populacja ofiar (zmienna *x*) osiąga maksimum przy wartości 600 i minimum przy wartości 50. Populacja drapieżników za to osiąga maksimum dla około 190 i minimum dla około 11. Warto dodać że domyślne użycie „ode45" dokonuje aproksymacji funkcji na podstawie 22 punktów w czasie.

Zad.2

W zadaniu drugim jako że przyjęliśmy krok całkowania h=0.005 wszystkie nasze wykresy będą miały 200 punktów na podstawie których będą one wykreślane.

Rozwiązując nasz układ równań otwartą metodą Eulera możemy zaobserwować, że wyznaczone wartości *x* i *y* w punktach w czasie są zawyżone. Na początku jest to niewielka różnica jednak z każdą iteracją algorytmu narasta i dla chwili w czasie t=0.825 dla populacji ofiar różnica jest rzędu setek.

Rozwiązując układ równań zamkniętą metodą Eulera ponownie widzimy odstępstwo od naszego wykresu wzorcowego. Tym razem wartości są zaniżone ale ponownie jak dla poprzedniego przypadku błędy narastają wraz z czasem.

Rozwiązania otrzymane przy użyciu metody Heuna oraz metody Adamsa-Moultona 3 rzędu są jednak bardzo dokładne, błędy są na tyle małe że na podstawie otrzymanych wykresów nie jesteśmy w stanie ich zauważyć bo wykresy są niemal identyczne.

Zad.3

Wyznaczając zagregowane błędy dla naszych rozwiązań wartości *y* widzimy, że otwarta metoda Eulera obarczona jest największym błędem, zamknięta metoda Eulera także wykazuje spory błąd natomiast metoda Adamsa-Moultona ma średni błąd na poziomie części tysięcznych. Najmniejszym zagregowanym błędem średnim obarczona jest metoda Heuna gdzie błąd jest rzędu czwartej liczby po przecinku.

Zad.4

Dla każdej z metod widzimy mocną zależność – wraz ze wzrostem kroku całkowania wartość błędu rośnie w sposób potęgowy w stosunku do wzrostu kroku całkowania. Dla każdej wartości kroku metoda Heuna obarczona jest najmniejszym błędem. Błąd metody Adamsa-Moultona ma trochę większe wartości. Największe błędy dla każdego kroku całkowania wykazują metody Eulera. Ich błędy są bardzo zbliżone gdy krok całkowania jest mniejszy od 10^{-3}, po przekroczeniu tej wartości błąd otwartej metody Eulera nadal rośnie potęgowo jednak błąd metody zamkniętej zaczyna narastać wolniej – wykres traci charakter liniowy.

Zad.5

Nasze otrzymane wartości parametrów *p* dla których rozwiązania równań są obarczone najmniejszymi błędami, różnią się od wartości początkowych. Największe różnice widać dla współczynnika p_1 i p_4.

5.Wnioski ogólne

Rozwiązując równania różniczkowe ważne jest zastosowanie odpowiedniej metody. Funkcja ode45 wbudowana w program MATLAB dobrze radzi sobie z rozwiązywanie równań różniczkowych jednak także nie jest bezbłędna i jej dokładność może zależeć od współczynników użytych w równaniu. Metody Eulera są dość uproszczone przez co wykazują większe błędy, które dodatkowo narastają wraz z każdą iteracją. Bardziej zaawansowane metody takie jak metoda Heuna oraz metoda Adamsa-Moultona 3 rzędu bardzo dobrze radzą sobie z rozwiązywaniem równań różniczkowych i wykazują małe błędy. Używając większego rzędu dla metody Adamsa-Moultona moglibyśmy jeszcze dokładniej przybliżyć przebieg naszych funkcji.

6.Lista źródeł

[1] Paweł Mazurek – MNUB 24L, wykłady Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska

7.Listing programów

- plik Zad1_2_3

```
p1=9;
p2=0.14;
p3=0.05;
p4=11;
tspan=[0 1];
xy0(1)=450;
xy0(2)=20;

%Zad1
[t, xy]=ode45(@(t, xy)
LotVol(t,xy,p1,p2,p3,p4), tspan, xy0);

figure(1);
plot(t, xy(:,1), '-o', 'DisplayName',
'x');
grid on;
hold on;
plot(t, xy(:,2), '-x', 'DisplayName',
'y');
legend;
xlabel('Czas');
ylabel('Wartość');

%Zad2
h=0.005;
t2 = 0:h:1;
xy2 = zeros(2, length(t2));
xy2(:,1) = xy0;
xy2=openeul(t2,xy2,p1,p2,p3,p4,h);
xy3=closeeul(t2,xy2,p1,p2,p3,p4,h);
xy4=Heun(t2,xy2,p1,p2,p3,p4,h);
xy5=AdamsMoul(t2,xy2,p1,p2,p3,p4,h);

figure(2);
plot(t2, xy2(1,:), '-o',
'DisplayName', 'x');
grid on;
hold on;
plot(t2, xy2(2,:), '-x',
'DisplayName', 'y');
legend;
xlabel('Czas');
ylabel('Wartość');

figure(3);
plot(t2, xy3(1,:), '-o',
'DisplayName', 'x');
grid on;
hold on;
plot(t2, xy3(2,:), '-x',
'DisplayName', 'y');
grid on;
legend;
xlabel('Czas');
ylabel('Wartość');

figure(4);
plot(t2, xy4(1,:), '-o',
'DisplayName', 'x');
grid on;
hold on;
plot(t2, xy4(2,:), '-x',
'DisplayName', 'y');
legend;
xlabel('Czas');
ylabel('Wartość');
hold off;

figure(5);
plot(t2, xy5(1,:), '-o',
'DisplayName', 'x');
grid on;
hold on;
plot(t2, xy5(2,:), '-x',
'DisplayName', 'y');
legend;
xlabel('Czas');
ylabel('Wartość');
hold off;

%Zad3
[t, xyref]=ode45(@(t, xyref)
LotVol(t,xyref,p1,p2,p3,p4), t2, xy0,
odeset('RelTol', 1e-8, 'AbsTol', 1e-
12));
sigmaOpenEuly=norm(xyref(:,2)-
xy2(2,:)')/norm(xyref(:,2));
sigmaCloseEuly=norm(xyref(:,2)-
xy3(2,:)')/norm(xyref(:,2));
sigmaHeuny=norm(xyref(:,2)-
xy4(2,:)')/norm(xyref(:,2));
sigmaAMy=norm(xyref(:,2)-
xy5(2,:)')/norm(xyref(:,2));

%Zad4
xy2z4 = zeros(2, length(t2));
xy2z4(:,1) = xy0;
allsigmaOpenEuly=zeros(19,1);
allsigmaCloseEuly=zeros(19,1);
allsigmaHeuny=zeros(19,1);
allsigmaAMy=zeros(19,1);
hv = zeros(19,1);

figure(6);
grid on;
xlabel('k');
ylabel('\sigma y');
```

```
labels = {'Open Euler', 'Closed
Euler', 'Heun', 'Adams-Moulton'};

for k=1:19
    if k<=10
        h=10^(-4)*k;
    else
        h=10^(-3)*(k-9);
    end
    hv(k)=h;
    t2 = 0:h:1;
    xy2z4 = zeros(2, length(t2));
    xy2z4(:,1) = xy0;
    xy3z4=xy2z4;
    xy4z4=xy2z4;
    xy5z4=xy2z4;

    [t, xyref]=ode45(@(t, xyref)
LotVol(t,xyref,p1,p2,p3,p4), t2, xy0,
odeset('RelTol', 1e-8, 'AbsTol', 1e-
12));

xy2z4=openeul(t2,xy2z4,p1,p2,p3,p4,h);

xy3z4=closeeul(t2,xy2z4,p1,p2,p3,p4,h)
;

xy4z4=Heun(t2,xy2z4,p1,p2,p3,p4,h);

xy5z4=AdamsMoul(t2,xy2z4,p1,p2,p3,p4,h
);

allsigmaOpenEuly(k)=norm(xyref(:,2)-
xy2z4(2,:)')/norm(xyref(:,2));

allsigmaCloseEuly(k)=norm(xyref(:,2)-
xy3z4(2,:)')/norm(xyref(:,2));
    allsigmaHeuny(k)=norm(xyref(:,2)-
xy4z4(2,:)')/norm(xyref(:,2));
    allsigmaAMy(k)=norm(xyref(:,2)-
xy5z4(2,:)')/norm(xyref(:,2));
end

loglog(hv, allsigmaOpenEuly, '-*r',
'DisplayName', 'Open Euler');
hold on;
loglog(hv, allsigmaCloseEuly, '-*g',
'DisplayName', 'Closed Euler');
loglog(hv, allsigmaHeuny, '-*b',
'DisplayName', 'Heun');
loglog(hv, allsigmaAMy, '-*y',
'DisplayName', 'Adams-Moulton');
legend(labels);
hold off;


%funkcje

%otwarta metoda Eulera

function
dxydt=openeul(t2,xy2,p1,p2,p3,p4,h)

    for i=1:(length(t2)-1)
        xy2(:,i+1)=xy2(:,i) +
h*LotVol(t2(i), xy2(:,i), p1, p2, p3,
p4);
    end

    dxydt=xy2;
end

%zamknieta metoda Eulera
function
dxydt=closeeul(t2,xy3,p1,p2,p3,p4,h)

    for i=1:(length(t2)-1)

        fun=@(xynext)[
            xynext - xy3(:,i) - h *
LotVol(t2(i+1), xynext, p1,p2,p3,p4);
        ];

xy3(:,i+1)=fsolve(fun,xy3(:,i),optimop
tions('fsolve', 'Display', 'none'));

    end

    dxydt=xy3;
end

%otwarta metoda Heuna
function
dxydt=Heun(t2,xy4,p1,p2,p3,p4,h)

    for i=1:(length(t2)-1)
        xy4(:,i+1)=xy4(:,i) +
0.5*h*(LotVol(t2(i), xy4(:,i), p1, p2,
p3, p4) + LotVol(t2(i) + h, xy4(:,i) +
h*LotVol(t2(i), xy4(:,i), p1, p2, p3,
p4), p1, p2, p3, p4));
    end

    dxydt=xy4;
end

%zamknieta metoda Adamsa-Moultona
function
dxydt=AdamsMoul(t2,xy5,p1,p2,p3,p4,h)

    B=[5/12, 8/12, (-1/12)];

    for i=1:(length(t2)-1)
        if i<3
            fun=@(xynext)[
            xynext - xy5(:,i) - h *
LotVol(t2(i+1), xynext, p1,p2,p3,p4);
        ];
```

```
xy5(:,i+1)=fsolve(fun,xy5(:,i),optimop
tions('fsolve', 'Display', 'none'));
        else
            AM=@(xynext)[
                xynext - xy5(:,i) -
h*B(1)*LotVol(t2(i+1), xynext, p1, p2,
p3, p4) - h*B(2)*LotVol(t2(i),
xy5(:,i), p1, p2, p3, p4) -
h*B(3)*LotVol(t2(i-1), xy5(:,i-1), p1,
p2, p3, p4);
                ];

xy5(:,i+1)=fsolve(AM,xy5(:,i),optimopt
ions('fsolve', 'Display', 'none'));
        end
    end

    dxydt=xy5;
end

%funkcje
function dxydt = LotVol(t, xy, p1, p2,
p3, p4)
    dxydt=zeros(2,1);
    dxydt(1) = p1*xy(1) -
p2*xy(1)*xy(2);
    dxydt(2) = p3*xy(1)*xy(2) -
p4*xy(2);
end
```

- plik Zad4

```
%Zad4
p_0 = [9, 0.14, 0.05, 11];

data =
readmatrix('MNUB_24L_P3_dane32.csv
');
tdata = data(:,1);
xdata = data(:,2);
ydata = data(:,3);
p_opt = fminsearch(@(p) FunJ(p,
tdata, xdata, ydata), p_0);

function J = FunJ(p_current,
tdata, xdata, ydata)
    p1 = p_current(1);
    p2 = p_current(2);
    p3 = p_current(3);
    p4 = p_current(4);

    xy0(1)=xdata(1);
    xy0(2)=ydata(1);

    % Rozwiązanie równania
różniczkowego
[t, xy] = ode45(@(t, xy) LotVol(t,
xy, p1, p2, p3, p4), tdata, xy0);

    J = sum((xdata - xy(:,1)).^2)
+ sum((ydata - xy(:,2)).^2);
end

%funkcje
function dxydt = LotVol(t, xy, p1,
p2, p3, p4)
    dxydt=zeros(2,1);
    dxydt(1) = p1*xy(1) -
p2*xy(1)*xy(2);
    dxydt(2) = p3*xy(1)*xy(2) -
p4*xy(2);
end
```