MNUM – Projekt 4.15

Zadanie 1

Ruch punktu opisany jest równaniami:

$$x_1' = x_2 + x_1(0{,}5 - {x_1}^2 - {x_2}^2)$$
$$x_2' = -x_1 + x_2(0{,}5 - {x_1}^2 - {x_2}^2)$$

Obliczyć przebieg trajektorii ruchu na przedziale $[0, 20]$ dla podanych warunków początkowych, z użyciem metod: Rungego-Kutty czwartego rzędu oraz wielokrokowej predyktor-korektor Adamsa czwartego rzędu ze stałym krokiem.

Metoda RK4:

Metodę tę można zdefiniować następująco:

$$y_{n+1} = y_n + \frac{1}{6} h\,(k_1 + 2k_2 + 2k_3 + k_4)$$
$$k_1 = f(x_n, y_n)$$
$$k_2 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1)$$
$$k_3 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2)$$
$$k_4 = f(x_n + h, y_n + hk_3)$$

Współczynnik k_1 jest pochodną rozwiązania w punkcie (x_n, y_n). Wartość k_2 wyznaczamy jak w zmodyfikowanej metodzie Eulera - jako pochodną rozwiązania wyznaczanego zwykłą metodą Eulera w punkcie $(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_1)$ (środkowym przedziału). Następnie, wartość k_3 wyznaczamy podobnie, jak k_2, ale tym razem w punkcie $(x_n + \frac{1}{2}h, y_n + \frac{1}{2}hk_2)$, tzn. startując z początku przedziału z nachyleniem k_2. W końcu, z nachyleniem tej stycznej startujemy z punktu początkowego do punktu $(x_n + h, y_n + hk_3)$, tzn. wyznaczamy pochodną rozwiązania k_4 w punkcie $(x_n + h, y_n + hk_3)$. Mamy w ten sposób wyznaczone 4 wartości pochodnej rozwiązania: po jednej na końcach przedziału i dwie w jego środku. Aproksymacja pochodnej dla pełnego kroku metody wyznaczana jest jako ważona średnia arytmetyczna tych wartości, z wagami 1 na krańcach i wagą 2 w punkcie środkowym.

Krok był zmniejszany do momentu aż wykres prezentował wystarczającą dokładność (gładkość). Błąd pojedynczego kroku był szacowany na podstawie wzoru:

$$\delta_n(h) = \frac{2^p}{2^p - 1}({y_n}^{(2)} - {y_n}^{(1)})$$

Gdzie:

${y_n}^{(1)}$ – nowy punkt uzyskany w kroku o długości h

${y_n}^{(2)}$ – nowy punkt wyznaczony przez dwa dodatkowe kroki o długościach $0.5h$

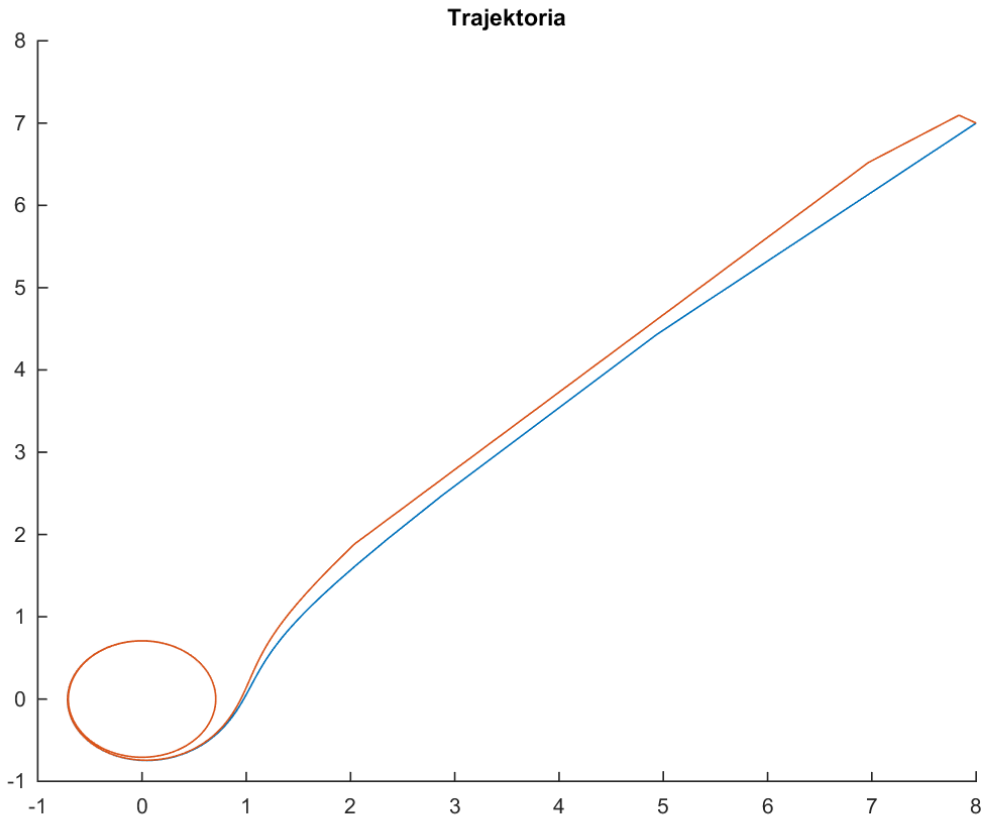
p – rząd metody.

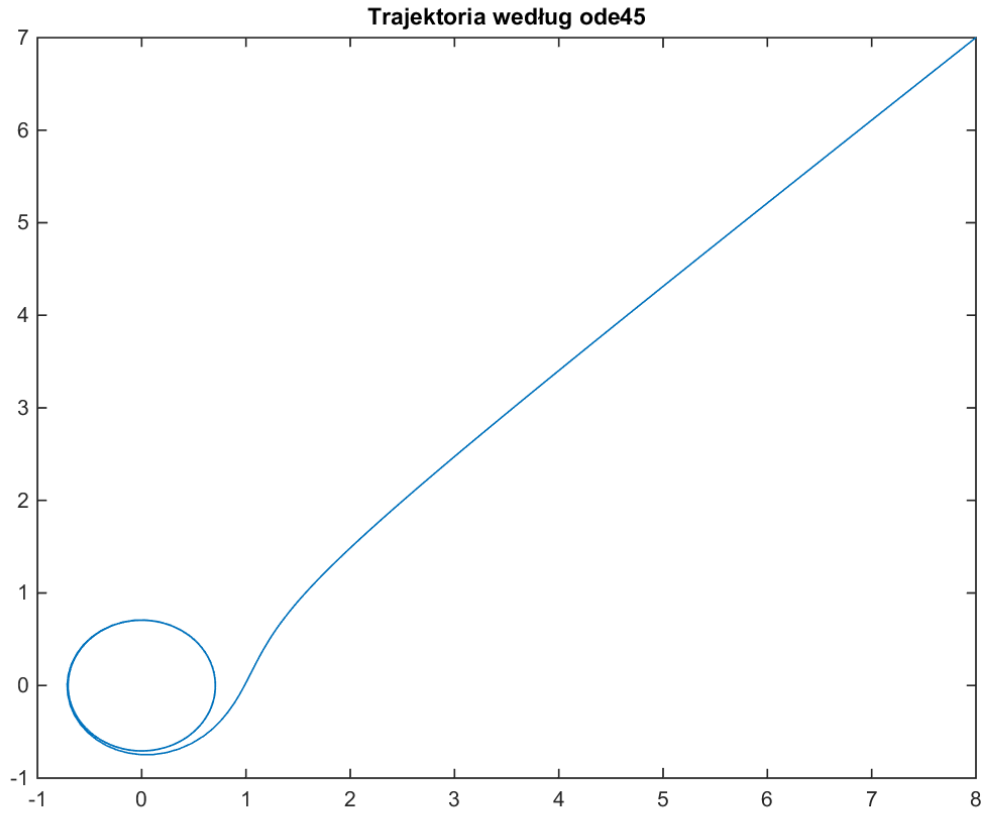
Kod algorytmu:

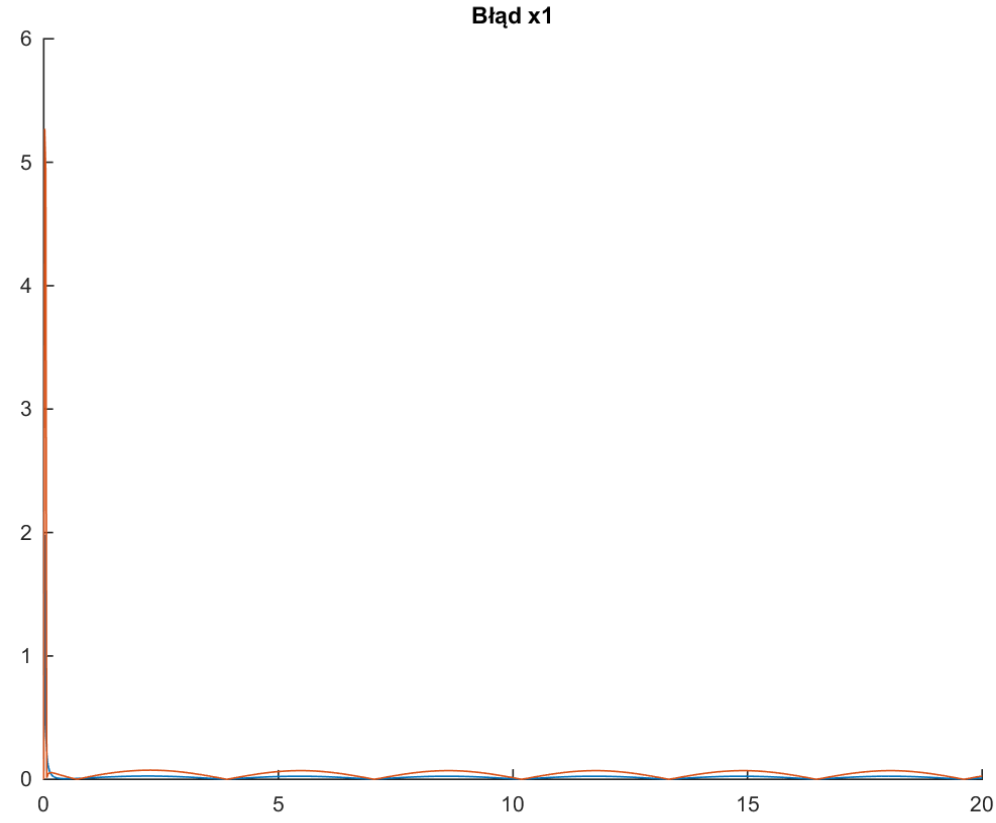
```
function [ x1,x2, err1, err2, t ] = rk4( podzial,war1, war2, podpkt)
skok=20/(podzial);
skokh=20/(podzial*2);
x1(1)=war1;
x2(1)=war2;
x1h(1)=war1;
x2h(1)=war2;
tic
for i = 1:(podzial)
    k11 = p1(x1(i),x2(i));
    k12 = p2(x1(i),x2(i));
    k21 = p1(x1(i) + 0.5*skok*k11, x2(i) + 0.5*skok*k12);
    k22 = p2(x1(i) + 0.5*skok*k11, x2(i) + 0.5*skok*k12);
    k31 = p1(x1(i) + 0.5*skok*k21, x2(i) + 0.5*skok*k22);
    k32 = p2(x1(i) + 0.5*skok*k21, x2(i) + 0.5*skok*k22);
    k41 = p1(x1(i) + skok*k31, x2(i) + skok*k32);
    k42 = p2(x1(i) + skok*k31, x2(i) + skok*k32);
    x1(i+1) = x1(i) + (1/6)*skok*(k11 + 2*k21 + 2*k31 + k41);
    x2(i+1) = x2(i) + (1/6)*skok*(k12 + 2*k22 + 2*k32 + k42);

    x1ht(1) = x1h(i);
    x2ht(1) = x2h(i);
    for j = 1:2
        k11 = p1(x1ht(j),x2ht(j));
        k12 = p2(x1ht(j),x2ht(j));
        k21 = p1(x1ht(j) + 0.5*skokh*k11, x2ht(j) + 0.5*skokh*k12);
        k22 = p2(x1ht(j) + 0.5*skokh*k11, x2ht(j) + 0.5*skokh*k12);
        k31 = p1(x1ht(j) + 0.5*skokh*k21, x2ht(j) + 0.5*skokh*k22);
        k32 = p2(x1ht(j) + 0.5*skokh*k21, x2ht(j) + 0.5*skokh*k22);
        k41 = p1(x1ht(j) + skokh*k31, x2ht(j) + skokh*k32);
        k42 = p2(x1ht(j) + skokh*k31, x2ht(j) + skokh*k32);
        x1ht(j+1) = x1ht(j) + (1/6)*skokh*(k11 + 2*k21 + 2*k31 + k41);
        x2ht(j+1) = x2ht(j) + (1/6)*skokh*(k12 + 2*k22 + 2*k32 + k42);
    end
    x1h(i+1)=x1ht(3);
    x2h(i+1)=x2ht(3);
end
err1 =(16/15) * abs(x1h - x1);
err2 =(16/15) * abs(x2h - x2);
t = toc;
end
```

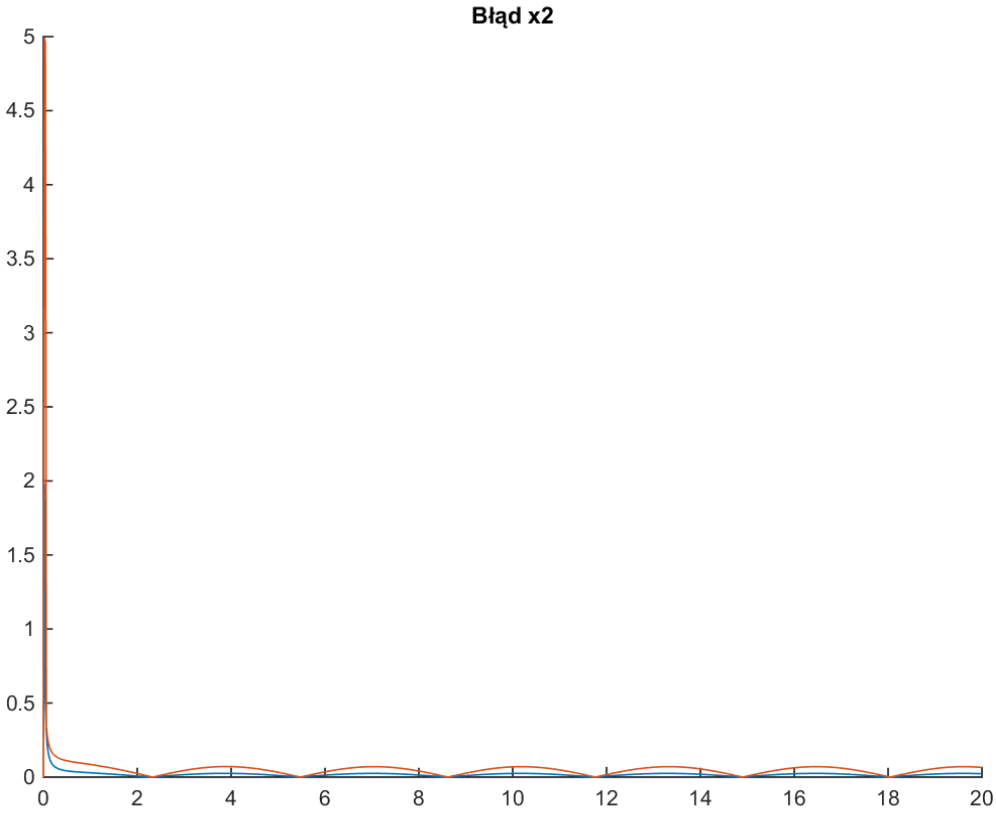
a) $x_1(0) = 8, \quad x_2(0) = 7$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.02



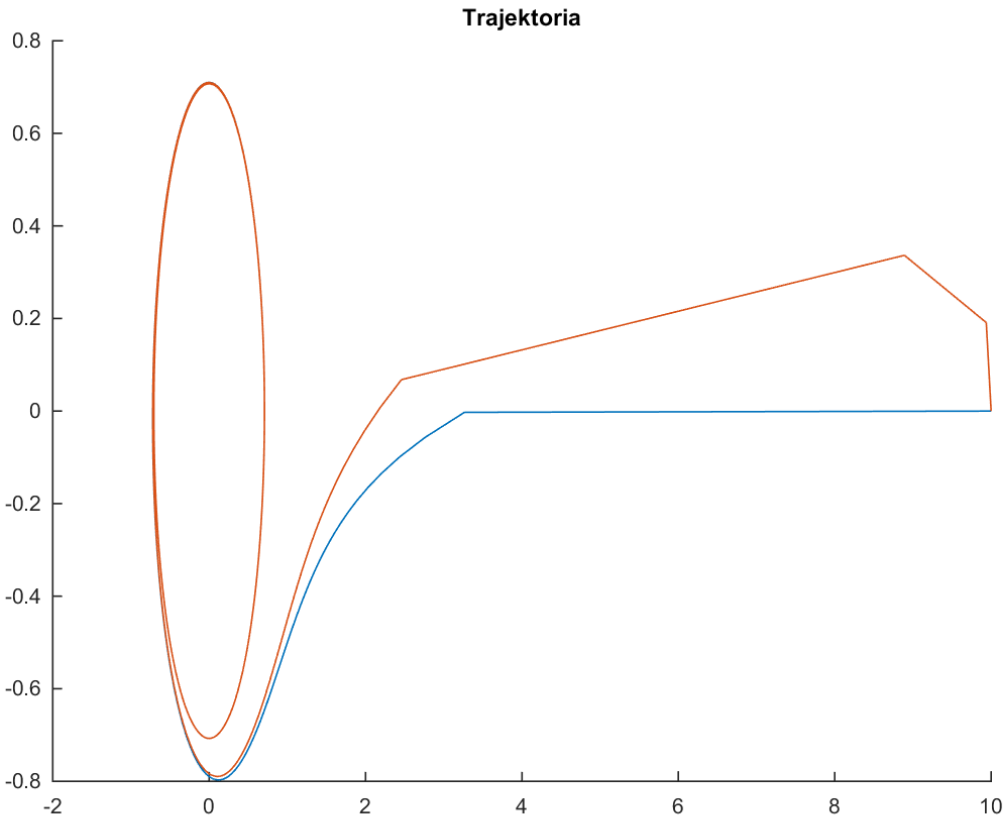


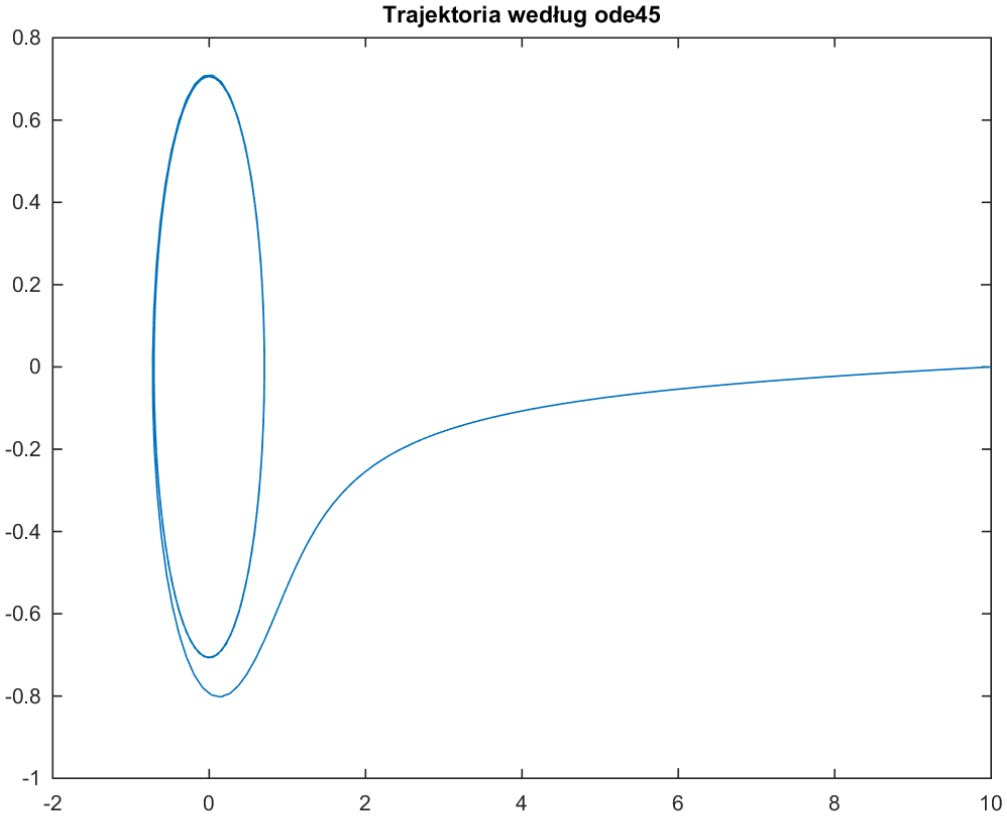

Błąd x1
6
5
4
3
2
1
0
0
5
10
15
20

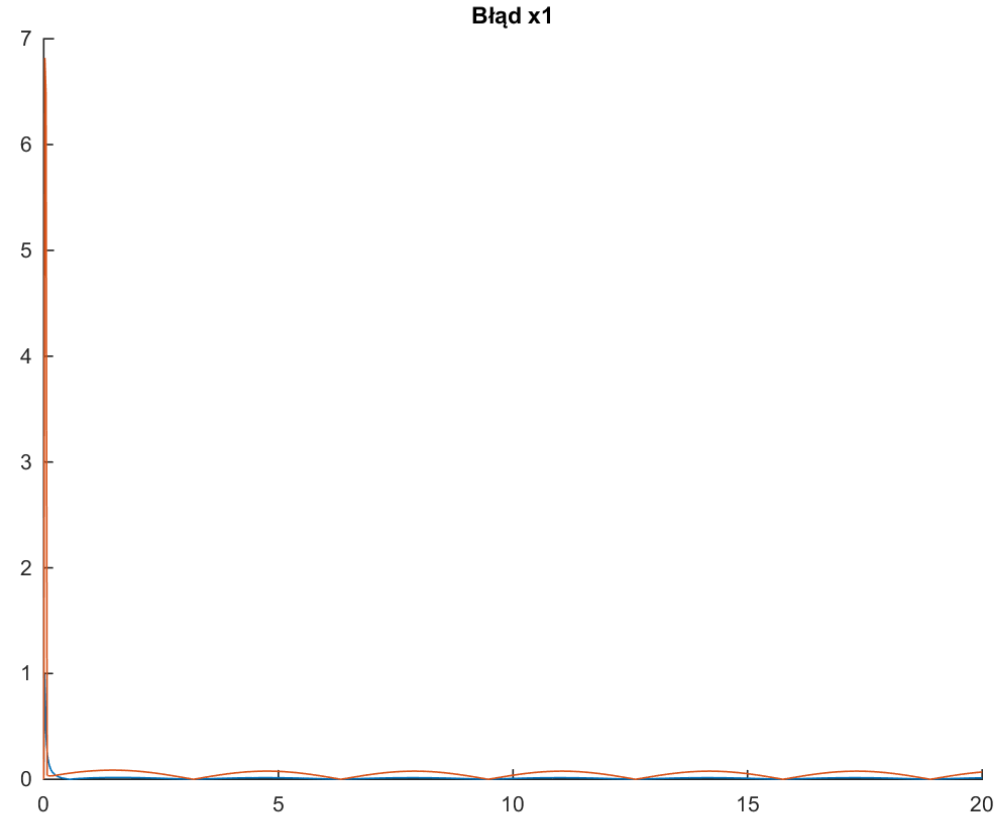

Błąd x2
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20

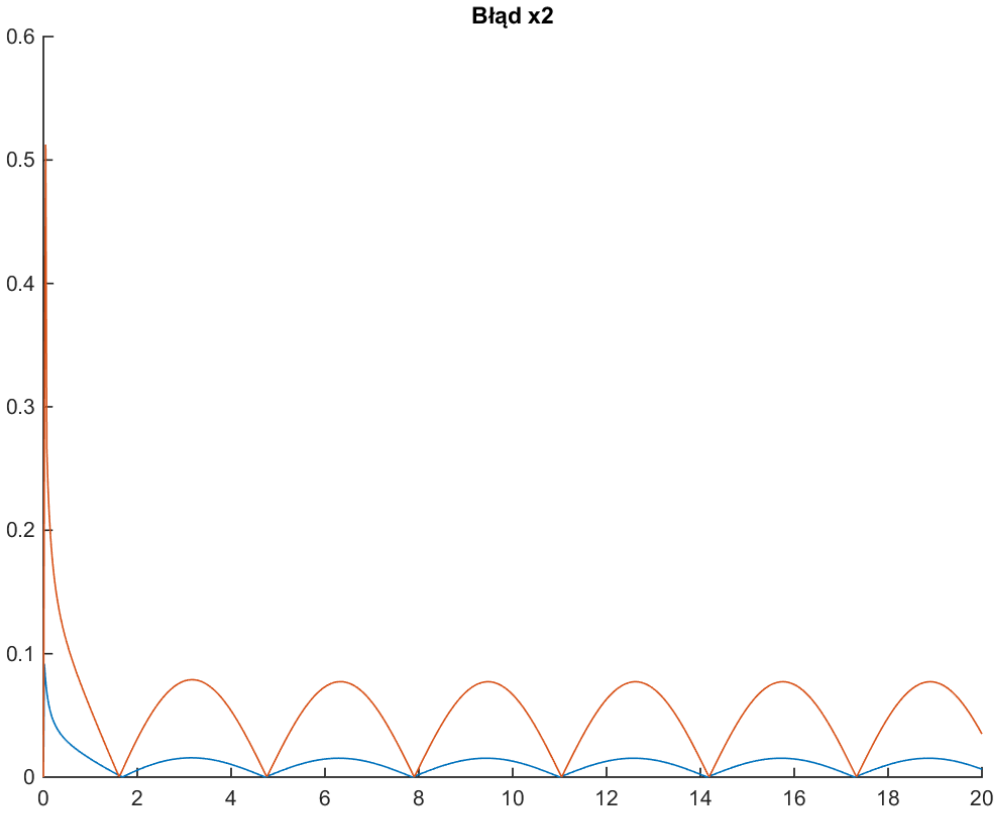
b) $x_1(0) = 0, \quad x_2(0) = 0.3$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.02



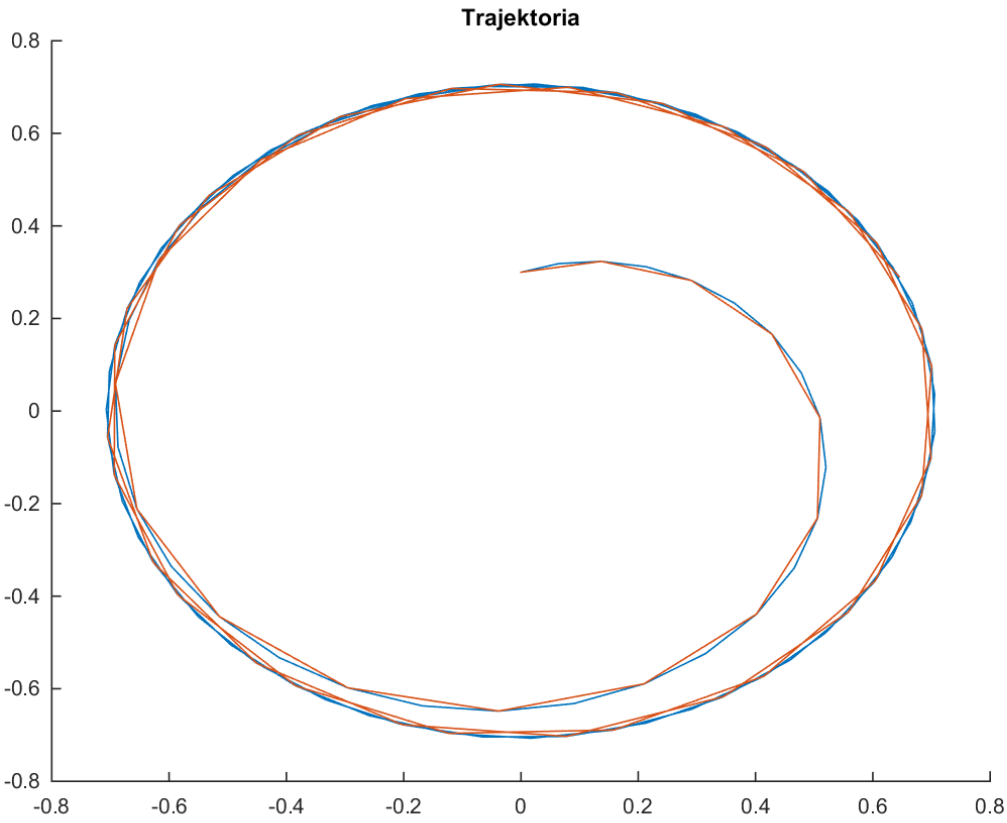


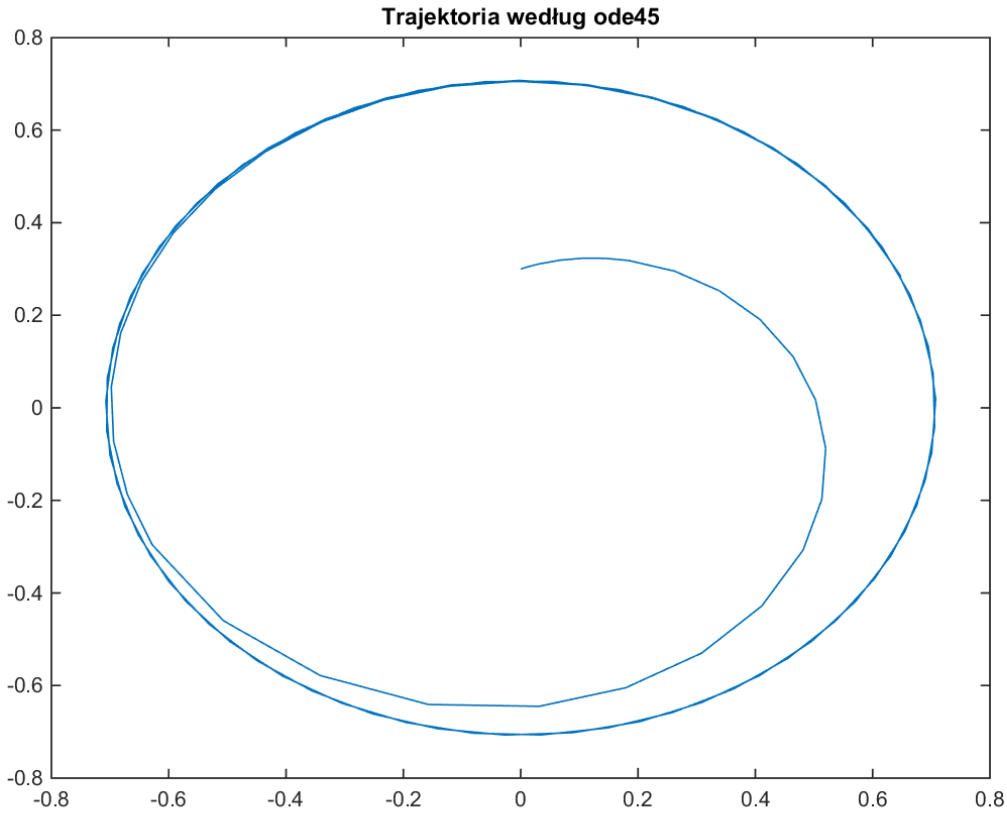

Błąd x1
7
6
5
4
3
2
1
0
0
5
10
15
20

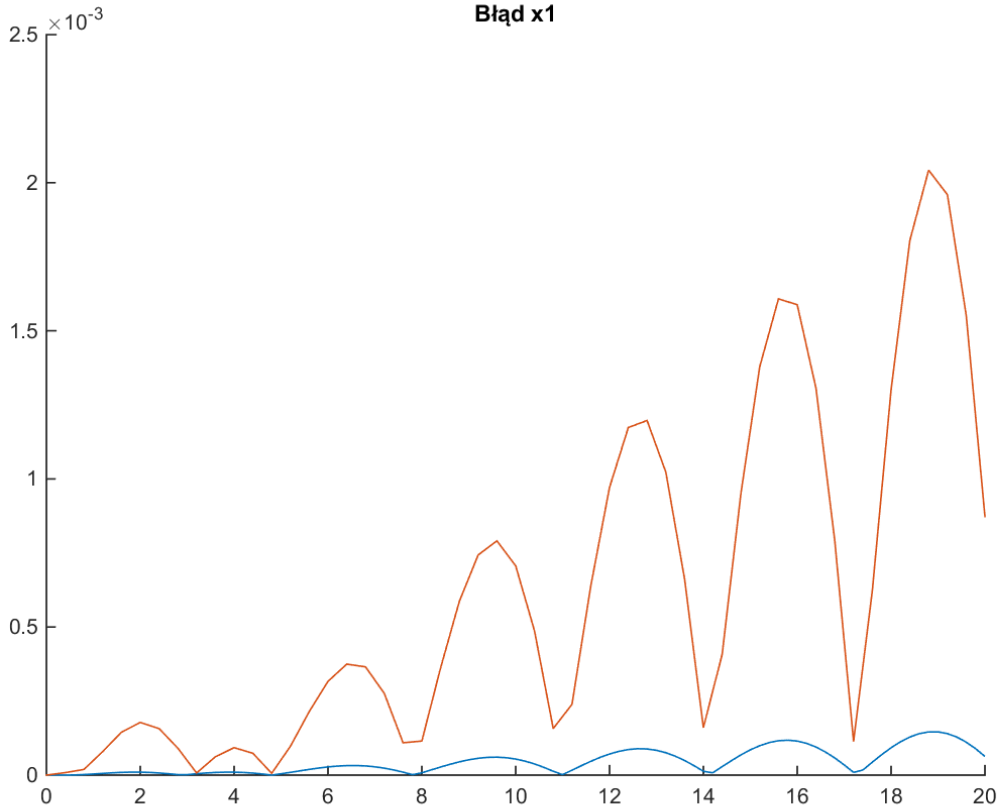

Błąd x2
0.6
0.5
0.4
0.3
0.2
0.1
0
0
2
4
6
8
10
12
14
16
18
20

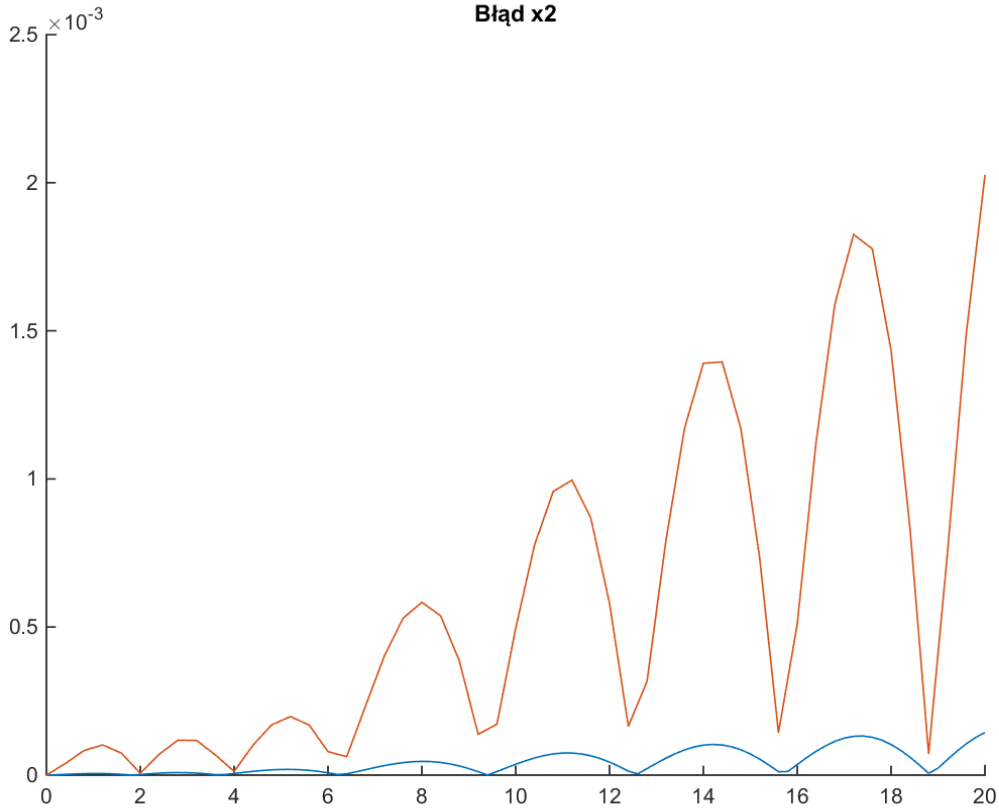
c) $x_1(0) = 10, \quad x_2(0) = 0$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2



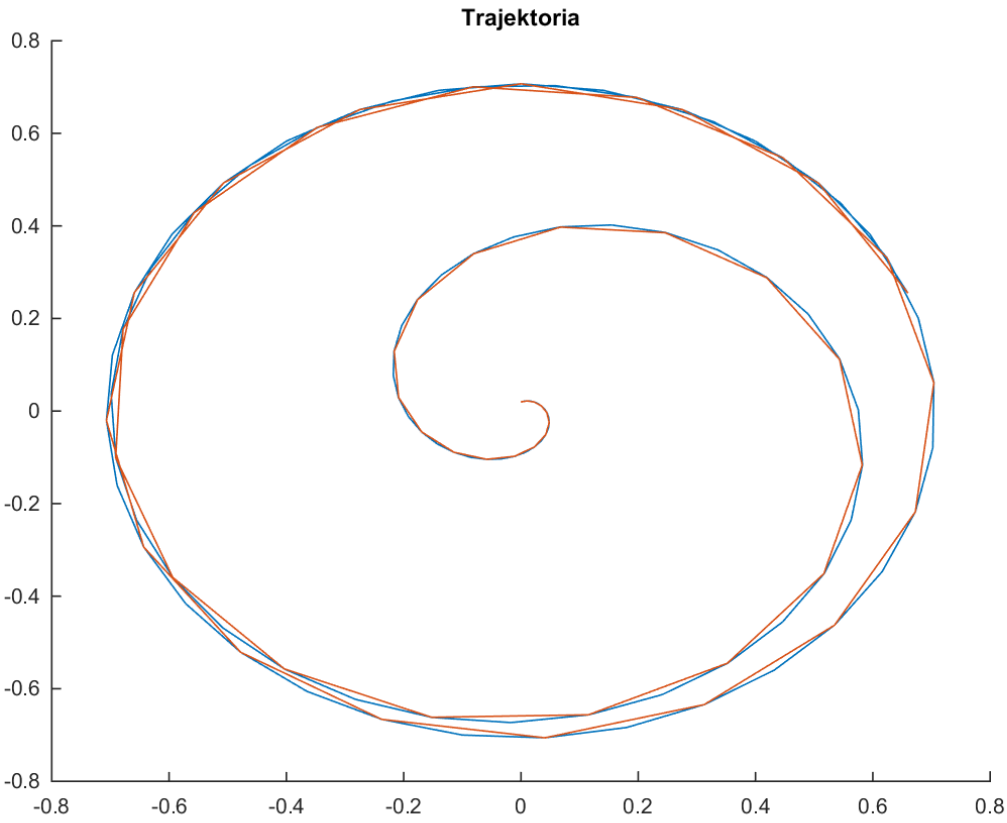


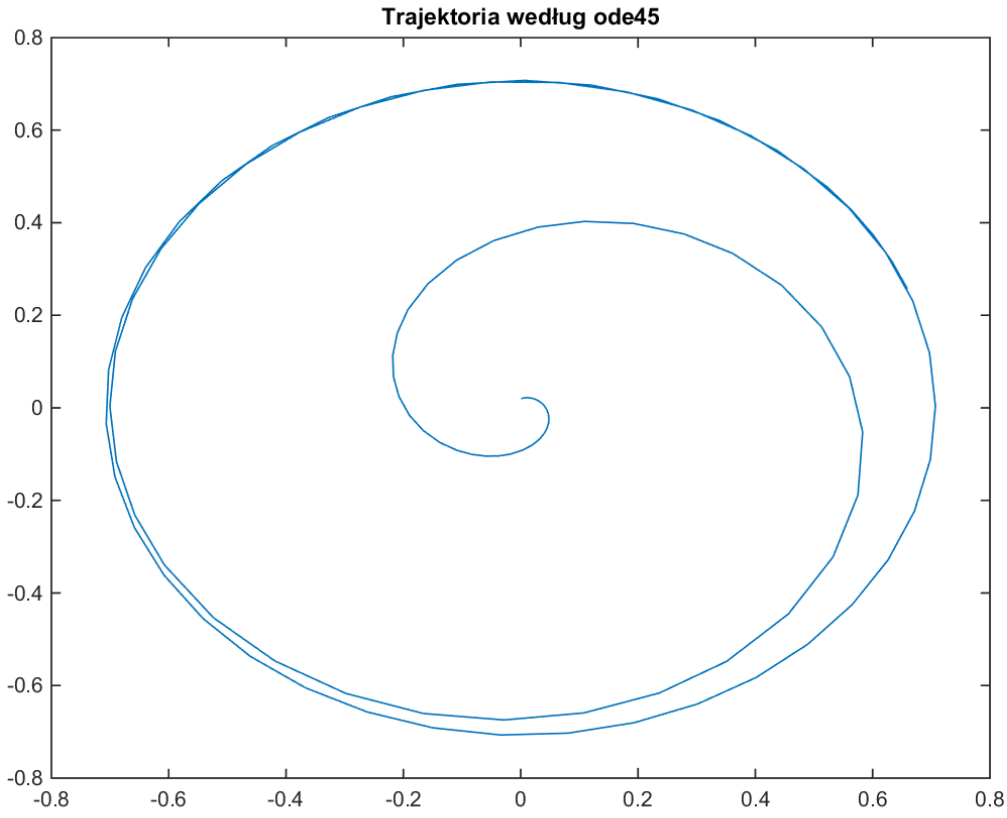

Błąd x1
×10^-3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20

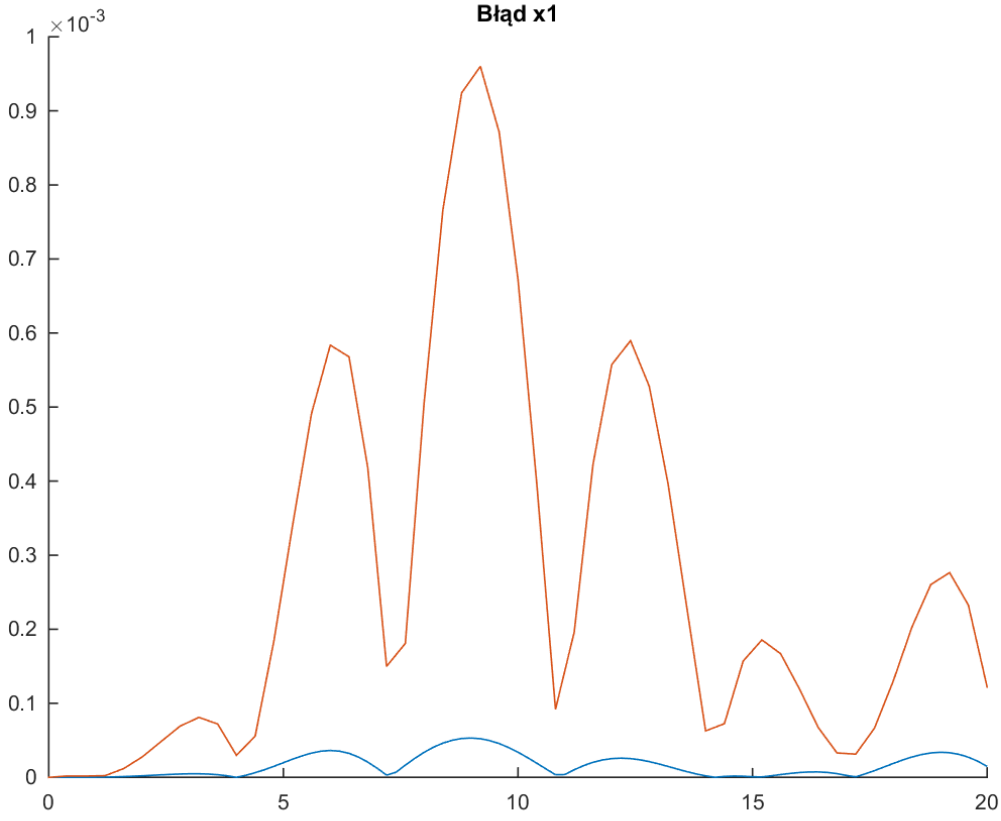

Błąd x2
×10^-3
2.5
2
1.5
1
0.5
0
0
2
4
6
8
10
12
14
16
18
20

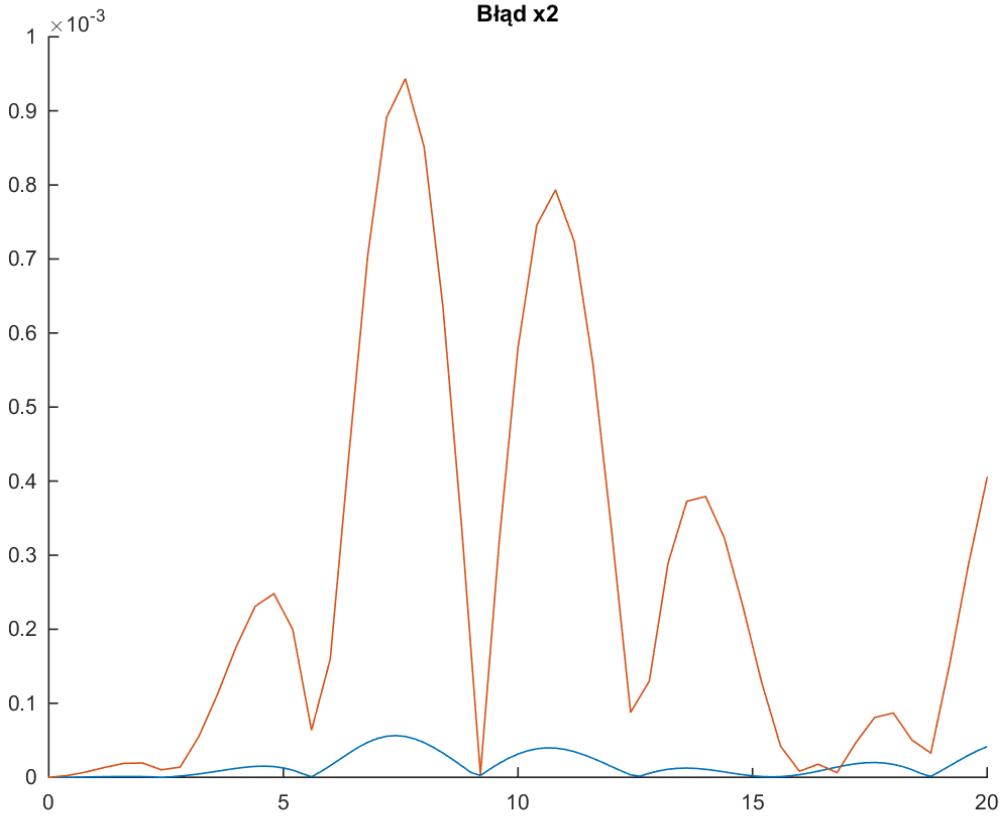
d) $x_1(0) = 0.001, \quad x_2(0) = 0.02$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2






Błąd x1
×10-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20


Błąd x2
×10-3
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
5
10
15
20

Metoda predyktor-korektor Adamsa czwartego rzędu:

Metody Adamsa:

Równanie różniczkowe

$$y'(x) = f\big(x, y(x)\big),$$

$$y(a) = y_a, x \in [a, b]$$

Równoważne jest równaniu całkowemu

$$y(x) = y(a) + \int_a^x f(t, y(t)dt$$

Metody Adamsa dostajemy, rozważając to równanie na przedziale $[x_{n-1}, x_n]$:

$$y(x_n) = y(x_{n-1}) + \int_{x_{n-1}}^{x_n} f(t, y(t)dt$$

Metody jawne:

Funkcję podcałkową przybliżamy wielomianem interpolacyjnym $W(x)$ stopnia co najwyżej k-1 opartym na węzłach $x_{n-1}, \dots, x_{n-k}$. Przyjmując przybliżenie Lagrange'a mamy:

$$f\big(x, y(x)\big) \approx W(x) = \sum_{j=1}^{k} f\big(x_{n-j}, y_{n-j}\big)\, L_j(x)$$

$$y_n = y_{n-1} + \sum_{j=1}^{k} f\big(x_{n-j}, y_{n-j}\big) \int_{x_{n-1}}^{x_n} L_j(t)dt$$

Gdzie $L_j(x)$ to wielomiany Lagrange'a:

$$L_j(x) = \prod_{m=1, m\neq j}^{k} \frac{x - x_{n-m}}{x_{n-j} - x_{n-m}}$$

Stąd po scałkowaniu, przy założeniu $x_{n-1} = x_n - hj, j = 1,2, \dots, k$ otrzymujemy:

$$y_n = y_{n-1} + h \sum_{j=1}^{k} \beta_j f\big(x_{n-j}, y_{n-j}\big)$$

Wartości β_j są stablicowane.

Metody niejawne:

Funkcję podcałkową przybliżamy wielomianem interpolacyjnym stopnia co najwyżej k opartym na węzłach $x_n, x_{n-1}, \dots, x_{n-k}$, z wartościami rozwiązania $y(x_{n-j}) \approx y_{n-j}$. Następnie postępując tak jak w przypadku metody jawnej otrzymujemy

$$y_n = y_{n-1} + h \sum_{j=0}^{k} \beta_j^* f\big(x_{n-j}, y_{n-j}\big) = y_{n-1} + \beta_0^* f(x_n, y_n) + h \sum_{j=1}^{k} \beta_j^* f\big(x_{n-j}, y_{n-j}\big)$$

Wartości β_j^* są stablicowane.

Metoda predyktor-korektor:

Realizacja metody PK polega na połączeniu metod jawnych i niejawnych w jeden algorytm. W naszym przypadku:

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

E: $f_n^{[0]} = f(x_n, y_n^{[0]})$

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

E: $f_n = f(x_n, y_n)$

Dla metody PK oszacowanie błędu:

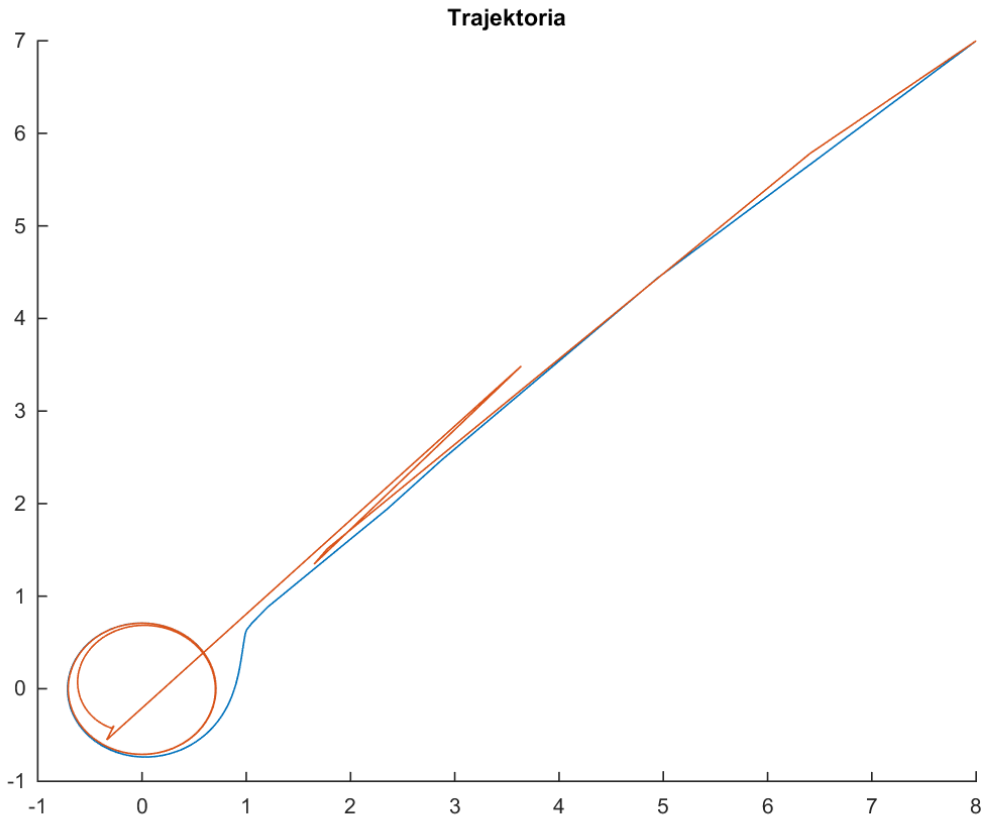
$$\delta_n(h_{n-1}) = -\frac{19}{270}(y_n^{[0]} - y_n)$$

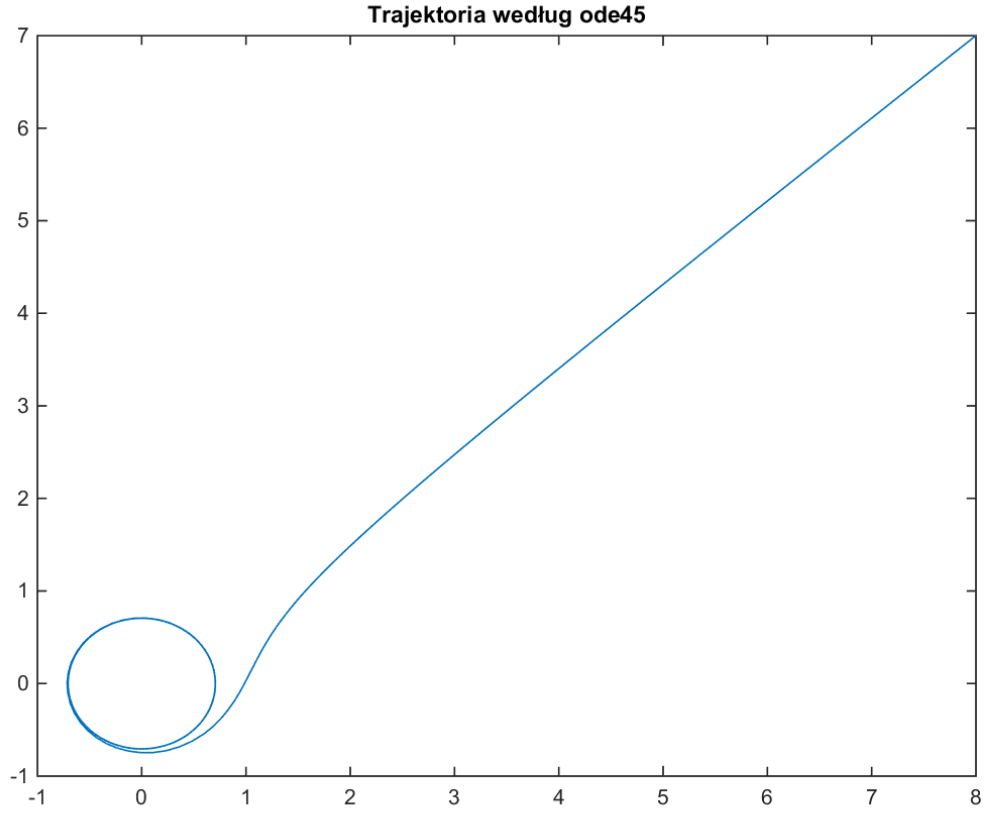
Kod algorytmu:

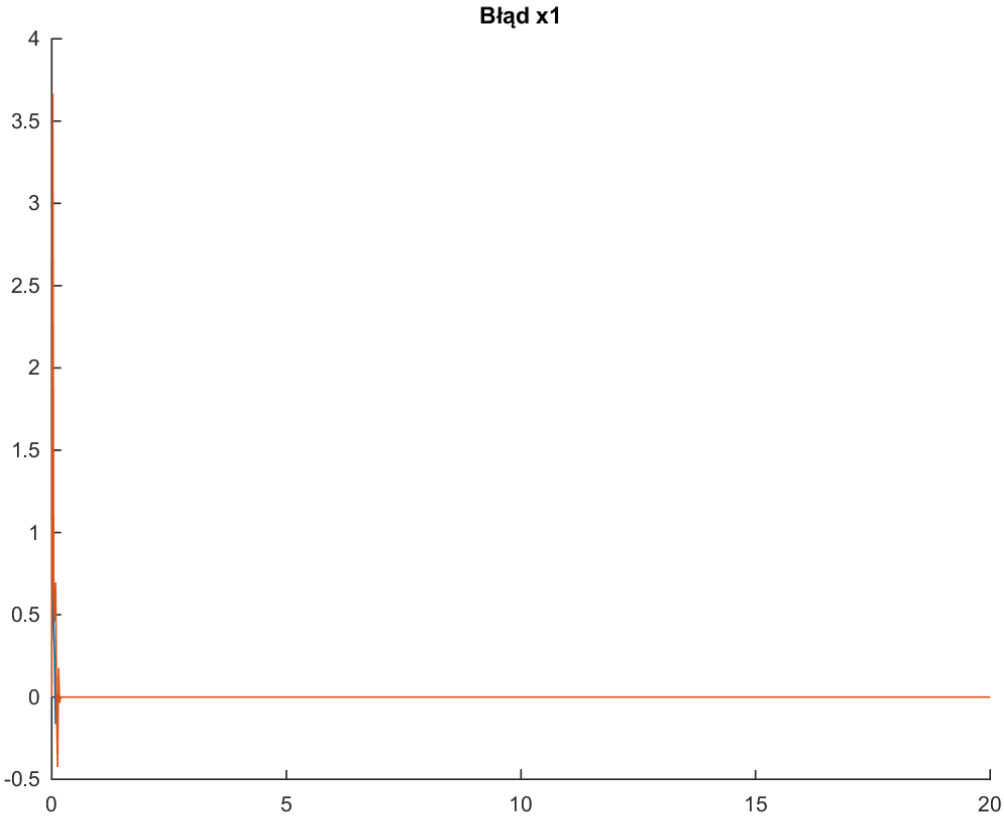
```
function [ x1,x2,err1, err2, t ] = pk( podzial,war1, war2, podpkt )
skok=20/(podzial);
skokh=20/(podzial*2);
x1(1)=war1;
x2(1)=war2;
x1h(1)=war1;
x2h(1)=war2;
beta=[55/24,-59/24,37/24,-9/24];
betag=[251/720,646/720,-264/720,106/720,-19/720];
tic;
for i = 1:3
    k11 = p1(x1(i),x2(i));
    k12 = p2(x1(i),x2(i));
    k21 = p1(x1(i) + 0.5*skok*k11, x2(i) + 0.5*skok*k12);
    k22 = p2(x1(i) + 0.5*skok*k11, x2(i) + 0.5*skok*k12);
    k31 = p1(x1(i) + 0.5*skok*k21, x2(i) + 0.5*skok*k22);
    k32 = p2(x1(i) + 0.5*skok*k21, x2(i) + 0.5*skok*k22);
    k41 = p1(x1(i) + skok*k31, x2(i) + skok*k32);
    k42 = p2(x1(i) + skok*k31, x2(i) + skok*k32);
    x1(i+1) = x1(i) + (1/6)*skok*(k11 + 2*k21 + 2*k31 + k41);
    x2(i+1) = x2(i) + (1/6)*skok*(k12 + 2*k22 + 2*k32 + k42);

    x1ht(1) = x1h(i);
    x2ht(1) = x2h(i);
    for j = 1:2
        k11 = p1(x1ht(j),x2ht(j));
        k12 = p2(x1ht(j),x2ht(j));
        k21 = p1(x1ht(j) + 0.5*skokh*k11, x2ht(j) + 0.5*skokh*k12);
        k22 = p2(x1ht(j) + 0.5*skokh*k11, x2ht(j) + 0.5*skokh*k12);
        k31 = p1(x1ht(j) + 0.5*skokh*k21, x2ht(j) + 0.5*skokh*k22);
        k32 = p2(x1ht(j) + 0.5*skokh*k21, x2ht(j) + 0.5*skokh*k22);
        k41 = p1(x1ht(j) + skokh*k31, x2ht(j) + skokh*k32);
        k42 = p2(x1ht(j) + skokh*k31, x2ht(j) + skokh*k32);
        x1ht(j+1) = x1ht(j) + (1/6)*skokh*(k11 + 2*k21 + 2*k31 + k41);
        x2ht(j+1) = x2ht(j) + (1/6)*skokh*(k12 + 2*k22 + 2*k32 + k42);
    end
    x1h(i+1)=x1ht(3);
    x2h(i+1)=x2ht(3);
end
err1 =(16/15) * abs(x1h - x1);
err2 =(16/15) * abs(x2h - x2);
for i=5:(podzial+1)
    suma1 = 0;
    suma2 = 0;
    for j =1:4
        suma1 = suma1 + beta(j)*p1(x1(i-j), x2(i-j));
        suma2 = suma2 + beta(j)*p2(x1(i-j), x2(i-j));
    end
    x10 = x1(i-1) + skok*suma1;
    x20 = x2(i-1) + skok*suma2;
    f1 = p1(x10,x20);
    f2 = p2(x10,x20);
    suma1 = 0;
    suma2 = 0;
    for j =1:3
        suma1 = suma1 + betag(j+1)*p1(x1(i-j), x2(i-j));
        suma2 = suma2 + betag(j+1)*p2(x1(i-j), x2(i-j));
    end
    x1(i) = x1(i-1) + skok*suma1 + skok*betag(1)*f1;
    x2(i) = x2(i-1) + skok*suma2 + skok*betag(1)*f2;
    err1(i) = -(19/270)*(x10 - x1(i));
    err2(i) = -(19/270)*(x20 - x2(i));
end
t = toc;
end
```

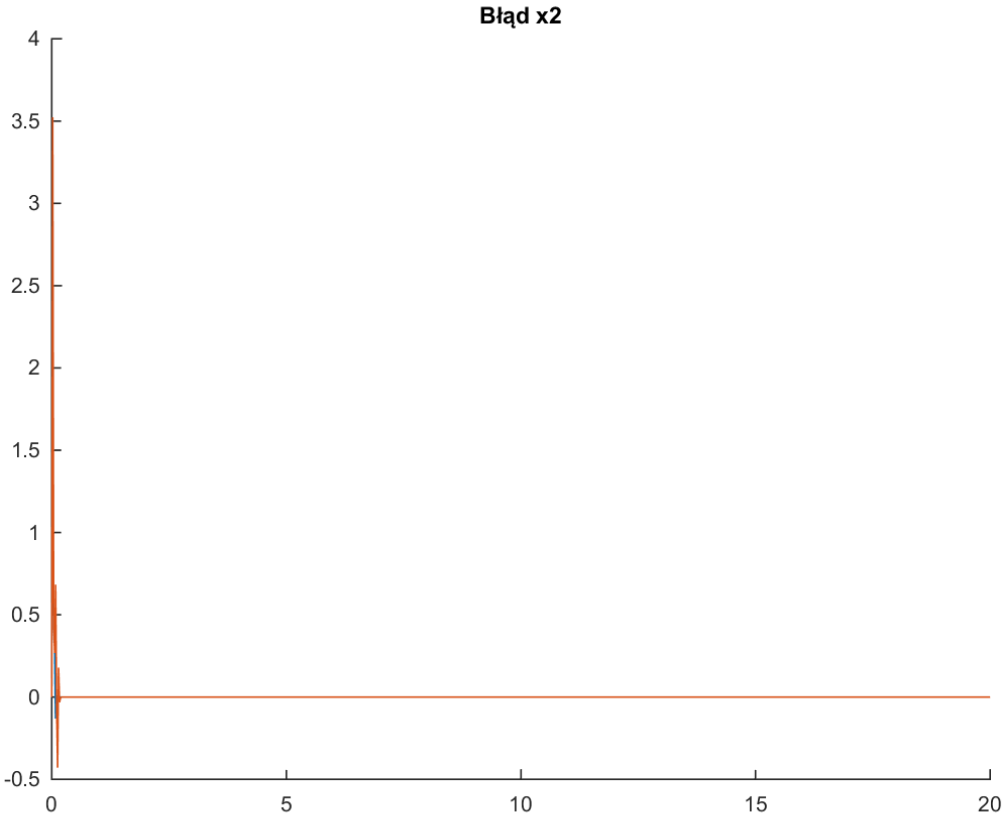
a) $x_1(0) = 8, \quad x_2(0) = 7$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.02



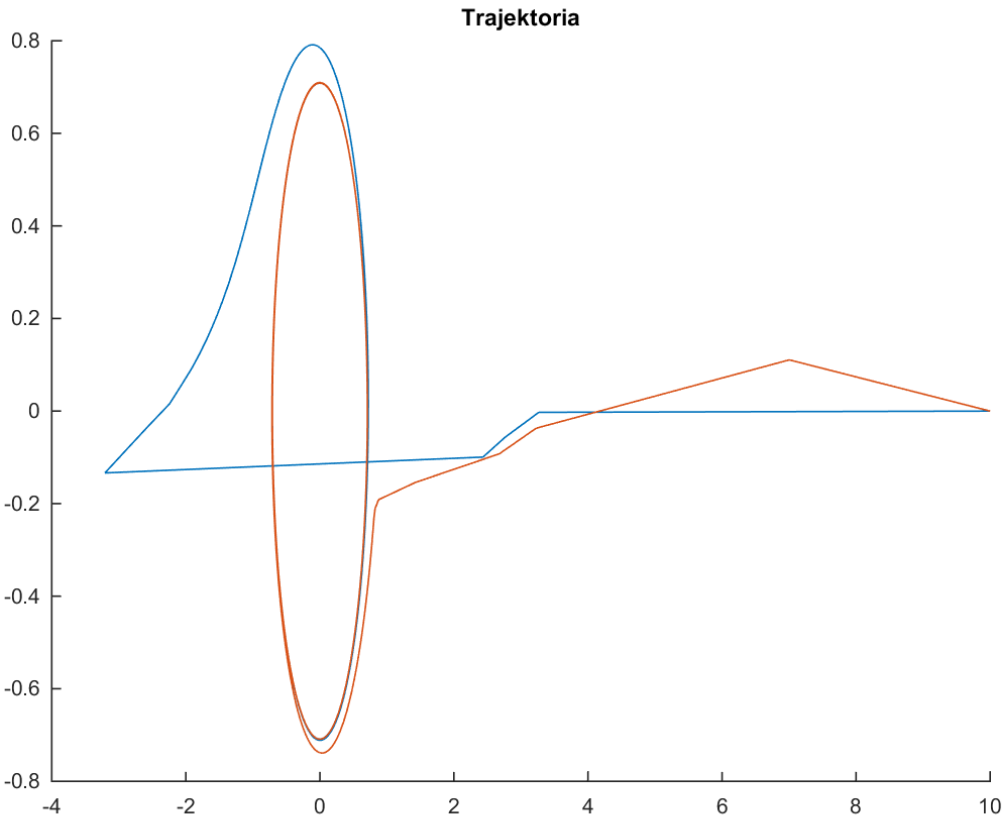


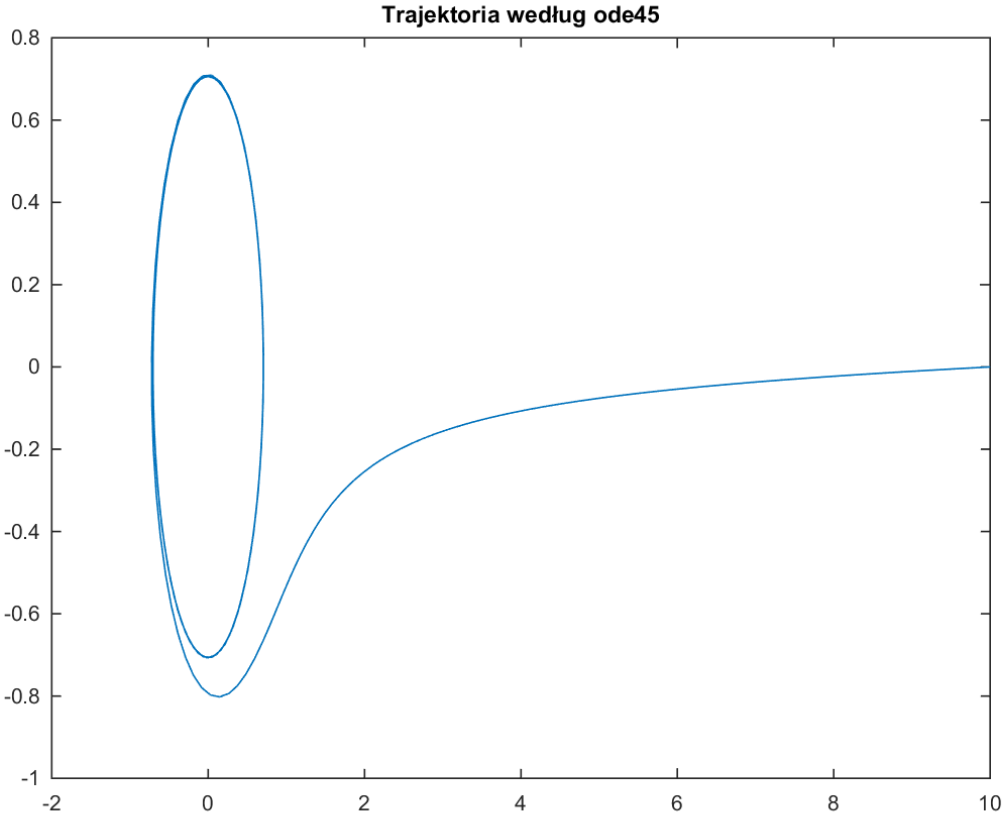

Błąd x1
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
5
10
15
20

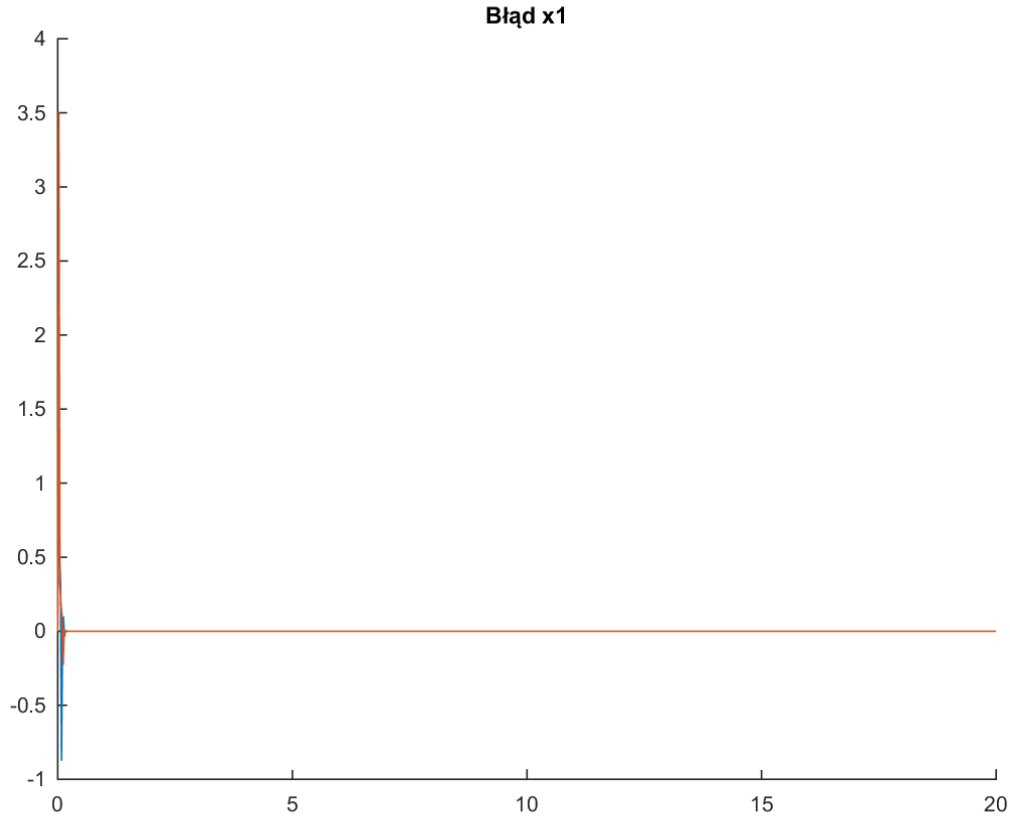

Błąd x2
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
5
10
15
20

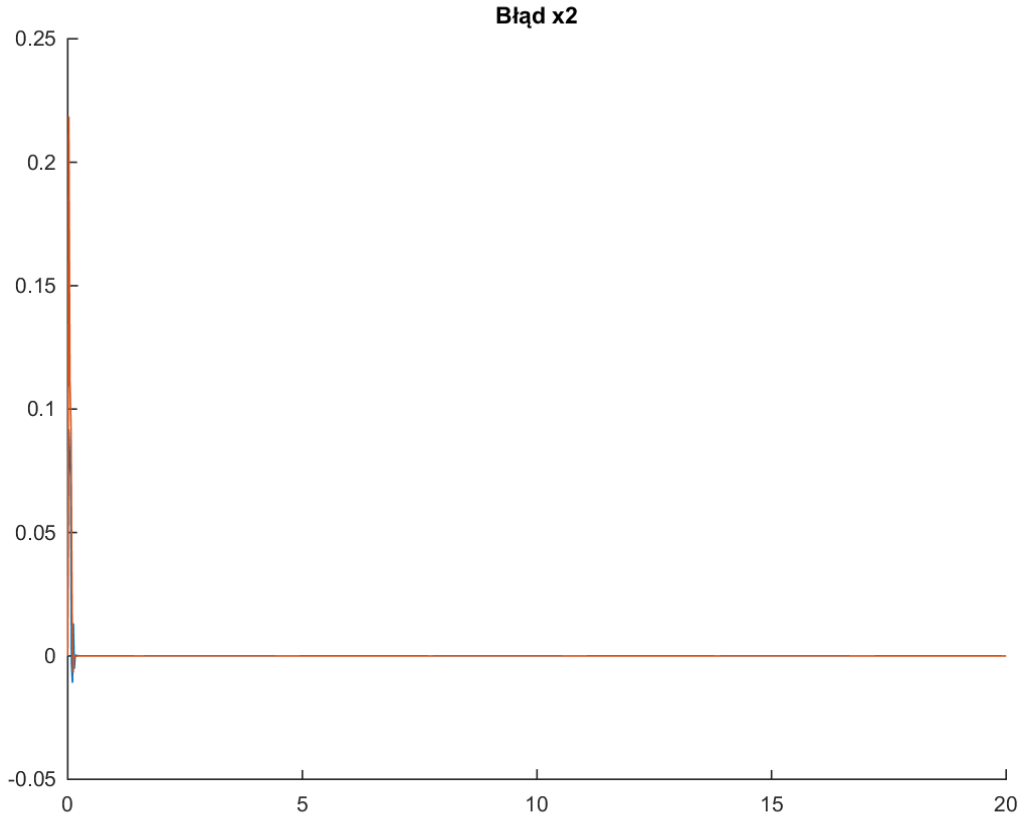
b) $x_1(0) = 0, \quad x_2(0) = 0.3$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.02



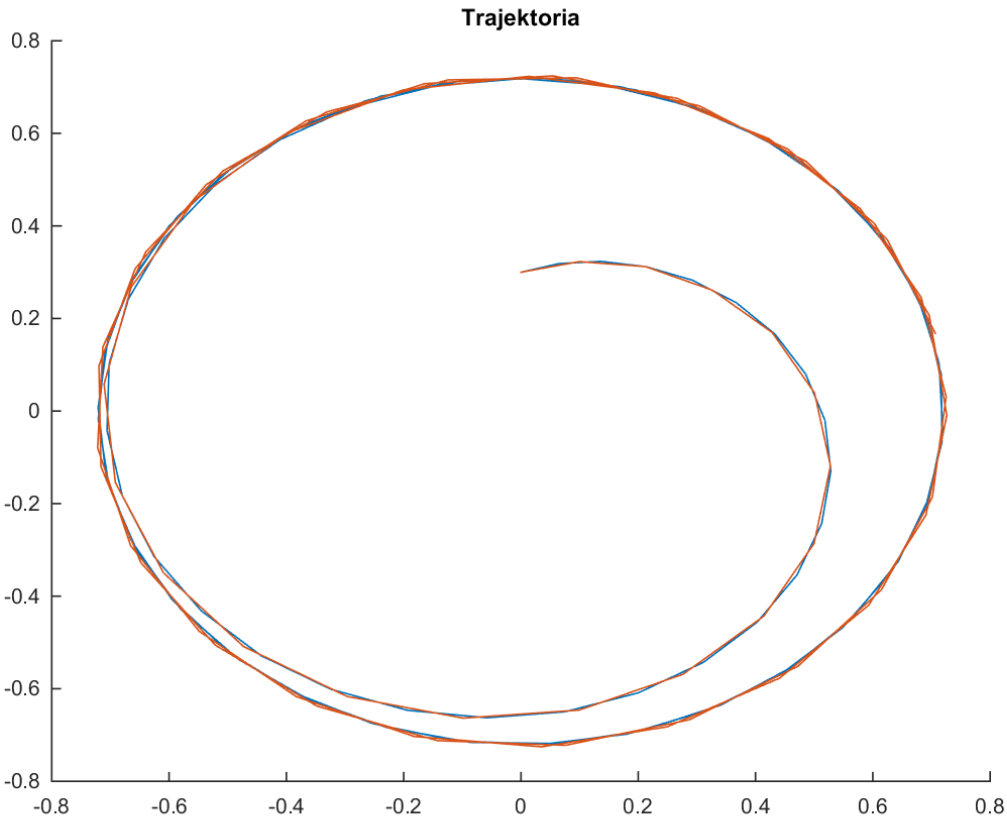


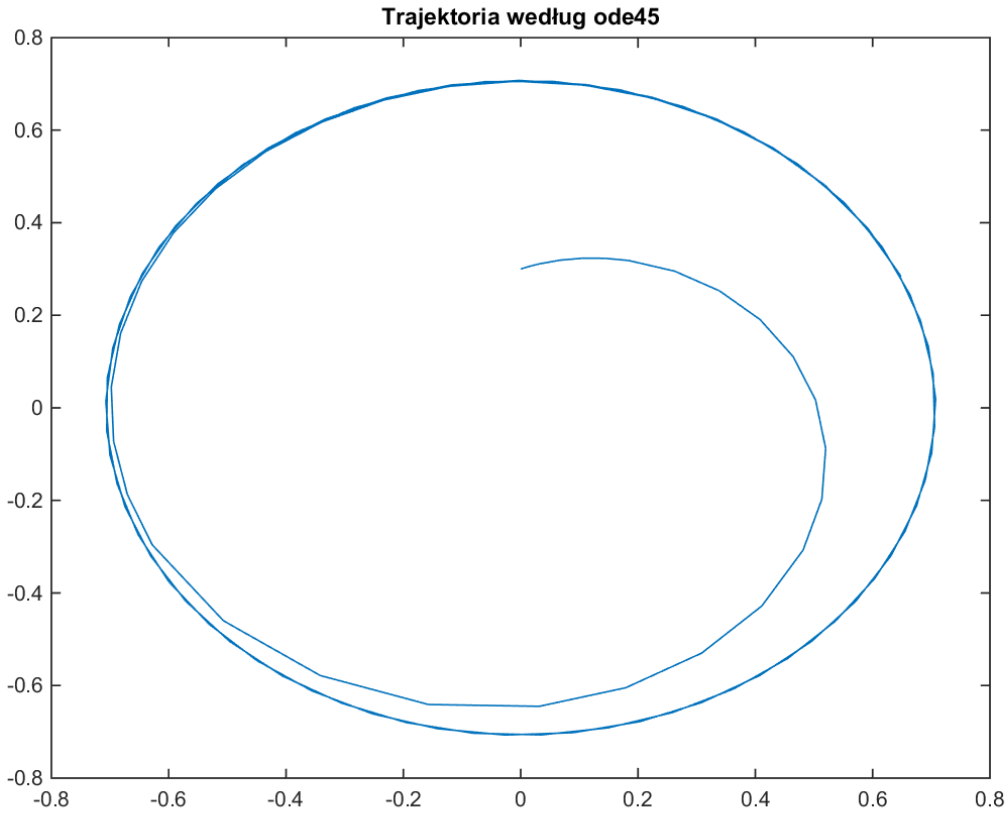

Błąd x1
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
-1
0
5
10
15
20

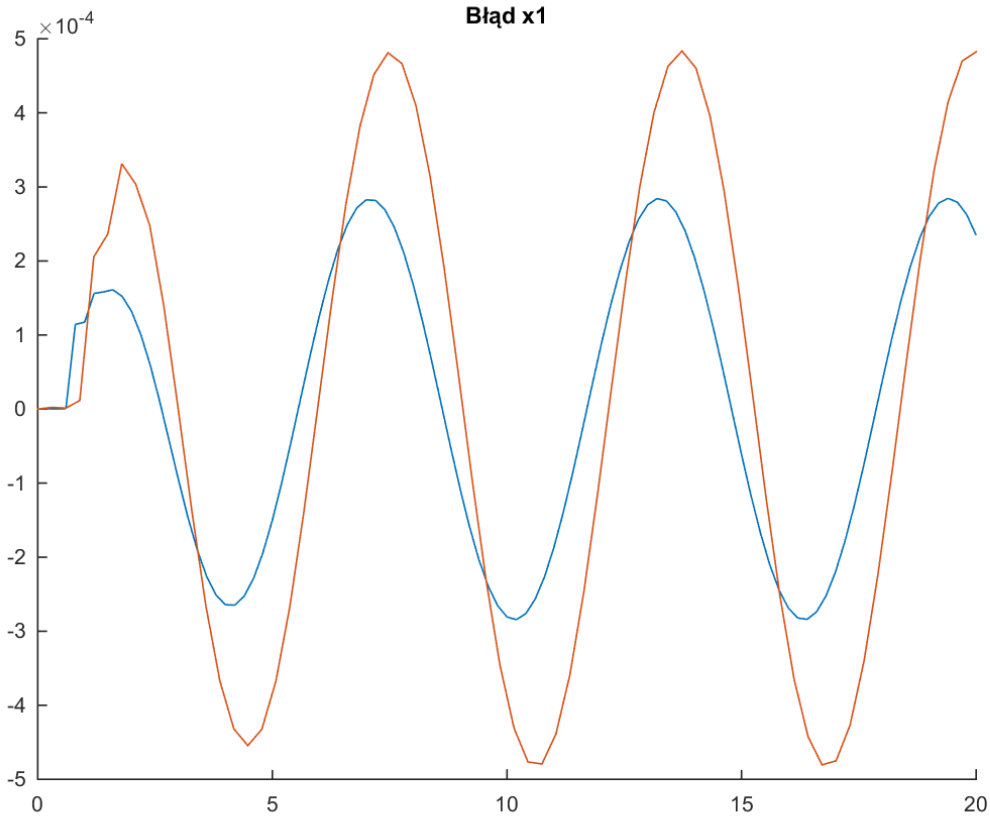

Błąd x2
0.25
0.2
0.15
0.1
0.05
0
-0.05
0
5
10
15
20

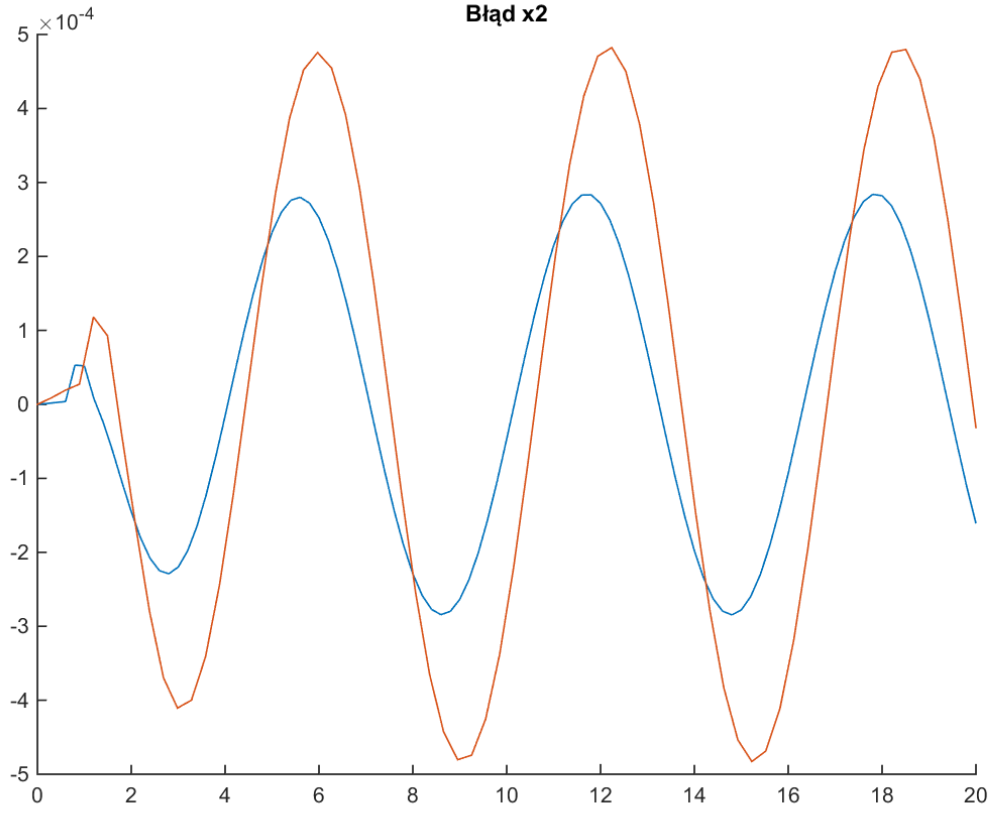
c) $x_1(0) = 10, \quad x_2(0) = 0$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2



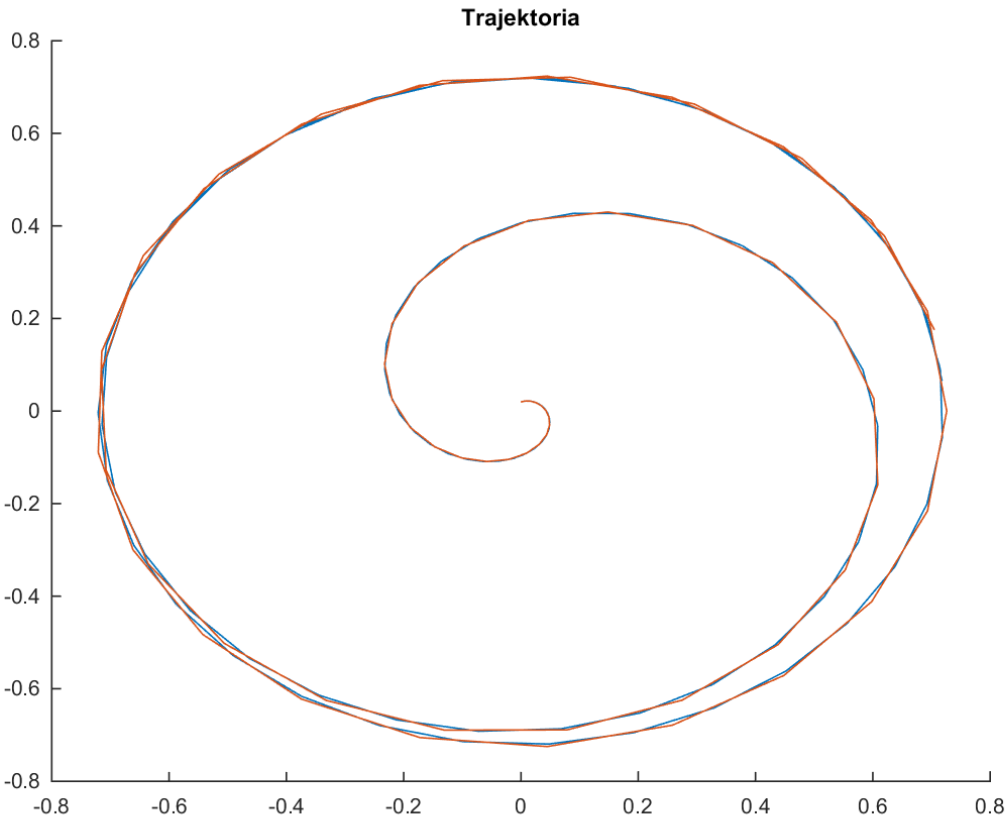


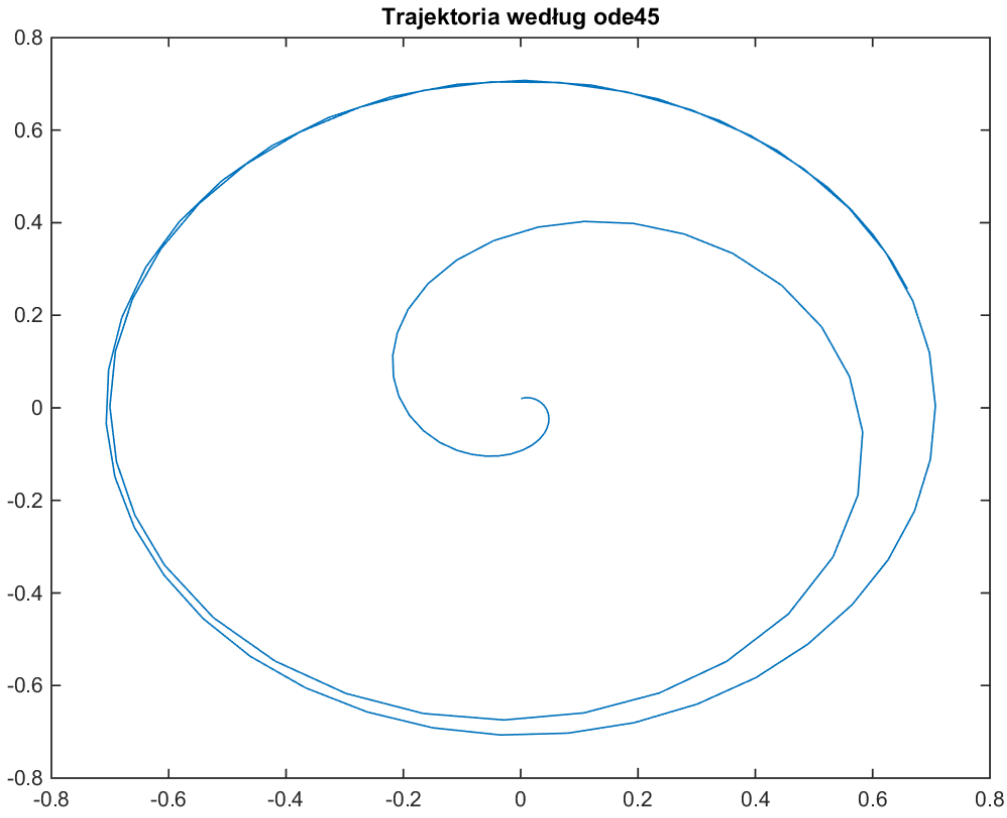

Błąd x1
×10^-4
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
5
10
15
20

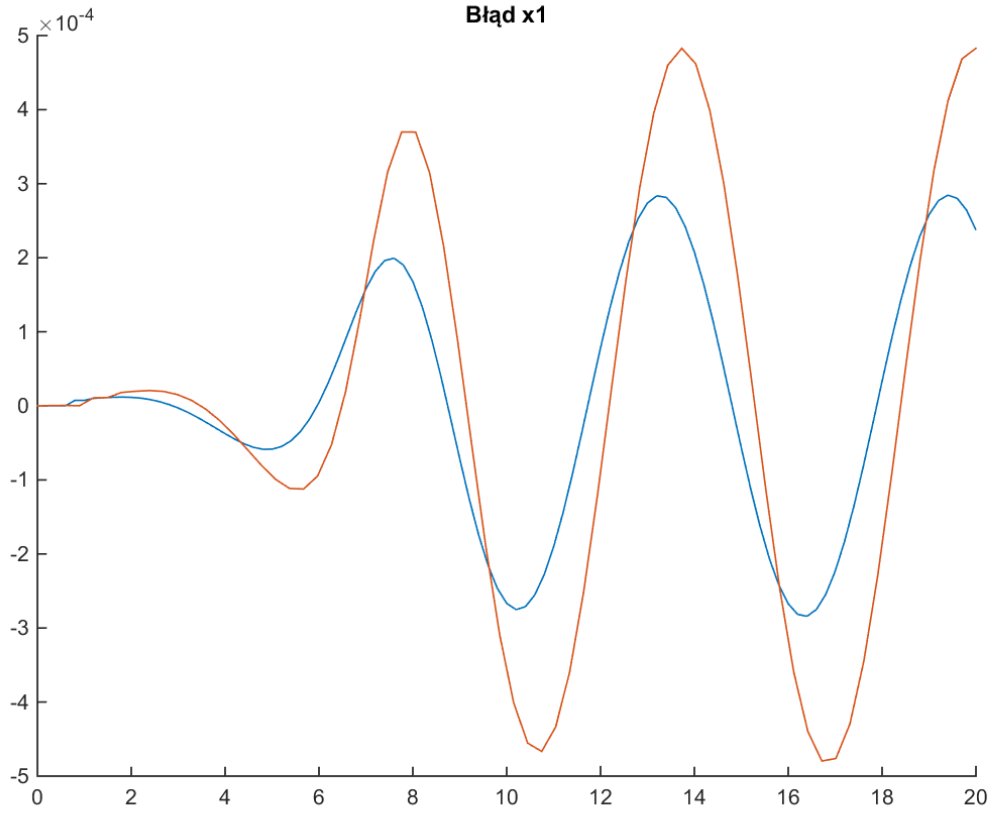

Błąd x2
×10^-4
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
2
4
6
8
10
12
14
16
18
20

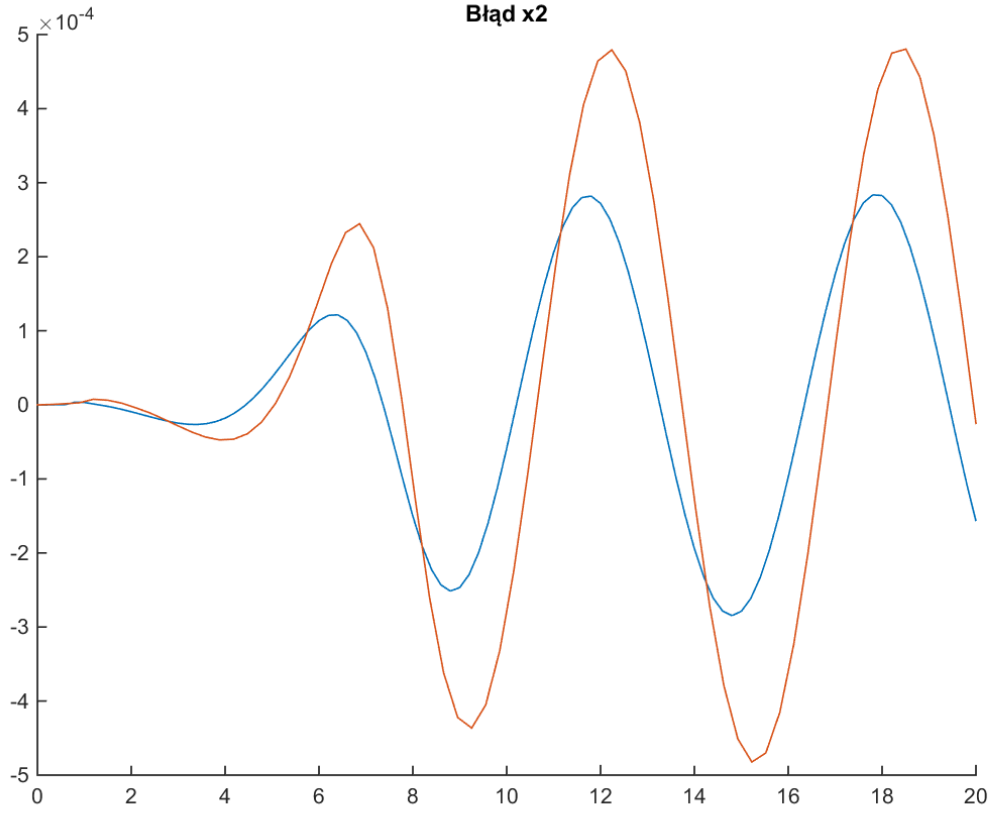
d) $x_1(0) = 0.001, \quad x_2(0) = 0.02$

W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2






Błąd x1
×10^-4
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
2
4
6
8
10
12
14
16
18
20


Błąd x2
×10^-4
5
4
3
2
1
0
-1
-2
-3
-4
-5
0
2
4
6
8
10
12
14
16
18
20

Porównanie czasów wykonania (w sekundach):

Podpunkt/Metoda	RK4	PK
a)	0,0210	0,0132
b)	0,0189	0,0130
c)	0,0020	0,0015
d)	0,0019	0,0013

Wnioski:

Jak widać w każdym przypadku metody dawały rezultaty podobne do rezultatów uzyskanych za pomocą polecenia *ode45().* Jednakże we wszystkich przypadkach metoda predyktor-korektor szybciej znajdowała wynik. Ponadto za każdym razem generowała ona mniejsze błędy: w przypadku pierwszych dwóch na samym początku są większe, lecz należy pamiętać że pierwsze cztery punkty w metodzie PK są obliczane za pomocą metody RK4, potem błędy szybko się stabilizują i osiągają bardzo małe wartości. W dwóch ostatnich podpunktach błędy są mniej więcej o rząd wielkości mniejsze. Jednakże metoda PK wymaga większego nakładu obliczeń niż RK4, ze względu na podwójną ewaluację wartości funkcji. Można ją zredukować badając zmienność funkcji i wprowadzając zmienny krok (większy dla mało zmiennych przedziałów i mniejszy dla przedziałów o dużej zmienności).