Ruch punktu jest opisany równaniami:

$$\frac{dx_1}{dt} = x_2 + x_1(0,7 - {x_1}^2 - {x_2}^2)$$

$$\frac{dx_2}{dt} = -x_1 + x_2(0,7 - {x_1}^2 - {x_2}^2)$$

Należy obliczyć przebieg trajektorii ruchu na przedziale [0, 15] dla warunków początkowych: x1(0) = 0, x2(0) = 10. Rozwiązanie proszę znaleźć korzystając z zaimplementowanych przez siebie w języku Matlaba (w formie funkcji) metod:

1. **Rungego–Kutty czwartego rzędu (RK4) ze stałym krokiem,**
2. **Wielokrokowej metody predyktor-korektor Adamsa czwartego rzędu ze stałym krokiem,**
3. **Rungego-Kutty czwartego rzędu (RK4) ze zmiennym krokiem zmienianym automatycznie przez algorytm wykorzystujący estymację błędu aproksymacji według zasady podwójnego kroku.**

1. **Metody Rungego-Kutty definiuje się następującym wzorem:**

$y_{n+1} = y_n + h \cdot \sum_{i=1}^{m} \omega_i \, k_i$

gdzie

$k_1 = f(x_n, y_n),$
$k_i = f(x_n + c_i h, y_n + h \cdot \sum_{j=1}^{i-1} a_{ij} k_j), \quad i = 2,3, \dots, m,$

przy czym

$\sum_{j=1}^{i=1} a_{ij} = c_i, i = 2,3, \dots, m.$

Aby wykonać jeden krok metody należy obliczyć wartości prawych stron podanych równań różniczkowych dokładnie m razy. W przypadku metody Rungego-Kutty czwartego rzędu ze stałym krokiem h w każdej iteracji wyliczamy rozwiązania dla obu równań posługując się następującymi wzorami:

$$k_{11} = f1(x_{1,i}, x_{2,i})$$
$$k_{12} = f2(x_{1,i}, x_{2,i})$$

$$k_{21} = f1(x_{1,i} + 0{,}5h \cdot k_{11}, x_{2,i} + 0{,}5h \cdot k_{12})$$
$$k_{22} = f2(x_{1,i} + 0{,}5h \cdot k_{11}, x_{2,i} + 0{,}5h \cdot k_{12})$$

$$k_{31} = f1(x_{1,i} + 0{,}5h \cdot k_{21}, x_{2,i} + 0{,}5h \cdot k_{22})$$
$$k_{32} = f2(x_{1,i} + 0{,}5h \cdot k_{21}, x_{2,i} + 0{,}5h \cdot k_{22})$$

$$k_{41} = f1(x_{1,i} + h \cdot k_{31}, x_{2,i} + h \cdot k_{32})$$
$$k_{42} = f2(x_{1,i} + h \cdot k_{31}, x_{2,i} + h \cdot k_{32})$$

Gdzie rozwiązaniem w danej iteracji jest:

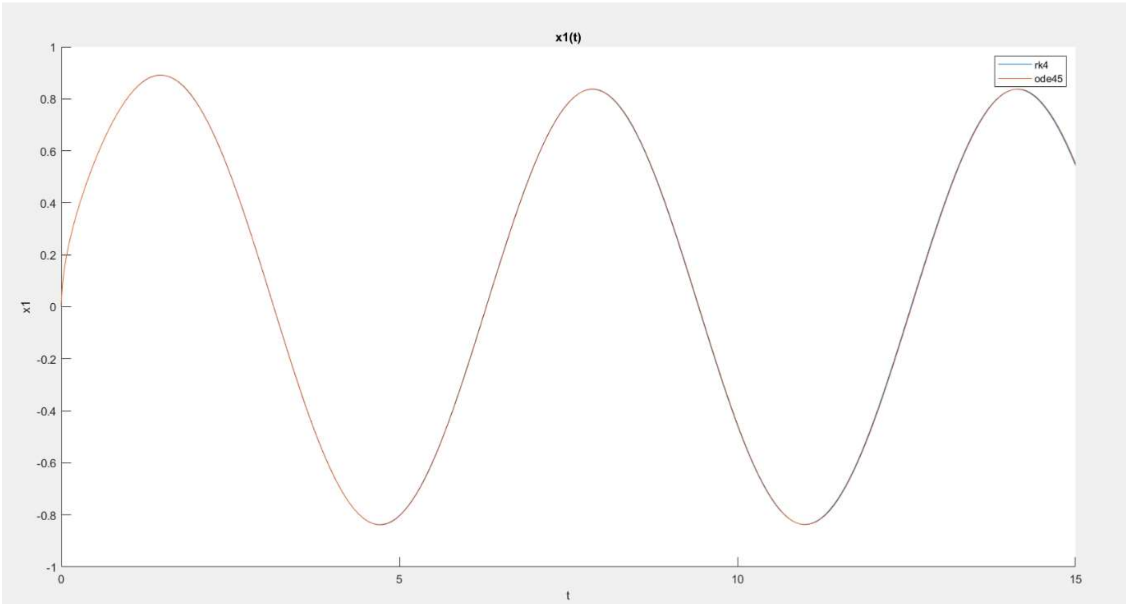
$$x_{1,i+1} = x_{1,i} + \frac{1}{6}(k_{11} + 2k_{21} + 2k_{31} + k_{41}) \cdot h$$
$$x_{2,i+1} = x_{2,i} + \frac{1}{6}(k_{12} + 2k_{22} + 2k_{32} + k_{42}) \cdot h$$

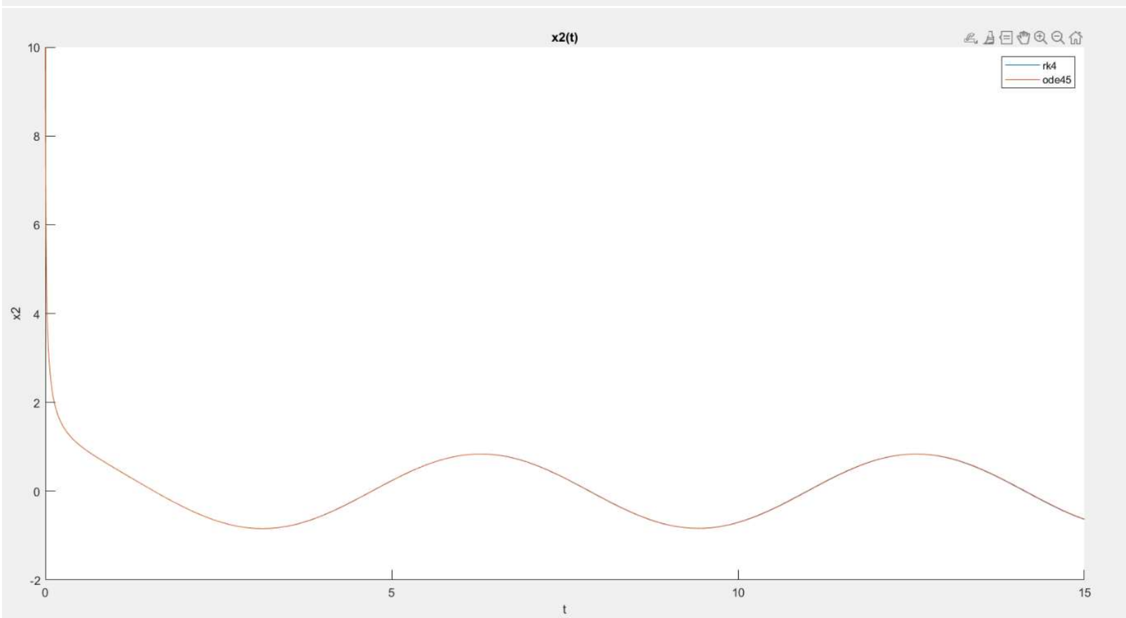
Zbiory punktów x_1 i x_2 są rozwiązaniami podanych równań różniczkowych, gdzie występuje dziedzina czasu na przedziale $[a, b]$ a wykonywanych jest $\frac{b-a}{h}$ iteracji. Przed rozpoczęciem iteracji punktami x_1 i x_2 są odpowiednio podane warunki początkowe tych równań, w naszym wypadku $x_1 = 0$, $x_2 = 10$.

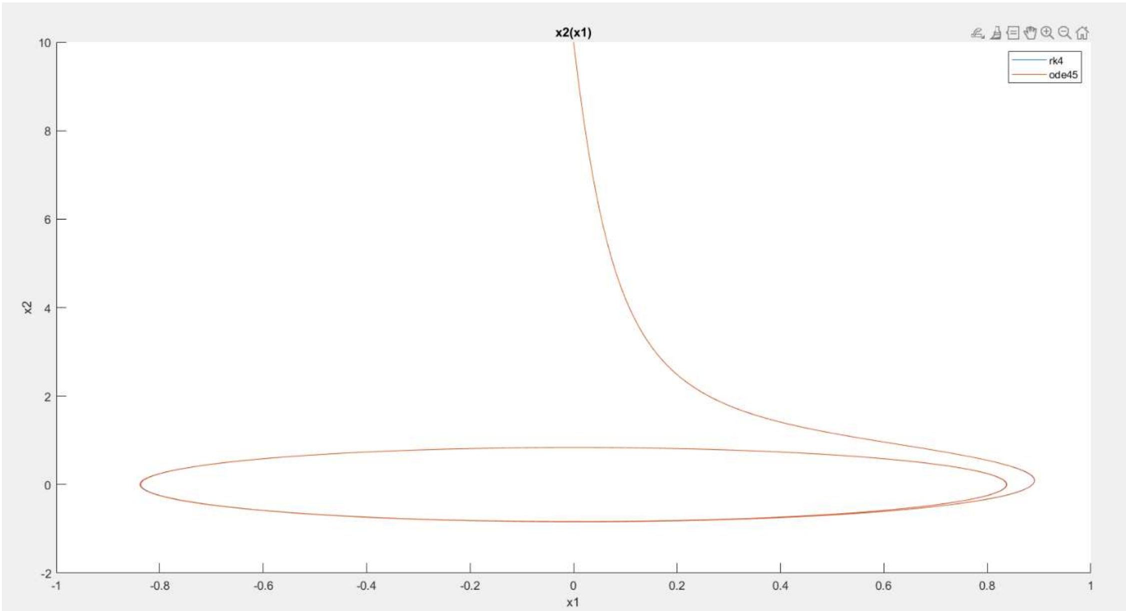
Funkcja wykorzystująca metodę Rungego-Kutty ze stałym krokiem:

```
% Deklaracja funkcji dla metody Rungego-Kutty ze stałym krokiem:
function [x1v,x2v] = rungekutta(f1,f2,x01,x02,a,b,h)
% Dziedzina czasu:
t = (a:h:b);
% Warunki początkowe:
x1 = x01;
x2 = x02;
% Wektory rozwiązań:
x1v = (x01);
x2v = (x02);
% Pętla główna:
for i=1:(length(t))

    % Wyznaczenie pierwszej pary współczynników:
    k11 = f1(x1, x2);
    k12 = f2(x1, x2);

    % Wyznaczenie drugiej pary współczynników:
    k21 = f1(x1+0.5*h*k11, x2 + 0.5*h*k12);
    k22 = f2(x1+0.5*h*k11, x2 + 0.5*h*k12);

    % Wyznaczenie trzeciej pary współczynników:
    k31 = f1(x1+0.5*h*k21, x2 + 0.5*h*k22);
    k32 = f2(x1+0.5*h*k21, x2 + 0.5*h*k22);

    % Wyznaczenie czwartej pary współczynników:
    k41 = f1(x1+h*k31, x2 + h*k32);
    k42 = f2(x1+h*k31, x2 + h*k32);

    % Wyznaczenie i-tego rozwiązania:
    x1 = x1 + (1/6)*(k11+2*k21+2*k31+k41)*h;
    x2 = x2 + (1/6)*(k12+2*k22+2*k32+k42)*h;

    % Dodanie rozwiązań do wektorów rozwiązań:
    x1v(i+1) = x1;
    x2v(i+1) = x2;
end
```

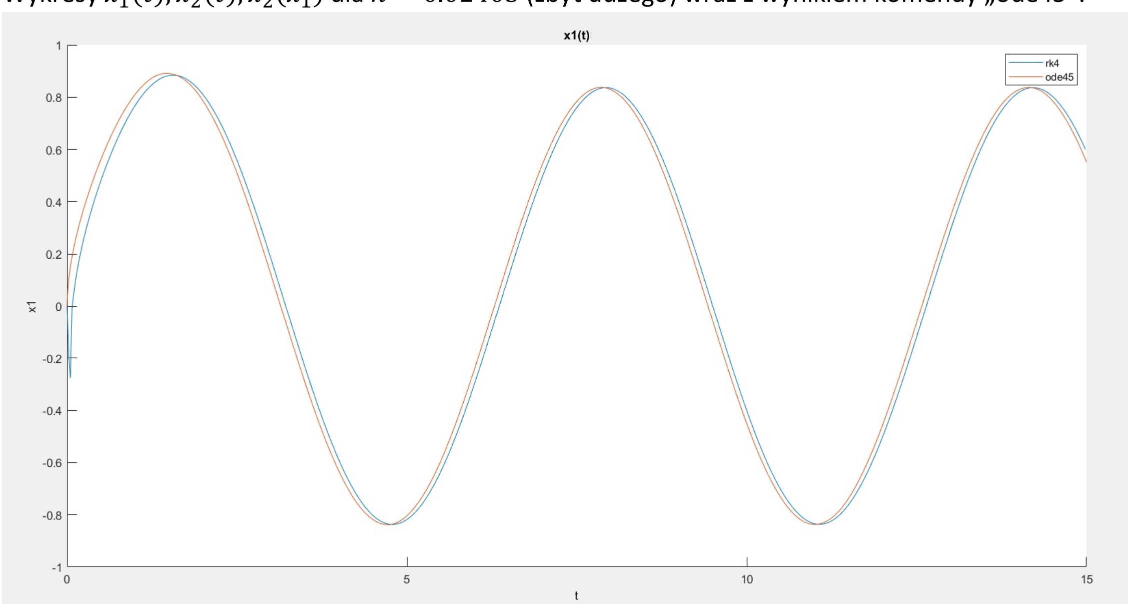
Wykresy $x_1(t), x_2(t), x_2(x_1)$ dla $h = 0.001$ (wybranego) wraz z wynikiem komendy „ode45”:

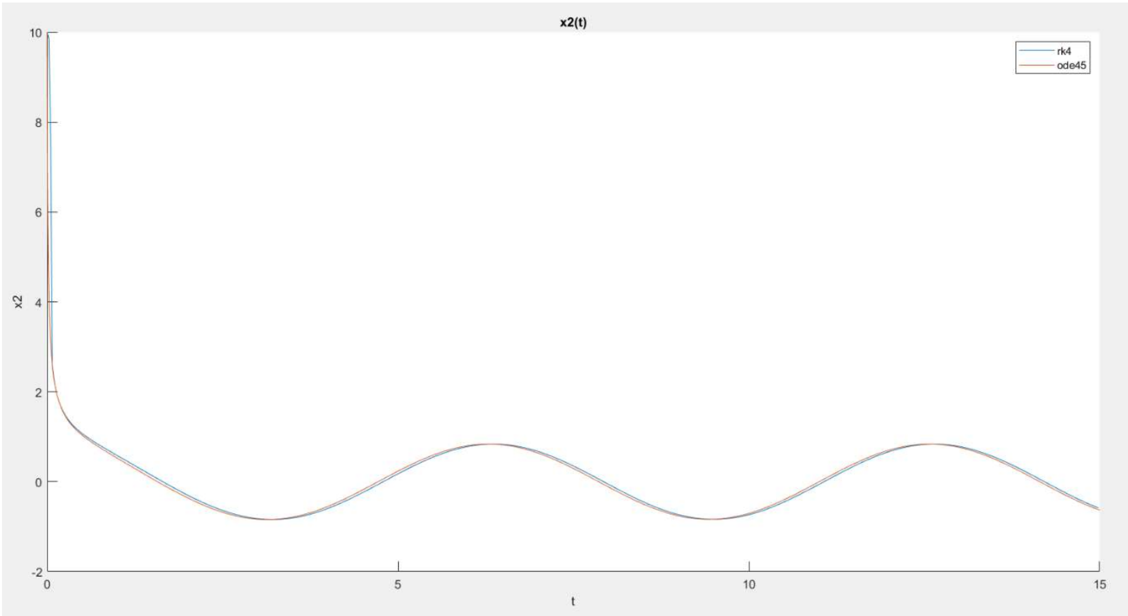

x1(t)
rk4
ode45
x1
t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15

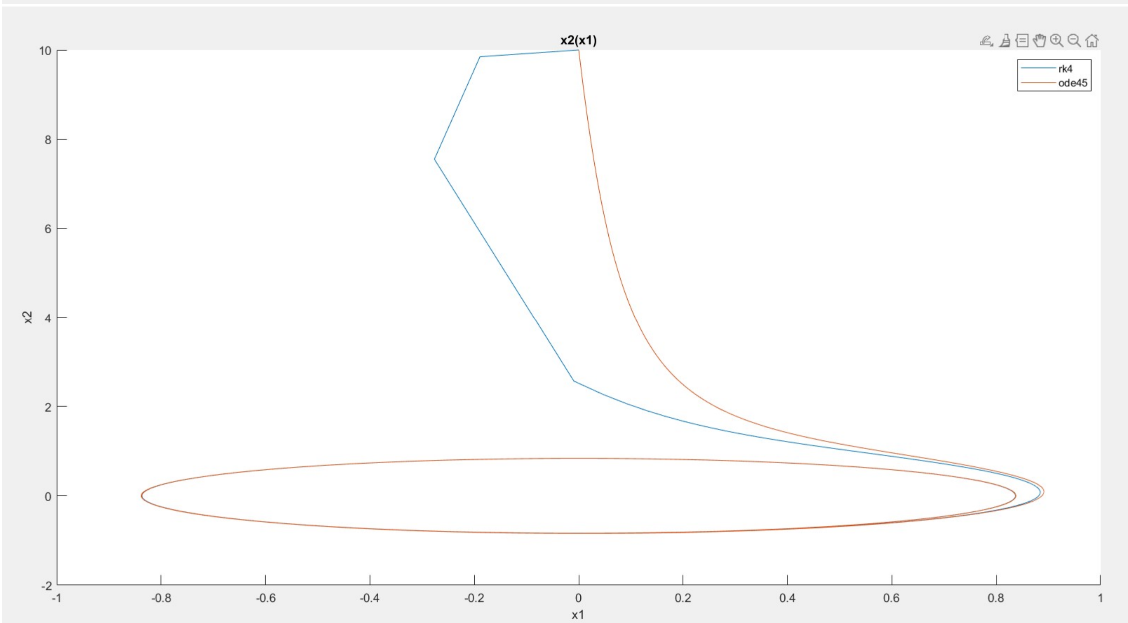

x2(t)
rk4
ode45
x2
t
10
8
6
4
2
0
-2
0
5
10
15



Wykresy $x_1(t), x_2(t), x_2(x_1)$ dla $h = 0.02405$ (zbyt dużego) wraz z wynikiem komendy „ode45”:







Wnioski do metody:

- przy doborze odpowiednio małego kroku metoda daje taki sam efekt jak metoda wbudowana „ode45”,

- im mniejszy krok tym mniejszy błąd metody a co za tym idzie dokładniejsze rozwiązanie,

- jednakże im mniejszy krok tym również większa liczba iteracji, a w rezultacie dłuższy czas wykonywania programu oraz kumulacja błędów numerycznych.

2. Metoda predyktor – korektor Adamsa czwartego rzędu jest metodą wielokrokową, a także stałokrokową. Algorytm predyktor – korektor jest połączeniem metody jawnej (Adamsa - Bashfortha) dla segmentu predyktora i metody niejawnej (Adamsa – Moultona)

dla segmentu korektora. Metoda ta wykonuje mniej obliczeń na iterację, ma większy obszar absolutnej stabilności oraz posiada małą stałą błędu. Struktura takiej metody ma postać:

P – predykcja: $\boldsymbol{y_n}^{[0]} = \sum_{j=1}^{k} \boldsymbol{\alpha_i y_{n-j}} + \boldsymbol{h} \cdot \sum_{j=1}^{k} \boldsymbol{\beta_j f_{n-j}}$

E - ewaluacja: $\boldsymbol{f_n}^{[0]} = \boldsymbol{f}\left(\boldsymbol{x_n}, \boldsymbol{y_n}^{[0]}\right)$

K - korekcja: $\boldsymbol{y_n}^{[0]} = \sum_{j=1}^{k} \boldsymbol{\alpha_i}^* \boldsymbol{y_{n-j}} + \boldsymbol{h} \cdot \sum_{j=1}^{k} \boldsymbol{\beta_j}^* \boldsymbol{f_{n-j}} + \boldsymbol{h\beta_0}^* \boldsymbol{f_n}^{[0]}$

E - ewaluacja: $\boldsymbol{f_n} = \boldsymbol{f}(\boldsymbol{x_n}, \boldsymbol{y_n})$

Aby rozwiązać zadany układ równań najpierw należy policzyć pierwsze cztery rozwiązania x_1, x_2 na przykład za pomocą metody Rungego – Kutty czwartego rzędu, gdzie oczywiście zaczynamy od wartości z warunków początkowych. Następne rozwiązania wyliczamy używając wzorów:

$$p_1 = x_{1,i} + \frac{h}{24}(55f1(x_{1,i}, x_{2,i}) - 59f1(x_{1,i-1}, x_{2,i-1}) + 37f1(x_{1,i-2}, x_{2,i-2}) - 9f1(x_{1,i-3}, x_{2,i-3}))$$

$$p_2 = x_{2,i} + \frac{h}{24}(55f1(x_{1,i}, x_{2,i}) - 59f1(x_{1,i-1}, x_{2,i-1}) + 37f1(x_{1,i-2}, x_{2,i-2}) - 9f1(x_{1,i-3}, x_{2,i-3}))$$

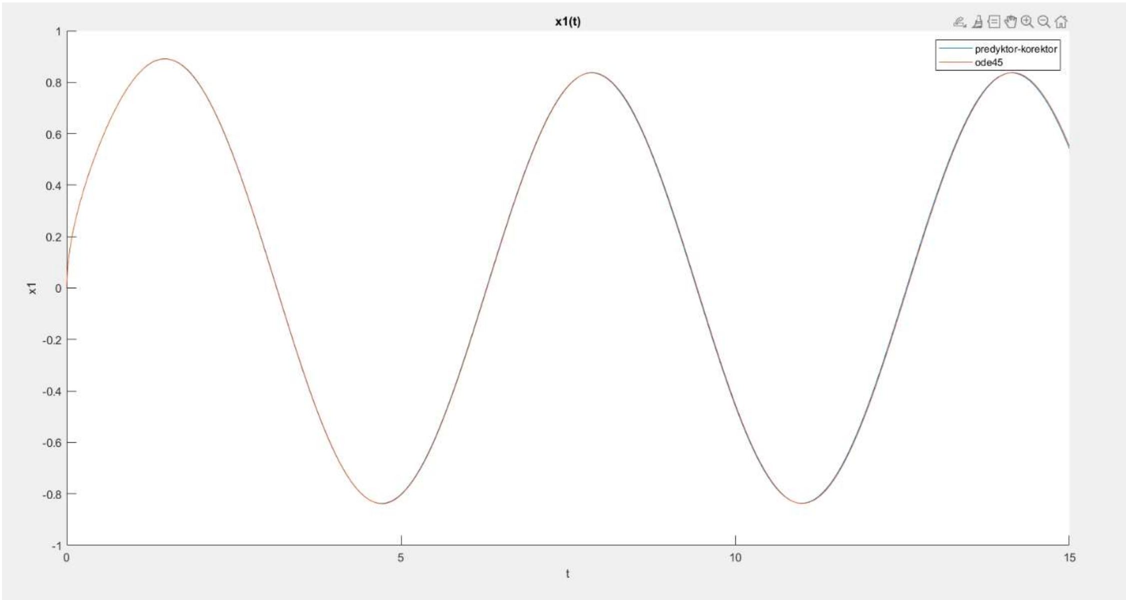
$$\begin{aligned} x_{1,i+1} = x_{1,i} + \frac{h}{720}(&646f1(x_{1,i}, x_{2,i}) - 264f1(x_{1,i-1}, x_{2,i-1}) + 106f1(x_{1,i-2}, x_{2,i-2}) \\ &- 19f1(x_{1,i-3}, x_{2,i-3}) + 251f1(p_1, p_2)) \end{aligned}$$

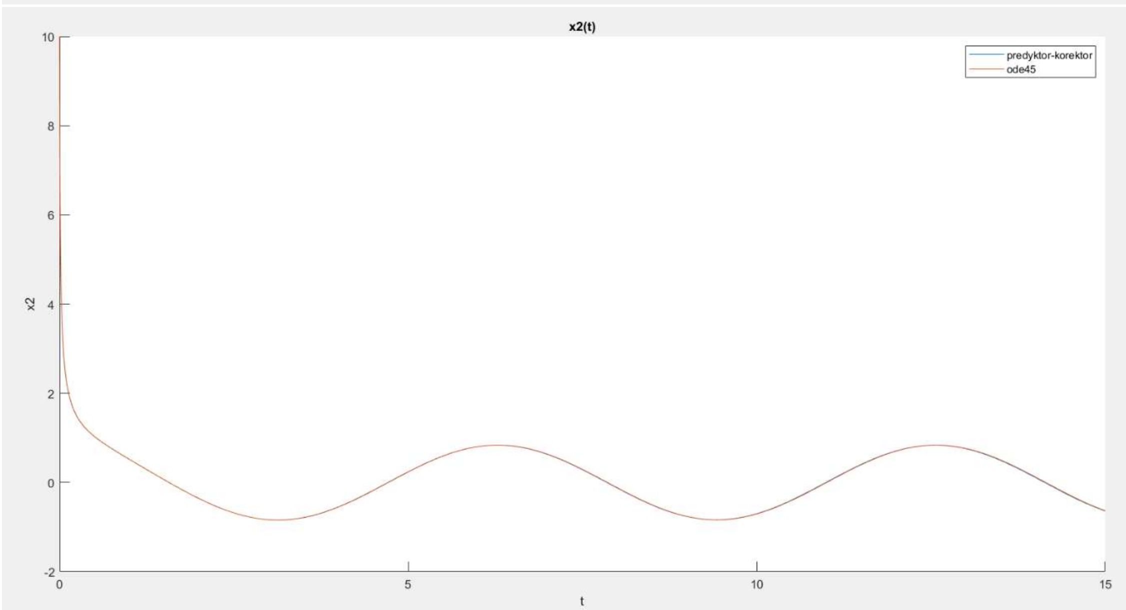
$$\begin{aligned} x_{2,i+1} = x_{2,i} + \frac{h}{720}(&646f1(x_{1,i}, x_{2,i}) - 264f1(x_{1,i-1}, x_{2,i-1}) + 106f1(x_{1,i-2}, x_{2,i-2}) \\ &- 19f1(x_{1,i-3}, x_{2,i-3}) + 251f1(p_1, p_2)) \end{aligned}$$

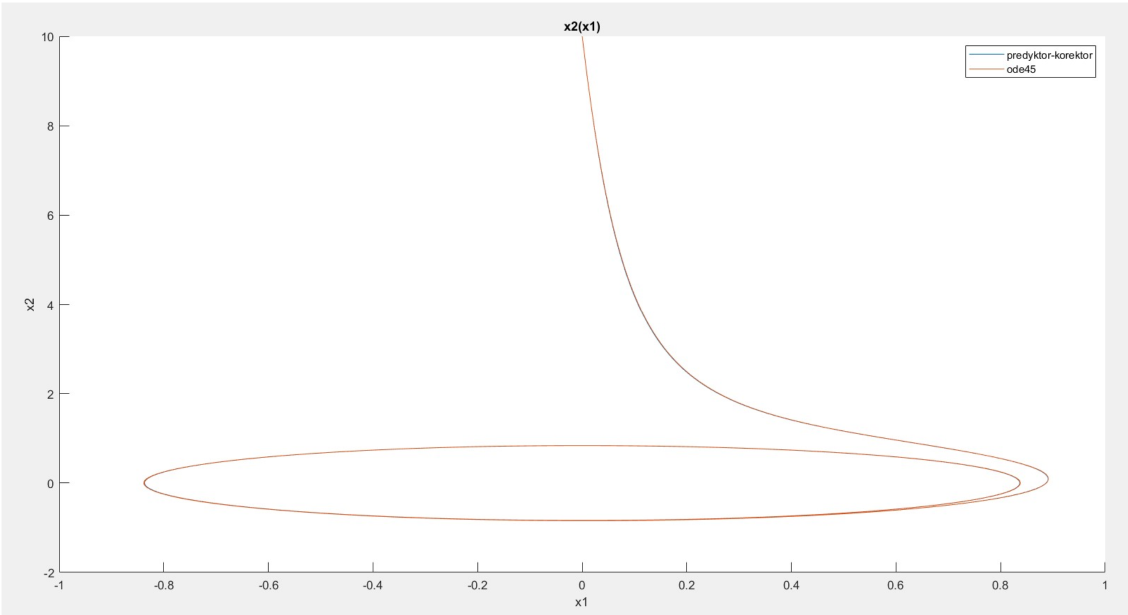
Funkcja wykorzystująca metodę predyktor – korektor ze stałym krokiem:

```
% Deklaracja funkcji dla metody predyktor-korektor ze stałym krokiem:
function [x1v, x2v] = predcorr(f1,f2,x01,x02,a,b,h)
% Dziedzina czasu:
t = (a:h:b);
% Warunki początkowe:
y(:,1) = [x01 x02];
x1v(:,1) = x01;
x2v(:,1) = x02;
% Wyliczenie czterech pierwszych punktów za pomocą metody Rungego-Kutty:
[y1, y2] = rungekutta(f1,f2,x01,x02,t(1),t(4),h);
for i=1:3
    x1 = y1(i);
    x2 = y2(i);
    x1v(i) = x1;
    x2v(i) = x2;
    y(:,i+1)=[x1 x2];
end
% Pętla główna:
for i=4:length(t)
% Wyliczenie współczynników predyktorów:
p1 = x1 + (h/24)*(55*f1(x1,x2) - 59*f1(x1v(i-1),x2v(i-1)) + 37*f1(x1v(i-2),x2v(i-2)) - 9*f1(x1v(i-3),x2v(i-3)));
p2 = x2 + (h/24)*(55*f2(x1,x2) - 59*f2(x1v(i-1),x2v(i-1)) + 37*f2(x1v(i-2),x2v(i-2)) - 9*f2(x1v(i-3),x2v(i-3)));
% Korekcja, która oblicza kolejne wartości punktów:
x1 = x1 + (h/720)*(646*f1(x1,x2) - 264*f1(x1v(i-1),x2v(i-1)) + 106*f1(x1v(i-2),x2v(i-2)) - 19*f1(x1v(i-3),x2v(i-3)) + 251*f1(p1, p2));
 x2 = x2 + (h/720)*(646*f2(x1,x2) - 264*f2(x1v(i-1),x2v(i-1)) + 106*f2(x1v(i-2),x2v(i-2)) - 19*f2(x1v(i-3),x2v(i-3)) + 251*f2(p1, p2));
 % Dodanie punktów do wektora rozwiązań:
 x1v(i) = x1;
 x2v(i) = x2;
 end
```

Wykresy $x_1(t), x_2(t), x_2(x_1)$ dla $h = 0.001$ (wybranego) wraz z wynikiem komendy „ode45”:

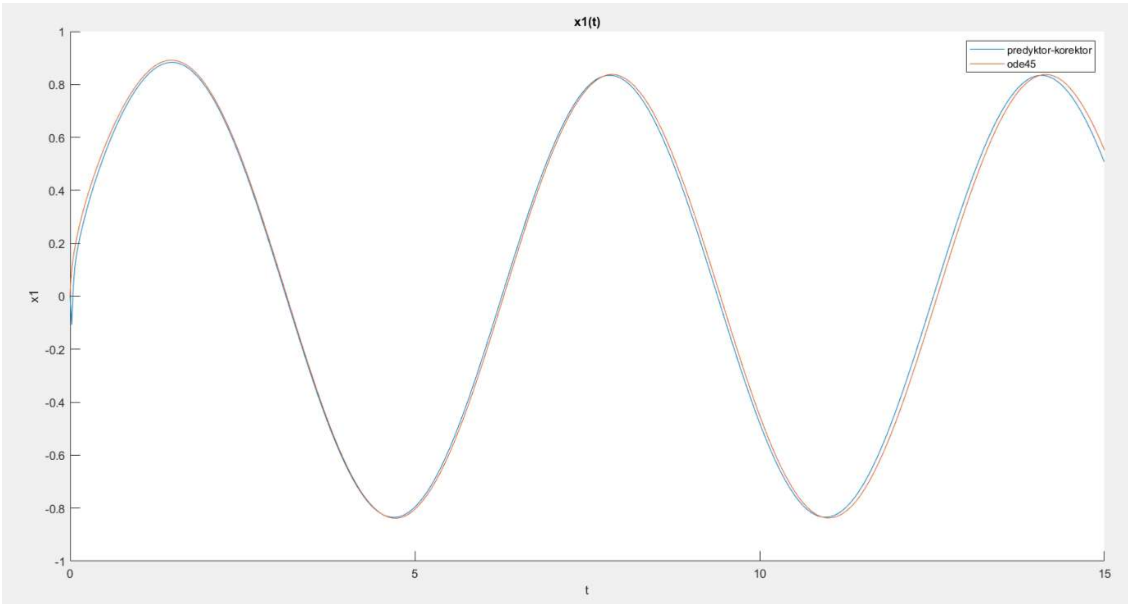

x1(t)
predyktor-korektor
ode45
x1
t

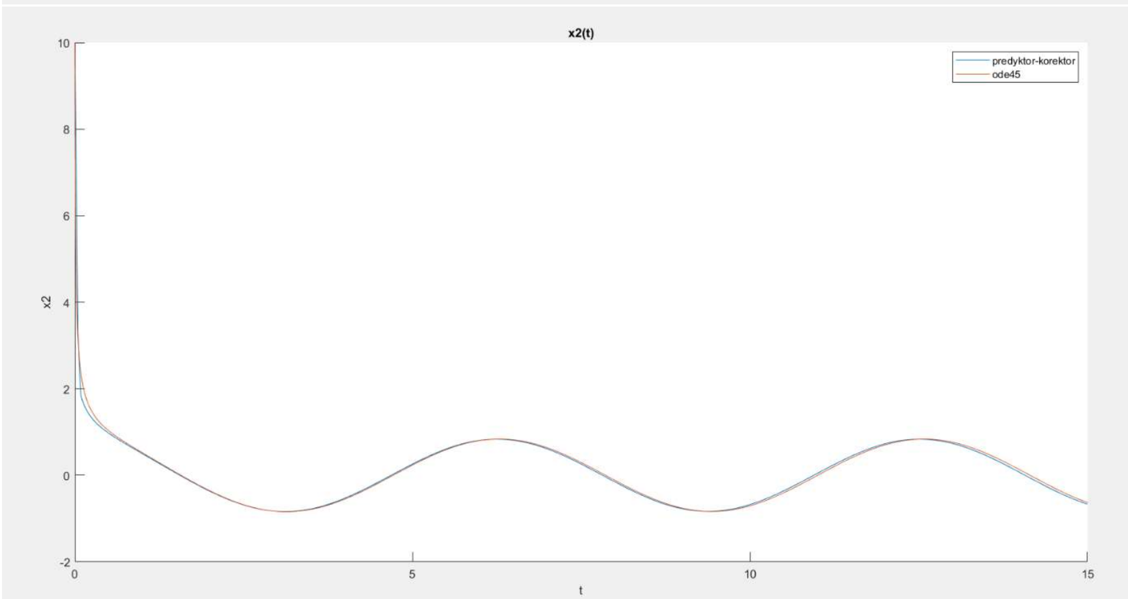

x2(t)
predyktor-korektor
ode45
x2
t

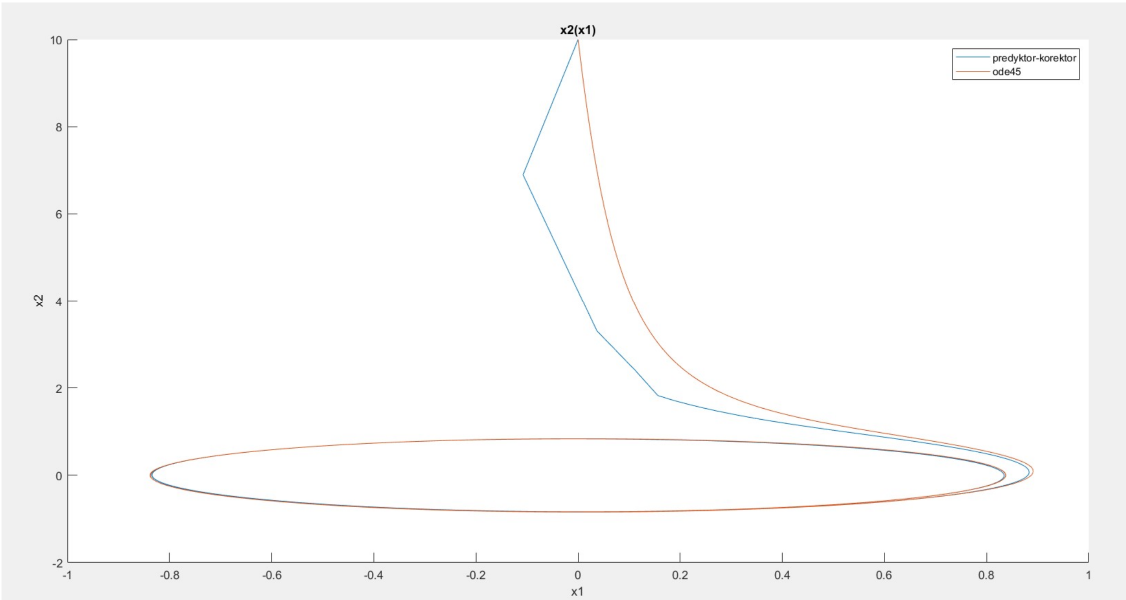


Wykresy $x_1(t), x_2(t), x_2(x_1)$ dla $h = 0.023$ (zbyt dużego) wraz z wynikiem komendy „ode45”:

m


x1(t)
predyktor-korektor
ode45
x1
t
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
0
5
10
15


x2(t)
predyktor-korektor
ode45
x2
t
10
8
6
4
2
0
-2
0
5
10
15



Wnioski do metody:

- metoda predyktor – korektor, aby zaczęła działać w miarę poprawnie potrzebuje mniejszego kroku niż metoda Rungego-Kutty czwartego rzędu ze stałym krokiem,

- metoda predyktor – korektor pomimo powyższego faktu daje bardziej dokładne rozwiązanie szczególnie w początkowych fazach daje rozwiązanie bardziej zbliżone do wyników otrzymanych komendą „ode45” niż metoda Rungego-Kutty ze stałym krokiem.

3. W metodzie Rungego-Kutty czwartego rzędu ze zmiennym krokiem w każdej iteracji wyznaczany jest nowy krok na podstawie estymacji błędu według zasady podwójnego kroku. Aby wyznaczyć nowy krok do następnej iteracji wpierw startując od punktów początkowych (a w kolejnych od punktu x_i) wyznaczamy kolejne punkty z krokiem h tak jak w metodzie Rungego-Kutty ze stałym krokiem a równolegle do niego wykonujemy dwa dodatkowe kroki o długości $0,5h$. Jako punkt otrzymany w pojedynczym kroku oznaczmy x_{kp} a w podwójnym x_{kd} gdzie $k = \{1, 2\}$.

Następnie dokonujemy szacowania błędu dla zarówno równania pierwszego jak i drugiego według wzoru:

$$\delta_k = \frac{|x_{kd} - x_{kp}|}{16}$$

Potem wyliczamy błędy dla tych równań według wzoru:

$$\varepsilon_k = |x_{kd}| \cdot \varepsilon_w + \varepsilon_b$$

gdzie ε_w oznacza dokładność względną, a ε_b dokładność bezwzględną. Oba parametry to parametry zdefiniowane przez użytkownika.

Po dokonaniu tychże obliczeń wyznaczamy parametr α_k dla każdego równania według następującego wzoru:

$$\alpha_k = (\frac{\varepsilon_k}{\delta_k})^{\frac{1}{5}}$$

Spośród wyliczonych parametrów a_k wybieramy ten, który jest najmniejszy i następnie używamy go do wyliczenia korekty kroku.

$\alpha = \min(a_k)$ dla $\mathrm{k} = \{1,2\}$

$$h_{i+1} = 0{,}9 \cdot \alpha \cdot h_i$$

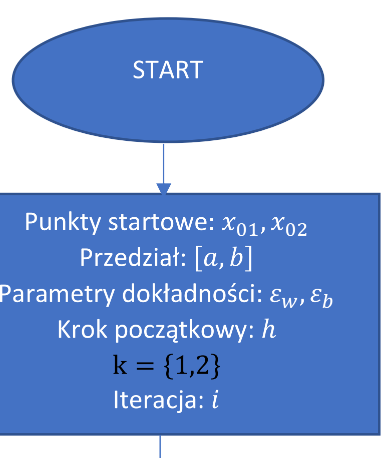
Po obliczeniu korekty kroku należy sprawdzić czy iloczyn $0{,}9 \cdot \alpha \geq 1$. Jeśli tak to sprawdzany jest kolejny warunek $x_i + h_i = b$. Jeśli ten warunek jest również spełniony to kończymy algorytm, jeśli nie to wykonujemy obliczenia: $t_{i+1} = t_i + h_i$, $h_{i+1} = \min(h_{i+1}, \ \beta h_i, \ b - t_i)$, gdzie β to dowolna liczba wybrana przez użytkownika. W tym wypadku krok jest zwiększany i przechodzimy do wyliczenia następnego punktu. Jeśli natomiast $0{,}9 \cdot \alpha \leq 1$ to następnie sprawdzamy warunek $h_{i+1} < h_{min}$. Jeśli ten warunek jest spełniony to niemożliwe jest rozwiązanie z zadaną dokładnością, natomiast jeśli nie jest spełniony to $h_i = h_{i+1}$, czyli krok jest zmniejszany oraz jeszcze raz liczymy wartości x_1 i x_2. Algorytm wykonujemy, dopóki nie osiągniemy końca przedziału.

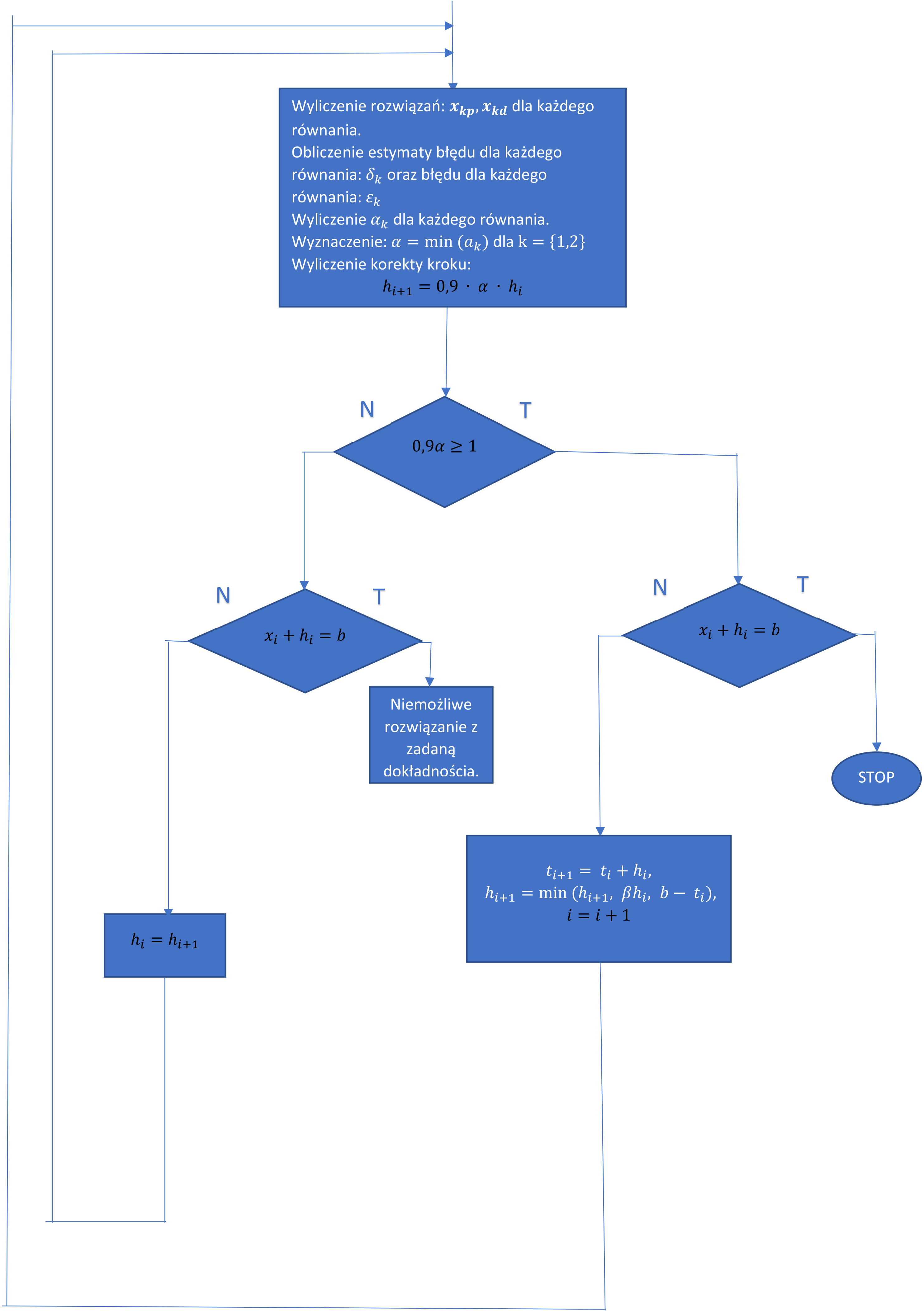
Funkcja wykorzystująca metodę Rungego-Kutty czwartego rzędu ze stałym krokiem:

```
% Deklaracja funkcji dla metody Rungego-Kutty ze zmiennym krokiem:
function [X1,X2,t, H, E1, E2] = rungekutta4z(f1,f2,x01,x02,h,a,b,epsw, epsb, hmin)
% Dziedzina czasu:
t(1,1) = a;
% Warunki początkowe:
x1=x01;
x2=x02;
% Wektory rozwiązań, estymaty błędu i długości kroku:
X1(1,1) = x01;
X2(1,1) = x02;
H(1,1) = h;
E1(1,1) = 0;
E2(1,1) = 0;
i = 1;

% Pętla główna:
while(t(i,1)+h<b

    % Wyliczenie punktu dla pojedyńczego kroku o długości h:
    k11 = f1(x1, x2);
    k12 = f2(x1, x2);

    k21 = f1(x1+0.5*h*k11, x2 + 0.5*h*k12);
    k22 = f2(x1+0.5*h*k11, x2 + 0.5*h*k12);
    k31 = f1(x1+0.5*h*k21, x2 + 0.5*h*k22);
    k32 = f2(x1+0.5*h*k21, x2 + 0.5*h*k22);

    k41 = f1(x1+h*k31, x2 + h*k32);
    k42 = f2(x1+h*k31, x2 + h*k32);

    x1p = x1 + (1/6)*(k11+2*k21+2*k31+k41)*h;
    x2p = x2 + (1/6)*(k12+2*k22+2*k32+k42)*h;

    x1d = x1;
    x2d = x2;

    % Wyliczenie punktu dla podwójnego kroku o długości 0,5h:
    for j=1:2
        k11d = f1(x1d, x2d);
        k12d = f2(x1d, x2d);

        k21d = f1(x1d+0.25*h*k11d, x2d + 0.25*h*k12d);
        k22d = f2(x1d+0.25*h*k11d, x2d + 0.25*h*k12d);

        k31d = f1(x1d+0.25*h*k21d, x2d + 0.25*h*k22d);
        k32d = f2(x1d+0.25*h*k21d, x2d + 0.25*h*k22d);
```

```
        k41d = f1(x1d+0.5*h*k31d, x2d + 0.5*h*k32d);
        k42d = f2(x1d+0.5*h*k31d, x2d + 0.5*h*k32d);

        x1d = x1d + (1/6)*(k11d+2*k21d+2*k31d+k41d)*0.5*h;
        x2d = x2d + (1/6)*(k12d+2*k22d+2*k32d+k42d)*0.5*h;

    end

    % Szacowanie błędu:
    error1=(abs(x1d-x1p))/15;
    error2=(abs(x2d-x2p))/15;

    % Wyliczenie błędu:
    e1 = abs(x1d) * epsw + epsb;
    e2 = abs(x2d) * epsw + epsb;

    % Obliczenie parametru alfa i wybranie najmniejszego z nich:
    alfa1 = (e1/error1)^(1/5);
    alfa2 = (e2/error2)^(1/5);
    alfa = min(alfa1, alfa2);

    %Wyliczenie korekty kroku:
    corr = 0.9 * alfa * h;
    % Sprawdzenie odpowiednich warunków:
    if 0.9 * alfa >= 1
        % Zwiększenie kroku i przejście do obliczania następnego punktu:
        if (t(i,1) + h <= b)
            h = min([b - t(i,1); 5*h; corr]);
            x1=x1d;
            x2=x2d;
            X1(i,1) = x1;
            X2(i,1) = x2;
            t(i+1,1) = t(i,1)+h;
            E1(i,1) = error1;
            E2(i,1) = error2;
            H(i,1) = h;
            i= i+1;
        end
    else
        % Zmniejszenie punktu lub przerwanie algorytmu:
        if h < hmin
            break;
        else
            h = corr;
        end
    end
end
```

Schemat blokowy algorytmu:

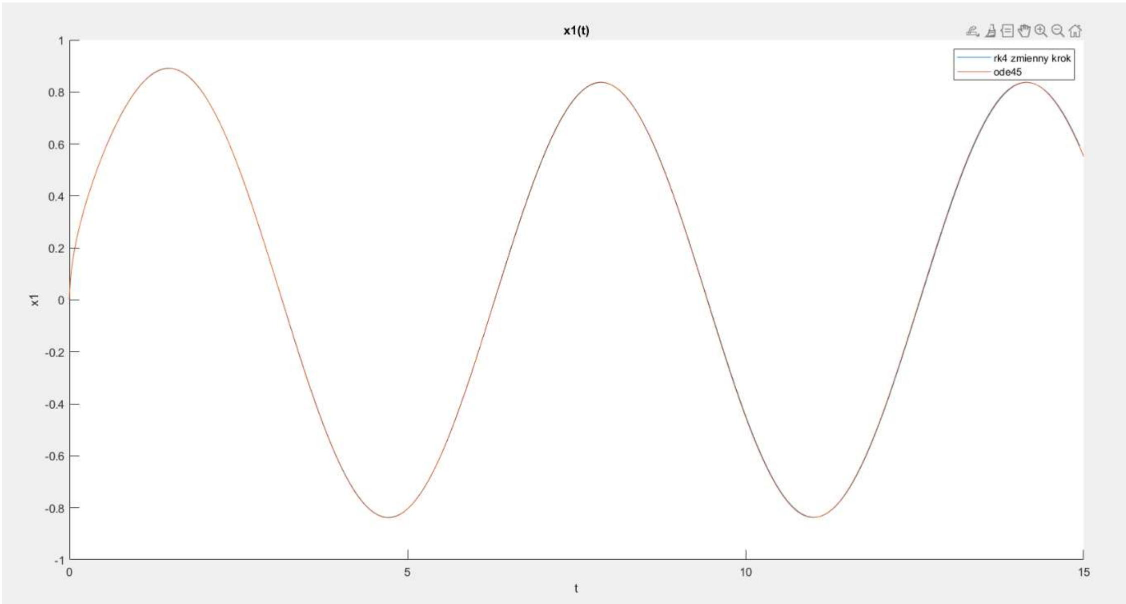


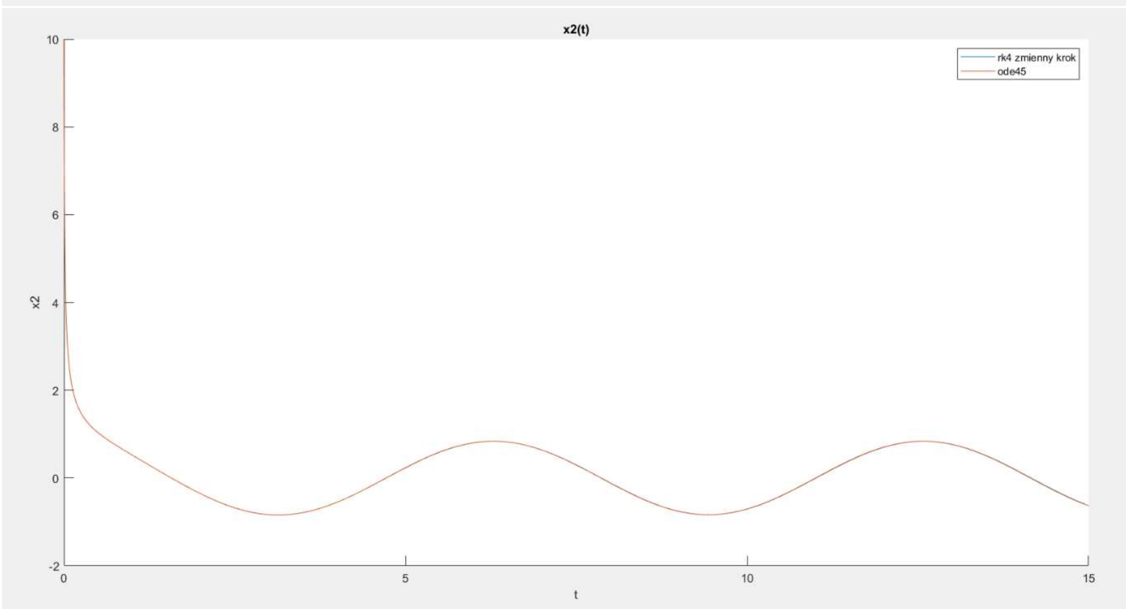


Wyliczenie rozwiązań: $\boldsymbol{x_{kp}}, \boldsymbol{x_{kd}}$ dla każdego równania.
Obliczenie estymaty błędu dla każdego równania: δ_k oraz błędu dla każdego równania: ε_k
Wyliczenie α_k dla każdego równania.
Wyznaczenie: $\alpha = \min(a_k)$ dla k = {1,2}
Wyliczenie korekty kroku:
$h_{i+1} = 0{,}9 \cdot \alpha \cdot h_i$
N
T
$0{,}9\alpha \geq 1$
N
T
$x_i + h_i = b$
N
T
$x_i + h_i = b$
Niemożliwe rozwiązanie z zadaną dokładnością.
STOP
$t_{i+1} = t_i + h_i,$
$h_{i+1} = \min(h_{i+1}, \beta h_i, b - t_i),$
$i = i + 1$
$h_i = h_{i+1}$

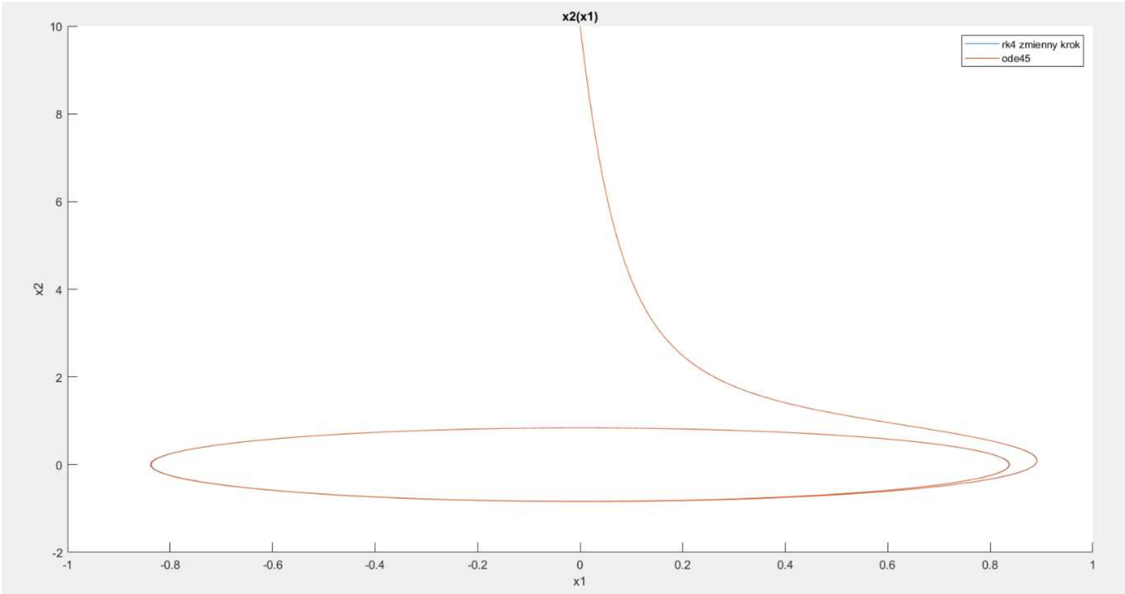
Wykresy dla metody Rungego-Kutty czwartego rzędu ze zmiennym krokiem:

$x_1(t), x_2(t)$:

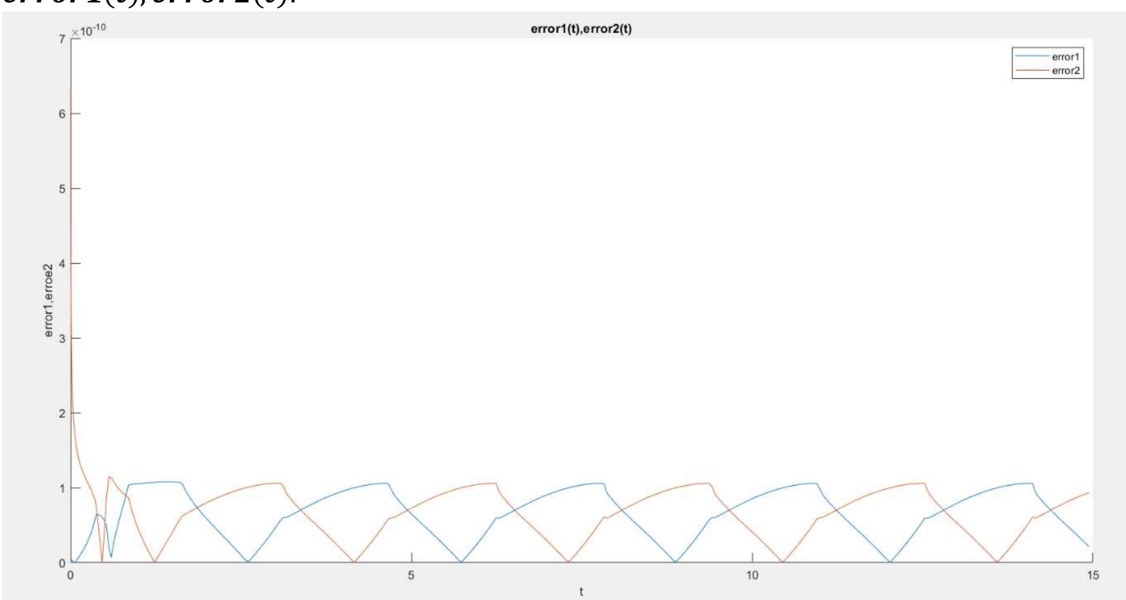




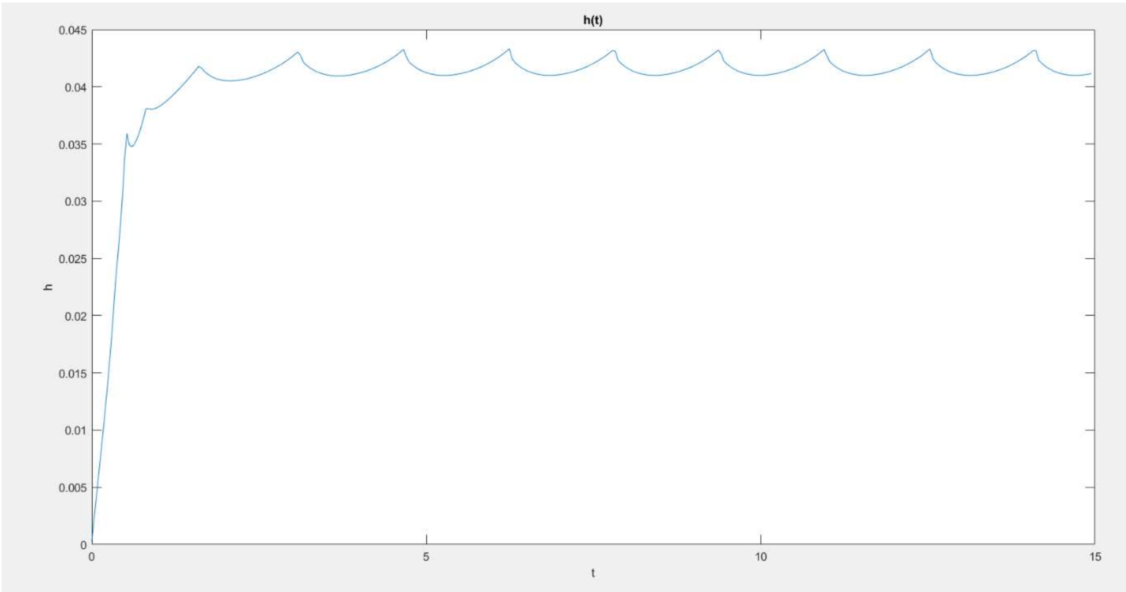
$x_2(x_1)$:



$error1(t), error2(t)$:



$h(t)$:



Wybrane parametry dokładności i kroku minimalnego:

$$\varepsilon_w = \mathbf{10^{-10}}$$

$$\varepsilon_b = \mathbf{10^{-10}}$$

$$h_{min} = \mathbf{10^{-16}}$$

Wnioski do metody:

- zmiana kroku pozwala zmniejszyć liczbę iteracji,

- zmiana kroku pozwala zmniejszyć nakład wykonywanych obliczeń, a co za tym idzie przyspieszyć algorytm

Wnioski:

Metoda Rungego-Kutty czwartego rzędu ze stałym krokiem wymaga doboru odpowiednio małego kroku, aby uzyskać zadowalające rezultaty. Tego samego wymaga metoda predyktor – korektor Adamsa. Z powodu małego kroku wydłuża się czas obliczeń, jednak w przypadku metody predyktor – korektor Adamsa nakład obliczeń jest mniejszy niż dla metody Rungego-Kutty ze stałym krokiem. W metodzie Rungego-Kutty czwartego rzędu ze zmiennym krokiem dobór kroku nie ma tak dużego znaczenia jak w metodach stałokrokowych, gdyż algorytm sam ustawia odpowiedni krok. Metoda zmiennokrokowa pozwala zmniejszyć liczbę wykonywanych iteracji oraz przyspieszyć działanie programu. Natomiast metody stałokrokowe są łatwiejsze do zaimplementowania, więc wybór metody zależy od tego, które z kryteriów jest dla nas decydujące, jednak pod względem obliczeń numerycznych najlepszą metodą jest metoda Rungego-Kutty ze zmiennym krokiem.