

Sprawozdanie MNUM Projekt 04

Autor: **TOMASZ SACHANOWSKI**

Grupa: **czwartek 8-10**

Nr. Indexu: **276467**

Nr. Zadania: **4.55**

Spis treści

Treść zadań.....	2
Cel:.....	3
Teoria:.....	3
Metody jednokrokowe	3
Metoda Rungego-Kutty ze stałym krokiem	4
Prezentacja wyników - metoda Rungego-Kutty ze stałym krokiem:	6
Metoda Rungego-Kutty ze zmiennym krokiem:.....	9
Prezentacja wyników - metoda RK4 ze zmiennym krokiem:.....	10
Metody wielokrokowe	12
Metody Adamsa	13
Metody jawne (Adamsa-Bashforta)	14
Metody niejawne (Adamsa-Moultona)	14
Prezentacja wyników - metoda wielokrokowa predyktor-korektor Adamsa:	16
Wynik:.....	19
$x_1(0)=10$ $x_2(0)=8$	19
$x_1(0)=8$ $x_2(0)=0$	19
$x_1(0)=0.001$, $x_2(0)=0.001$	19
Podsumowanie:	19
Dodatek:	20

Treść zadań

MNUM-PROJEKT zadanie 4.55

Ruch punktu jest opisany równaniami:

$$\begin{aligned}x_1' &= x_2 + x_1(0,9 - x_1^2 - x_2^2) \\ x_2' &= -x_1 + x_2(0,9 - x_1^2 - x_2^2)\end{aligned}$$

Należy obliczyć przebieg trajektorii ruchu na przedziale $[0, 20]$ dla następujących warunków początkowych:

- | | | | |
|----------------|--------------|-------------------|------------------|
| a) $x_1(0)=10$ | $x_2(0)=8$; | b) $x_1(0)=0$ | $x_2(0)=9$; |
| c) $x_1(0)=8$ | $x_2(0)=0$; | d) $x_1(0)=0,001$ | $x_2(0)=0,001$. |

Do rozwiązania zadania należy użyć zaimplementowanych przez siebie metod:

1. Rungego–Kutty czwartego rzędu (RK4) ze stałym krokiem. Proszę przy tym wykonać tyle prób (kilka – kilkanaście), ile będzie potrzebnych do znalezienia takiego kroku, którego zmniejszanie nie wpływa znacząco na rozwiązanie, podczas gdy zwiększanie – już wpływa;
2. Wielokrokowej predyktor–korektor Adamsa czwartego rzędu ze stałym krokiem, który należy dobrać w sposób podany dla metody z punktu 1;
3. Rungego–Kutty czwartego rzędu (RK4) ze zmiennym krokiem;
4. Metody powinny być tak zaimplementowane, aby w każdym kroku błąd aproksymacji był szacowany.

Sprawozdanie powinno zawierać:

- a) krótki opis zastosowanych algorytmów;
- b) opis sposobu szacowania błędów i znajdowania nowego rozwiązania dla każdej z metod;
- c) dla metod stałokrokowych RK4 i predyktor–korektor Adamsa komentarze i wnioski dotyczące doboru długości kroku zilustrowane wykresami rozwiązań otrzymanych przy wybranym i przy zbyt dużym kroku;
- d) porównanie rozwiązań otrzymanych przy wybranym kroku z rozwiązaniami wyznaczonymi poleceniem *ode45* programu Matlab;
- e) odpowiedź na pytanie, z powołaniem na wyniki eksperymentów, która metoda jest lepsza i dlaczego (proszę wziąć, w szczególności, pod uwagę liczbę iteracji/kroków, czas obliczeń, średni i maksymalny błąd);
- f) wydruk dobrze skomentowanych programów z implementacją metod.

Sprawozdanie powinno być wysłane na adres prowadzącego: a.krzemienowski@elka.pw.edu.pl.

Cel:

Celem zadania jest znalezienie przebiegu trajektorii punktu na danym przedziale i przy początkowych warunkach.

Teoria:

Równania różniczkowe służą do modelowania fizycznych układów dynamicznych. Metody numeryczne są sposobem rozwiązywania tych nieliniowych układów równań.

Wyróżniamy następujące metody numerycznego rozwiązywania równań różniczkowych:

- metody jednokrokowe.
- metody wielokrokowe.

Metody jednokrokowe

Metody jednokrokowe są zdefiniowane poprzez następujący wzór:

$$y_{n+1} = y_n + h\phi_f(x_n, y_n; h)$$

gdzie:

$\phi_f(x_n, y_n; h)$ – funkcja definiująca metodę

h – długość kroku

Metoda jest zbieżna gdy:

$$h \rightarrow 0 \Rightarrow y(x_n; h) \rightarrow y(x)$$

Jeżeli spełnione są założenia:

- funkcje są ciągłe na zbiorze: $D = \{(x, y) : a \leq x \leq b, y \in \mathbb{R}\}$
- funkcja spełnia warunki Lipschitza względem y ,
tzn.: $\|f(x, y) - f(x, \bar{y})\| \leq L \|y - \bar{y}\|$,

to warunek powyższy jest koniecznym i dostatecznym warunkiem zbieżności metody jednokrokowej.

Metody te służą do rozwiązywania równań różniczkowych zwyczajnych podanych z warunkiem początkowym:

$$\frac{dy}{dx} = f(x, y(x))$$

$$y(x_0) = y_0$$

Mówimy, że metoda jest rzędu p , jeśli zachodzą równości:

$$r_n(0) = 0, r'_n(0) = 0, r''_n(0) = 0, \dots, r_n^p(0) = 0, r_n^p(0) \neq 0$$

Metoda Rungego-Kutty ze stałym krokiem

Jest ona jedną z metod jednokrokowych. Do wykonania jednego kroku metody należy obliczyć wartości prawych stron dokładnie m razy (metoda m -etapowa).

Jeśli przez $p(m)$ oznaczymy maksymalny rząd metody, to udowodniono, że:

$$p(m)=m \text{ dla } m=1,2,3,4$$

$$p(m)=m-1 \text{ dla } m=5,6,7$$

$$p(m) \leq m-2 \text{ dla } m \geq 8.$$

Największe znaczenie praktyczne mają metody z $m=4$ i rzędu 4 – jest to kompromis między dokładnością (rząd metody) a nakładem obliczeń na jedną iterację i związanym z tym wpływem błędów zaokrągleń.

Jest ona metodą samosterującą, tzn. znajomość warunku początkowego wystarcza, by rozpocząć obliczenia. Niestety jest to też raczej metoda kosztowna czasowo (wymaga wielokrotnego obliczania wartości funkcji).

Metoda RK4

$$y_{n+1} = y_n + \frac{1}{6}h(k_1 + 2k_2 + 3k_3 + k_4)$$

$$k_1 = f(x_n, y_n)$$

$$k_2 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right)$$

$$k_3 = f\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right)$$

$$k_4 = f(x_n + h, y_n + hk_3)$$

Gdzie:

x, y – argumenty funkcji f ,

h – długość kroku.

Współczynnik k_1 jest pochodną rozwiązania w punkcie (x_n, y_n) .

Wartość k_2 pochodną rozwiązania wyznaczanego zwykłą metodą

Eulera w punkcie $(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1)$ (środkowym przedziału). Z k_3 i k_4

jest podobnie. W taki sposób mamy wyznaczone cztery wartości pochodnych obliczonych w końcach przedziału i w jego środku.

Aproksymacja pochodnej dla pewnego kroku wyznaczana jest jako średnia arytmetyczna tych wartości z wagami 1 dla wartości krańcowych i odpowiednio 2 i 3 dla wartości środkowych.

Wybór kroku jest jednym z trudniejszych zadań w tej metodzie. Krok powinien być wystarczający do uzyskania założonej dokładności, lecz nie powinien być znacznie mniejszy przy której wymagana dokładność jest osiągnięta.

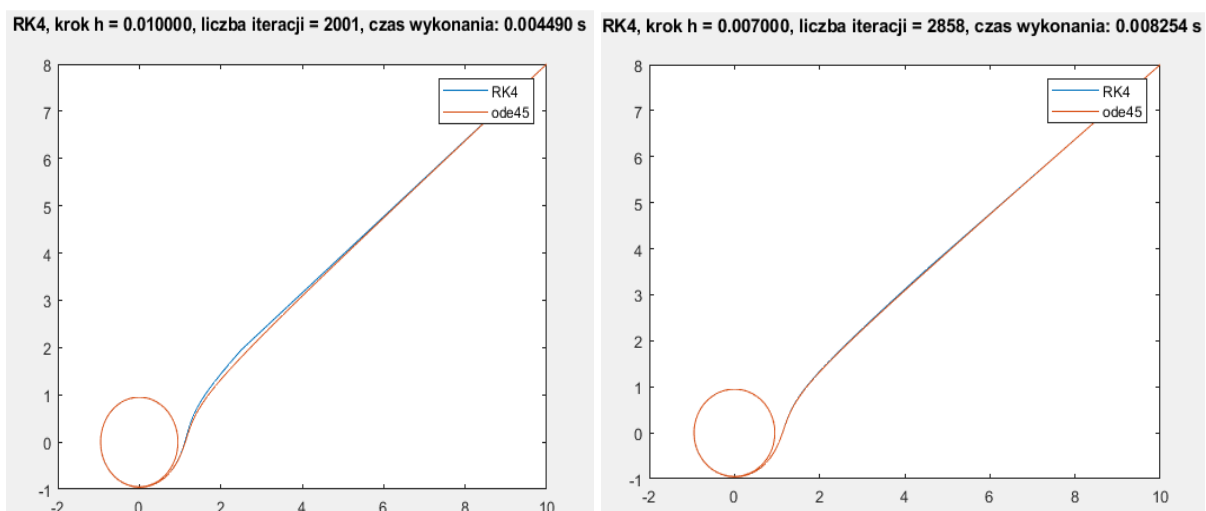
Przy wyznaczaniu długości kroku występują dwie przeciwstawne tendencje:

- jeśli krok h_n maleje, to maleje błąd metody (błąd aproksymacji), dla metody zbieżnej błąd maleje do zera przy h dążącym do zera,
- jeśli krok h_n maleje, to zwiększa się liczba iteracji (liczba kroków) potrzebnych do wyznaczenia rozwiązania na zadanym odcinku $[a,b]$, a stąd liczba obliczeń i związanych z nimi błędów numerycznych.

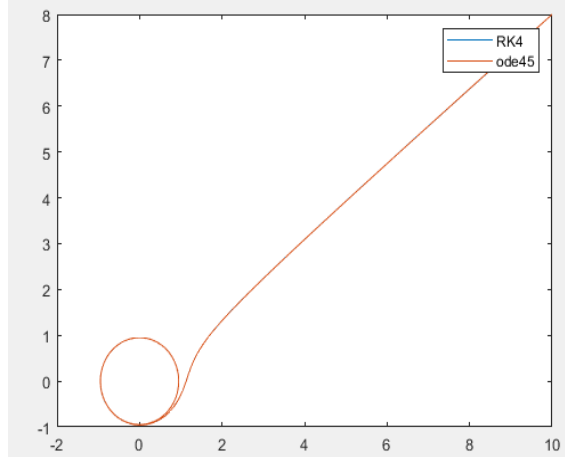
Wybór długości kroku h jest bardzo istotny. Jeśli maleje krok h to maleje błąd metody, ale zwiększa się liczba iteracji, a zatem również wielkość błędów numerycznych. Wynika z tego, że istnieje pewny optymalny krok dla tej metody. Wyznaczyć go można empirycznie. Krok dobrany zostanie tak aby jego zmniejszenie nie wpływało znacząco na rozwiązanie, podczas gdy zwiększenie wpływa.

Prezentacja wyników - metoda Rungego-Kutty ze stałym krokiem:

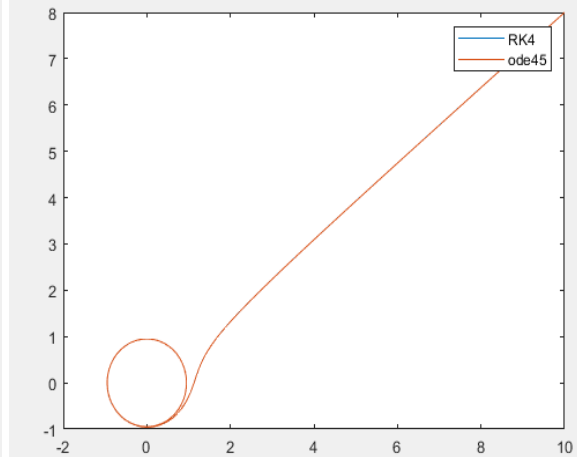
a. $x_1(0)=10$ $x_2(0)=8$



RK4, krok $h = 0.004000$, liczba iteracji = 5001, czas wykonania: 0.018592 s



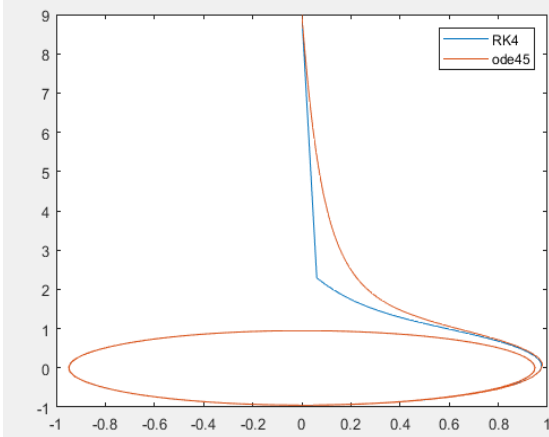
RK4, krok $h = 0.001000$, liczba iteracji = 20001, czas wykonania: 0.219943 s



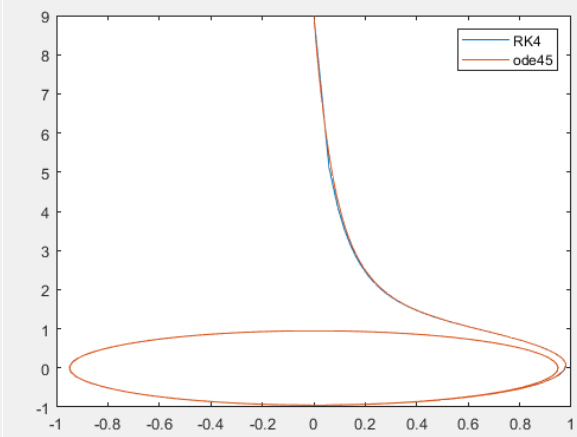
Właściwa trajektoria dla $h = 0.007$:

b. $x_1(0)=0$ $x_2(0)=9$

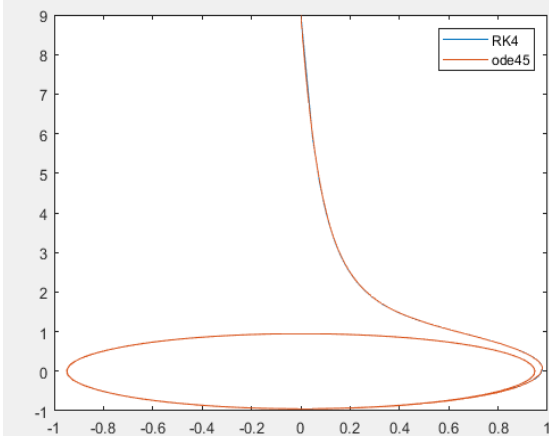
RK4, krok $h = 0.020000$, liczba iteracji = 1001, czas wykonania: 0.001825 s



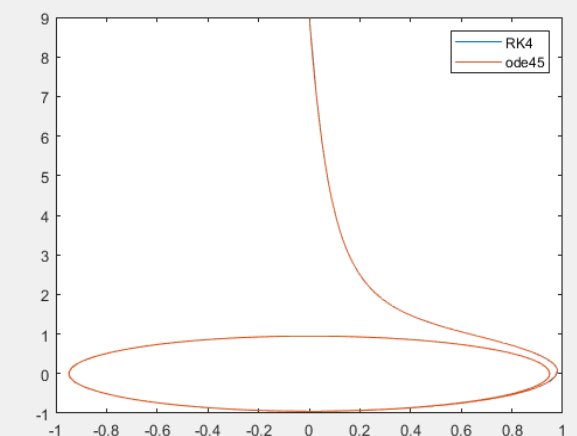
RK4, krok $h = 0.011000$, liczba iteracji = 1819, czas wykonania: 0.002476 s



RK4, krok $h = 0.008000$, liczba iteracji = 2501, czas wykonania: 0.006677 s



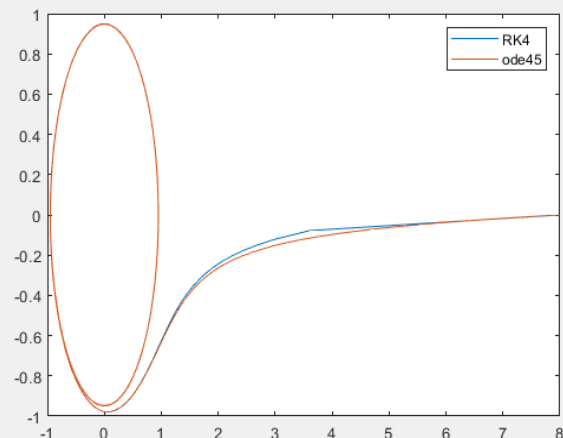
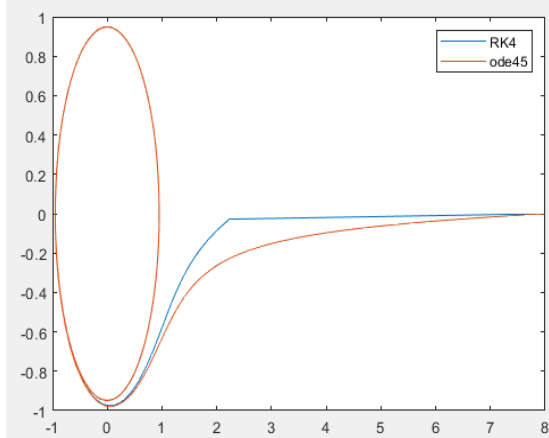
RK4, krok $h = 0.004000$, liczba iteracji = 5001, czas wykonania: 0.017284 s



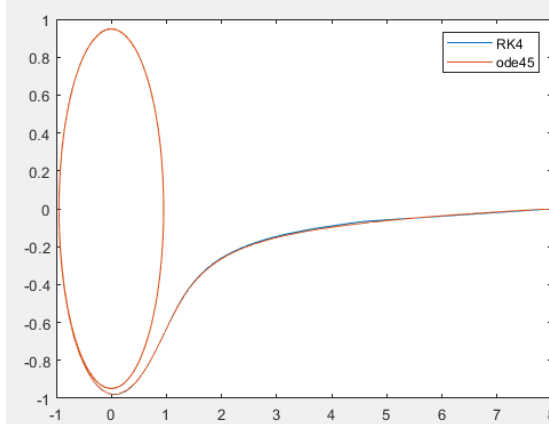
Właściwa trajektoria dla $h = 0.011$:

c. $x_1(0)=8 \ x_2(0)=0$

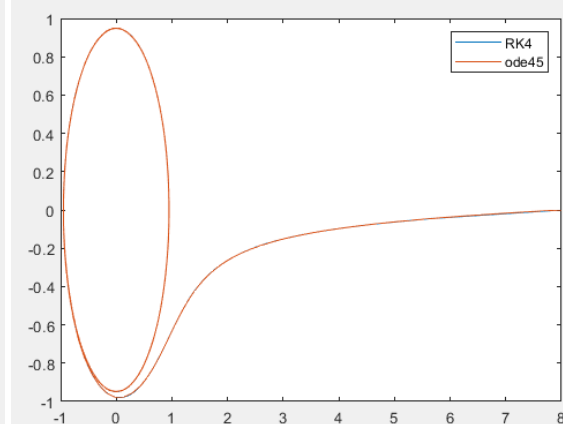
RK4, krok $h = 0.030000$, liczba iteracji = 667, czas wykonania: 0.001233 s RK4, krok $h = 0.018000$, liczba iteracji = 1112, czas wykonania: 0.002130 s



RK4, krok $h = 0.014000$, liczba iteracji = 1429, czas wykonania: 0.002808 s



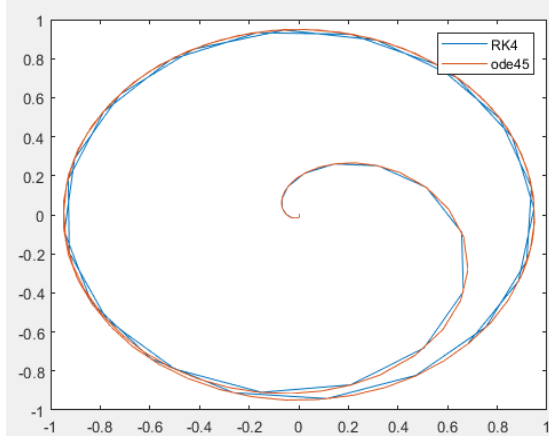
RK4, krok $h = 0.008000$, liczba iteracji = 2501, czas wykonania: 0.006183 s



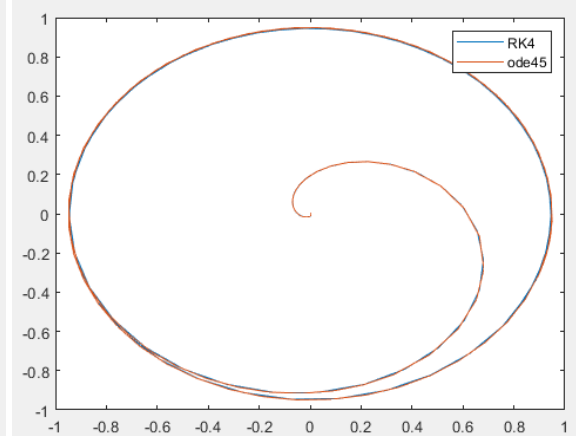
Właściwa trajektoria dla $h = 0.014$:

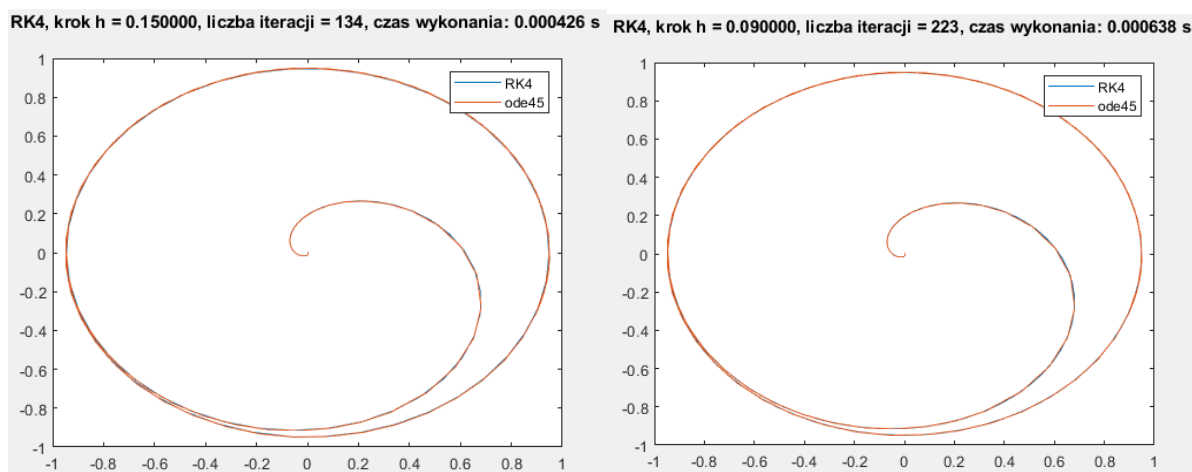
d. $x_1(0)=0.001, x_2(0)=0.001$

RK4, krok $h = 0.400000$, liczba iteracji = 51, czas wykonania: 0.000332 s



RK4, krok $h = 0.200000$, liczba iteracji = 101, czas wykonania: 0.000326 s





Właściwa trajektoria dla $h = 0.015$:

Metoda Rungego-Kutty ze zmiennym krokiem:

Wiadomo, że zmniejszając krok h , to maleje błąd metody, lecz wówczas zwiększa się liczba iteracji potrzebnych do wyznaczenia rozwiązania na żądanym odcinku, więc i liczba obliczeń i związanych z nimi błędów numerycznych.

W takim razie warto poszukać złotego środka, który będzie korzystał starał się zoptymalizować.

Lokalny błąd metody szacowany jest w każdym kroku. Definiuje się on w następujący sposób:

$$r_n(h) = y(x_n + h) - [y(x_n) + h\Phi_f(x_n, y_n; h)]$$

Przekształcając otrzymujemy:

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

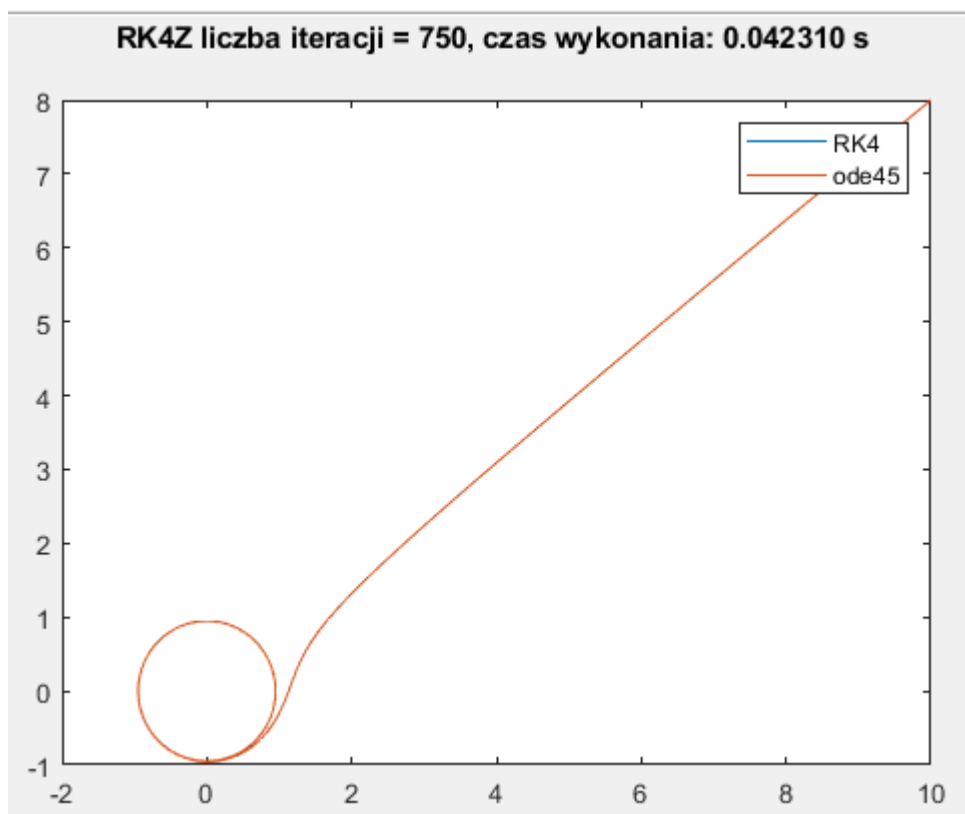
gdzie:

- h – wartość kroku,
- p – rząd metody,
- $y_n^{(2)}$ – nowy punkt wyznaczony metodą z 2 krokami o długości $h/2$,
- $y_n^{(1)}$ – nowy punkt wyznaczony metodą z krokiem o długości h .

Metoda ta polega na obliczaniu błędu aproksymacji w każdej iteracji. Na jego podstawie obliczany jest współczynnik przez który mnożony jest wcześniejszy krok. Jeśli okazuje się, że obecny krok nie jest wystarczająco dokładny to należy powtórzyć iterację z krokiem pomniejszonym o wyliczony współczynnik.

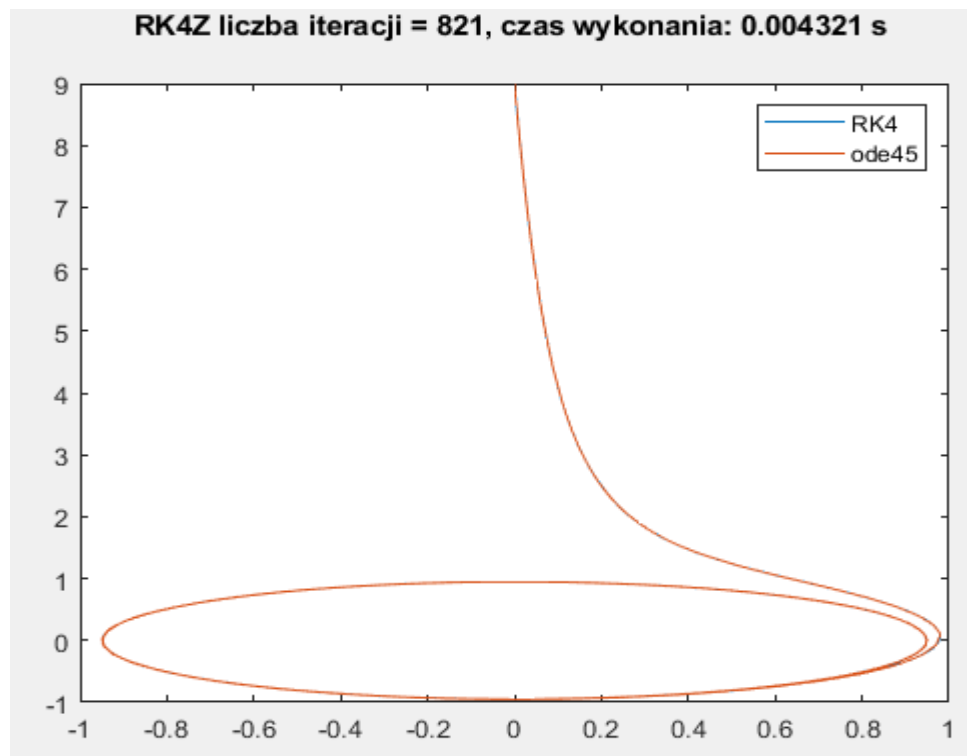
Prezentacja wyników - metoda RK4 ze zmiennym krokiem:

a. $x_1(0)=10$ $x_2(0)=8$



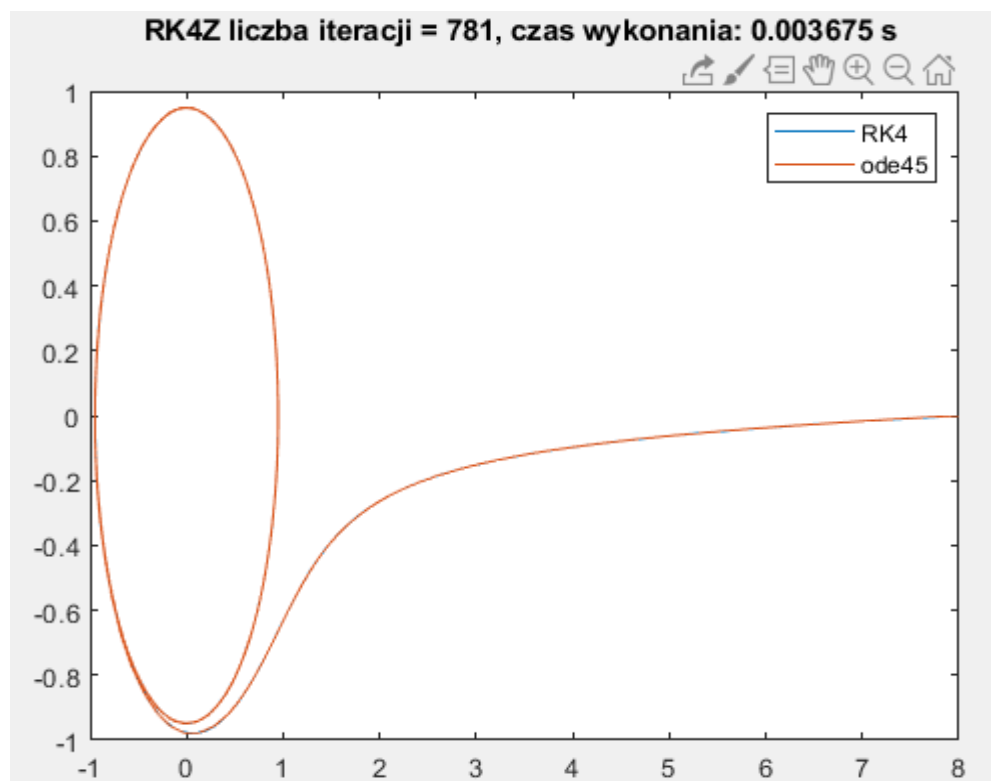
Właściwa trajektoria dla $h = 0.009$:

b. $x_1(0)=0$ $x_2(0)=9$



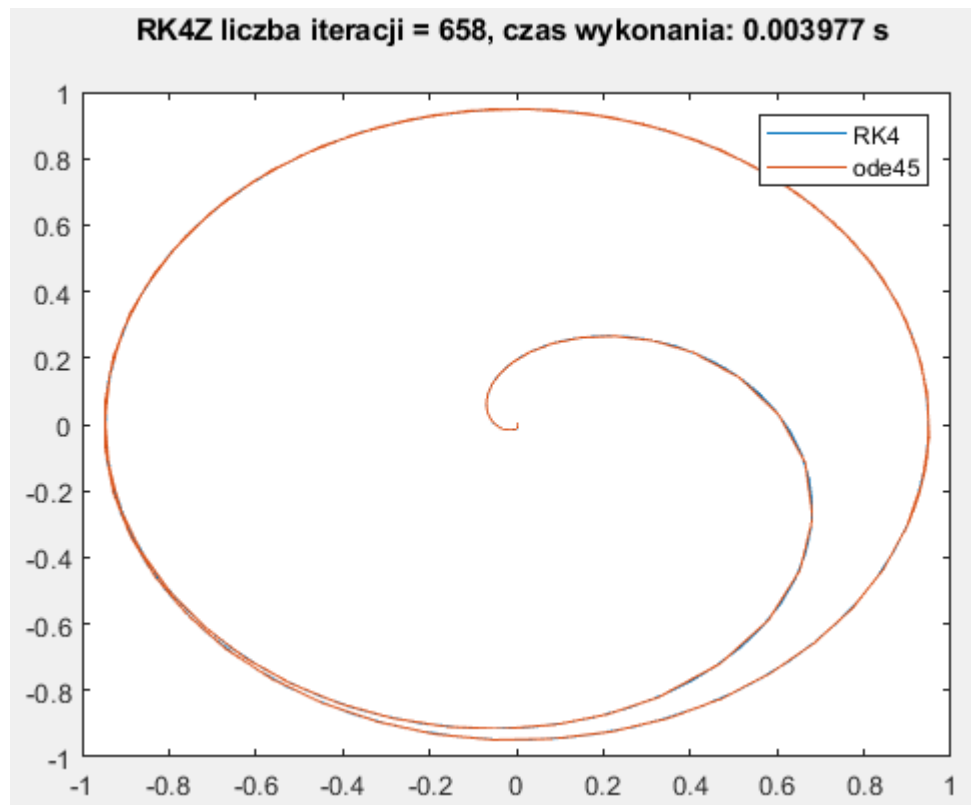
Właściwa trajektoria dla $h = 0.025$:

c. $x_1(0)=8$ $x_2(0)=0$



Właściwa trajektoria dla $h = 0.014$:

d. $x_1(0)=0.001, x_2(0)=0.001$



Właściwa trajektoria dla $h = 0.55$

Metody wielokrokowe

Ogólna postać wzoru definiującego krok (iterację) metody k -krokowej liniowej, ze stałą długością kroku h jest następująca:

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

Gdzie:

- $y_0=y(x_0)=y_a$
- $x_n=x_0+nh$

- $x_0=a$
- $x \in [a,b]$.

Metoda wielokrokowa jest jawna, jeśli $\beta_0 = 0$. Dla metody jawnej wartość y_n zależy jawnie od wartości y i $f(x, y)$ jedynie w poprzednich (już obliczonych) punktach.

Metoda wielokrokowa jest niejawna, jeśli $\beta_0 \neq 0$. Dla metody niejawnej wartość y_n obliczana jest na podstawie k poprzednich wartości y i $f(x, y)$ oraz dodatkowo również wartości w punkcie bieżącym $f_n = f(x_n, y_n)$ tj. do wyznaczenia y_n trzeba w istocie rozwiązać równanie algebraiczne $\varphi(y_n) = 0$ (nieliniowe, jeśli funkcja prawej strony f jest nieliniowa), gdzie:

$$\varphi(y_n) \stackrel{\text{def}}{=} -y_n + \sum_{j=1}^k \alpha_j y_{n-j} + h \sum_{j=1}^k \beta_j f(x_{n-j}, y_{n-j}) + h\beta_0 f(x_n, y_n).$$

Metody Adamsa

Równanie różniczkowe:

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

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

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 (Adamsa-Bashforta)

Funkcję podcałkową f z powyższego wzoru przybliżamy wielomianem interpolacyjnym $W(x)$ stopnia co najwyżej $k-1$ opartym na węzłach x_{n-1}, \dots, x_{n-k} . Przyjmując przybliżenie $y(x_{n-j}) \approx y_{n-j}$ i stosując wzór interpolacyjny Lagrange'a mamy:

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

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

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:

$$x_{n-j} = x_n - jh, \quad j=1, 2, \dots, k,$$

Otrzymujemy:

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

Wartości współczynników β odczytujemy z tabeli parametrów metod jawnych Adamsa.

Aby otrzymać rząd metody równy 4 przyjmujemy $k=4$.

Metody niejawne (Adamsa-Moultona)

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 metod Adamsa-Bashforta, otrzymamy:

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

Wartości parametrów β_j^* odczytujemy z tabeli parametrów metod niejawnych Adamsa.

Aby otrzymać rząd metody równy 4 przyjmujemy $k=3$.

Metody predyktor – korektor

Najpraktyczniejszą metodą wielokrokową byłaby metoda o:

- wysokim rzędzie i małej stałej błędu,
- możliwie dużym obszarze absolutnej stabilności,
- możliwie małej liczbie obliczeń na iterację.

Metody jawne gorzej spełniają dwa pierwsze warunki, natomiast metody niejawne spełniają je znacznie lepiej, ale nie wypełniają warunku trzeciego, gdyż w każdej iteracji trzeba rozwiązywać względem y_n równanie nieliniowe:

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

Praktyczne realizacje metod wielokrokowych to algorytmy typu predyktor – korektor (PK) będące połączeniem metod jawnych i niejawnych. Dla metody k -krokowej realizacja w postaci struktury predyktor-korektor $P_k E K_k E$ ma postać:

$$P \text{ (predykcja)} : y_n^{[0]} = \sum_{j=1}^k \alpha_j y_{n-j} + h \sum_{j=1}^k \beta_j f_{n-j},$$

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

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

$$E \text{ (ewaluacja)} : f_n = f(x_n, y_n).$$

Metoda PK to w istocie przybliżony sposób realizacji metody niejawnej (korektora). Algorytm predyktora gra tu rolę pomocniczą polegającą na efektywnym wyliczeniu dobrego punktu startowego. Dla metod Adamsa algorytm $P_k E K_k E$ ma postać:

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

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

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

$$E \text{ (ewaluacja)} : f_n = f(x_n, y_n).$$

Iteracja predyktora ma na celu obliczenie dobrego punktu początkowego dla iteracji korektora rozwiążemy nieliniowe równanie algebraiczne metody niejawnej korektora.

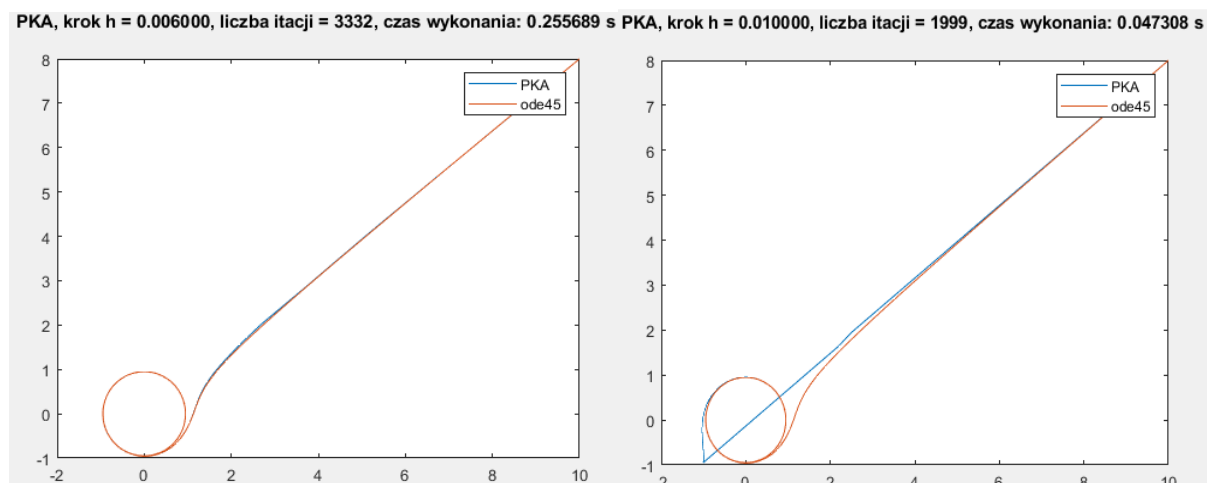
Jeżeli predyktor jest dostatecznie dokładny, to dla dostatecznie małych wartości kroku h , uzyskanie maksymalnego rzędu następuje w algorytmie PK już po jednej iteracji korektora. Natomiast dla mniej dokładnego predyktora potrzebna jest większa ilość iteracji.

Dla metody PK Adamsa z 4-etapowym predykatorem i 3-etapowym korektorem błąd metody określony jest wzorem:

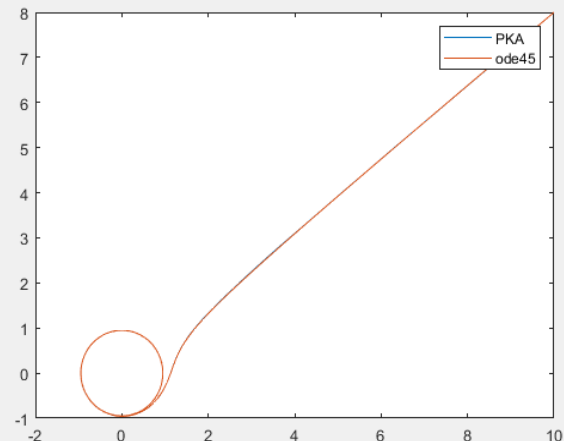
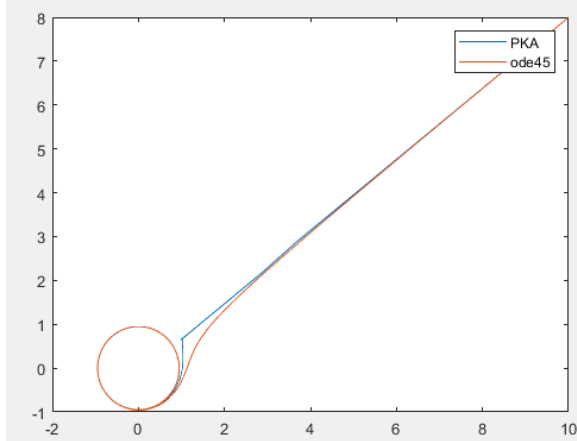
$$y_n - y(x_n) = -\frac{19}{210} (y_n^{[0]} - y_n)$$

Prezentacja wyników - metoda wielokrokowa predyktor-korektor Adamsa:

a. $x_1(0)=10$ $x_2(0)=8$



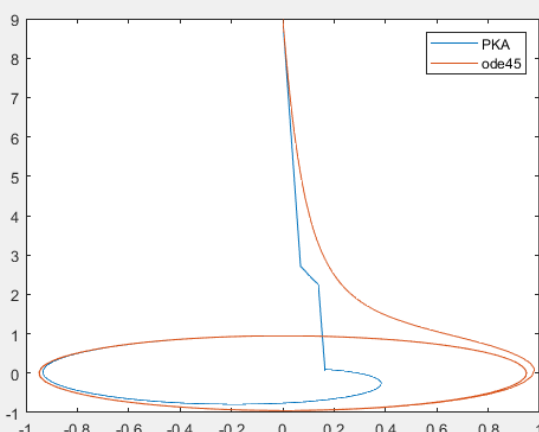
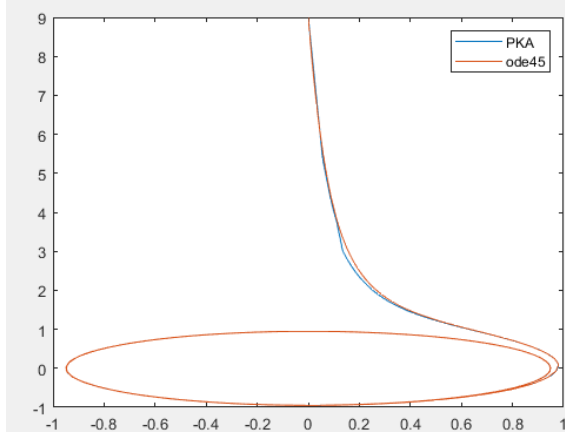
PKA, krok $h = 0.008000$, liczba itacji = 2499, czas wykonania: 0.059930 s PKA, krok $h = 0.004500$, liczba itacji = 4443, czas wykonania: 0.046959 s



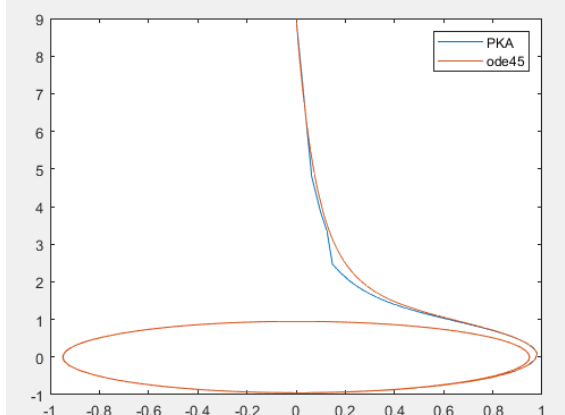
Właściwa trajektoria dla $h = 0.006$:

b. $x_1(0)=0$ $x_2(0)=9$

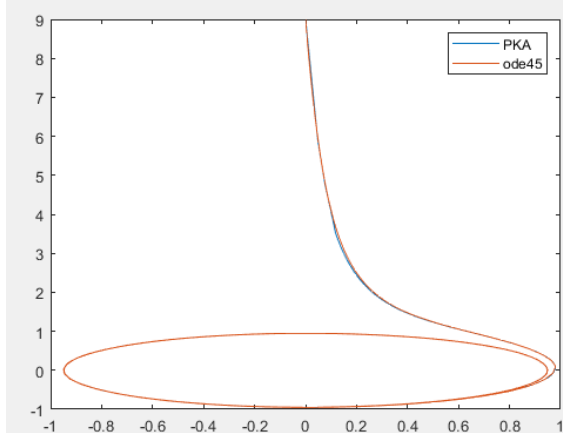
PKA, krok $h = 0.010000$, liczba itacji = 1999, czas wykonania: 0.019485 s PKA, krok $h = 0.018000$, liczba itacji = 1110, czas wykonania: 0.010566 s



PKA, krok $h = 0.012000$, liczba itacji = 1665, czas wykonania: 0.021622 s



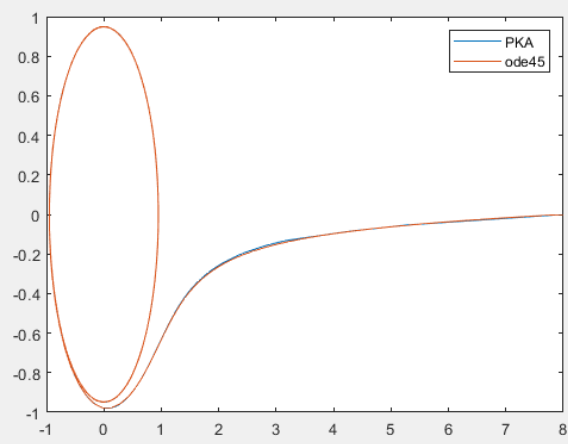
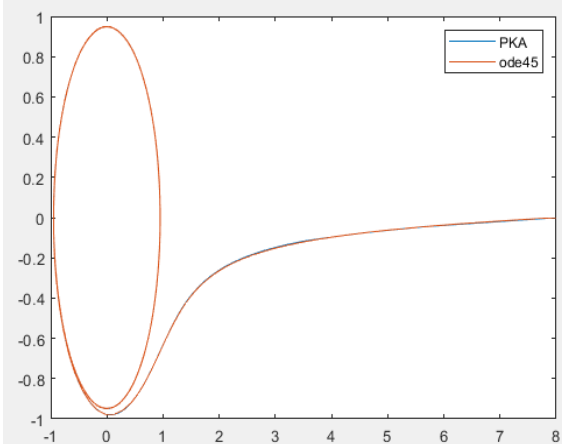
PKA, krok $h = 0.008000$, liczba itacji = 2499, czas wykonania: 0.035307 s



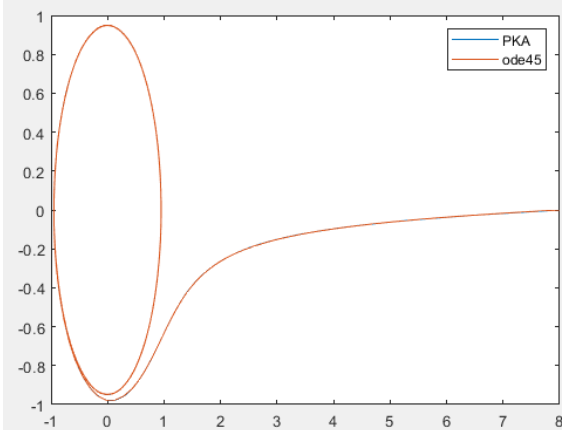
Właściwa trajektoria dla $h = 0.008$:

c. $x_1(0)=8$ $x_2(0)=0$

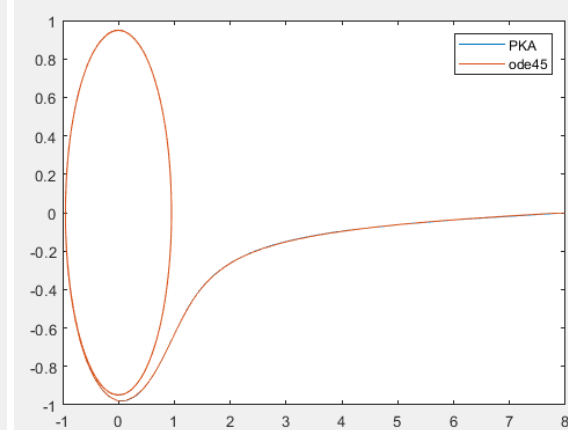
PKA, krok $h = 0.008000$, liczba itacji = 2499, czas wykonania: 0.028842 s PKA, krok $h = 0.010000$, liczba itacji = 1999, czas wykonania: 0.020354 s



PKA, krok $h = 0.004000$, liczba itacji = 4999, czas wykonania: 0.068114 s



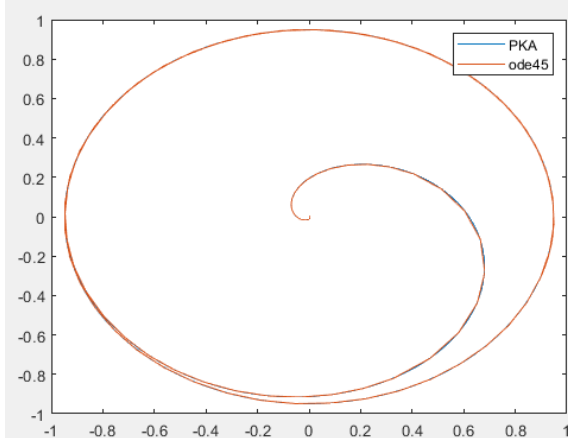
PKA, krok $h = 0.006000$, liczba itacji = 3332, czas wykonania: 0.039170 s



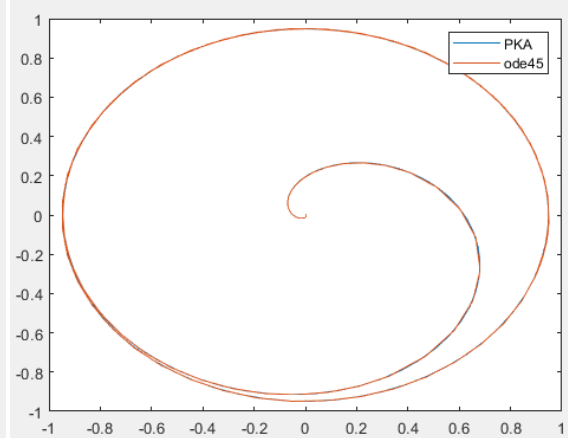
Właściwa trajektoria dla $h = 0.008$:

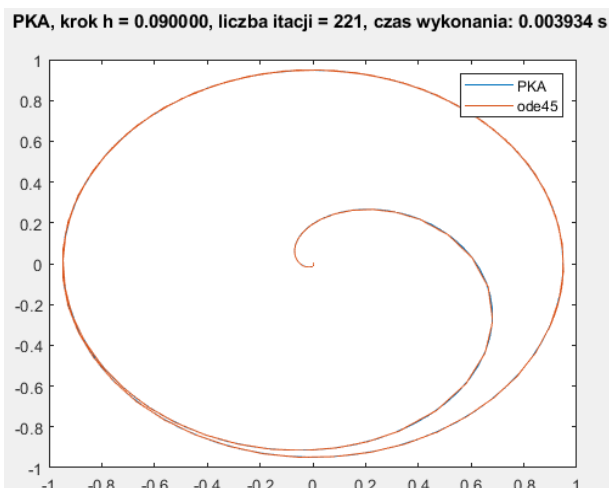
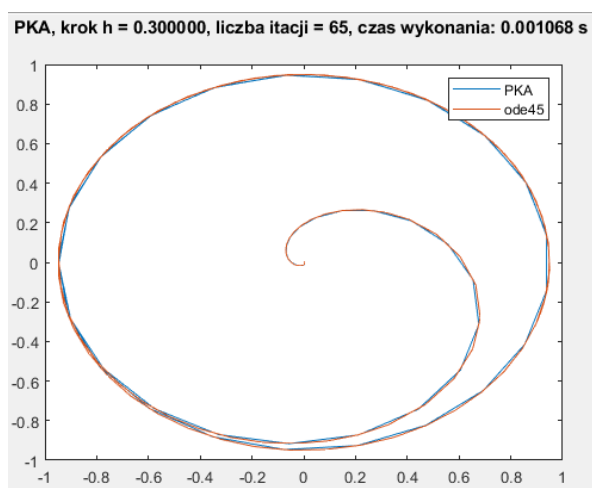
d. $x_1(0)=0.001$, $x_2(0)=0.001$

PKA, krok $h = 0.008000$, liczba itacji = 2499, czas wykonania: 0.034454 s



PKA, krok $h = 0.100000$, liczba itacji = 199, czas wykonania: 0.002730 s





Właściwa trajektoria dla $h = 0.09$:

Wynik:

	RK4		RK4Z		ADAMS	
	iteracje	czas	iteracje	czas	iteracje	czas
$x_1(0)=10$ $x_2(0)=8$	2858	0.008254	750	0.042310	3332	0.255689
$x_1(0)=0$ $x_2(0)=9$	1819	0.002476	821	0.004321	2499	0.035307
$x_1(0)=8$ $x_2(0)=0$	1429	0.002808	781	0.003675	2499	0.028842
$x_1(0)=0.001$, $x_2(0)=0.001$	134	0.000426	658	0.003977	221	0.003934

Podsumowanie:

Wyniki zostały porównywane z wbudowaną funkcją ode45. Na podstawie jej zostały dobrane parametry dla poszczególnych metod i podpunktów.

Nie ma jednej metody, która byłaby najlepsza we wszystkich czterech przypadkach.

Metoda ze zmiennym krokiem była najlepsza w 3/ 4 przypadków.
Okazała się słabsza w ostatnim przypadku.

Metody RK4 i Adamsa potrzebują podobnej liczby iteracji, jednak iteracje w tej drugiej metodzie są bardziej czasochłonne.

Metoda predyktor-korektor Adamsa, której rezultaty są dość podobne do metody RK4 ze stałym krokiem (są jednak zauważalne, iż metoda ta daje prawidłowy wynik trochę dla większych kroków niż RK4).

Metoda RK4 ze zmiennym krokiem, zmniejszająca ilość iteracji w porównaniu do metody RK4 ze stałym krokiem (rosnący krok zmniejsza liczbę iteracji). Jednak coś za coś – w każdej iteracji wykonywanych jest więcej obliczeń. Zdecydowaną zaletą tej metody jest automatycznie dobierany krok w zależności od danej trajektorii (np. w miejscach przegięcia wykresu zwiększa on swoją dokładność itd.). Najlepsza z tych metod!!

Dodatek:

```
function [y1,y2] = func(x1,x2)
%Funkcje opisujące ruch punktu
% y1 do pochodnej pierwszej
% y2 do drugiej pochodnej
y1 = x2 + x1*(0.9 - x1^2 - x2^2);
y2 = -x1 + x2*(0.9 - x1^2 - x2^2);
end
```

```
function [dydt] = odefun(t, x)
dydt = [x(2)+x(1)*(0.9-x(1)^2-x(2)^2) ; -x(1)+x(2)*(0.9-x(1)^2-
x(2)^2)];

end
```

```

function [] = RK4(x1, x2, h)
    it = 1;
    tic;
    for i = 0:h:20
        % wpisujemy do macierzy wyników
        results(it,1) = x1;
        results(it,2) = x2;
        it = it + 1;
        % obliczenie współczynników k
        [k11,k12] = func(x1,x2);
        [k21,k22] = func(x1 + 1/2 * h * k11, x2 + 1/2 * h * k12);
        [k31,k32] = func(x1 + 1/2 * h * k21, x2 + 1/2 * h * k22);
        [k41,k42] = func(x1 + h * k31, x2 + h * k32);
        x1 = x1 + (1/6) * h * (k11 + 2 * k21 + 2 * k31 + k41);
        x2 = x2 + (1/6) * h * (k12 + 2 * k22 + 2 * k32 + k42);
    end
    results(it,1) = x1;
    results(it,2) = x2;
    it = it - 1;
    t = toc;
    % porównanie z funkcją ode45 programu Matlab
    [~,y] = ode45(@odefun, [0 20], [results(1,1) results(1,2)]);
    plot(results(:,1), results(:,2));
    hold on;
    plot(y(:,1), y(:,2));
    legend('RK4', 'ode45');
    hold off;
    plotTitle = sprintf('RK4, krok h = %f, liczba iteracji = %d, czas
wykonania: %f s \n', h, it, t);
    title(plotTitle);
end

```

```

function [w,it] = RK4Z(x1,x2,h)
    it = 1; s = 0.9; % współczynnik bezpieczeństwa
    ew = 10^(-8); % dokładność względna
    eb = 10^(-8); % dokładność bezwzględna
    beta = 5; %maksymalna zmiana kroku w jednej iteracji
    hMin = 10^(-12);
    i=0;
    tic;
    while(true)
        w(it,1) = x1;
        w(it,2) = x2;
        w(it,3) = h;
        w(it,6) = i;
        [k11,k12] = func(x1,x2);
        [k21,k22] = func(x1 + 1/2 * h * k11, x2 + 1/2 * h * k12);
        [k31,k32] = func(x1 + 1/2 * h * k21, x2 + 1/2 * h * k22);
        [k41,k42] = func(x1 + h * k31, x2 + h * k32);
        x1e = x1;
        x2e = x2;
        h=h/2;
        for j=1:2
            [k11e,k12e] = func(x1e,x2e);

```

```

        [k21e,k22e] = func(x1e + 1/2 * h * k11e, x2e + 1/2 * h * k12e);
        [k31e,k32e] = func(x1e + 1/2 * h * k21e, x2e + 1/2 * h * k22e);
        [k41e,k42e] = func(x1e + h * k31e, x2e + h * k32e);
        x1e = x1e + (1/6) * h * (k11e + 2 * k21e + 2 * k31e + k41e);
        x2e = x2e + (1/6) * h * (k12e + 2 * k22e + 2 * k32e + k42e);
    end
    h=h*2;
    x1 = x1 + (1/6) * h * (k11 + 2 * k21 + 2 * k31 + k41);
    x2 = x2 + (1/6) * h * (k12 + 2 * k22 + 2 * k32 + k42);
    % błędy aproksymacji
    d1 = (x1e-x1)/15;
    d2 = (x2e-x2)/15;
    w(it,4)=d1;
    w(it,5)=d2;
    e1 = abs(x1e) * ew + eb;
    e2 = abs(x2e) * ew + eb;
    alpha = min([(e1/abs(d1))^(1/5)), ((e2/abs(d2))^(1/5))]); % 1/p+1 =
1/5
    hNext = s * alpha * h;
    x1 = x1e;
    x2 = x2e;
    if (s * alpha >= 1)
        if ( i + h >= 20 )
            break;
        else
            i = i + h;
            h = min([hNext, beta * h, 20 - i]);
        end
    else
        if (hNext < hMin)
            error('Nie można rozwiązać zadaną dokładnością');
        else
            h = hNext;
            x1 = w(it,1);
            x2 = w(it,2);
        end
    end
    end
    it = it + 1;
end
it = it - 1;
t = toc;
[~,y] = ode45(@odefun, [0 20], [w(1,1) w(1,2)]);
plot(w(:,1), w(:,2));
hold on;
plot(y(:,1),y(:,2));
legend('RK4', 'ode45');
hold off;
plotTitle = sprintf('RK4Z liczba iteracji = %d, czas wykonania: %f s
\n', it, t);
title(plotTitle);
end

```

```

function Adams(x1,x2,h)
    it = 1;
    tic;
    for i = 1:3
        % wpisujemy do macierzy wyników
        results(it,1) = x1;
    end

```

```

results(it,2) = x2;
it = it + 1;
% obliczenie współczynników k
[k11,k12] = func(x1,x2);
[k21,k22] = func(x1 + 1/2 * h * k11, x2 + 1/2 * h * k12);
[k31,k32] = func(x1 + 1/2 * h * k21, x2 + 1/2 * h * k22);
[k41,k42] = func(x1 + h * k31, x2 + h * k32);
x1 = x1 + (1/6) * h * (k11 + 2 * k21 + 2 * k31 + k41);
x2 = x2 + (1/6) * h * (k12 + 2 * k22 + 2 * k32 + k42);
end
for i = 5:floor(20/h)
    results(it,1) = x1;
    results(it,2) = x2;
    x = [x1 x2]';
    sigma = ((55/24) * odefun(0,[x1 x2]) - (59/24) *
odefun(0,[results(it-1,1) results(it-1,2)]) + (37/24) *
odefun(0,[results(it-2,1) results(it-2,2)]) - (9/24) *
odefun(0,[results(it-3,1) results(it-3,2)])));
    pe = x + h * sigma;
    sigma = ((9/24) * odefun(0,pe) + (19/24) * odefun(0,[x1,x2]) -
(5/24) * odefun(0,[results(it-1,1) results(it-1,2)]) + (1/24) *
odefun(0,[results(it-2,1) results(it-2,2)])));
    x = x + h * sigma;
    x1 = x(1);
    x2 = x(2);
    it = it+1;
end
results(it,1) = x1;
results(it,2) = x2;
it = it - 1;
t = toc;
% porównanie z funkcją ode45 programu Matlab
[~,y] = ode45(@odefun, [0 20], [results(1,1) results(1,2)]);
plot(results(:,1), results(:,2));
hold on;
plot(y(:,1),y(:,2));
legend('PKA', 'ode45');
hold off;
plotTitle = sprintf('PKA, krok h = %f, liczba itacji = %d, czas
wykonania: %f s \n', h, it, t);
title(plotTitle);
end

```