**MNUM – Projekt 4.15**

**Zadanie 1**  
Ruch punktu opisany jest równaniami:

Obliczyć przebieg trajektorii ruchu na przedziale 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:

Współczynnik jest pochodną rozwiązania w punkcie . Wartość wyznaczamy jak w zmodyfikowanej metodzie Eulera - jako pochodną rozwiązania wyznaczanego zwykłą metodą Eulera w punkcie (środkowym przedziału). Następnie, wartość wyznaczamy podobnie, jak , ale tym razem w punkcie , tzn. startując z początku przedziału z nachyleniem . W końcu, z nachyleniem tej stycznej startujemy z punktu początkowego do punktu , tzn. wyznaczamy pochodną rozwiązania w punkcie . 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:

Gdzie:

– nowy punkt uzyskany w kroku o długości

– nowy punkt wyznaczony przez dwa dodatkowe kroki o długościach

– 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

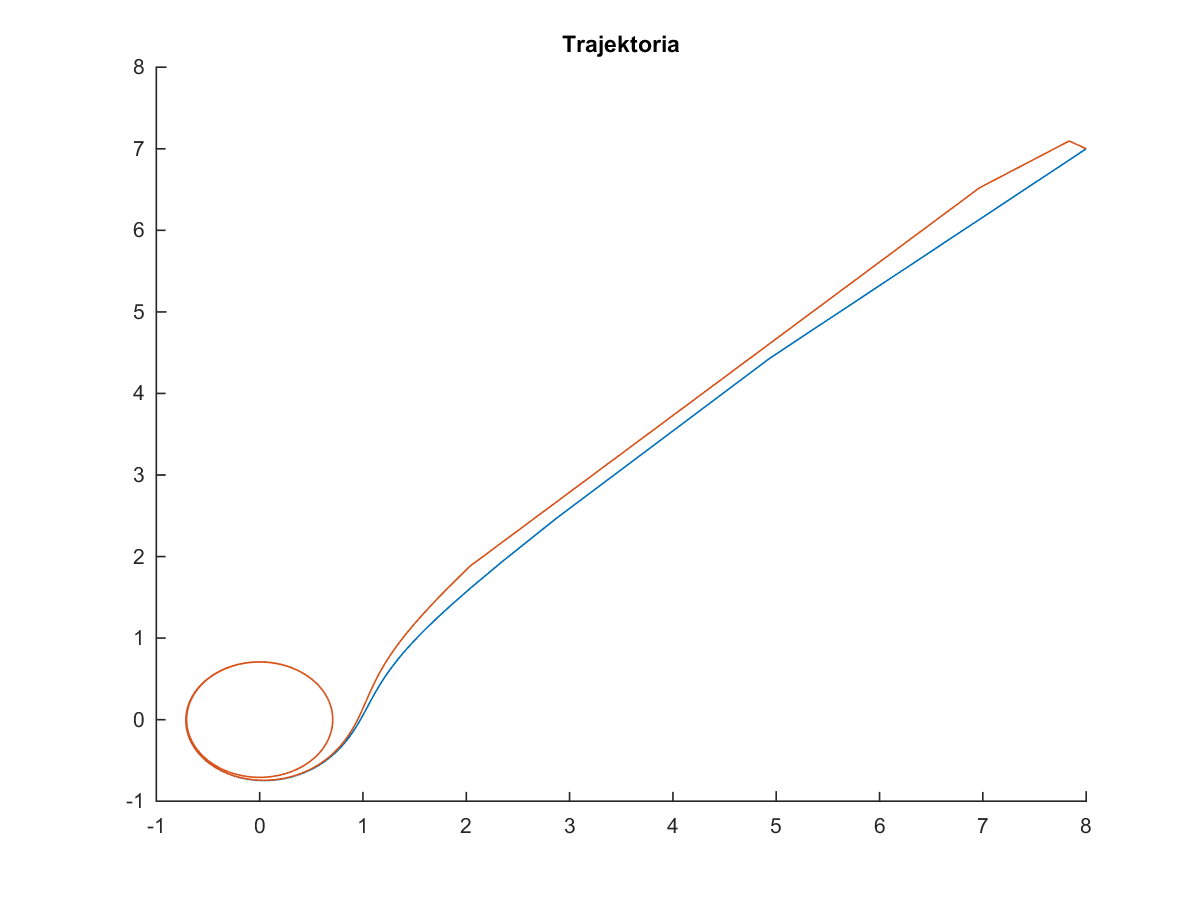
err1 =(16/15) \* abs(x1h - x1);

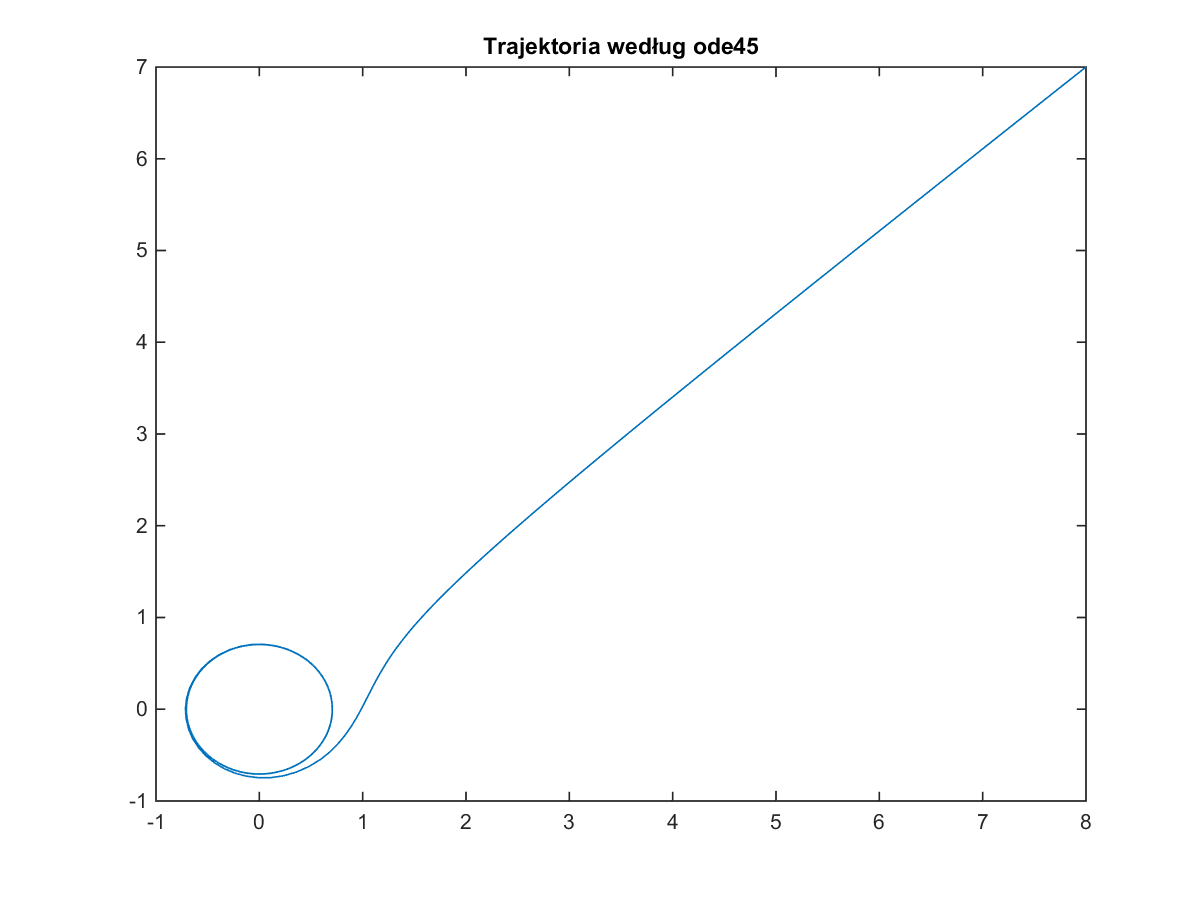
err2 =(16/15) \* abs(x2h - x2);

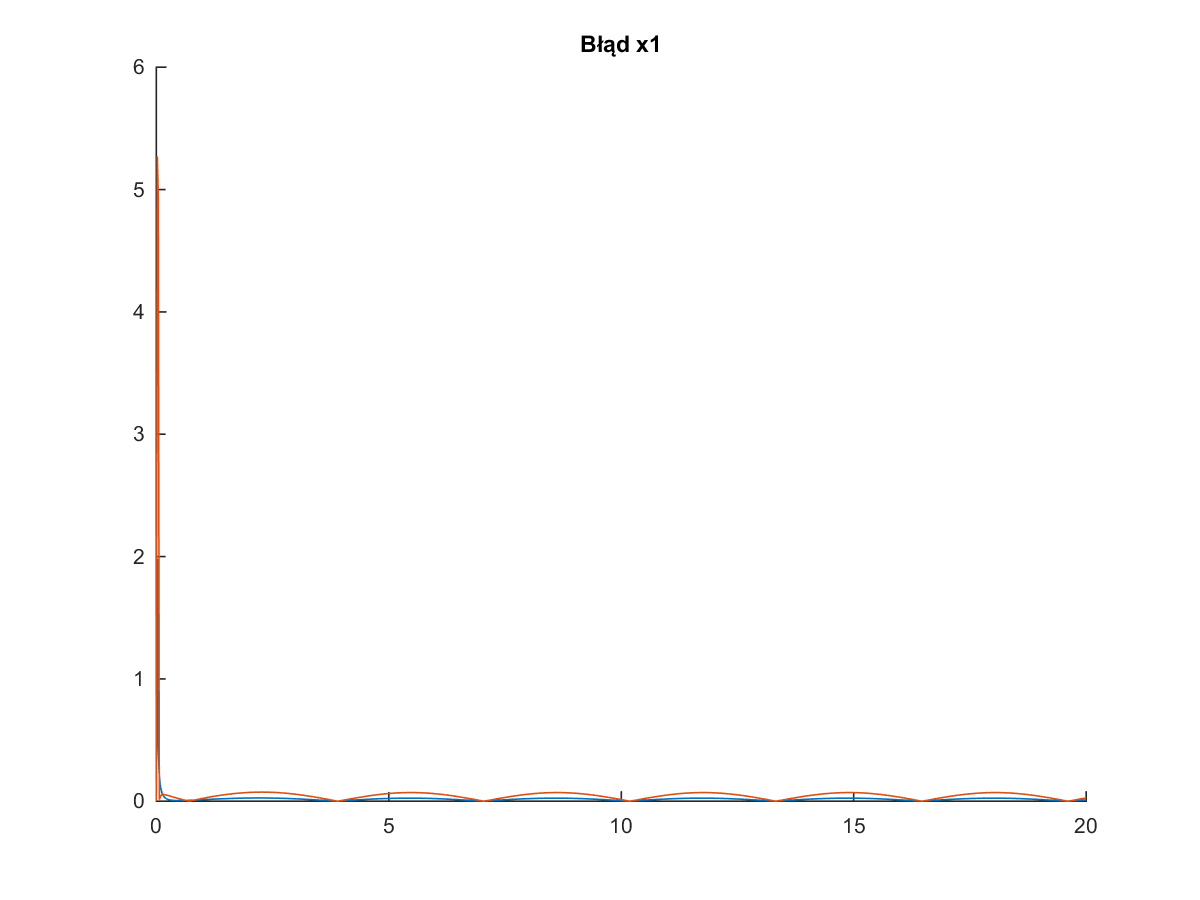
t = toc;

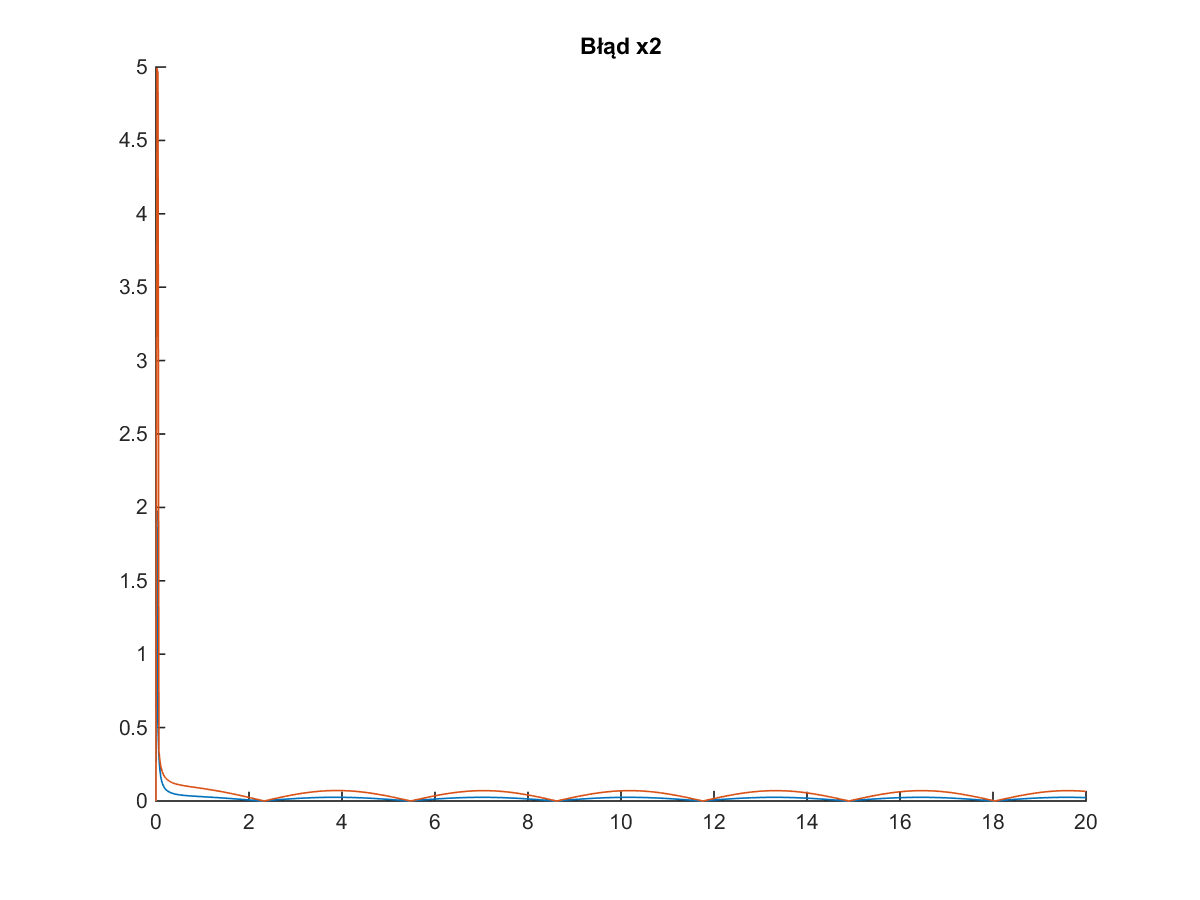
end

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

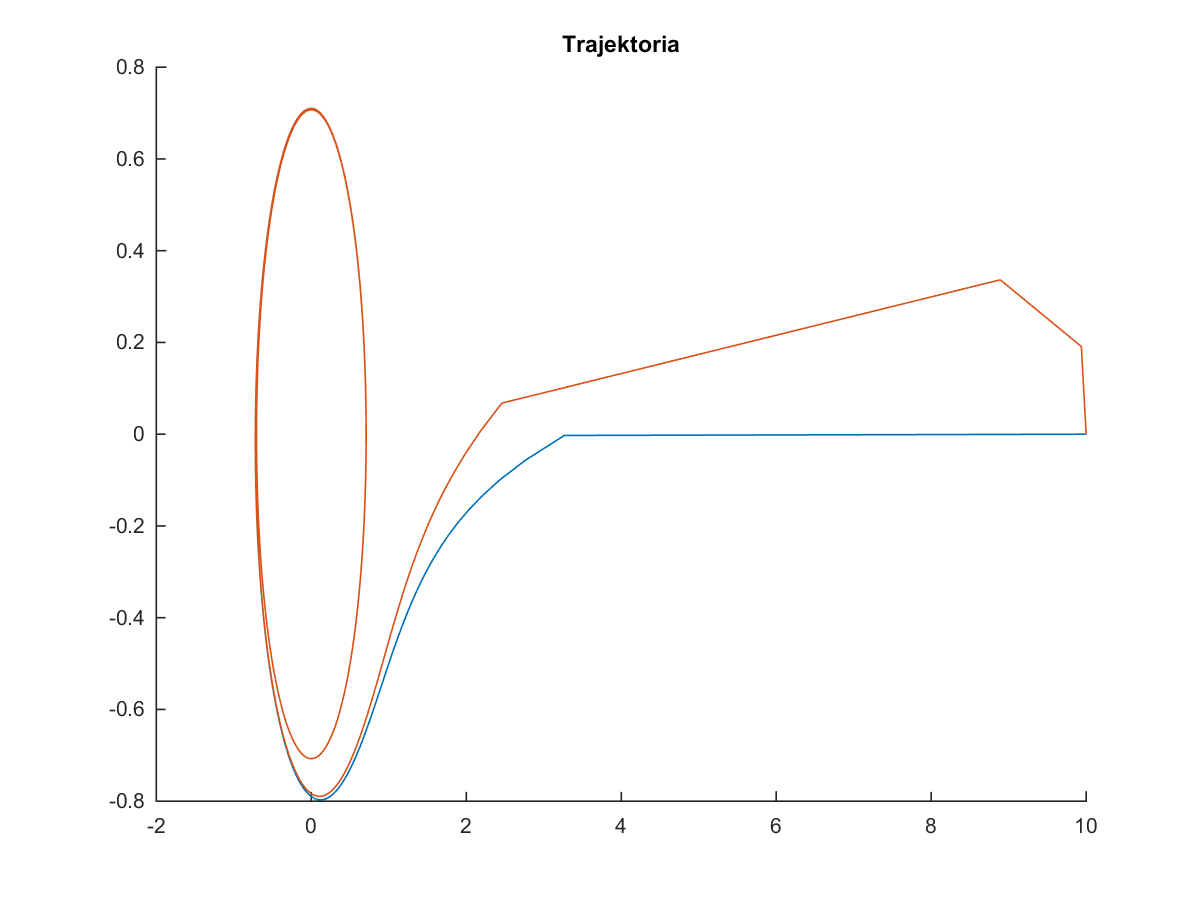


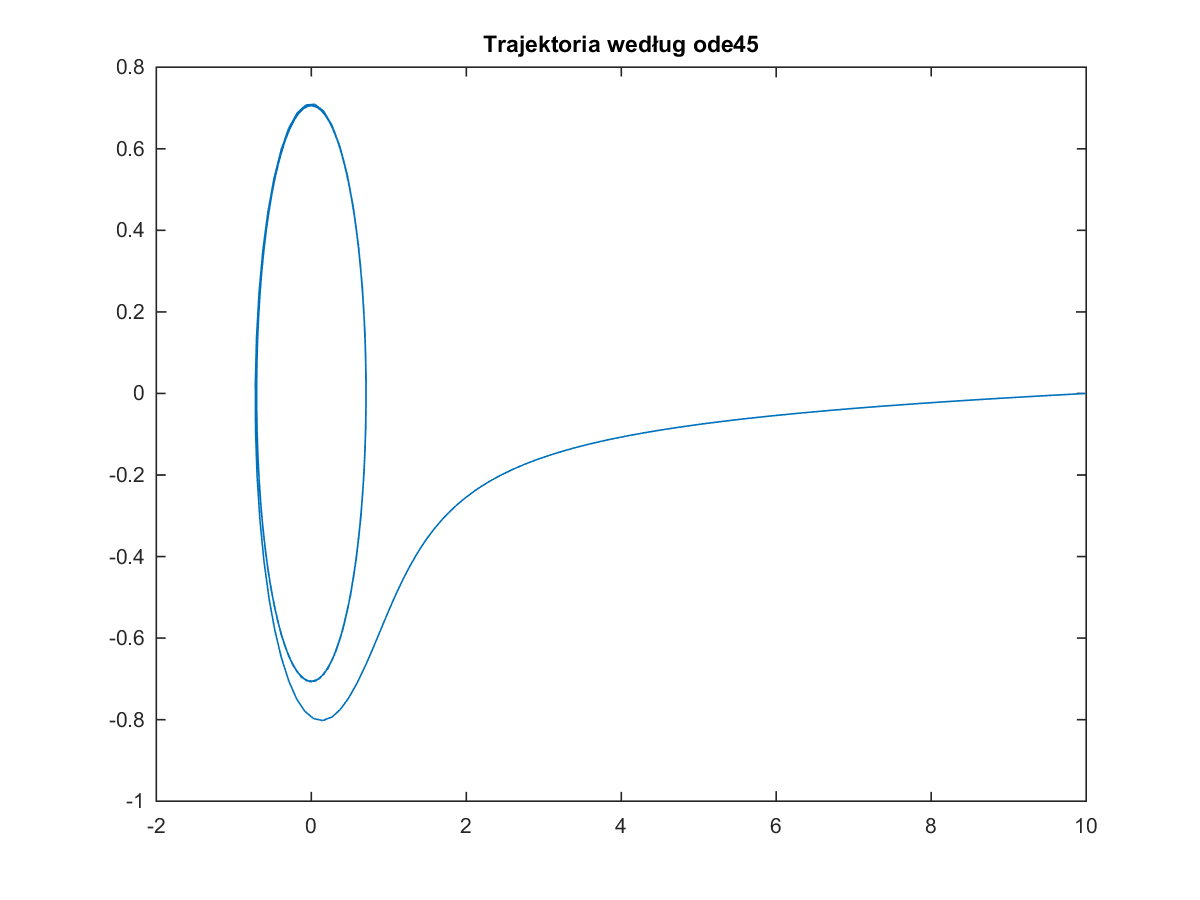


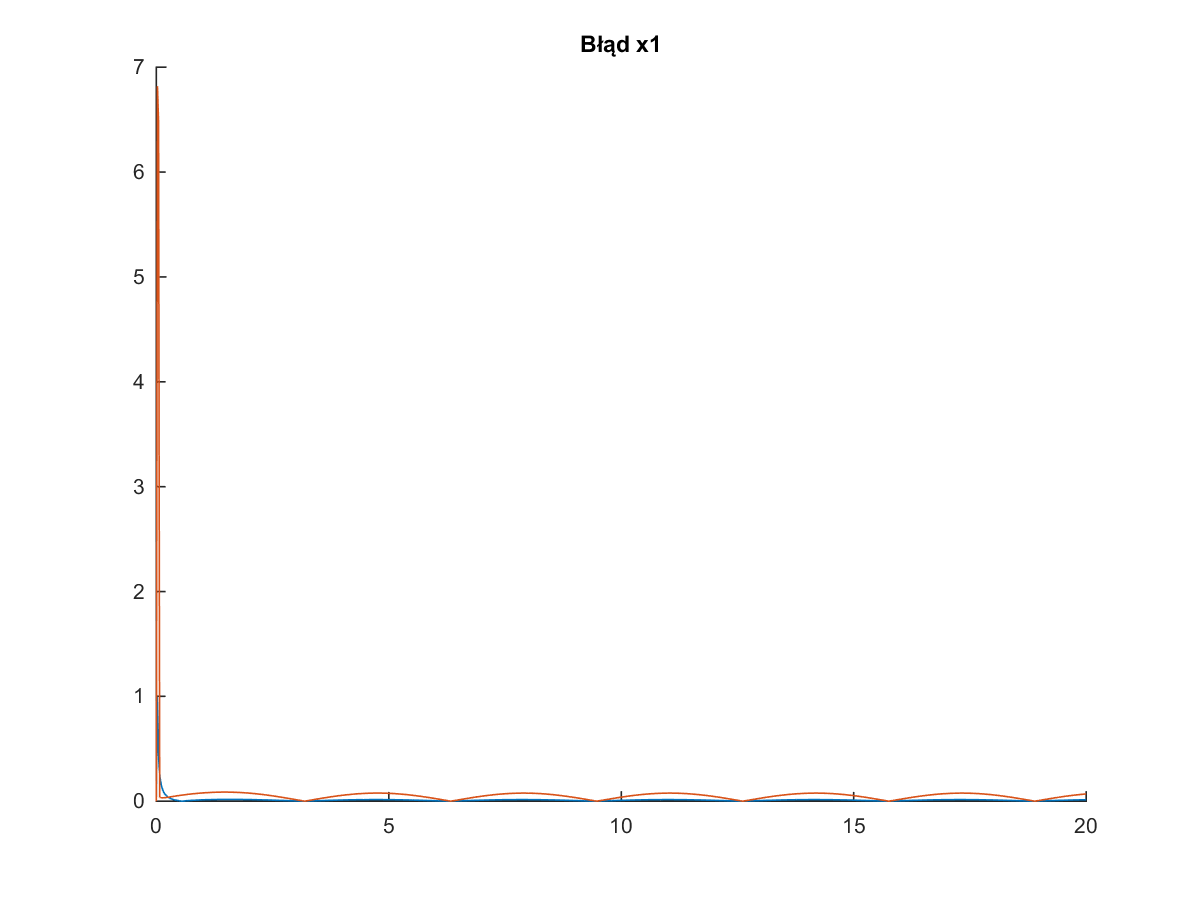


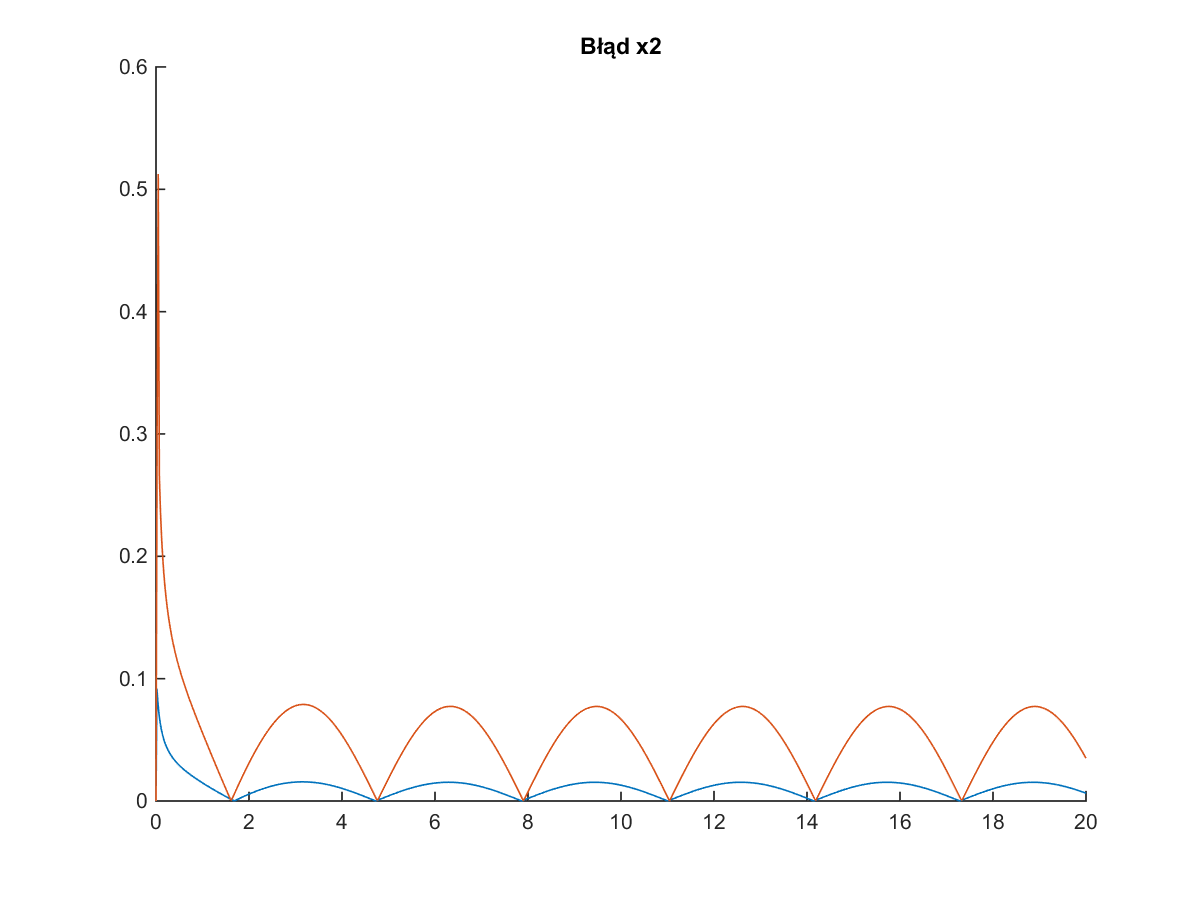


W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.02

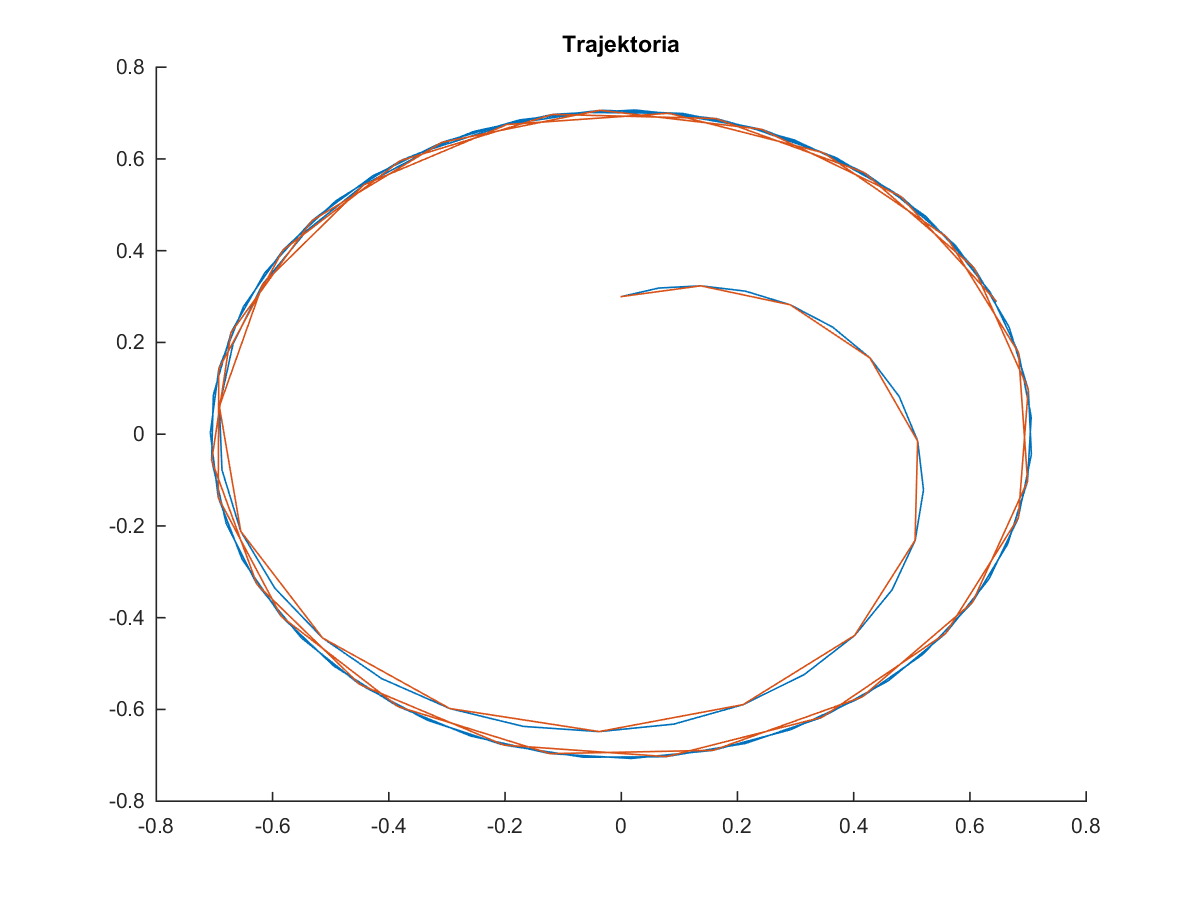


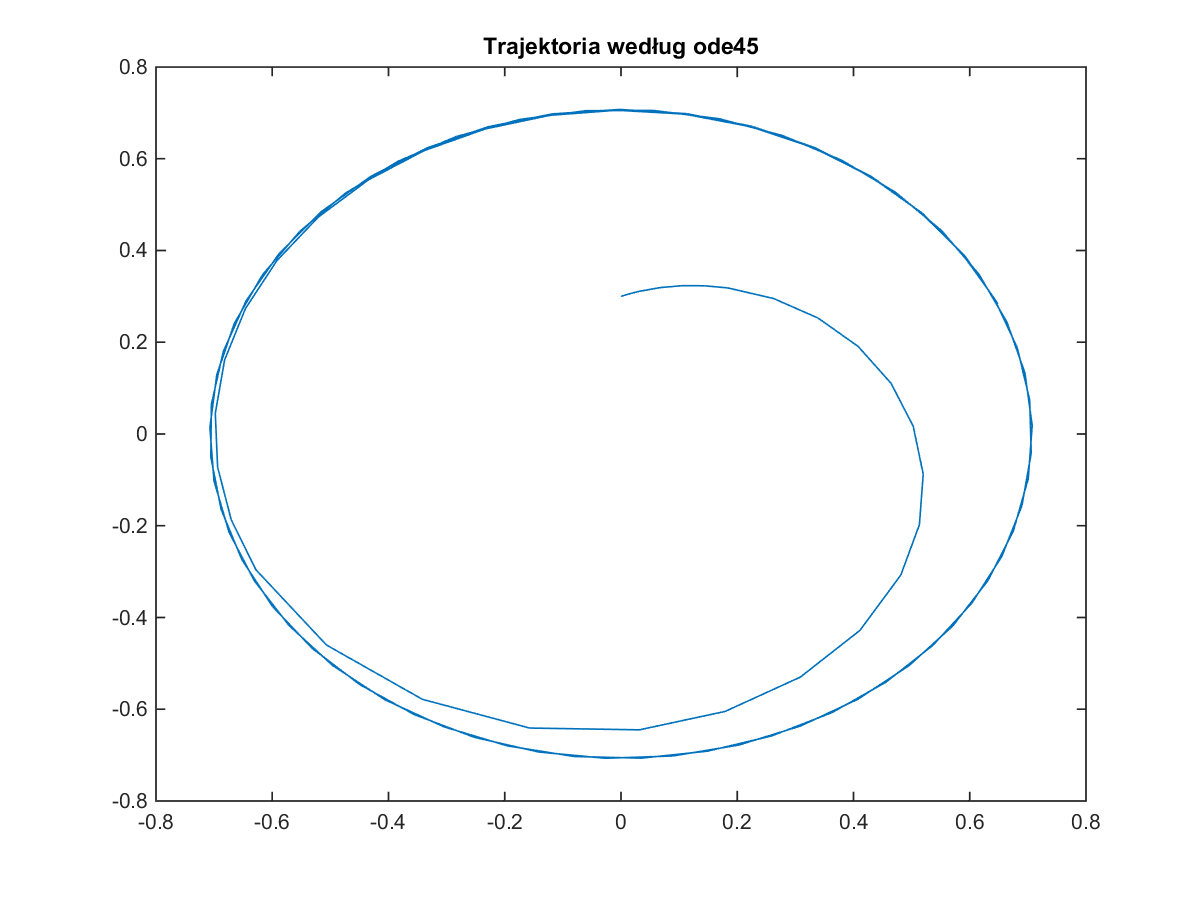


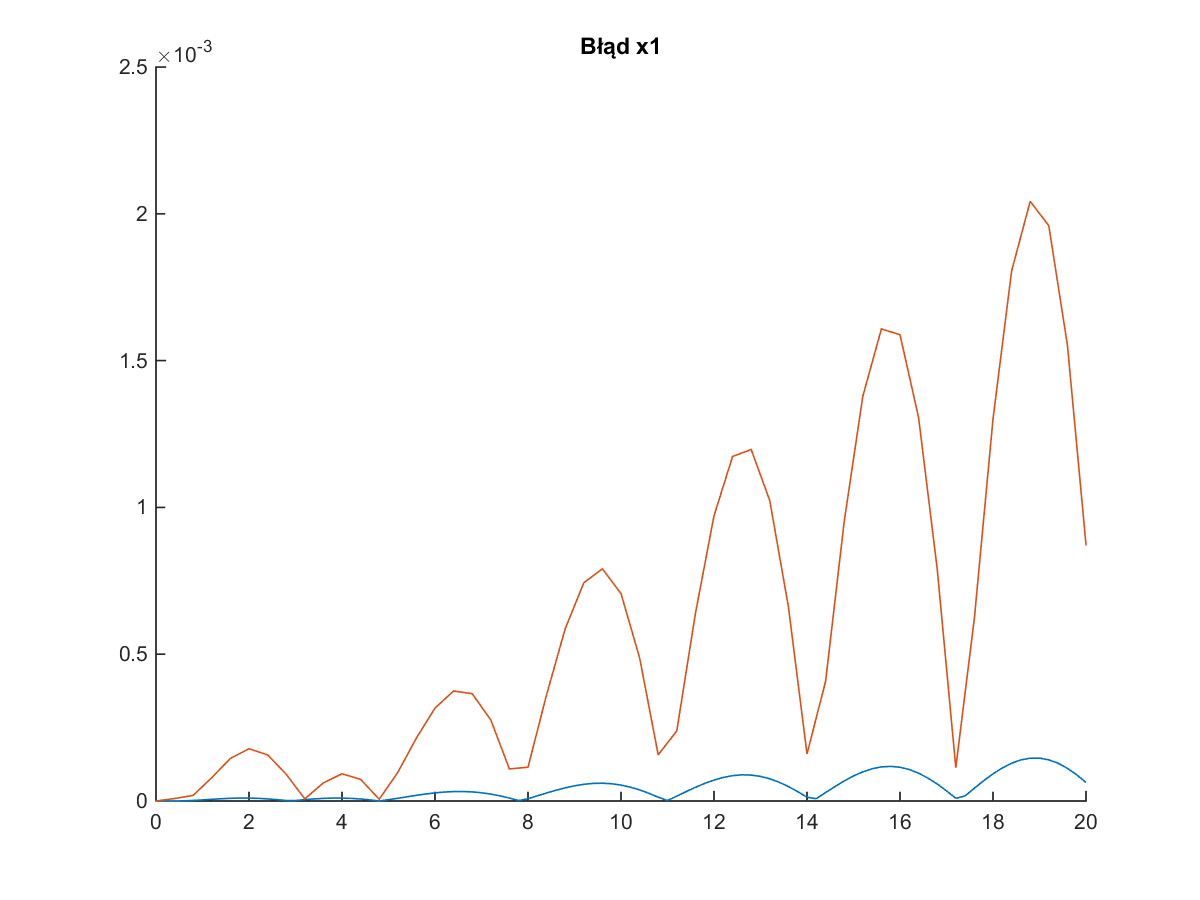


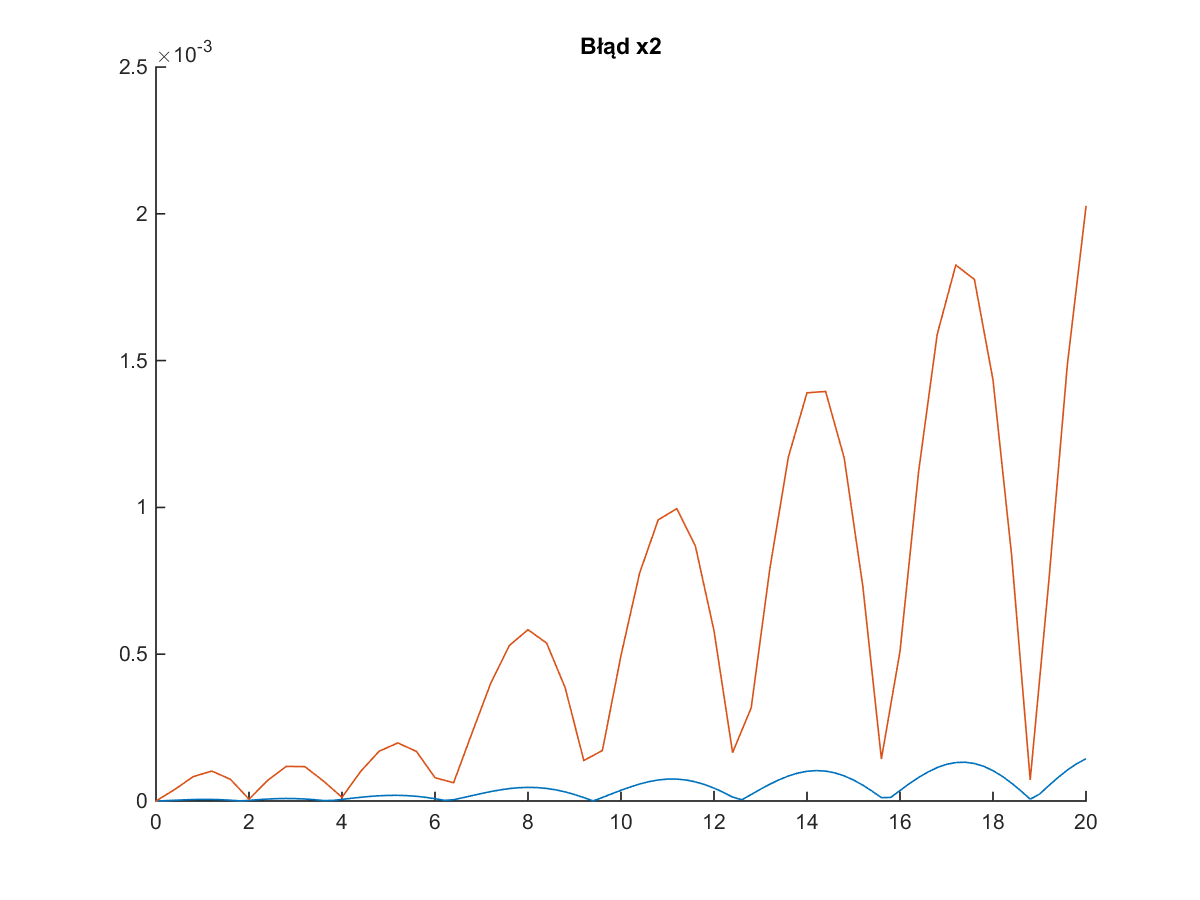


W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2

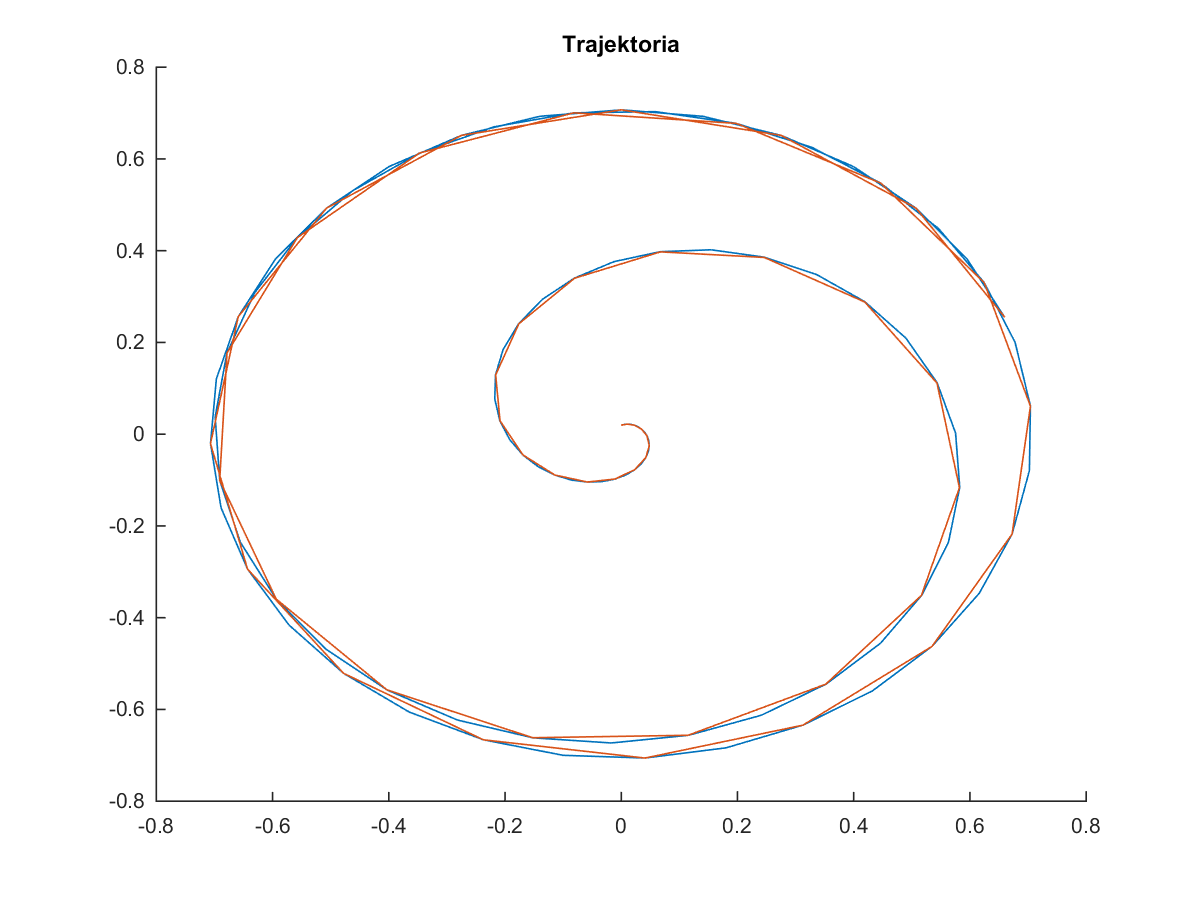


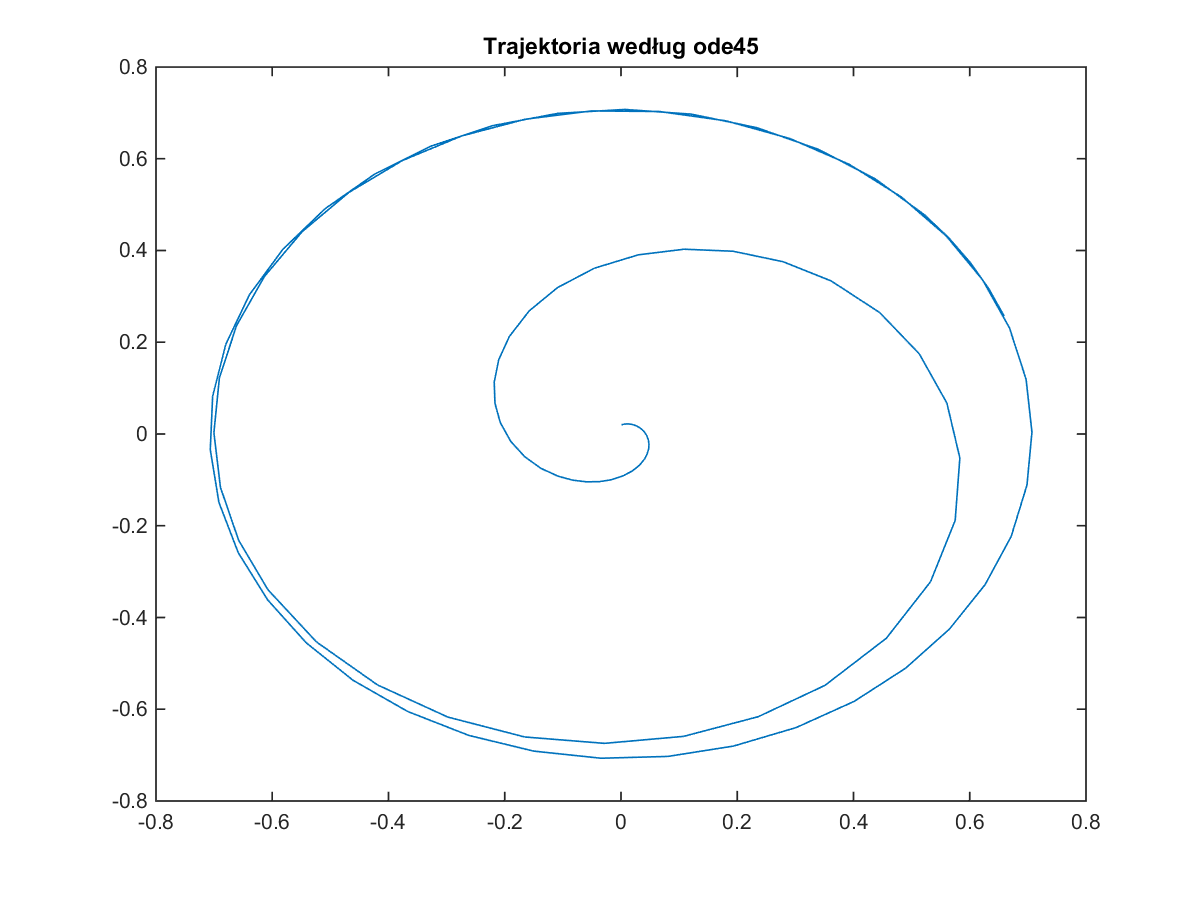


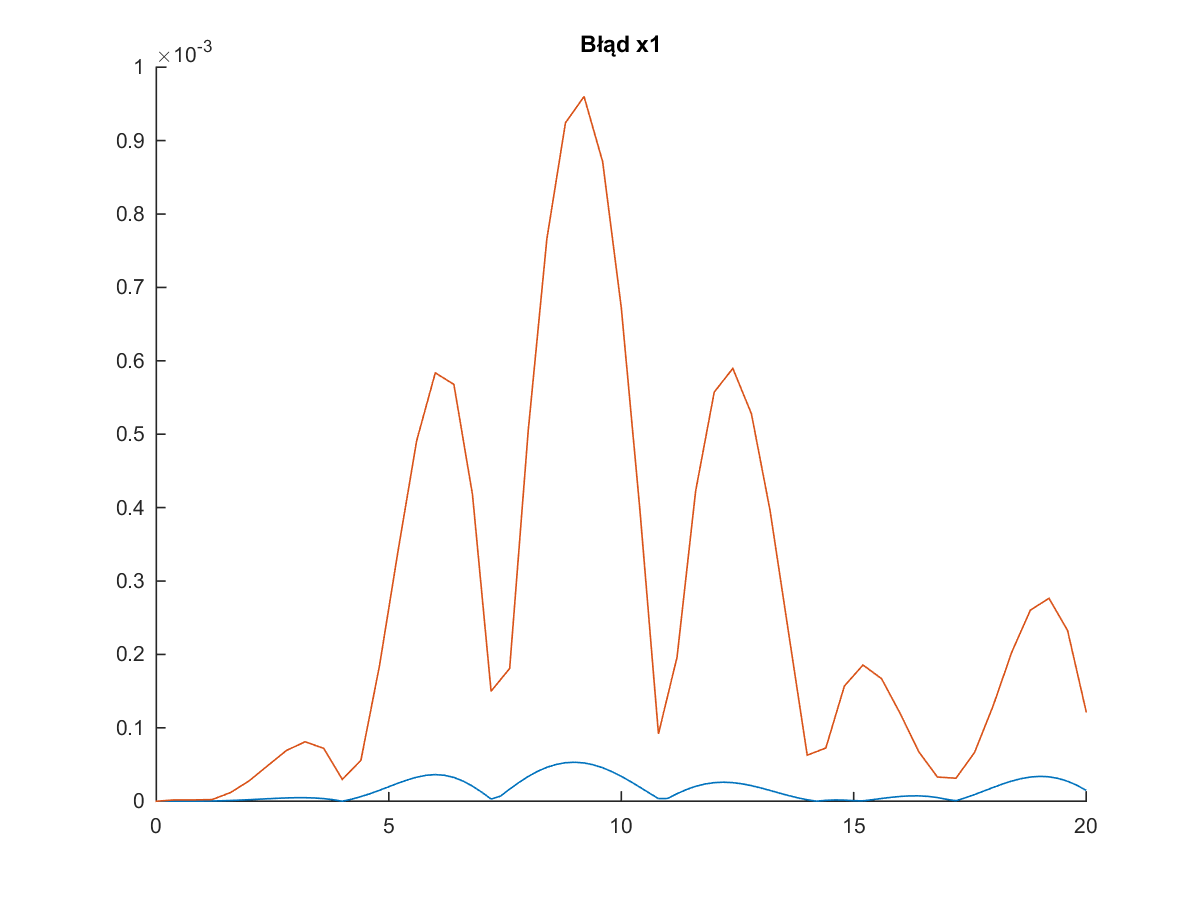


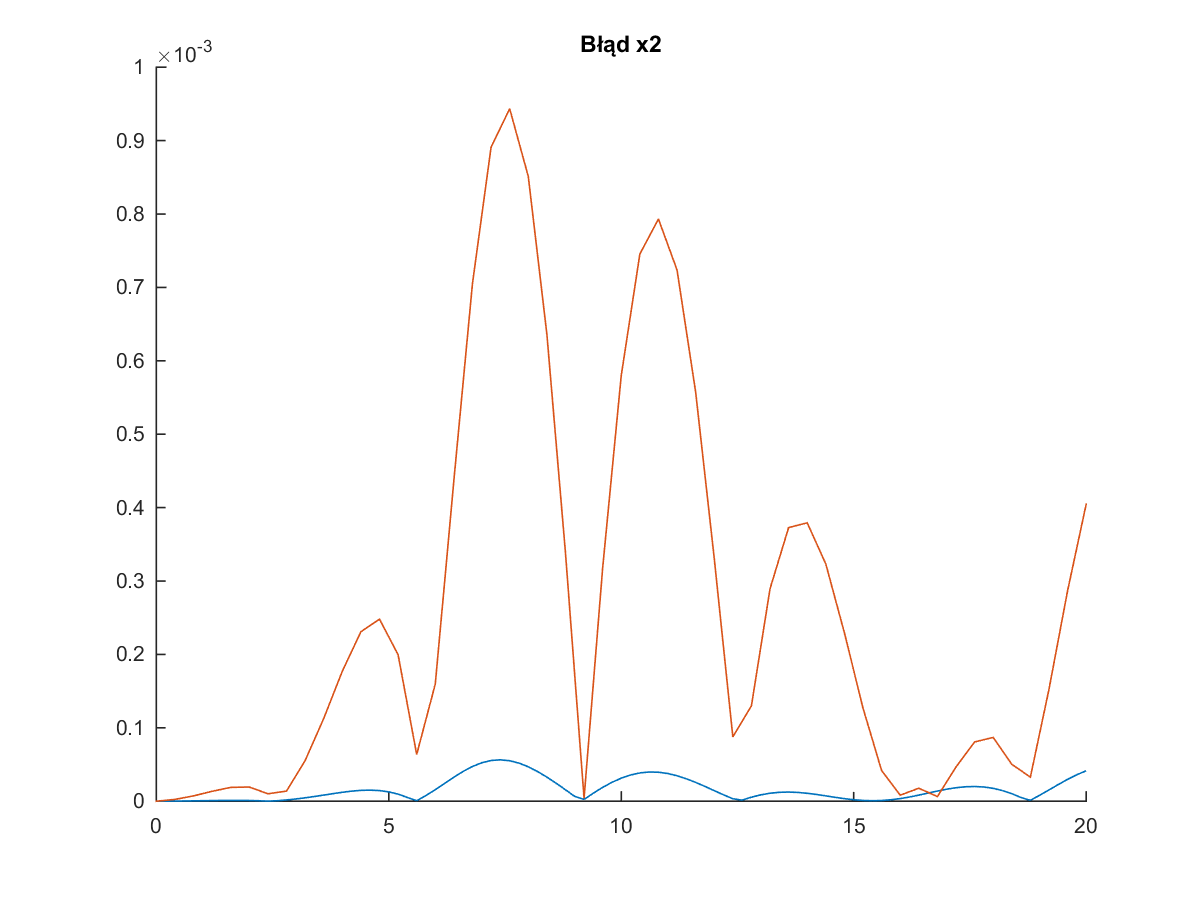


W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2









**Metoda predyktor-korektor Adamsa czwartego rzędu:**

Metody Adamsa:

Równanie różniczkowe

Równoważne jest równaniu całkowemu

Metody Adamsa dostajemy, rozważając to równanie na przedziale :

Metody jawne:

Funkcję podcałkową przybliżamy wielomianem interpolacyjnym stopnia co najwyżej k-1 opartym na węzłach . Przyjmując przybliżenie Lagrange’a mamy:

Gdzie to wielomiany Lagrange’a:

Stąd po scałkowaniu, przy założeniu otrzymujemy:

Wartości są stablicowane.

Metody niejawne:

Funkcję podcałkową przybliżamy wielomianem interpolacyjnym stopnia co najwyżej k opartym na węzłach , z wartościami rozwiązania . Następnie postępując tak jak w przypadku metody jawnej otrzymujemy

Wartości są stablicowane.

Metoda predyktor-korektor:

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

P:

E:

K:

E:

Dla metody PK oszacowanie błędu:

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));

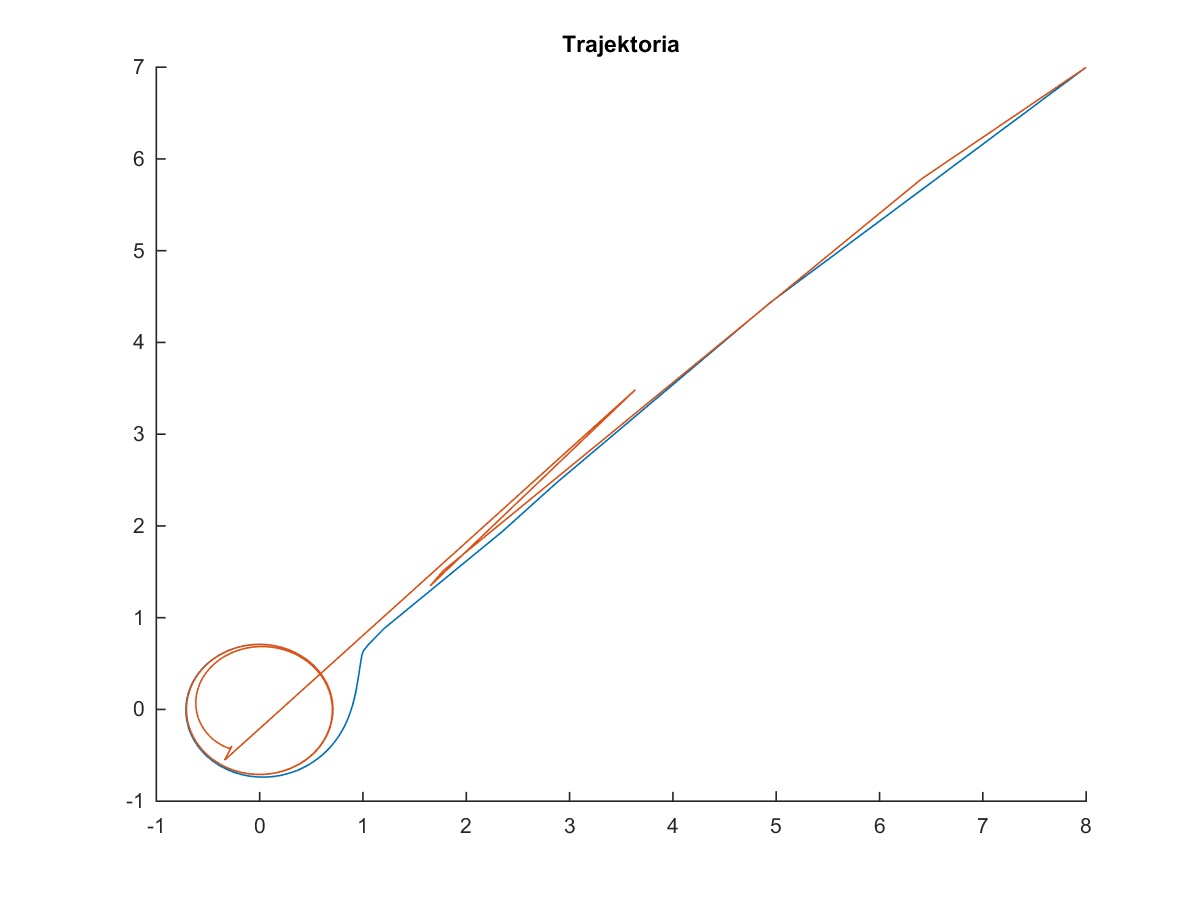
err2(i) = -(19/270)\*(x20 - x2(i));

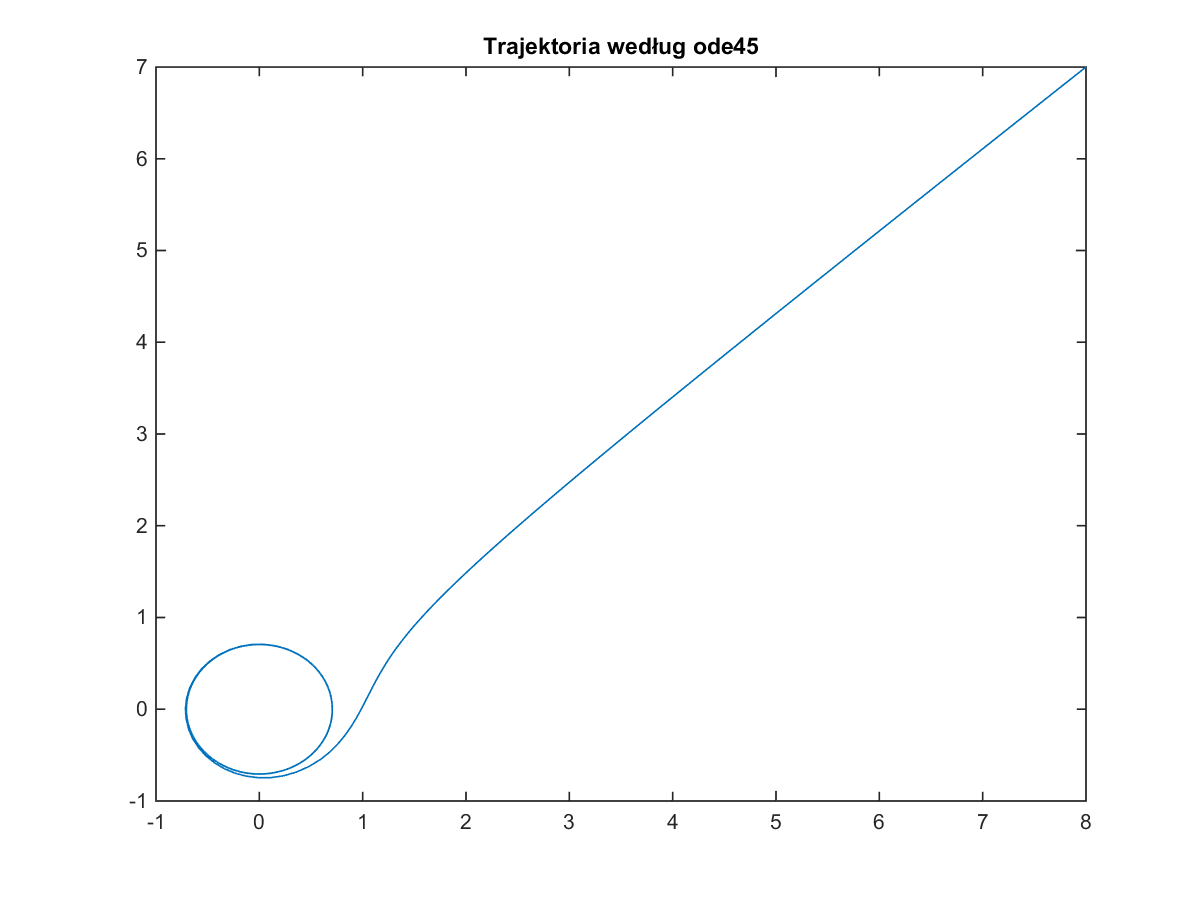
end

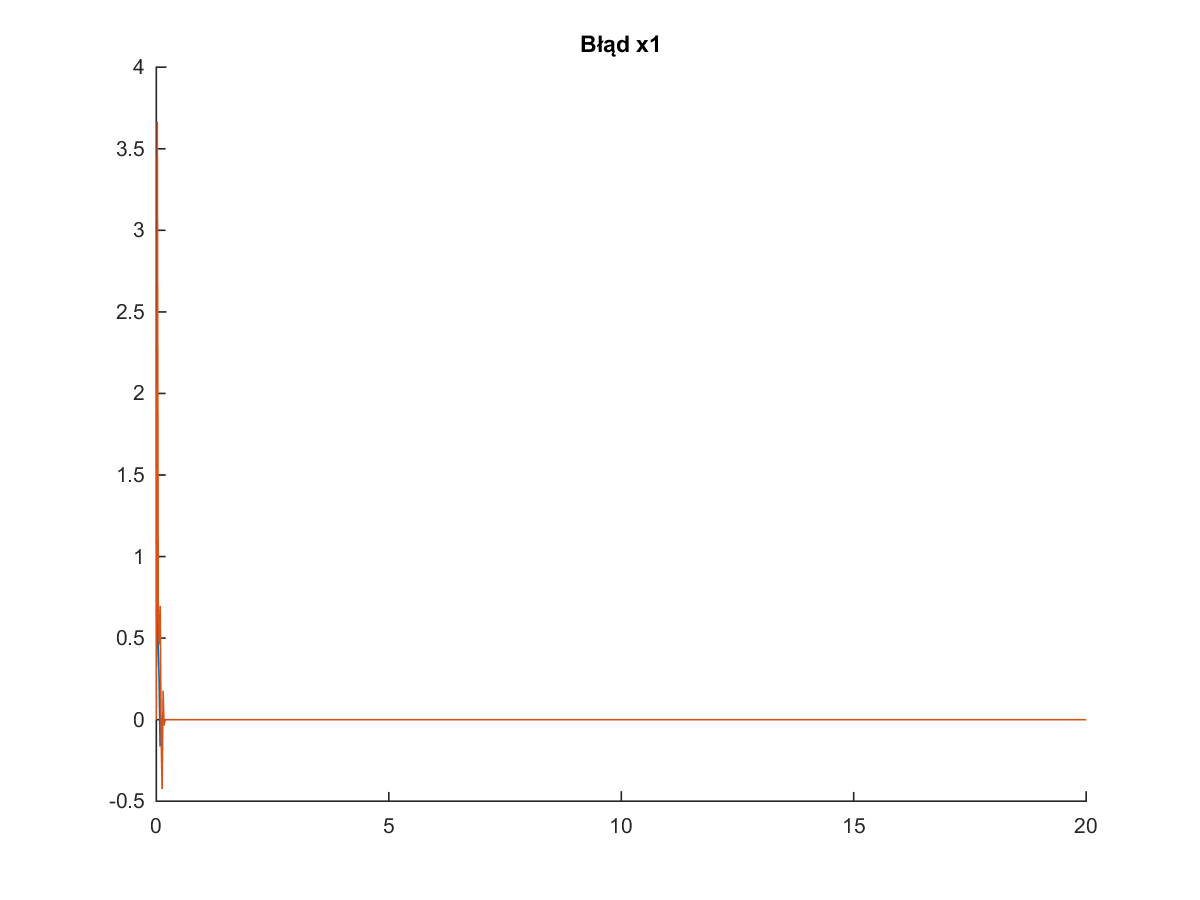
t = toc;

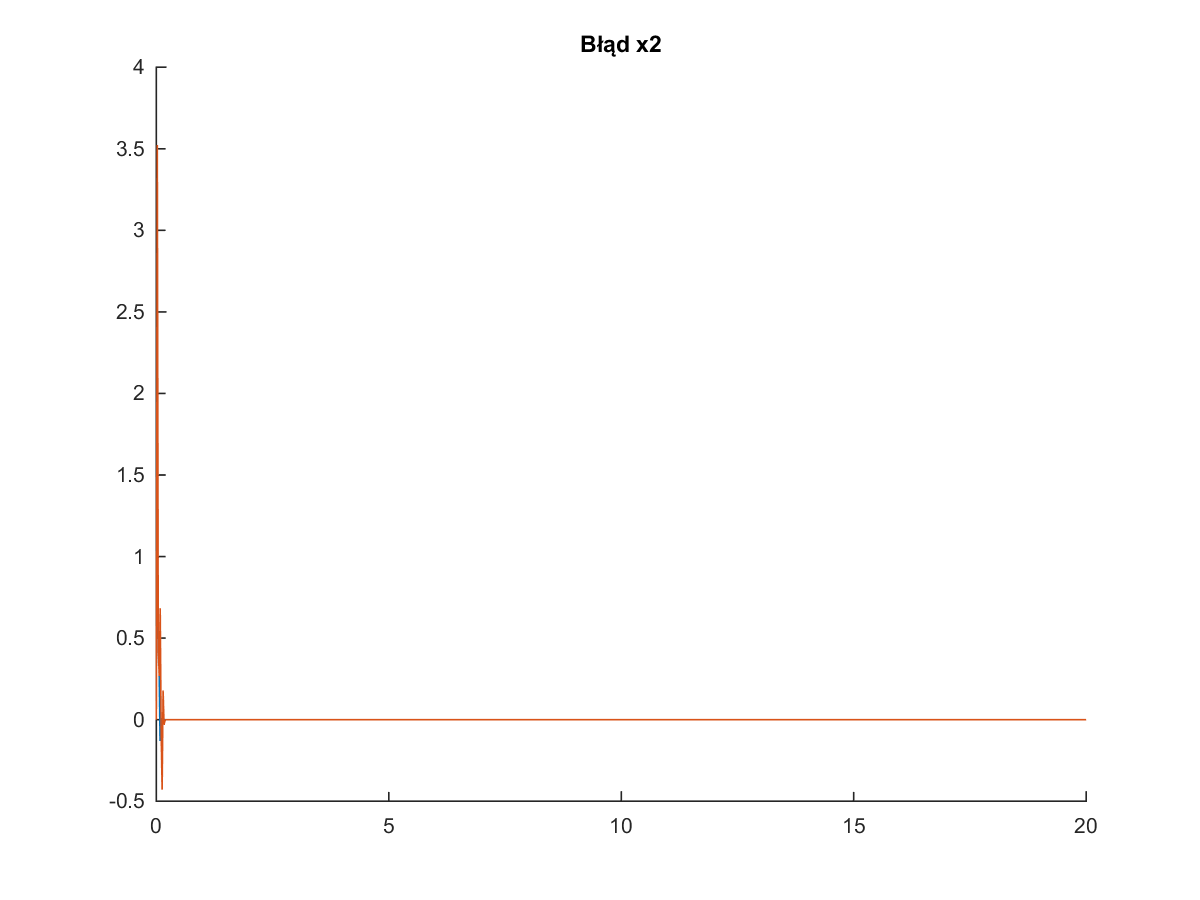
end

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

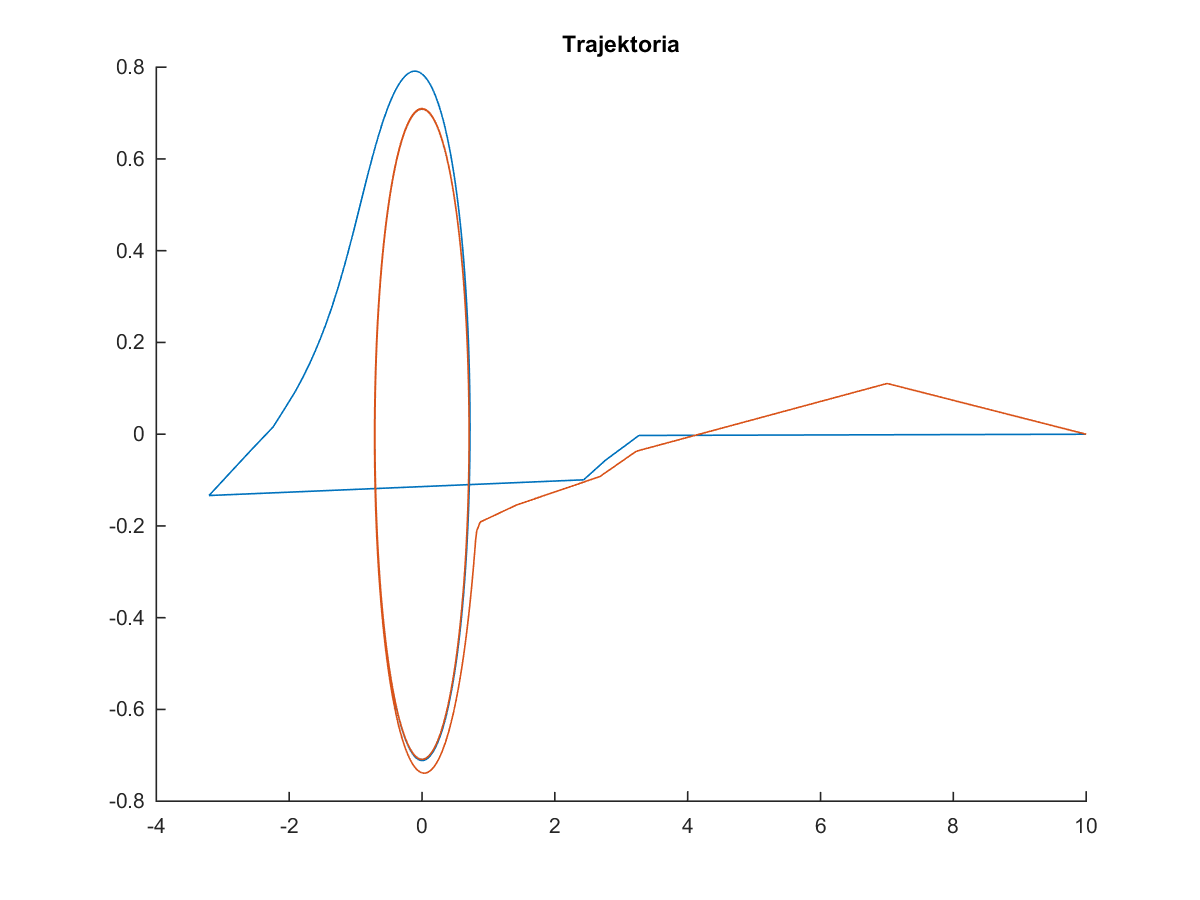


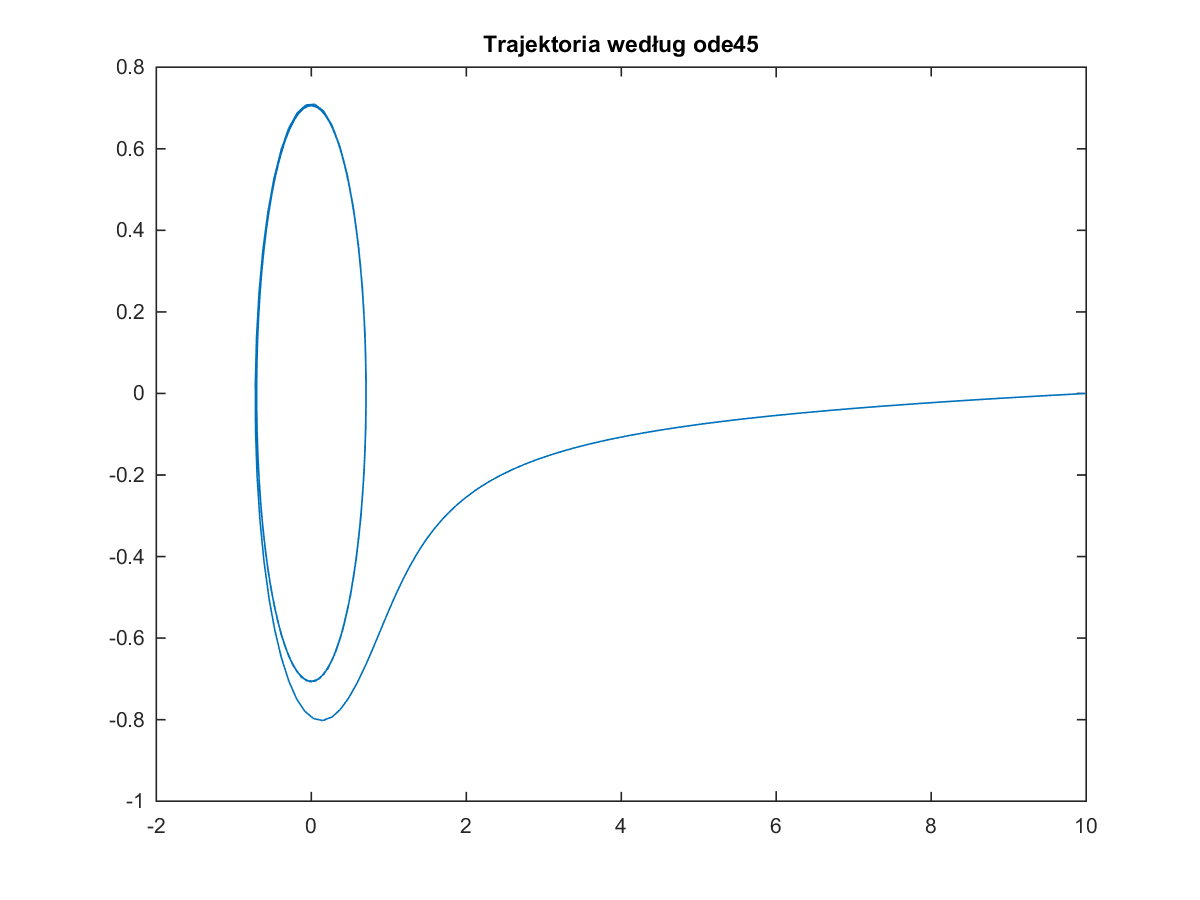


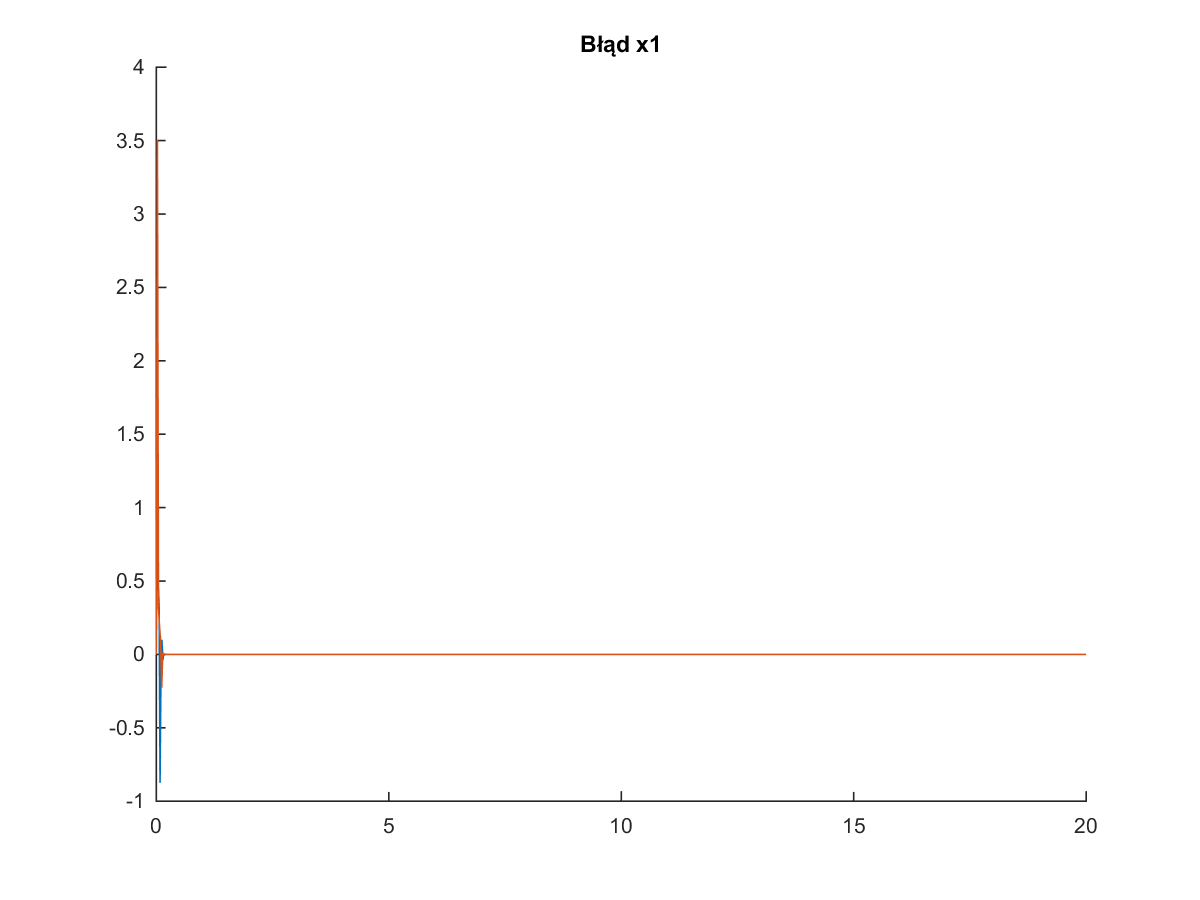


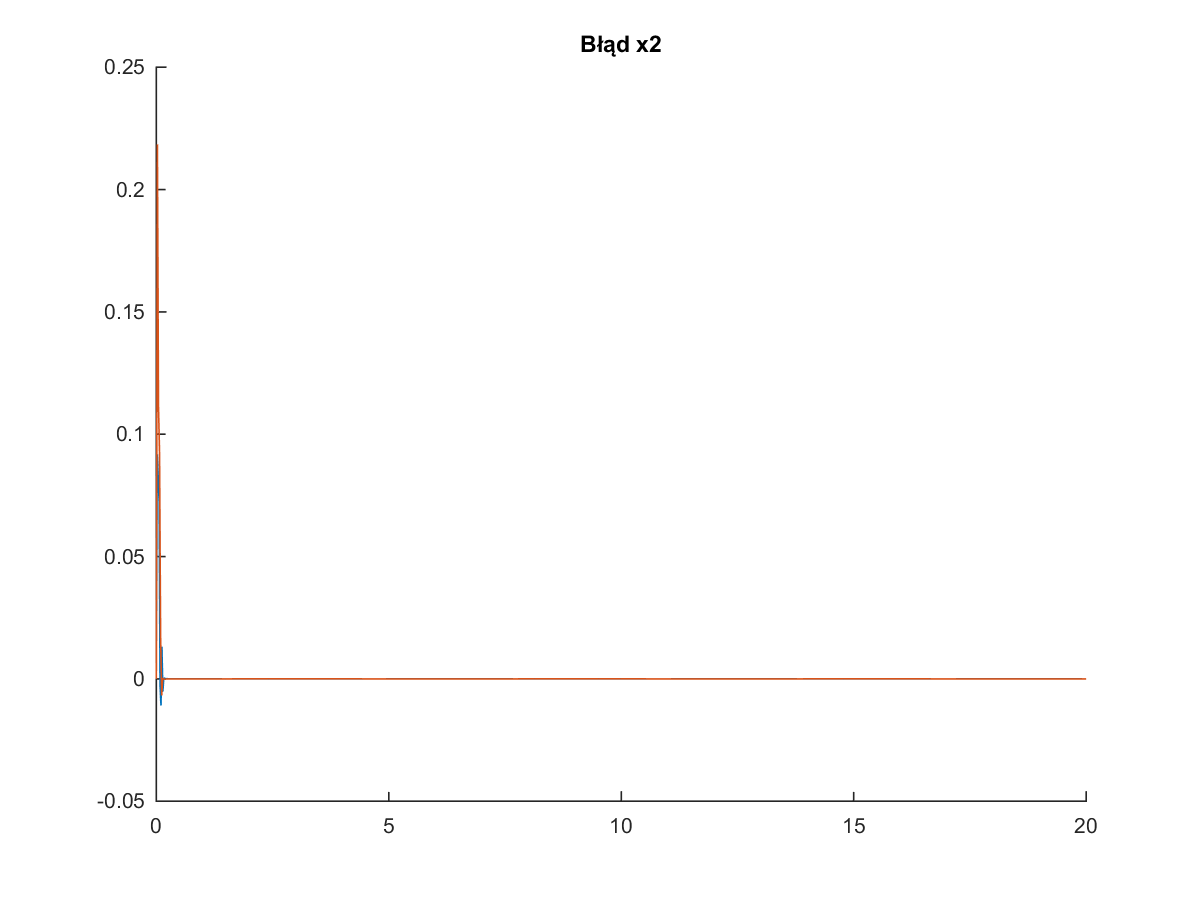


W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.02

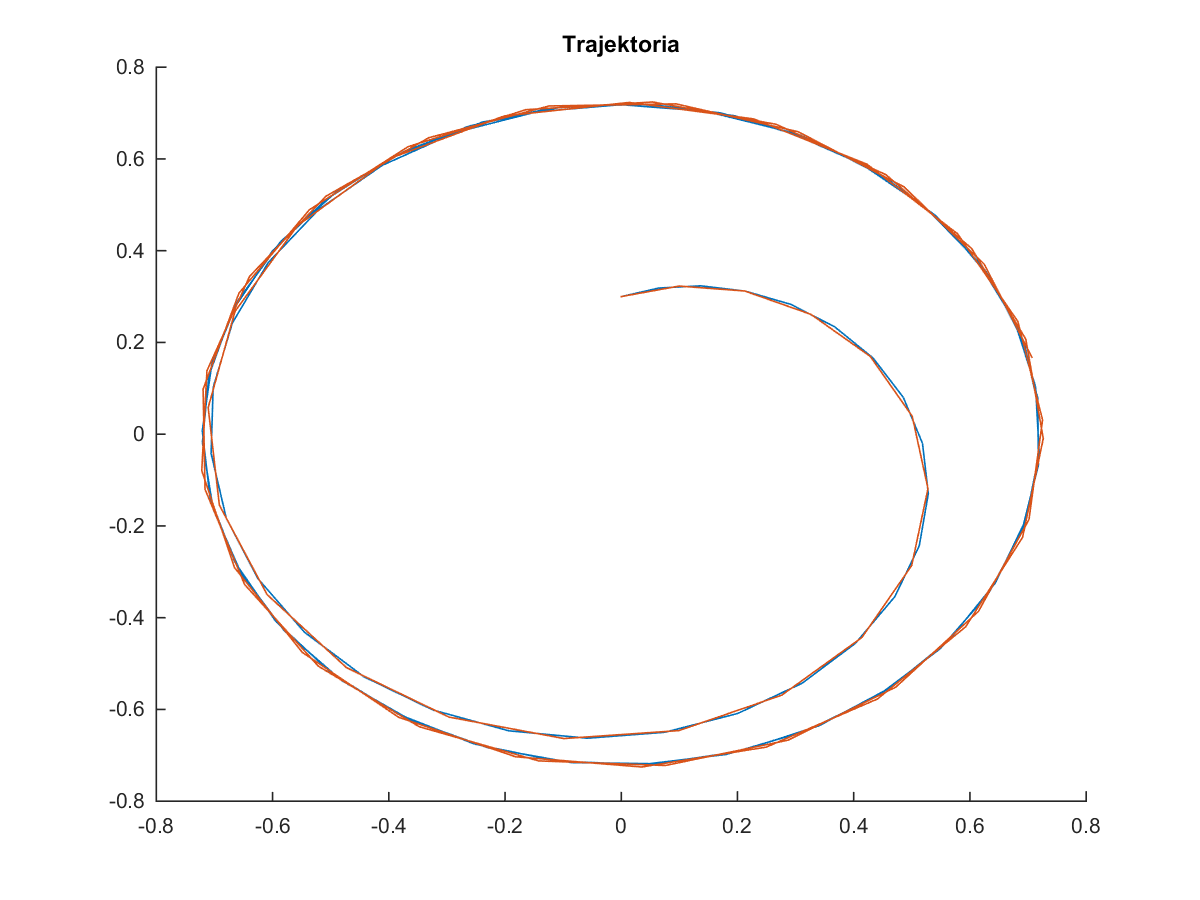


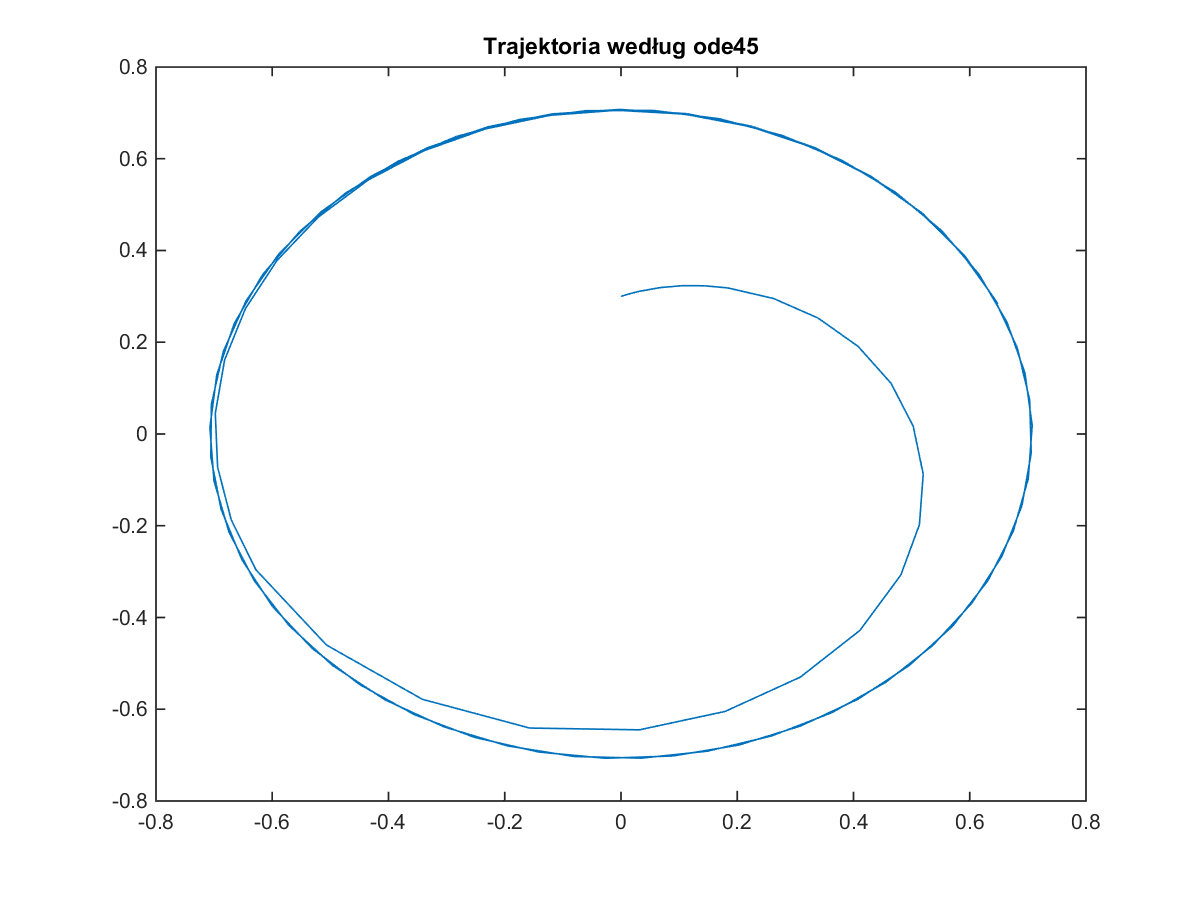


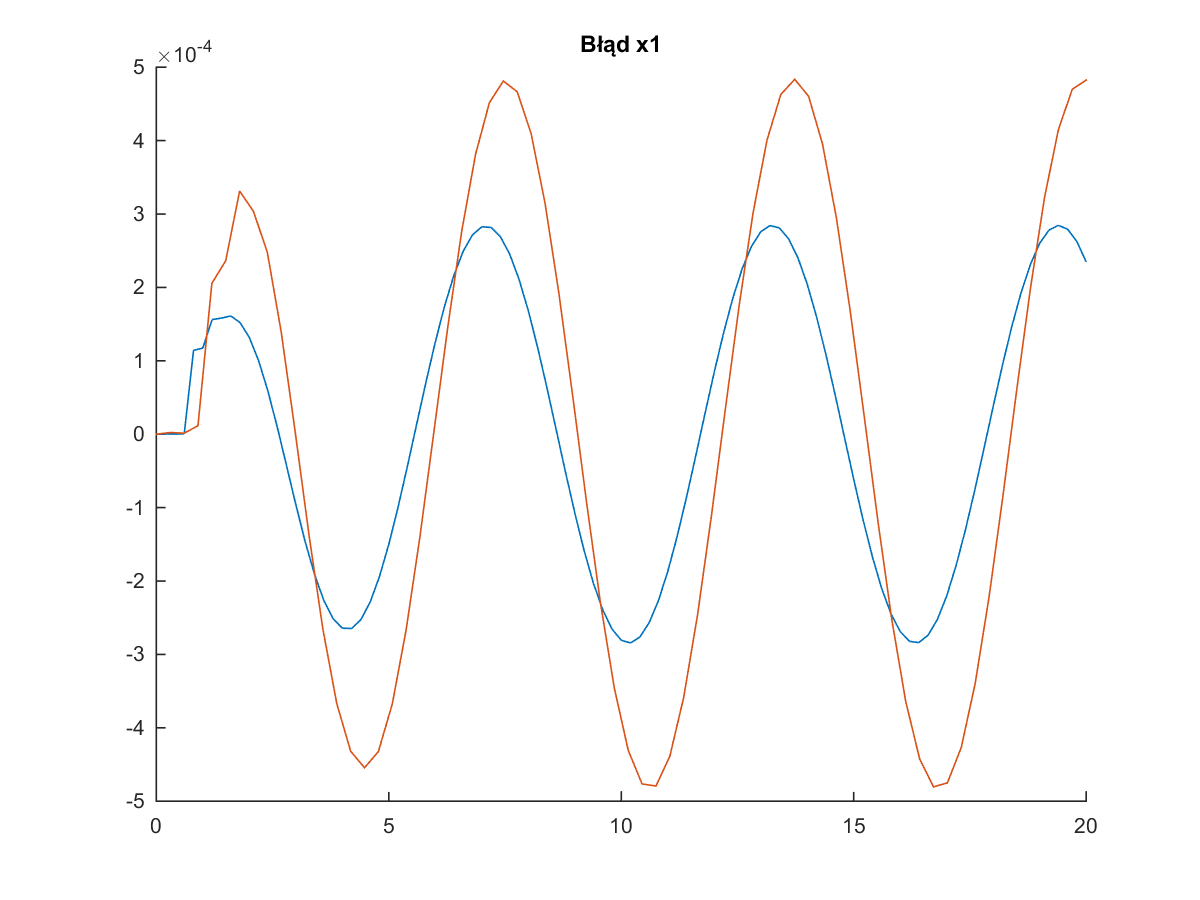


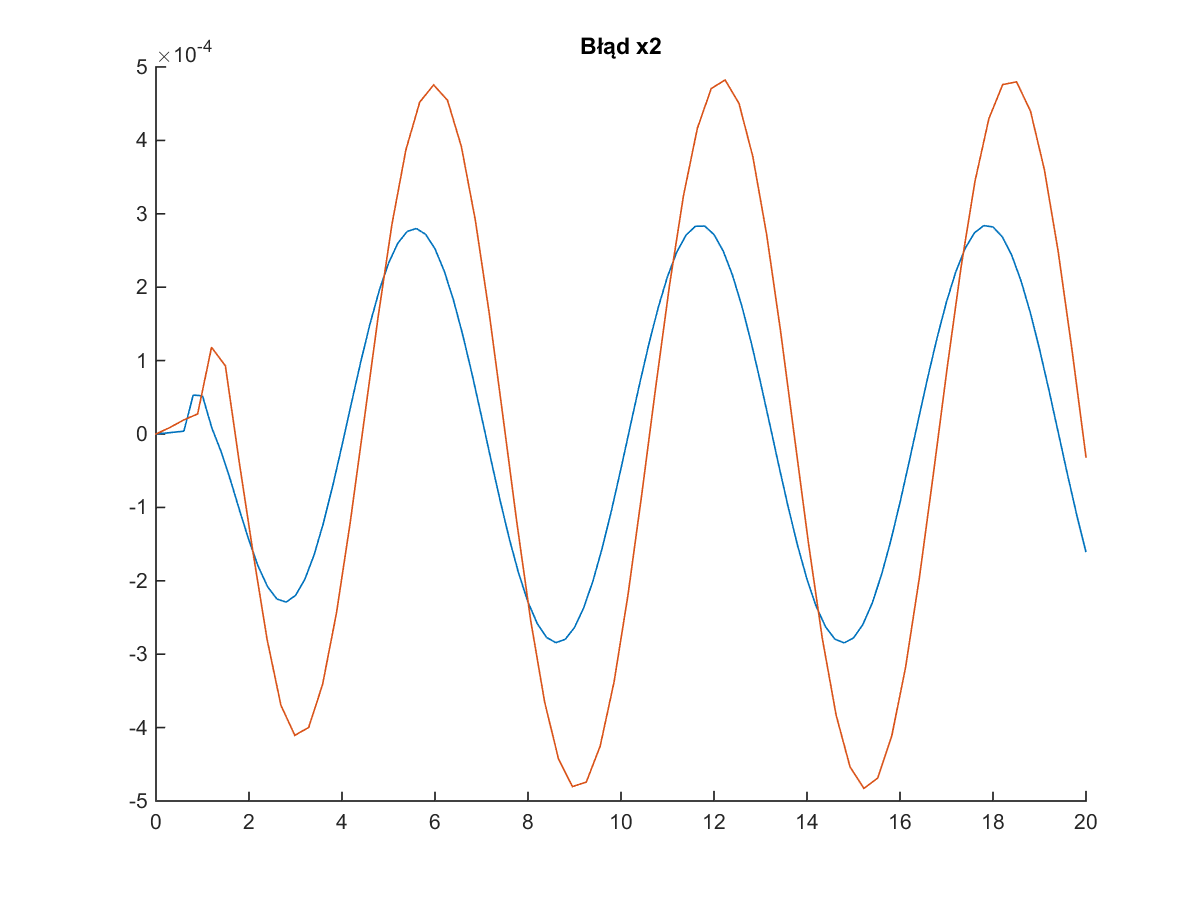


W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2

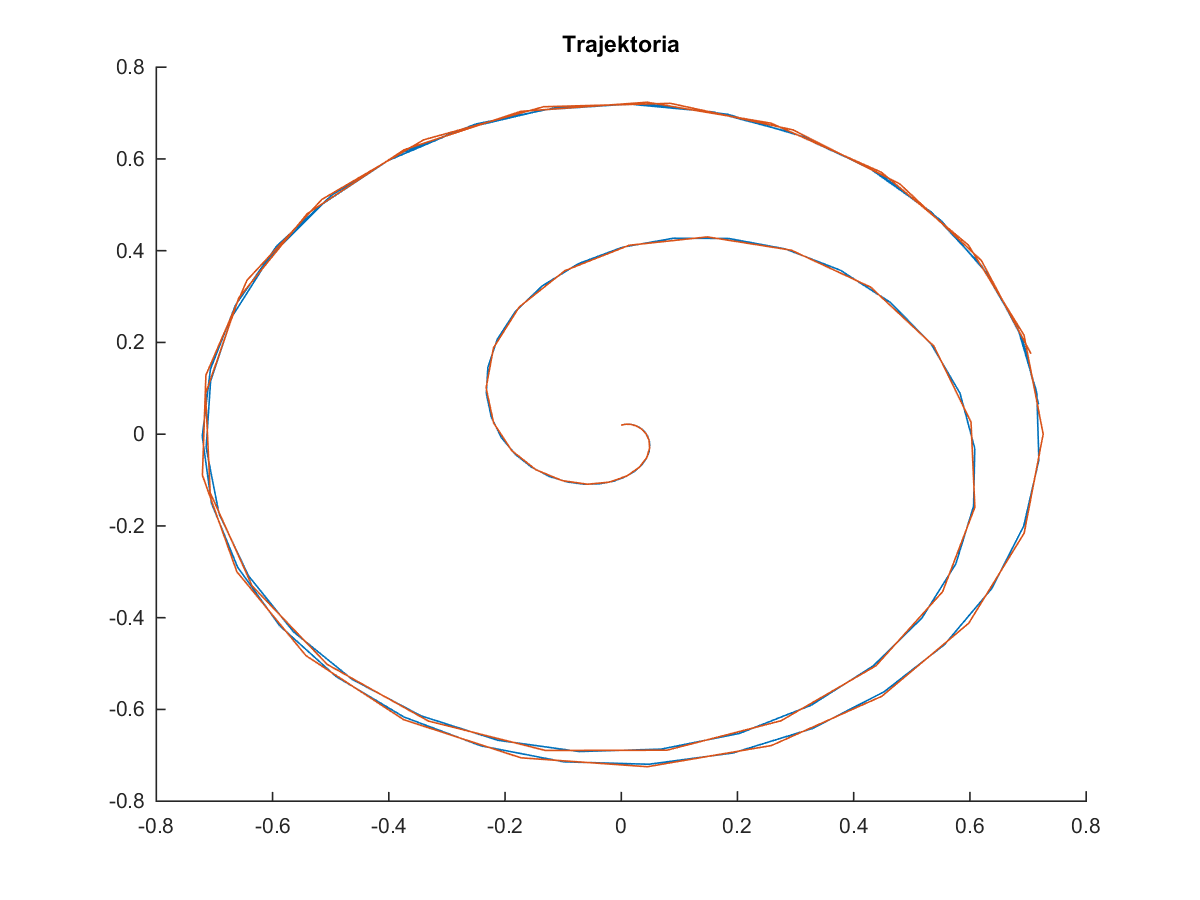


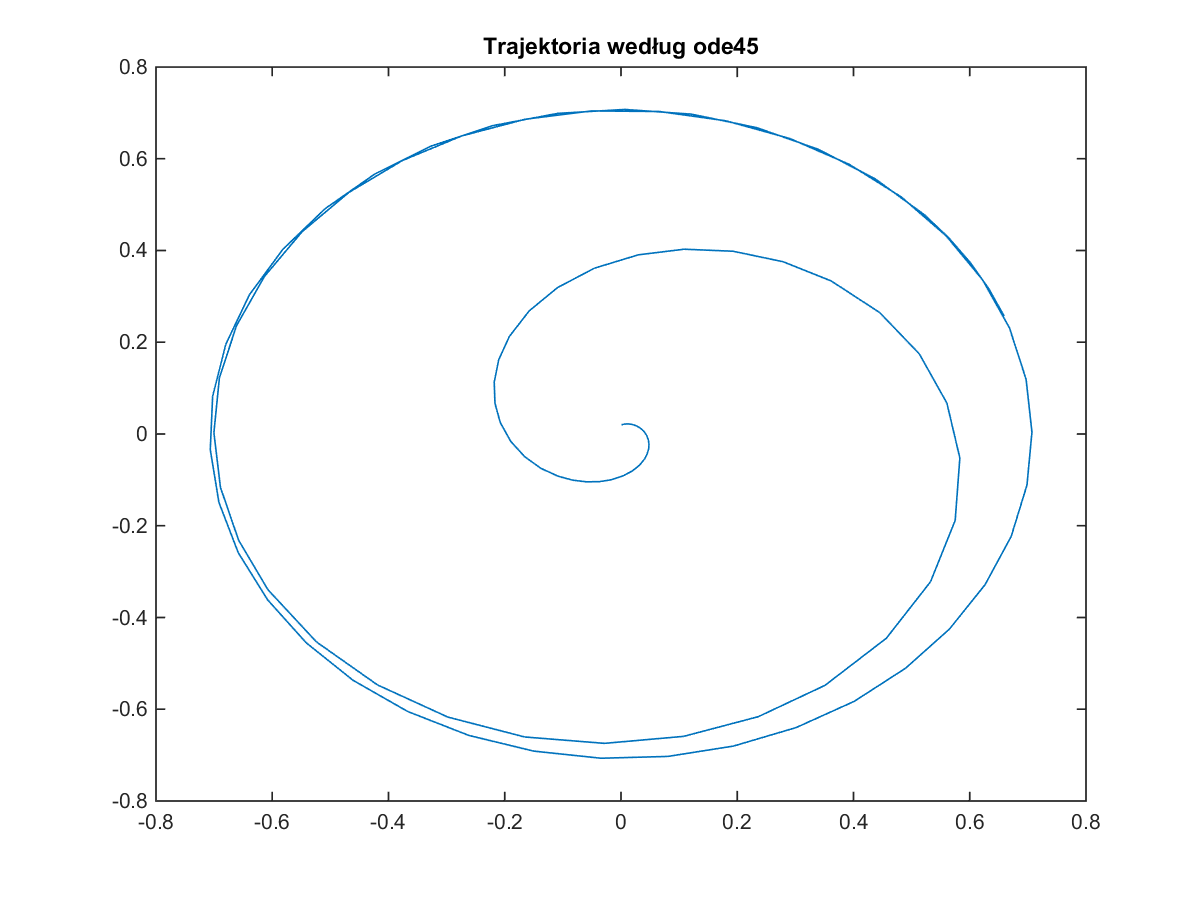


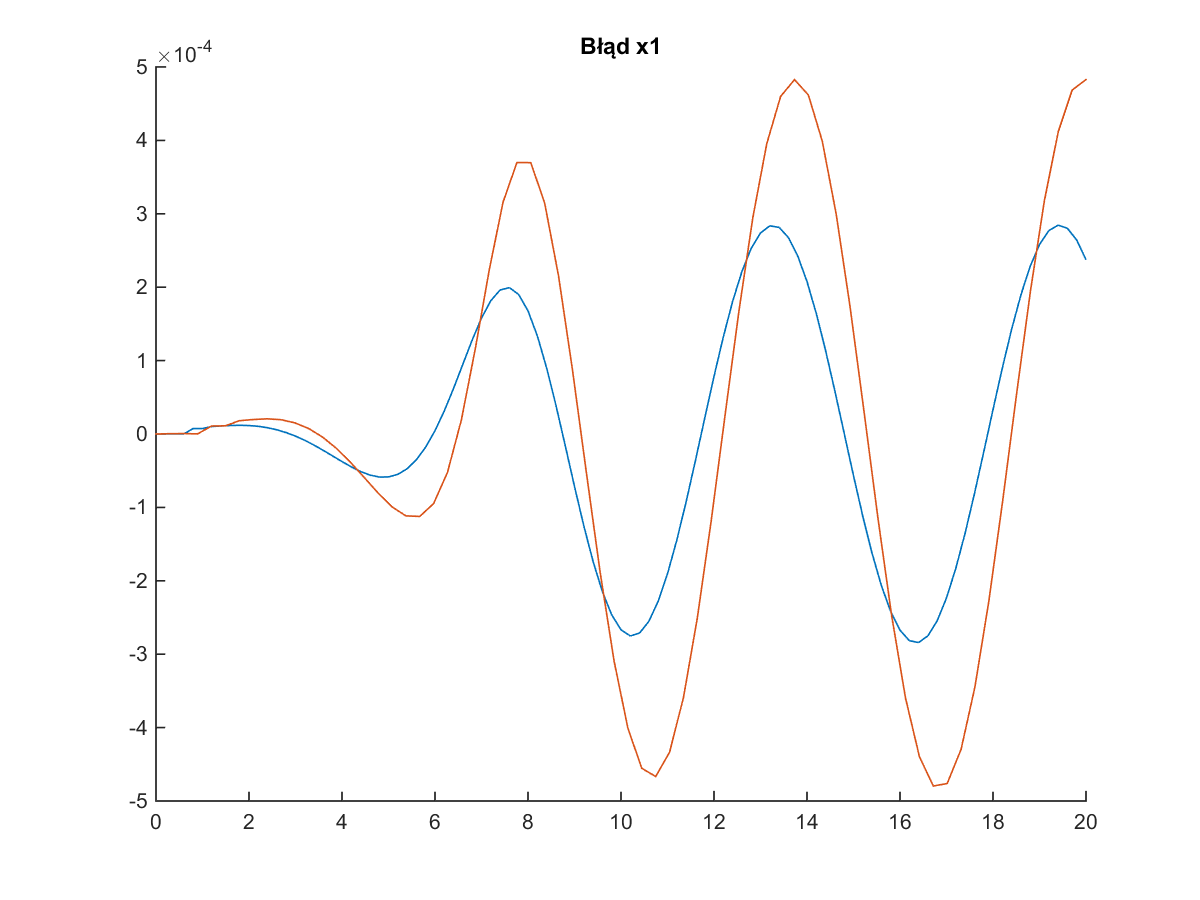


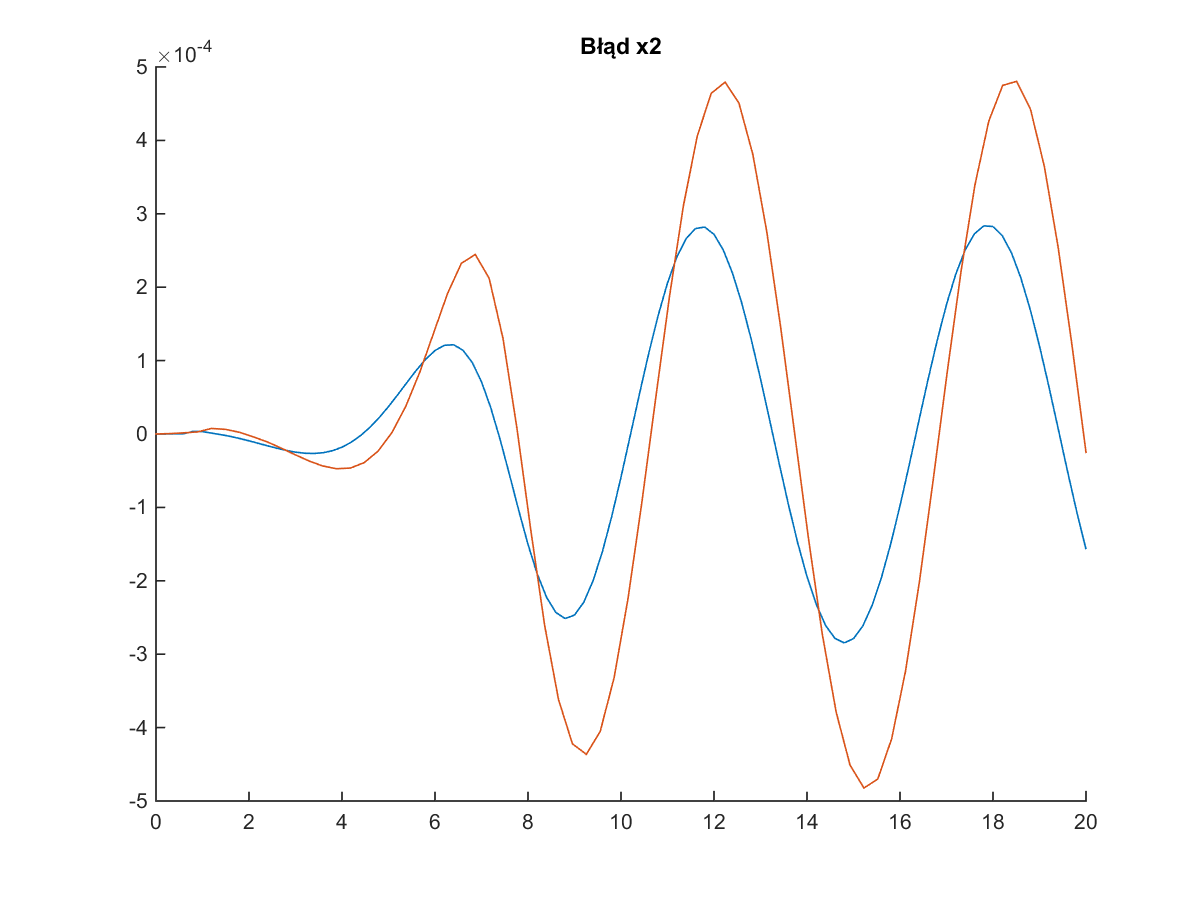


W tym przypadku krok, dla którego metoda dawała poprawne rezultaty wynosił 0.2









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).