

## MNUM Projekt 4.

Autor:	Numer projektu:	Prowadzący:	Liczba punktów:
Mariusz Słapek	4.42	dr inż. Adam Krzemienowski	

Kody do poszczególnych programów zostały umieszczone na końcu sprawozdania.

### Spis treści

Zadanie 1 .....	2
Treść: .....	2
Ogólny zarys: .....	2
Metoda Rungego-Kutty ze stałym krokiem: .....	4
Prezentacja wyników - metoda Rungego-Kutty ze stałym krokiem: .....	5
Metoda predyktor-korektor Adamsa czwartego rzędu .....	10
Prezentacja wyników - metoda wielokrokowa predyktor-korektor Adamsa: .....	11
Metoda Rungego-Kutty ze zmiennym krokiem: .....	15
Prezentacja wyników - metoda RK4 ze zmiennym krokiem: .....	16
Wnioski: .....	19
Kod do programów.....	21
Zadanie 1: .....	21
Kształt za pomocą programu: .....	25
.....	25

## Zadanie 1

### Treść:

Ruch punktu jest opisany równaniami. Należy obliczyć przebieg trajektorii ruchu na danych przedziale dla poszczególnych warunków początkowych.

Należy wykorzystać dane metody:

- a) Rungego-Kutty czwartego rzędu (RK4) ze stałym krokiem.
- b) Wielokrokowej predyktor-korektor Adamsa czwartego rzędu ze stałym krokiem.
- c) Rungego-Kutty czwartego rzędu (RK4) ze zmiennym krokiem.

### Ogólny zarys:

Równania różniczkowe są powszechnie stosowane do modelowania fizycznych układów dynamicznych. Dane układy równań są zazwyczaj równaniami nieliniowymi ze względu na nieliniowość świata w którym żyjemy.

Metoda numeryczna (jedne ze sposobów rozwiązywania tych układów równań) rozwiązania tego zagadnienia stanowią podstawę algorytmiczną pakietów *symulacji ciągłych układów dynamicznych*.

Wyróżnia się następujące metody numerycznego rozwiązywania równań różniczkowych:

- a) metody jednokrokowe;
- b) metody wielokrokowe.

Metody numeryczne znajdowania rozwiązań układu równań różniczkowych są metody różnicowe (przybliżona wartość rozwiązania obliczana jest w kolejnych, dyskretnych punktach).

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:  $D = \{(x, y): a \leq x \leq b, y \in \mathbb{R}\}$

oraz 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.

## 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).

Metody typu Rungego-Kutty są metodami samosterującymi, tzn. znajomość warunku początkowego wystarcza, by rozpocząć obliczenia. Niestety są to też raczej metody kosztowne czasowo (wymaga wielokrotnego obliczania wartości funkcji).

Największe znaczenie praktyczne mają metody czwartego rzędu, tzn.:

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

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

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

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

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

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

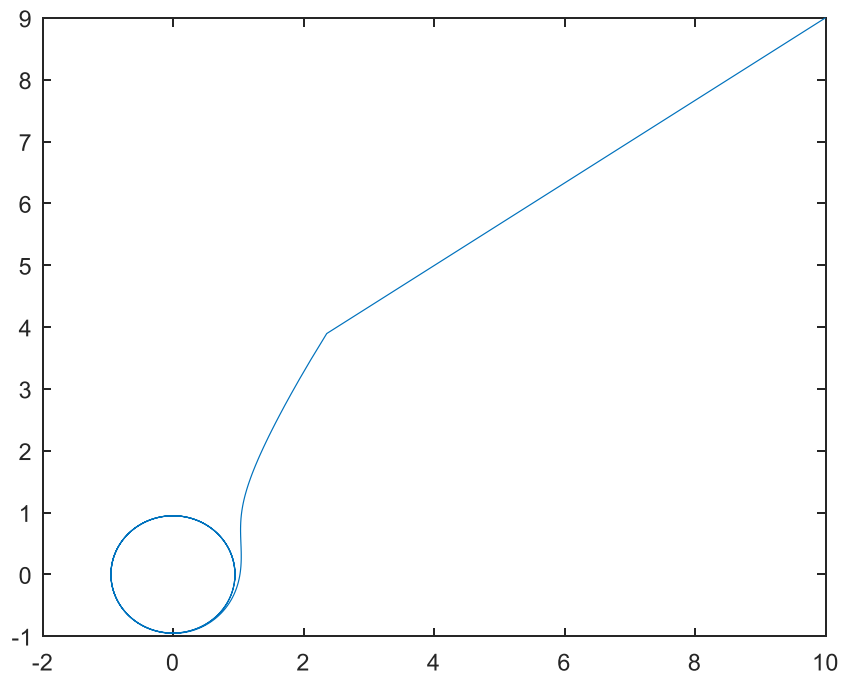
Wybór kroku jest jednym z trudniejszych zadań w tej metodzie. Krok powinien być wystarczający do uzyskania założonej dokładności, lecz nie powinien być znacznie mniejszy przy której wymagana dokładność jest osiągnięta.

Prezentacja wyników- metoda Rungego-Kutty ze stałym krokiem:

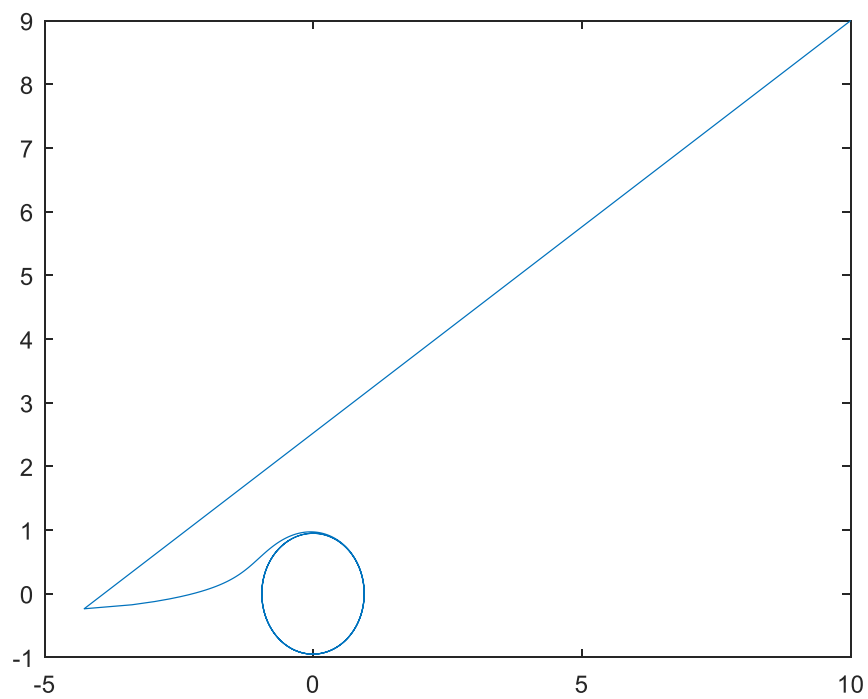
Poszczególne podpunkty oznaczają wyniki dla danych warunków początkowych.

a)

Właściwa trajektoria dla  $h = 0.006$ :

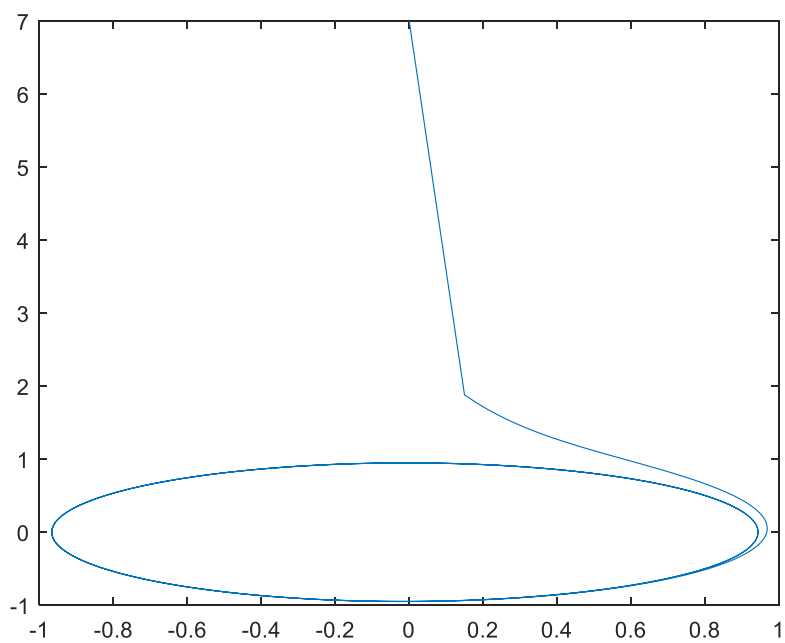


Błędna trajektoria dla  $h = 0.03$ :

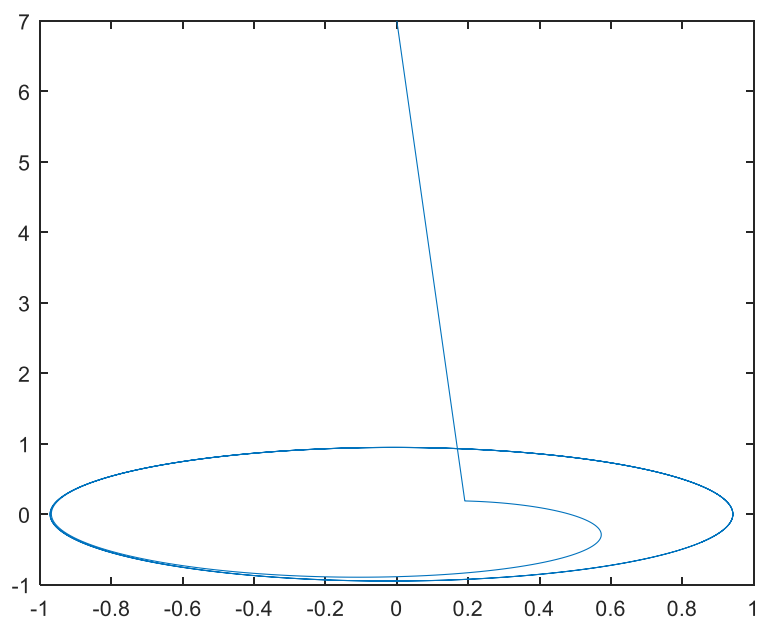


b)

Właściwa trajektoria dla  $h = 0.23$ :

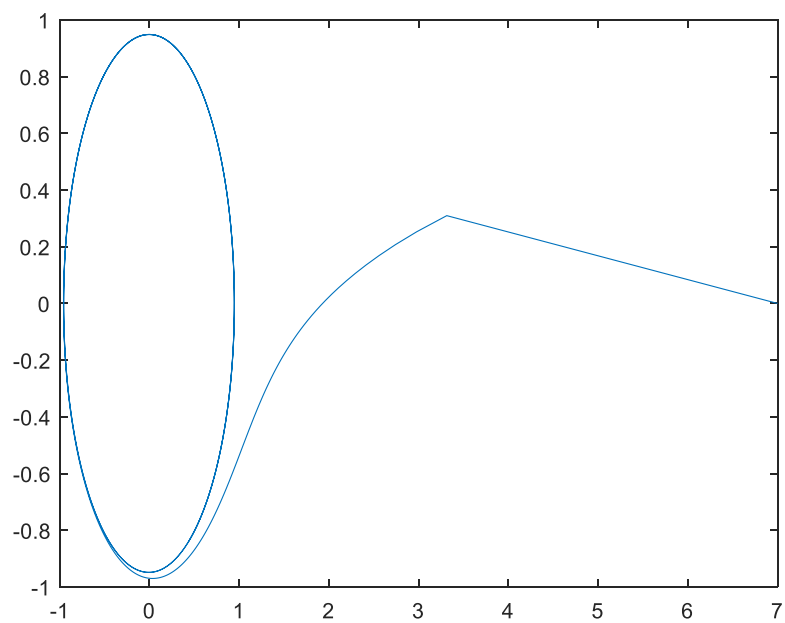


Błędna trajektoria dla  $h = 0.3$ :

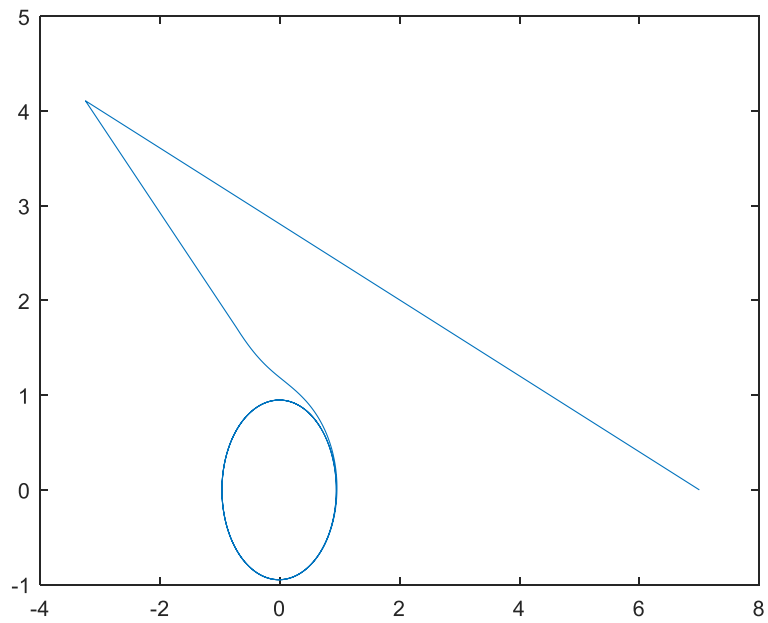


c)

Właściwa trajektoria dla  $h = 0.01$ :

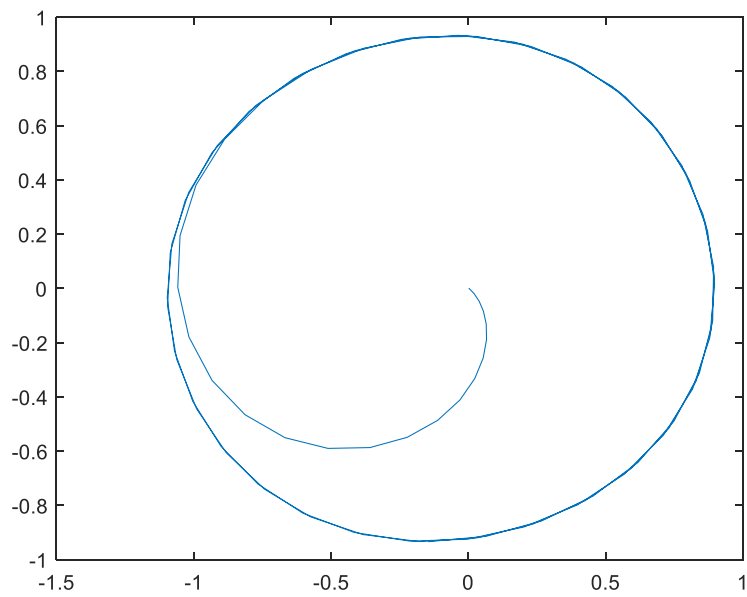


Niewłaściwa trajektoria dla  $h = 0.2$ :



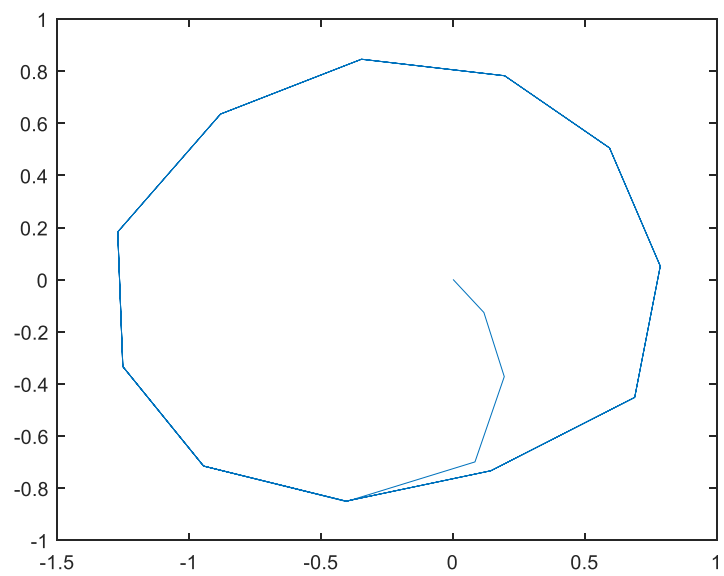
d)

Właściwa trajektoria dla  $h = 0.2$ :



Niewłaściwa trajektoria dla  $h = 0.5$ :





## Metoda predyktor-korektor Adamsa czwartego rzędu

Jest to jedna z metod wielokrokowych. Do metod wielokrokowych potrzebujemy pierwszych  $k$  – pierwszych wartości rozwiązania. W tym celu możemy posłużyć się już znaną metodą Rungego-Kutty.

Ogólnie metoda ta opiera się na algorytmach predyktor – korektor, który jest połączeniem metody jawnej i niejawnej. W metodach predyktor-korektor wykorzystujemy oba sposoby: najpierw używamy metody jawnej, aby obliczyć dobry punkt początkowy, a następnie wykorzystujemy metody niejawne. Ma on następujące zalety:

- a) wysoki rząd i mała stała błędów,
- b) możliwie duży obszar absolutnej stabilności,
- c) możliwie mała liczba obliczeń na iterację.

Dla metody  $k$ -krokowej realizacja w postaci struktury PK:

$$\begin{aligned} P: y_n^{[0]} &= \sum_{j=0}^k \alpha_j y_{n-j} + h \sum_{j=1}^k \beta_j f_{n-j} \\ E: f_n^{[0]} &= f(x_n, y_n^{[0]}) \\ K: y_n &= \sum_{j=0}^k \alpha_j^* y_{n-j} + h \sum_{j=1}^k \beta_j^* f_{n-j} + h \beta_0^* f_n^{[0]} \\ E: f_n &= f(x_n, y_n) \end{aligned}$$

,gdzie:

P – predykcja

E – ewaluacja

K – korekcja

Iteracja predyktora ma na celu obliczenie dobrego punktu początkowego dla iteracji korektora rozwiązujemy 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.

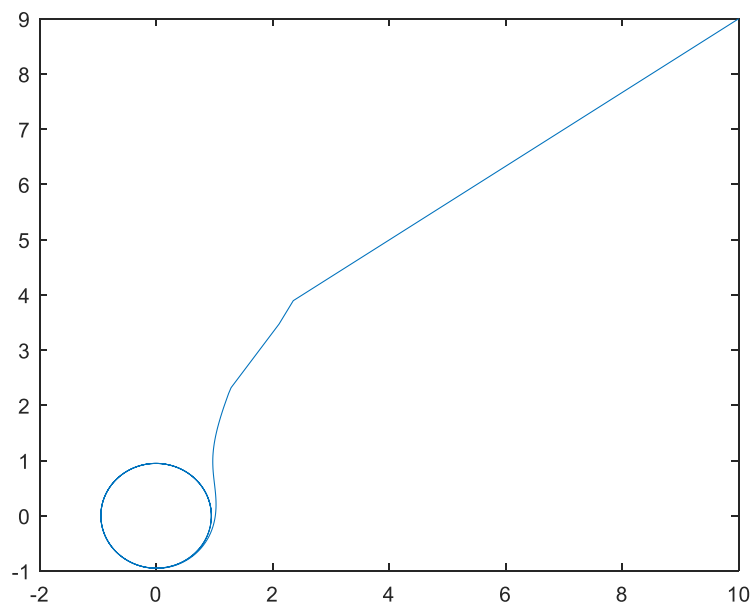
Dla metody PK Adamsa z 4-etapowym predykatorem i 3-etapowym korektorem błąd metody określony jest wzorem:

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

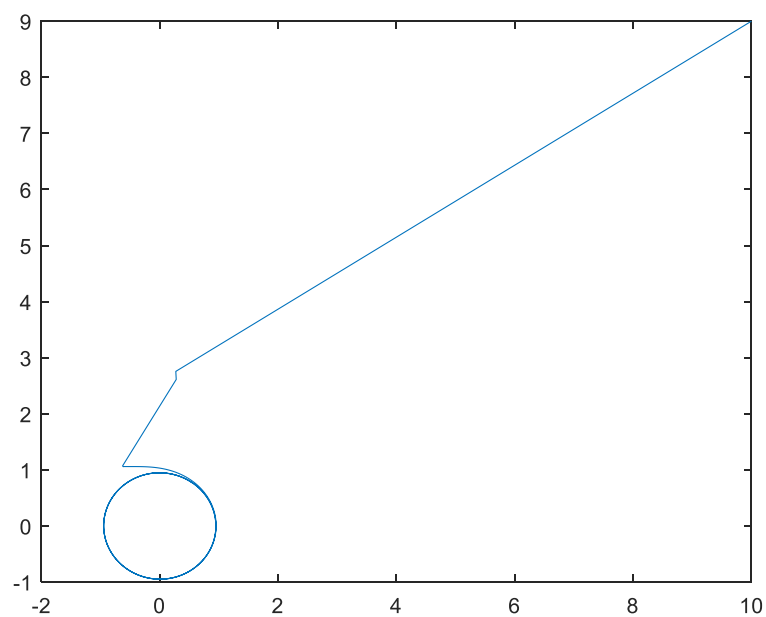
## Prezentacja wyników- metoda wielokrokowa predyktor-korektor Adamsa:

a)

Właściwa trajektoria dla  $h = 0.006$ :

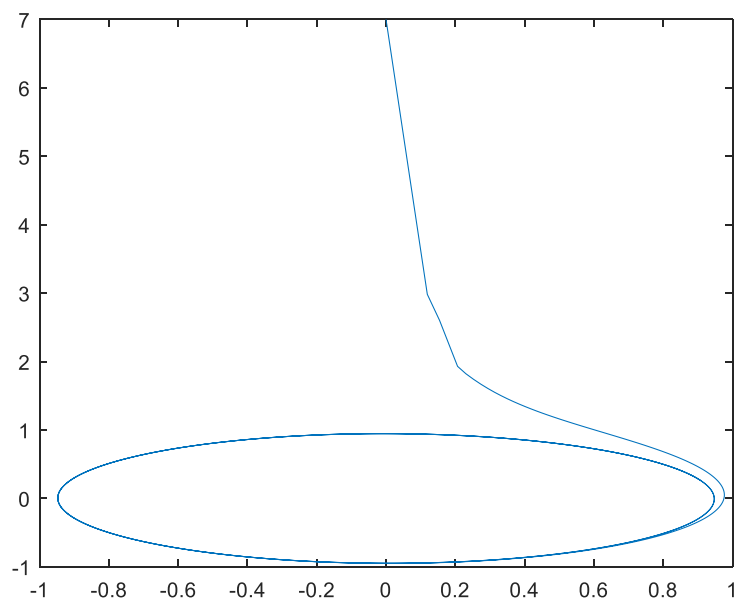


Niewłaściwa trajektoria dla  $h = 0.008$ :

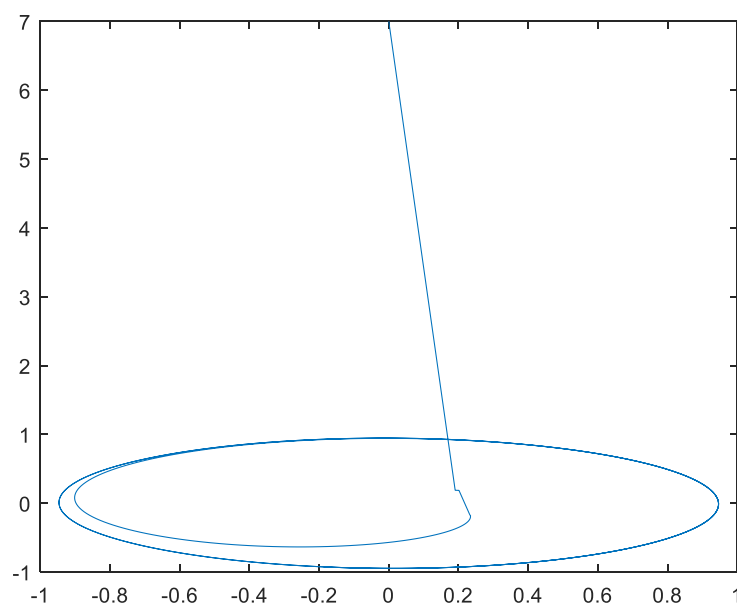


b)

Właściwa trajektoria dla  $h = 0.018$ :

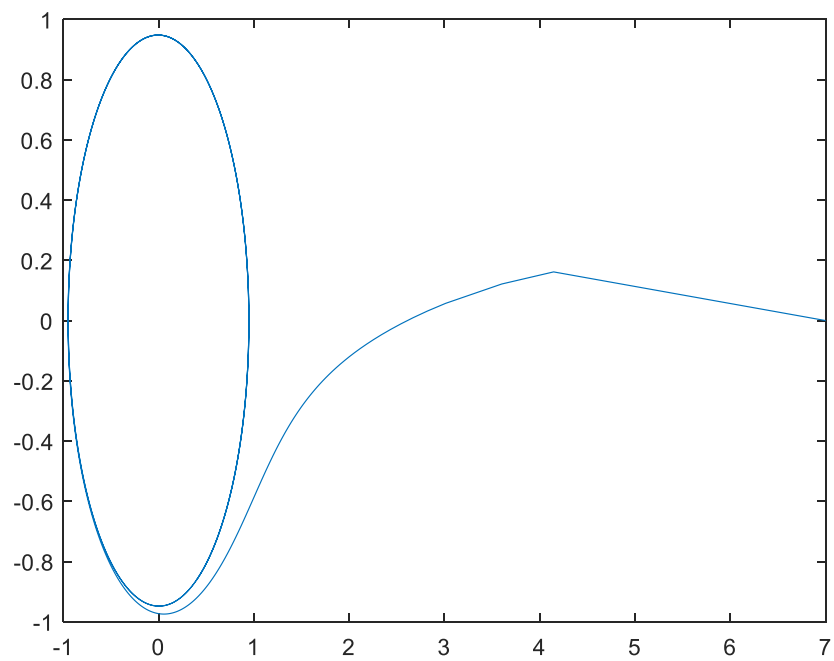


Niewłaściwa trajektoria dla  $h = 0.03$ :

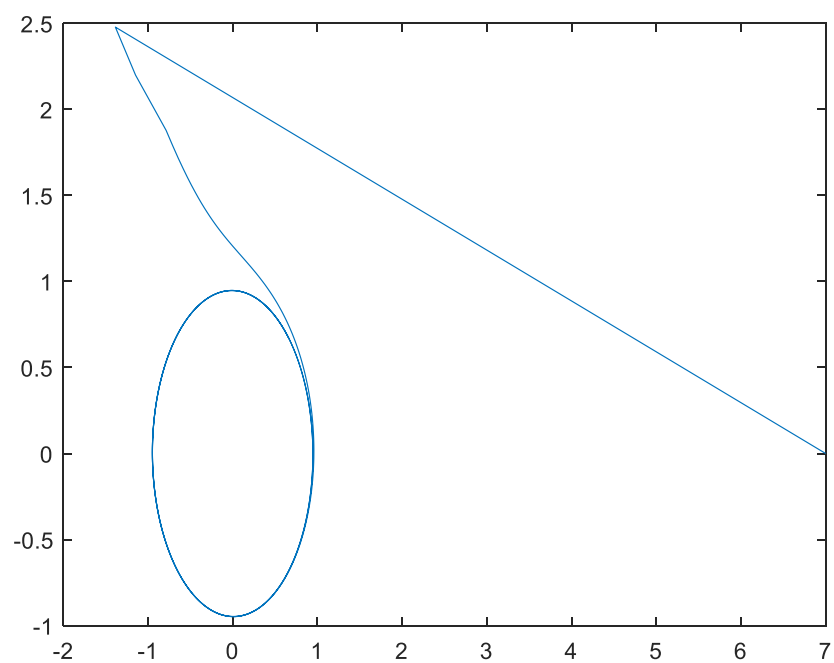


c)

Właściwa trajektoria dla  $h = 0.008$ :

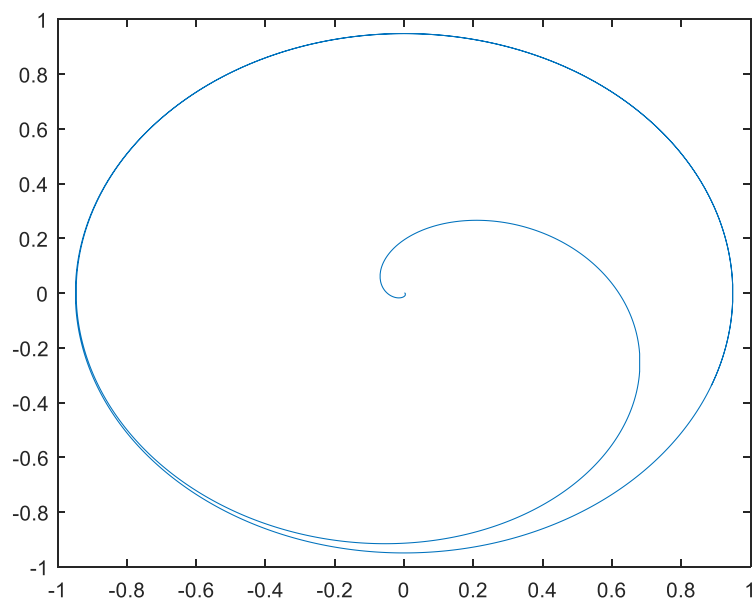


Niewłaściwa trajektoria dla  $h = 0.018$ :

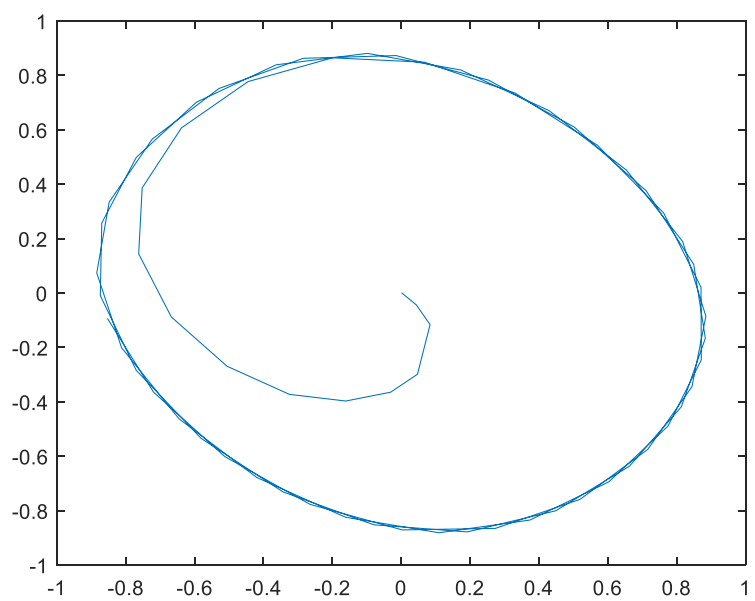


d)

Właściwa trajektoria dla  $h = 0.1$ :



Niewłaściwa trajektoria dla  $h = 0.3$ :



## 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  $h/2$ ,

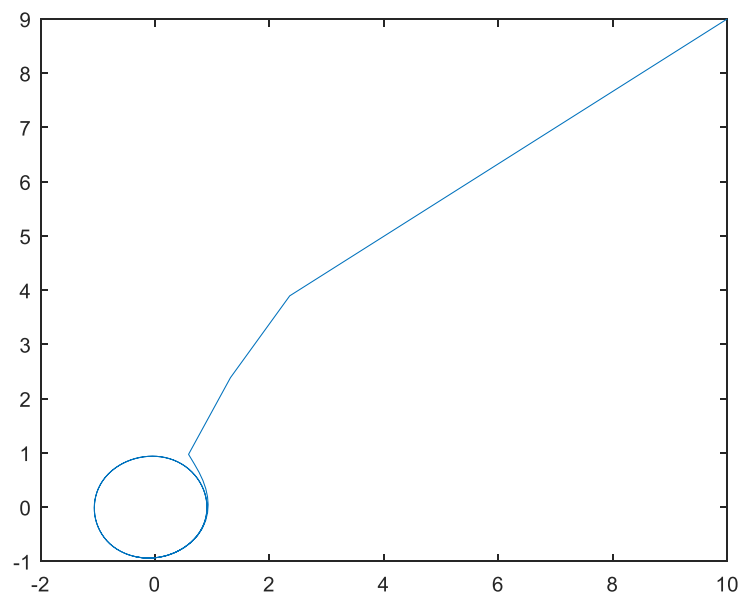
$y_n^{(1)}$  – nowy punkt wyznaczony metodą z krokiem o długości  $h$ .

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.

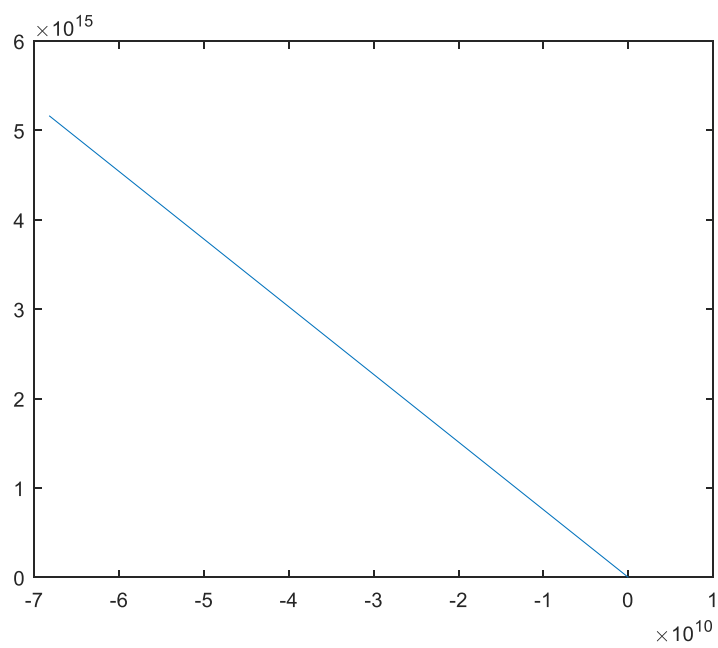
## Prezentacja wyników- metoda RK4 ze zmiennym krokiem:

a)

Właściwa trajektoria dla  $h = 0.009$ :



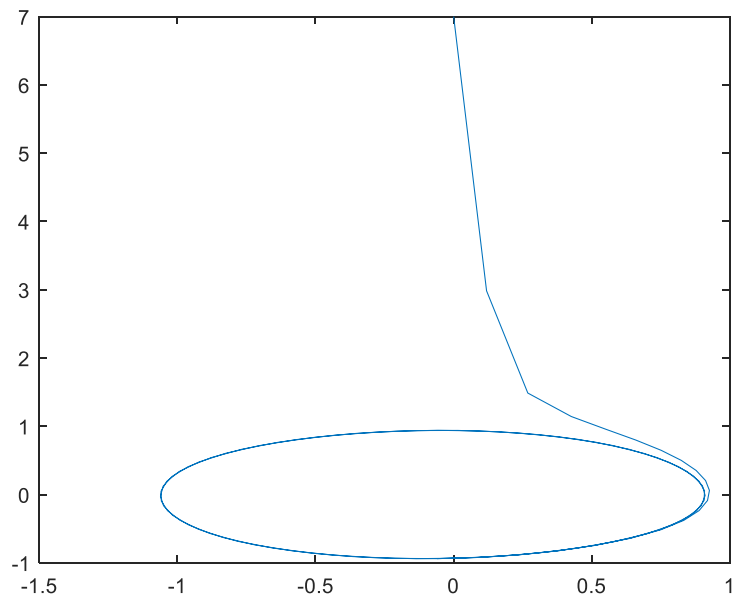
Niewłaściwa trajektoria dla  $h = 0.07$ :



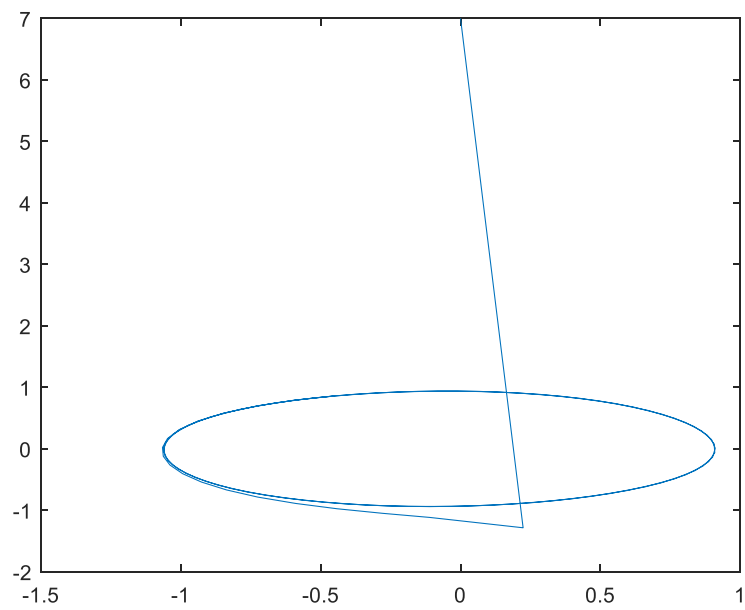


b)

Właściwa trajektoria dla  $h = 0.025$ :

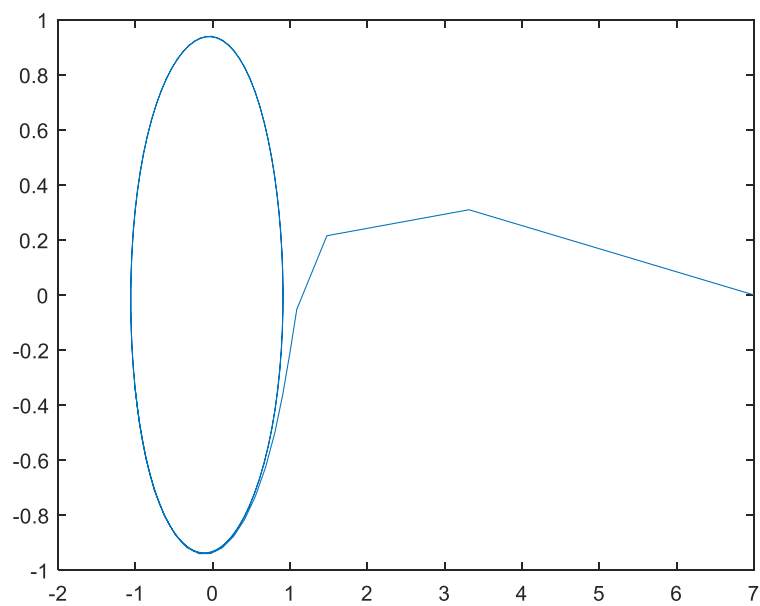


Niewłaściwa trajektoria dla  $h = 0.36$ :

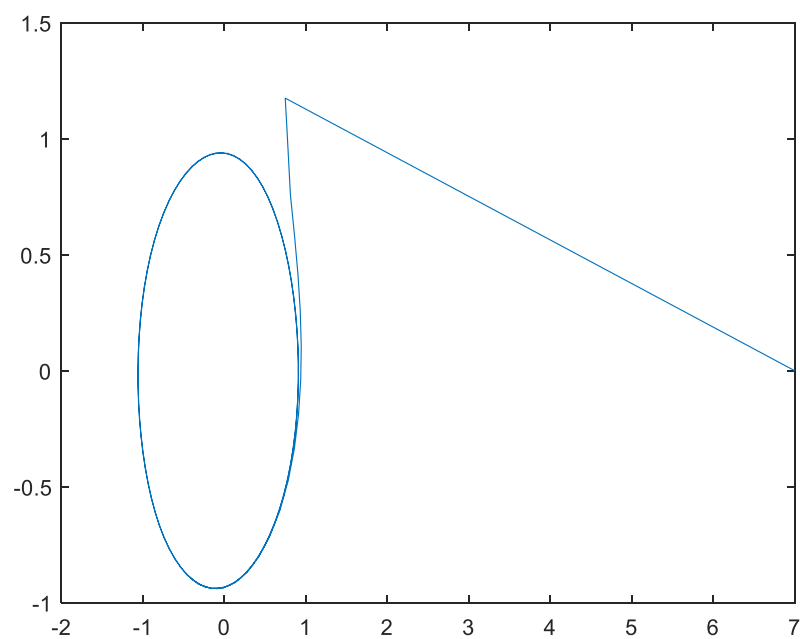


c)

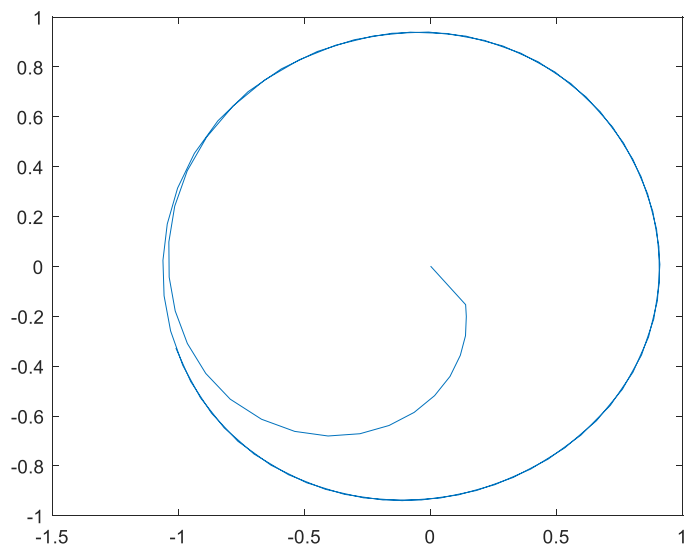
Właściwa trajektoria dla  $h = 0.012$ :



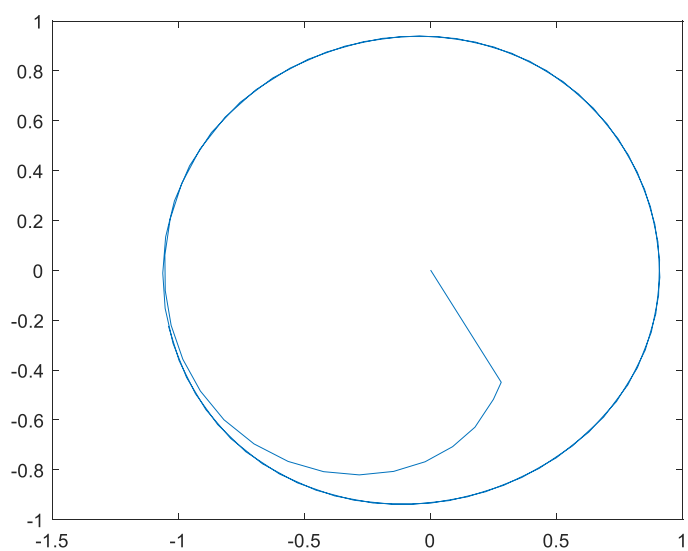
Niewłaściwa trajektoria dla  $h = 0.015$ :



d)  
Właściwa trajektoria dla  $h = 0.55$ :



Niewłaściwa trajektoria dla  $h = 0.7$ :



## Wnioski:

Wyniki sprawdziłem z wbudowaną funkcją ode45 i są one z nią zgodne.

Najbardziej dokładną metodą jest metoda RK4 ze zmiennym krokiem. Czasy działania są bardzo porównywalne, jednak metoda RK4 ze stałym krokiem, która nie należy do zbyt najbardziej dokładnych, czasowo wypada najbardziej korzystnie.

### Metoda RK4 ze stałym krokiem:

Widzimy, iż duże znaczenie w ten metodzie ma wybranie punktu początkowego ma na trajektorię ruchu oraz błędy do długości kroku. Gdy punkt początkowy jest „dość” blisko początku układu współrzędnych możemy wybierać stosunkowo duże długości kroku. Natomiast jeśli punkt ten znajduje się dalej konieczne jest wybranie kroku o większej długości.

### Metoda predyktor-korektor Adamsa:

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

### Metoda RK4 ze zmiennym krokiem:

Metoda 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.

## Kod do programów

### Zadanie 1:

#### a) funkcje:

```
function [fx1] = fx1(x1, x2)
    fx1 = x2 + x1*(0.9-(x1)^2-(x2)^2);
end

function [fx2] = fx2(x1,x2)
    fx2 = -x1 + x2*(0.9-(x1)^2-(x2)^2);
end
```

#### b) metoda Rungego-Kutty ze stałym krokiem:

```
function[fun] = rk4()
%fl,f2 - funkcje z zadania %
% h -krok %
% x1, x2 - warunki poczatkowe %

h = 0.5; %krok
x1=0.001; %pierwszy pkt
x2=0.001; %drugi pkt
%% 10 9; 0 7; 7 0; 0.001 0.001 - dane warunki początkowe
tic;
t=0:h:20; %liczba kroków

fun(:,1) = [x1 x2]; %dane do wykresu

for i=1:(length(t)-1)
    k11=fx1(x1,x2);
    k12=fx2(x1,x2);

    k21=fx1(x1+0.5*h,x2+0.5*h*k11);
    k22=fx2(x1+0.5*h,x2+0.5*h*k12);

    k31=fx1(x1+0.5*h,x2+0.5*h*k21);
    k32=fx2(x1+0.5*h,x2+0.5*h*k22);

    k41=fx1(x1+h,x2+h*k31);
    k42=fx2(x1+h,x2+h*k31);

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

    fun(:,i+1)=[x1 x2]; %zapis danych
end
plot(fun(1,:),fun(2,:));
toc;
end
```

#### c) metoda wielokrokowa predyktor-korektor Adamsa czwartego rzędu:

```
function [fun] = pcAdamsa()
tic;
```

```

% f1, f2 - funkcje z zadania %
% h - krok %
% x1, x2 - warunki początkowe %

h = 0.006; %krok
x1=10; %pierwszy pkt
x2=9; %drugi pkt
%% 10 9; 0 7; 7 0; 0.001 0.001 - dane warunki początkowe

t=0:h:20; %ilość kroków

fun(:,1) = [x1 x2]; %zapis pierwszych danych do wykresu

for i=1:3
    k11=fx1(x1,x2);
    k12=fx2(x1,x2);

    k21=fx1(x1+0.5*h,x2+0.5*h*k11);
    k22=fx2(x1+0.5*h,x2+0.5*h*k12);

    k31=fx1(x1+0.5*h,x2+0.5*h*k21);
    k32=fx2(x1+0.5*h,x2+0.5*h*k22);

    k41=fx1(x1+h,x2+h*k31);
    k42=fx2(x1+h,x2+h*k31);

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

    fun(:,i+1)=[x1 x2]; %zapisanie danych
end
for i = 4:(length(t))

    tmp1 = x1 + (h/24)*55*fx1(x1,x2) - 59*(h/24)*fx1(fun(1,i-1),fun(2,i-1))
+ 37*(h/24)*fx1(fun(1,i-2),fun(2,i-2)) - 9*(h/24)*fx1(fun(1,i-3),fun(2,i-3));
    tmp2 = x2 + (h/24)*55*fx2(x1,x2) - 59*(h/24)*fx2(fun(1,i-1),fun(2,i-1))
+ 37*(h/24)*fx2(fun(1,i-2),fun(2,i-2)) - 9*(h/24)*fx2(fun(1,i-3),fun(2,i-3));

    x1 = x1 + (h/720)*646*fx1(x1,x2) - 264*(h/720)*fx1(fun(1,i-1),fun(2,i-1))
+ 106*(h/720)*fx1(fun(1,i-2),fun(2,i-2)) - 19*(h/720)*fx1(fun(1,i-3),fun(2,i-3))
+ h*(251/720)*fx1(tmp1, tmp2);
    x2 = x2 + (h/720)*646*fx2(x1,x2) - 264*(h/720)*fx2(fun(1,i-1),fun(2,i-1))
+ 106*(h/720)*fx2(fun(1,i-2),fun(2,i-2)) - 19*(h/720)*fx2(fun(1,i-3),fun(2,i-3))
+ h*(251/720)*fx2(tmp1, tmp2);

    fun(:,i)=[x1 x2];
end
plot(fun(1,:),fun(2,:));

toc;
end

```

d) metoda RK4 ze zmiennym krokiem

```

% h -krok

```

```
% x1, x2 - warunki poczatkowe
```

```
function [fun] = rk_zmienny()
    i = 1;
    epsw = 0.000001;
    epsb = 0.000001;

    %menu
    h = 0.9;
    x1=0.001; %pierwszy pkt
    x2=0.001; %drugi pkt

    fun(:,i)=[x1 x2];
    t(i)=0;
    while(t(i)<=15)

        k11=fx1(x1,x2);
        k12=fx2(x1,x2);

        k21=fx1(x1+0.5*h,x2+0.5*h*k11);
        k22=fx2(x1+0.5*h,x2+0.5*h*k12);

        k31=fx1(x1+0.5*h,x2+0.5*h*k21);
        k32=fx2(x1+0.5*h,x2+0.5*h*k22);

        k41=fx1(x1+h,x2+h*k31);
        k42=fx2(x1+h,x2+h*k31);

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

        fun(:,i+1)=[x1 x2];

        %1. krok
        h=0.5*h;
        k11=fx1(x1,x2);
        k12=fx2(x1,x2);

        k21=fx1(x1+0.5*h,x2+0.5*h*k11);
        k22=fx2(x1+0.5*h,x2+0.5*h*k12);

        k31=fx1(x1+0.5*h,x2+0.5*h*k21);
        k32=fx2(x1+0.5*h,x2+0.5*h*k22);

        k41=fx1(x1+h,x2+h*k31);
        k42=fx2(x1+h,x2+h*k31);

        tmp1=x1+(h/6)*(k11+k41+2*(k21+k31));
        tmp2=x2+(h/6)*(k12+k42+2*(k22+k32));

        % 2. krok
        k11=fx1(tmp1,tmp2);
        k12=fx2(tmp1,tmp2);
        k21=fx1(tmp1+0.5*h,tmp2+0.5*h*k11);
        k22=fx2(tmp1+0.5*h,tmp2+0.5*h*k12);
        k31=fx1(tmp1+0.5*h,tmp2+0.5*h*k21);
        k32=fx2(tmp1+0.5*h,tmp2+0.5*h*k22);
```

```

k41=fx1(tmp1+h,tmp2+h*k31);
k42=fx2(tmp1+h,tmp2+h*k31);
tmp1=tmp1+(h/6)*(k11+k41+2*(k21+k31));
tmp2=tmp2+(h/6)*(k12+k42+2*(k22+k32));

h=2*h;

delta1=(tmp1-x1)/15;
delta2=(tmp2-x2)/15;
eps1=abs(tmp1)*epsw+epsb;
eps2=abs(tmp2)*epsw+epsb;
alpha1=(eps1/abs(delta1*(h^5)))^(1/5);
alpha2=(eps2/abs(delta2*(h^5)))^(1/5);
if (alpha1>alpha2)
    alpha=alpha2;
else
    alpha=alpha1;
end;
hp = 0.9*alpha*h;
if(t(i) + h >= 15)
    break
else
    t(i+1) = t(i)+h;
    A = [hp, 5*h, 15-t(i)];
    h = min(A);
    i=i+1;
end
end

%narysowanie wykresu
plot(fun(1,:),fun(2,:));

end

```



Kształt za pomocą programu:

