Metody Numeryczne w inżynierii

Rozwiązywanie równań różniczkowych zwyczajnych
-
zagadnienie początkowe
-
z wykorzystaniem wybranych języków programowania

Spr. 2

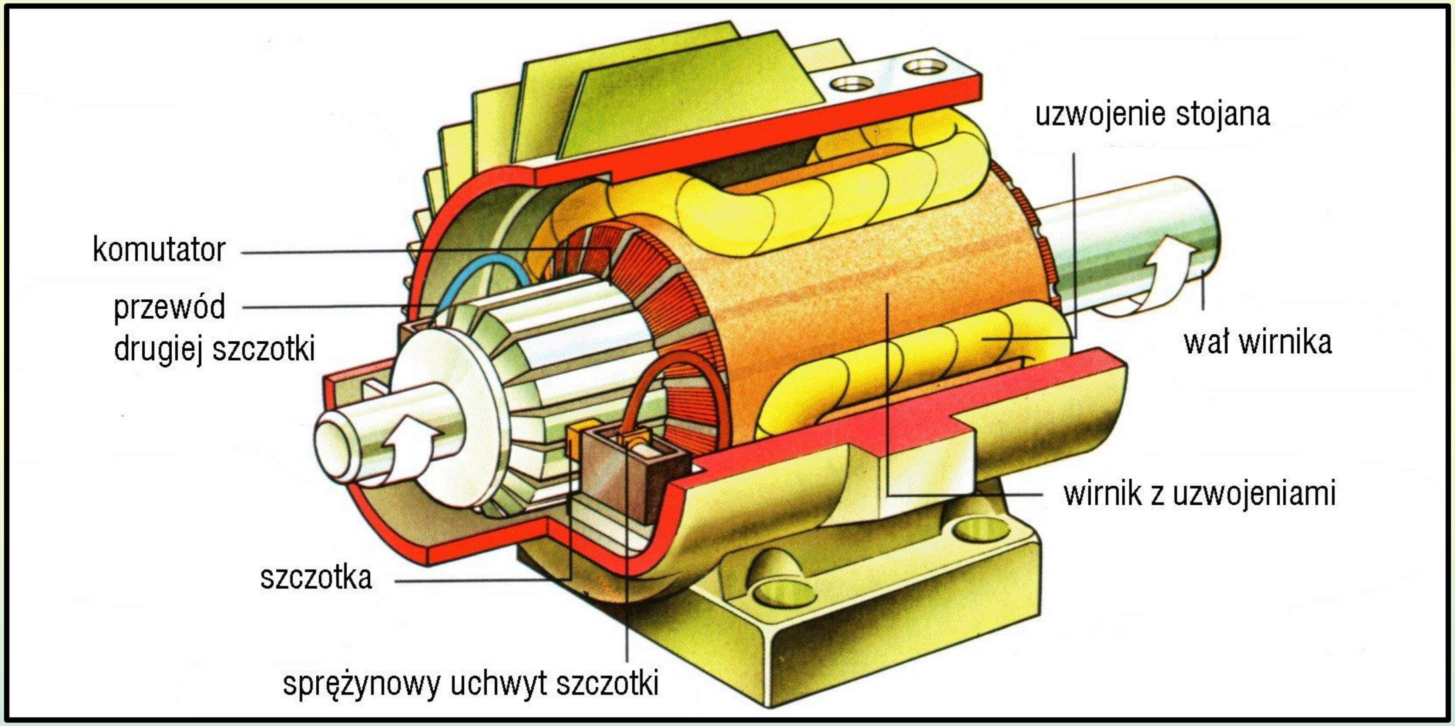
Silnik elektryczny obcowzbudny prądu stałego (DC)

To maszyna, w której uzwojenie wzbudzenia na stojanie zasilane jest z oddzielnego, niezależnego źródła napięcia niż uzwojenie twornika (wirnika). Umożliwia to precyzyjną, niezależną regulację prędkości obrotowej i momentu, zapewniając stabilną pracę, często wykorzystywaną w zaawansowanych napędach.

Kluczowe cechy i właściwości:

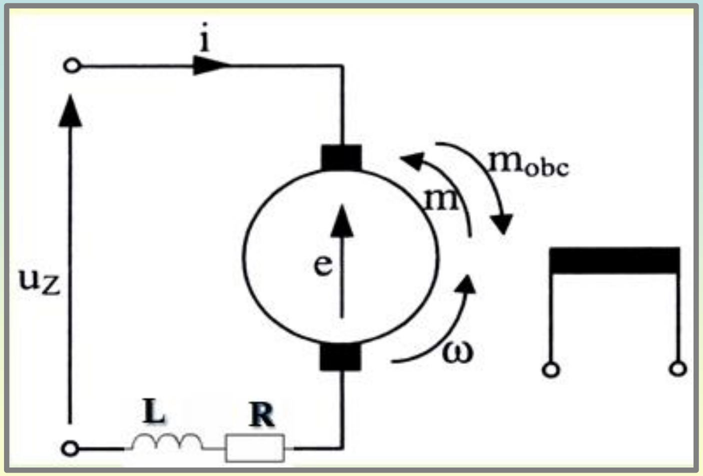
- **Niezależne zasilanie:** Uzwojenie wzbudzenia i twornika są zasilane z różnych źródeł, co odróżnia go od silników samowzbudnych (szeregowych lub bocznikowych).
- **Regulacja prędkości:** Prędkość obrotową reguluje się zazwyczaj poprzez zmianę napięcia twornika lub regulację prądu wzbudzenia.
- **Charakterystyka mechaniczna:** Silnik obcowzbudny posiada zazwyczaj sztywną charakterystykę mechaniczną, co oznacza, że prędkość obrotowa jest mało podatna na zmiany obciążenia.
- **Moment obrotowy:** Moment obrotowy jest wprost proporcjonalny do prądu twornika.
- **Zastosowanie:** Stosowany tam, gdzie wymagana jest szeroka i precyzyjna regulacja prędkości obrotowej, np. w systemach automatyki, obrabiarkach czy dźwigach.

Silnik elektryczny obcowzbudny prądu stałego (DC)
Rysunek poglądowy



Model matematyczny obcowzbudnego silnika elektrycznego prądu stałego

Sposób działania obcowzbudnego silnika prądu stałego opisuje prosty układ liniowych równań różniczkowych i algebraicznych postaci:



$$
\begin{cases}
\begin{cases}
u_z = R * i(t) + L * \dfrac{di(t)}{dt} + e(t) \\
e(t) = k_e * \omega(t)
\end{cases} \\
\begin{cases}
m(t) = m_{obc} + m_{dyn}(t) \\
m(t) = k_m * i(t) \\
m_{dyn}(t) = J * \dfrac{d\omega(t)}{dt}
\end{cases}
\end{cases}
$$

warunki początkowe: $i(t=0) = i_0 \qquad \omega(t=0) = \omega_0$

Symbole literowe w równaniach oznaczają:

$u_z,\ m_{obc}$	–	**napięcie zasilania i moment obciążenia silnika**
$i,\ \omega$	–	**prąd i prędkość kątowa wału silnika**
R, L, J, k_e, k_m	–	**stałe silnika (dane znamionowe)**
e	–	**siła elektromotoryczna indukowana w tworniku**
$m,\ m_{dyn}$	–	**moment obrotowy i moment dynamiczny silnika**
$i_0 = 0 \quad \omega_0 = 0$		**warunki początkowe**

Dane znamionowe:

R = 0.465 [Ω]

L = 0.015345 [H]

J = 2.7 [kg*m^2]

k_e = 2.62 [V*s]

k_m = 2.62 [Nm/A]

Zadanie 2.1

Korzystając z dowolnie wybranego języka programowania: C++, Python, Java zamodelować zachowanie się przebiegów i(t) oraz ω(t) w trakcie rozruchu silnika, którego model matematyczny jest zaprezentowany na slajdzie nr 4.

Rozwiązanie obejmuje:

1. zapis układu równań różniczkowych w postaci rekurencyjnej,
2. kod programu: do wyboru ulepszona metoda Eulera, RK2 lub RK4,
3. sporządzenie wykresów i(t) oraz ω(t),
   a) wykorzystując język C++ wyniki konieczne do sporządzenia wykresu i(t) oraz ω(t) zapisujemy w plikach, a następnie korzystając z Excel'a i zapisanych danych sporządzamy wykresy i(t) oraz ω(t),
   b) korzystając z języka Python należy wykorzystać zaimportowaną bibliotekę:
      import matplotlib.pyplot
   c) do obliczeń przyjąć:
      - ✓ krok obliczeń h = 0.001 [s],
      - ✓ krok wydruku wyników Hdru = 20*h = 0.02 [s] dla przebiegów ω(t) oraz i(t),
      - ✓ czas obliczeń Tkon = 1.5 [s],
4. wykonanie obliczeń tak aby uzyskać wykresy i(t) i ω(t) dla danych:
   a) Uz=100+X [V], Mobc=200+(Y+1)*10 [Nm],
   b) Uz=200+X/2 [V], Mobc=200+(Y+1)*15 [Nm],
   c) Uz=0 [V], Mobc=200+(Y+1)*10 [Nm],
   d) załączenie napięcia **Uz=100+X [V]** jest opóźnione o 0.3 [s] oraz o 0.5 [s], natomiast moment obciążający **Mobc=200+2*Y** [Nm] jest załączany w chwili t=0,
5. wyciągnięcie wniosków odnośnie zachowania się obiektu w każdym z przypadków, tzn wyjaśnić przebieg prądu oraz obrotów; wyjaśnić zachowanie się obiektu w pkt. 4c) oraz 4d).

gdzie X to przedostatnia a Y to ostatni cyfra numeru indeksu.

Zadanie 2.2

Korzystając z pakietu SciLab-Xcos zamodelować zachowanie się przebiegów i(t) oraz ω(t) w trakcie rozruchu silnika, którego model matematyczny jest zaprezentowany na slajdzie nr 4.

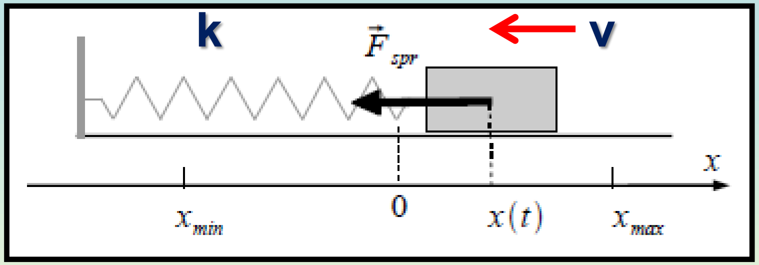
Rozwiązanie obejmuje:

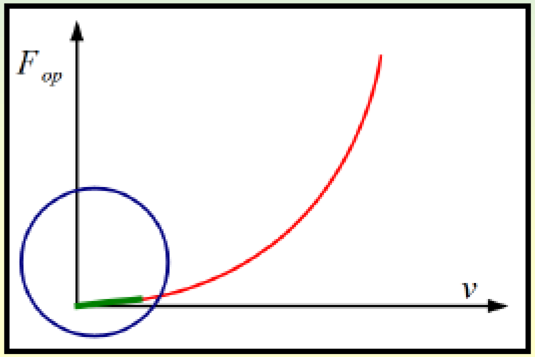
1. opracowanie modelu matematycznego jako układu dwóch równań różniczkowych, tj. $\begin{cases} \frac{di}{dt} = \ldots \\ \frac{d\omega}{dt} = \ldots \end{cases}$ z zadanymi warunkami początkowymi,
2. opracowanie modelu symulacyjnego przy użyciu pakietu SciLab-Xcos,
3. sporządzenie wykresów i(t) oraz ω(t),
4. do obliczeń przyjąć:
   a) krok całkowania h = 0.001 [s],
   b) czas obliczeń Tkon = 1.5 [s],
5. wykonanie symulacji modelu tak aby uzyskać wykresy i(t) i ω(t) dla danych:
   a) Uz=(100+X*10) [V], Mobc=(200+(Y+1)*10) [Nm],
   b) Uz=(200+X*10) [V], Mobc=(200+(Y+1)*15) [Nm],
   c) Uz=0 [V], Mobc=(200+(Y+1)*10) [Nm],
   d) załączenie napięcia **Uz=(100+X*10) [V]** jest opóźnione o 0.3 [s] oraz o 0.5 [s], natomiast moment obciążający **Mobc=200+(Y+1)*10) [Nm**] jest załączany w chwili t=0,
6. wyciągnięcie wniosków odnośnie zachowania się obiektu w każdym z przypadków, tzn wyjaśnić przebieg prądu oraz obrotów; wyjaśnić zachowanie się obiektu w pkt. 5c) oraz 5d).
7. Porównanie uzyskanych wyników z wynikami zadania 2.1.

gdzie X to przedostatnia a Y to ostatni cyfra numeru indeksu.

Zadanie przykładowe

zad. 1 Rozpatrzmy przypadek drgań harmonicznych tłumionych





$$F_{op} = \beta * v$$

Założenia modelu:

Rozpatrzymy jedynie przypadek gdy siła oporu (tłumienia) jest proporcjonalna do prędkości.

m – masa ciężarka, k – współczynnik sprężystości sprężyny, β współczynnik tłumienia, v=dx/dt prędkość oraz a – wstępne wychylenie.

Równanie ruchu drgającego (prawo Newtona) z tłumieniem przyjmie wtedy postać:

$$m * \frac{d^2x}{dt^2} = -k * x - \beta * v$$

$$\frac{d^2x}{dt^2} + \frac{\beta}{m}\frac{dx}{dt} + \frac{k}{m}x = 0$$

$$\frac{d^2x}{dt^2} = -\frac{\beta}{m}\frac{dx}{dt} - \frac{k}{m}x$$

warunki początkowe:

$$\begin{cases} x(0) = a \\ \left.\dfrac{dx}{dt}\right|_{t=0} = 0 \end{cases}$$

gdzie

a – wstępne wychylenie

Ostatnią postać równania

$$\frac{d^2x}{dt^2} = -\frac{\beta}{m}\frac{dx}{dt} - \frac{k}{m}x$$

możemy zastąpić układem dwóch równań różniczkowych I-go rzędu, postaci:

$$\begin{cases} \dfrac{dx1}{dt} = x2 \\ \dfrac{dx2}{dt} = -\dfrac{\beta}{m}x2 - \dfrac{k}{m}x1 \end{cases}$$

gdzie nowe zmienne to:

$$\begin{cases} x1 = x \\ x2 = \dfrac{dx}{dt} \end{cases}$$

a warunki początkowe:

$$\begin{cases} x1(0) = a \\ x2(0) = 0 \end{cases}$$

Wykonajmy obliczenia dla wartości liczbowych:

a=0.5 [m]; k=10[kg/s2]; m=1 [kg]; β=1.2;

krok obliczeniowy h=0.005[s]; krok drukowania Hdru=0.1 [s]; t_kon=6.5 [s]

[*] 07_met_RK4_z4.cpp

```
/* m. m. RK4 zadanie 4*/
#include <stdlib.h>
#include <iostream>
#include <math.h>
using namespace std;
double k, beta, m;
double px1(double t, double x1, double x2)
  { return  x2; }
double px2(double t, double x1, double x2)
  { return  -beta/m*x2-k/m*x1; }
int main()
{   double t, t0=0, tkon, x1, x2, a, h, k1, k2, k3, k4 ;
    int L, i, N;
    printf("\nDla rownania rozniczk. rzedu drugiego\n");
    cout << "wychylenie a  = ";        cin >> a;
    cout << "masa m  = ";              cin >> m;
    cout << "wsp. sprezyst. k  = ";    cin >> k;
    cout << "tlumienie beta  = ";      cin >> beta;
    cout << "czas kon.= ";             cin >> tkon;
    cout << "krok oblicz. h = ";       cin >> h;
    cout << "krok wydruku jako N*h, tzn. N = ";   cin >> N;
    printf("\nCzas t: |  Rozw.numer.  m.  RK-4:");
    printf("\n-------------------------------\n\n");
    x1=a;   x2=0;  t=t0;  L=0;  i=0;
    printf(" %5.3f %15.3f\n", t, x1);
    while(t<tkon)
     { k1=h*px1(t, x1, x2);       k2=h*px1(t+h/2, x1+k1/2, x2);
       k3=h*px1(t+h/2, x1+k2/2, x2); k4=h*px1(t, x1+k3, x2);
       x1=x1+(k1+2*k2+2*k3+k4)/6;

       k1=h*px2(t, x1, x2);       k2=h*px2(t+h/2, x1, x2+k1/2);
       k3=h*px2(t+h/2, x1, x2+k2/2); k4=h*px2(t, x1, x2+k3);
       x2=x2+(k1+2*k2+2*k3+k4)/6;

       if (++i == N) {printf(" %5.3f %15.3f\n",t+h, x1); i=0;};
       t=t0 + (++L)*h;
     };
   return 0;
}
```

$$\frac{d^2x}{dt^2} = -\frac{\beta}{m}\frac{dx}{dt} - \frac{k}{m}x$$

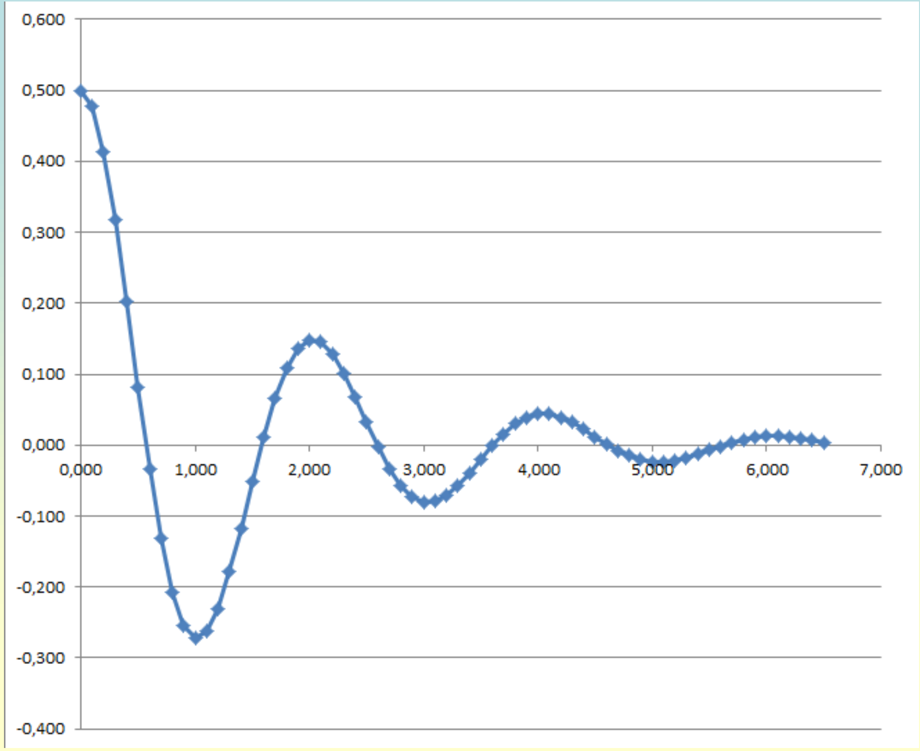
$$\begin{cases} x(0) = a \\ \left.\dfrac{dx}{dt}\right|_{t=0} = 0 \end{cases}$$

$$\begin{cases} \dfrac{dx1}{dt} = x2 \\ \dfrac{dx2}{dt} = -\dfrac{\beta}{m}x2 - \dfrac{k}{m}x1 \end{cases}$$

$$\begin{cases} x1(0) = a \\ x2(0) = 0 \end{cases}$$

```
Dla rownania rozniczk. rzedu drugiego
wychylenie a  = 0.5
masa m  = 1
wsp. sprezyst. k  = 10
tlumienie beta  = 1.2
czas kon.= 6.5
krok oblicz. h = 0.005
krok wydruku jako N*h, tzn. N = 20
```

Czas t:	Rozw.numer. m. RK-4:
0.000	0.500
0.100	0.477
0.200	0.413
0.300	0.317
0.400	0.202
0.500	0.081
0.600	-0.033
0.700	-0.132
0.800	-0.207
0.900	-0.254
1.000	-0.272
1.100	-0.263
1.200	-0.230
1.300	-0.179
1.400	-0.118
1.500	-0.052
1.600	0.011
1.700	0.066
1.800	0.109
1.900	0.136
2.000	0.148
2.100	0.145
2.200	0.128
2.300	0.101
2.400	0.068
2.500	0.033
2.600	-0.002
2.700	-0.033
2.800	-0.057
2.900	-0.073
3.000	-0.080
3.100	-0.079
3.200	-0.071
3.300	-0.057
3.400	-0.039
3.500	-0.020
3.600	-0.001
3.700	0.016
3.800	0.030
3.900	0.039
4.000	0.044
4.100	0.044
4.200	0.039
4.300	0.032
4.400	0.023
4.500	0.012
4.600	0.002
4.700	-0.008
4.800	-0.015
4.900	-0.021
5.000	-0.024
5.100	-0.024
5.200	-0.022
5.300	-0.018
5.400	-0.013
5.500	-0.007
5.600	-0.002
5.700	0.004
5.800	0.008
5.900	0.011
6.000	0.013
6.100	0.013
6.200	0.012
6.300	0.010
6.400	0.007
6.500	0.004



Wariant z zapisem wyników do plików aby później zrobić wykresy np. używając pakietu Excel

wykorzystanie zapisu danych do pliku

```
int main()
{   double ……….;     int …….;
    FILE  *fp;                                    //definicja zmiennej wskaźnikowej do pliku
    char  fileName[128];                          //tablica znakowa na nazwę pliku (max dł. 128 zn.)

    printf("\nNazwa pliku wynikowego = ");
    scanf("%s",&fileName);
    if ( (fp = fopen(fileName, "w")) == NULL )              //otwarcie pliku
      {printf("Otwarcie pliku nie jest mozliwe!\n"); exit; }
        else {printf("Plik otwarty prawidłowo..\n");};
……………………………….

  printf("\nJakiś opis ……\n");                //zapis na ekran
  fprintf(fp, "\nJakiś opis ……\n");           //zapis do pliku
   ……………………………….

   while(t<tkon)
    { ………………………. //np. obliczenia w pętli
      printf(" %5.3f %15.3f %15.3f\n",t+h, x1, x2);          //zapis na ekran
      fprintf(fp, " %5.3f %15.3f %15.3f\n",t+h, x1, x2);     //zapis do pliku
     };
……………………………….

   fclose(fp);                  //zamknięcie pliku
   return 0;
}
```

09_met_RK4_z5.cpp

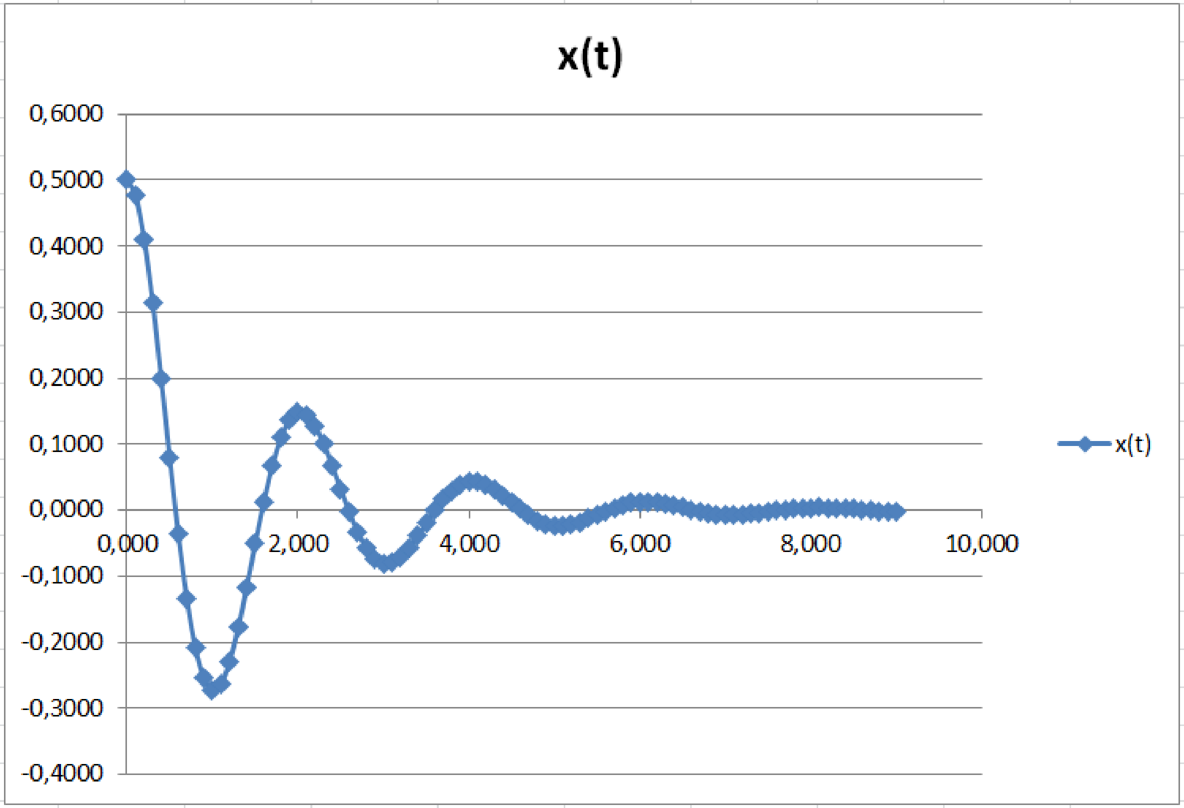
m. RK4

```cpp
/* met. RK4 zapis wyników do dwóch plików*/
#include <stdlib.h>
#include <iostream>
#include <math.h>
using namespace std;
double a=0.5, m=1, k=10, beta=1.2;
double px1(double t, double x1, double x2)
  { return  x2; }
double px2(double t, double x1, double x2)
  { return  -beta/m*x2-k/m*x1; }
int main()
{   double t, t0=0, tkon=9, h=0.001, x1, x2, k1, k2, k3, k4 ;
    int LL, i, N=100;
    FILE *plik1, *plik2;
    char nazwa1[64], nazwa2[64];
    printf("\nWprowadz nazwe pliku wynikowego dla t= ");
    scanf("%s",&nazwa1);
    printf("\nWprowadz nazwe pliku wynikowego dla x(t)= ");
    scanf("%s",&nazwa2);
    if ((plik1 = fopen(nazwa1, "w")) == NULL)
     {printf("Otwarcie pliku dla t nie jest mozliwe!\n"); exit;}
      else {printf("Plik dla t otwarty prawidlowo..\n");};
    if ((plik2 = fopen(nazwa2, "w")) == NULL)
     {printf("Otwarcie pliku dla x(t) nie jest mozliwe!\n"); exit;}
      else {printf("Plik dla x(t) otwarty prawidlowo..\n");};
    fprintf(plik1, "\n     t\n");
    fprintf(plik2, "\n  x(t)\n");

    printf("\n  t               x(t)              v(t)");
    printf("\n--------------------------------------\n\n");

    t=t0; x1=a;  x2=0;  LL=0;  i=0;

    printf(" %6.3f %10.4f %10.4f\n", t, x1, x2);
    fprintf(plik1, " %6.3f\n", t);
    fprintf(plik2, " %10.4f\n", x1);

```

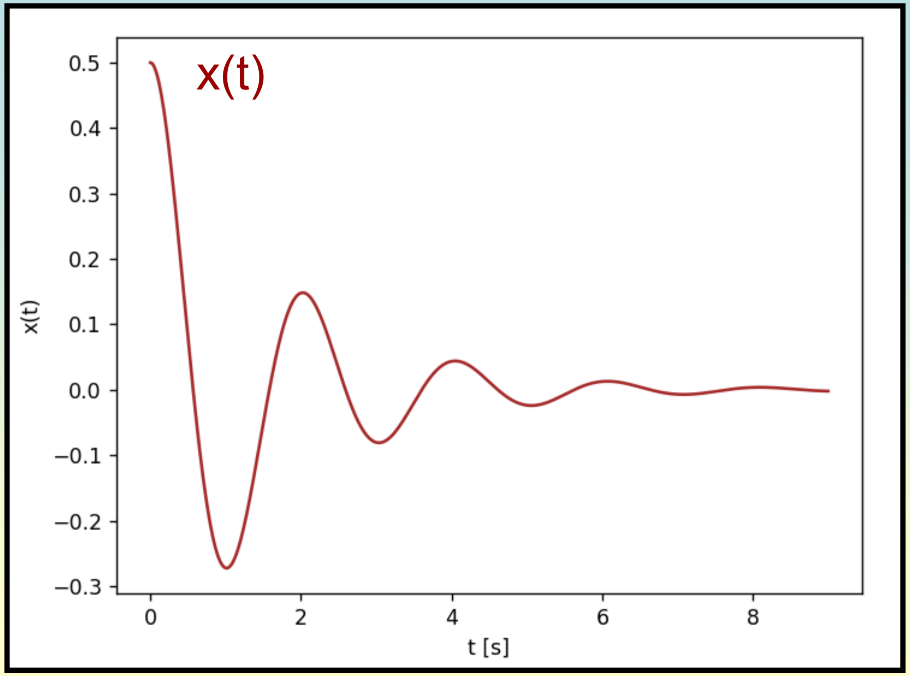
```
    while(t<tkon)
     { k1=h*px1(t, x1, x2);
       k2=h*px1(t+h/2, x1+k1/2, x2);
       k3=h*px1(t+h/2, x1+k2/2, x2);
       k4=h*px1(t, x1+k3, x2);
       x1=x1+(k1+2*k2+2*k3+k4)/6;

       k1=h*px2(t, x1, x2);
       k2=h*px2(t+h/2, x1, x2+k1/2);
       k3=h*px2(t+h/2, x1, x2+k2/2);
       k4=h*px2(t, x1, x2+k3);
       x2=x2+(k1+2*k2+2*k3+k4)/6;

       if (++i == N)
        {printf(" %6.3f %10.4f %10.4f\n",t+h, x1, x2);
         fprintf(plik1, " %6.3f\n",t+h);
         fprintf(plik2, " %10.4f\n", x1);
         i=0;
        };
       t=t0 + (++LL)*h;
     };
   fclose(plik1);
   fclose(plik2);
   return 0;
}
```

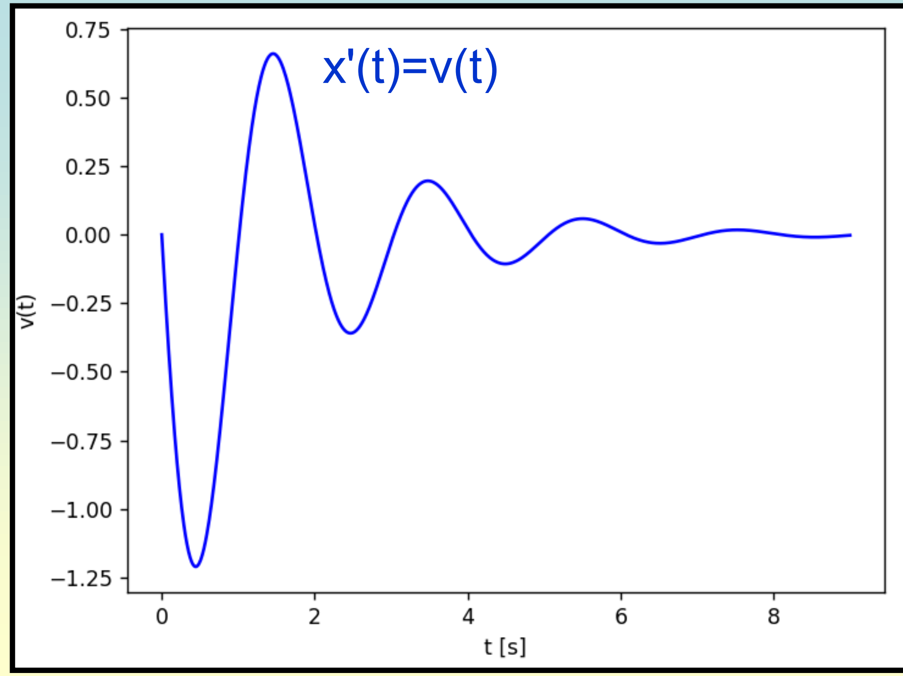
Prezentacja wykresu w Excel po niewielkiej obróbce plików wynikowych (zamiana w kolumnach znaku . na ,)

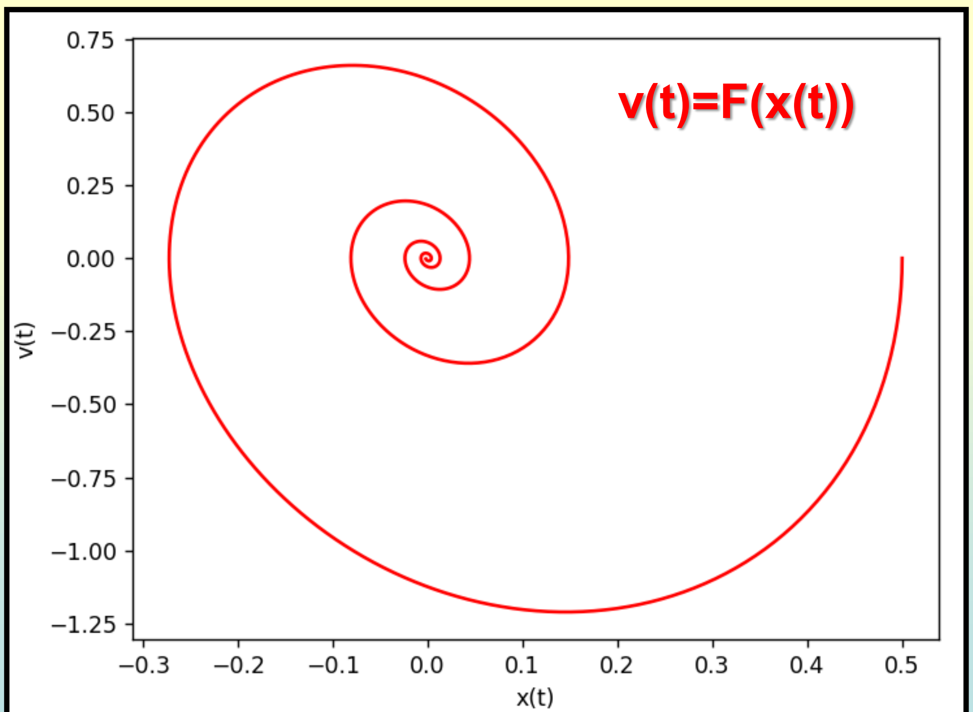
t	x(t)
0,000	0,5000
0,100	0,4764
0,200	0,4109
0,300	0,3145
0,400	0,1996
0,500	0,0790
0,600	-0,0354
0,700	-0,1335
0,800	-0,2080
0,900	-0,2547
1,000	-0,2723
1,100	-0,2624
1,200	-0,2292
1,300	-0,1783
1,400	-0,1165
1,500	-0,0508
1,600	0,0123
1,700	0,0670
1,800	0,1093
1,900	0,1366
2,000	0,1480
2,100	0,1444
2,200	0,1276
2,300	0,1008
2,400	0,0676
2,500	0,0319
2,600	-0,0028
2,700	-0,0333
2,800	-0,0572
2,900	-0,0731
3,000	-0,0804
3,100	-0,0793



Ruch harmoniczny tłumiony **k = 1.2**


x(t)
0.5
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0
2
4
6
8
t [s]


x'(t)=v(t)
v(t)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
0
2
4
6
8
t [s]


v(t)=F(x(t))
v(t)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.25
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
0.4
0.5
x(t)

m. RK4

Python

Koniec Spr. 2