Metody Numeryczne w inżynierii

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

Spr. 1

Zadanie 1.1

Korzystając z dowolnie wybranego języka programowania:

- ***C++***
- ***Python***
- ***Java***

zamodelować zachowanie się przebiegów i(t) oraz E(t) w obwodzie przedstawionym na kolejnym slajdzie.

Rozwiązanie obejmuje:

1. zapis układu równań różniczkowych w postaci rekurencyjnej,

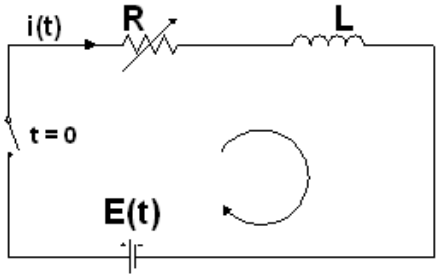
2. kod programu: ulepszona metoda Eulera oraz RK2 lub RK4,

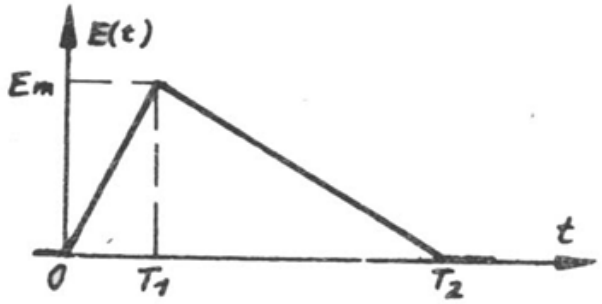
3. sporządzenie wykresów i(t) oraz E(t),
   a) wykorzystując język C++ wyniki konieczne do sporządzenia wykresu i(t) oraz E(t) zapisujemy w plikach, a następnie korzystając z Excel'a i zapisanych danych sporządzamy wykresy i(t) oraz E(t),
   b) korzystając z języka Python należy zaimportować zainstalowaną wcześniej bibliotekę:
      pip install nazwa_biblioteki,
      import matplotlib.pyplot,

4. do obliczeń przyjąć:
   a) krok obliczeń $h=0.005*10^{-6}$ [s],
   b) krok wydruku wyników [tj. E(t), i(t)] $Hdru=100*h=0.5*10^{-6}$ [s],
   c) czas obliczeń $Tkon=90*10^{-6}$ [s]

zad. 1.1 charakterystyki i(t) w nieliniowym układzie zasilonym impulsem napięciowym

W obwodzie jak na rysunku





źródło napięcia wytwarza impuls postaci

Należy wyprowadzić wzór na funkcję E(t), tj. E(t) = ...

Charakterystyka prądowo-napięciowa nieliniowego rezystora R jest podana

w postaci wzoru: $u = K * i^{\alpha}$

Przebieg prądu i(t) w układzie po załączeniu włącznika opisuje równanie:

$$L\frac{di}{dt} + K * i^{\alpha} = E(t)$$

warunek początkowy i(0)=0

Dane do obliczeń: Em=100; K=300; L=0.005; α=0.6; $T_1=10*10^{-6}$; $T_2=90*10^{-6}$

Zadania przykładowe

03_met_Eulera_z2.cpp

```
1   /* m. Eulera zadanie 2*/
2
3   #include <stdlib.h>
4   #include <iostream>
5   using namespace std;
6
7   double fpoch(double x, double y)
8   { return 2*x; }
9
10  int main()
11  {
12      double x, x0, xkon, y, y0, h;
13      int k, i, N;
14      printf("\n  dy/dx=2x\n");
15      cout << "\nx0  = ";  cin >> x0;
16      cout << "xkon= ";    cin >> xkon;
17      cout << "y0  = ";    cin >> y0;
18      cout << "krok oblicz. h = ";    cin >> h;
19      cout << "krok wydruku jako N*h, N = ";   cin >> N;
20      printf("\n   x    |   y(x)  |   y=x^2");
21      printf("\n--------------------------\n\n");
22      x=x0;   y=y0;
23      printf("%6.2f%10.3f%10.3f\n", x, y, x*x);          //wydruk war. poczatkowych
24      k=0;
25      i=0;
26      while(x<xkon)
27       { y=y+h*fpoch(x,y);
28         i++;
29         if (i==N) {printf("%6.2f%10.3f%10.3f\n",x+h,y, (x+h)*(x+h));  i=0;}
30         k++;
31         x=x0+k*h;}
32   return 0;
33  }
```

$$\frac{dy}{dx} = y' = f(x,y) = 2x$$

$$\mathbf{x(0) = x0 = 0} \quad \mathbf{i} \quad \mathbf{y(0) = y0 = 0}$$

oraz

$$rozw.analit. \quad \mathbf{y(t)} = x^2$$

metoda Eulera

h = 0.2

```
Dla rownania dy/dx=2x

x0  = 0
xkon= 20
y0  = 0
krok oblicz. h = 0.2
krok wydruku jako N*h, N = 5

   x    |   y(x)  |   y=x^2
--------------------------

  0.00     0.000     0.000
  1.00     0.800     1.000
  2.00     3.600     4.000
  3.00     8.400     9.000
  4.00    15.200    16.000
  5.00    24.000    25.000
  6.00    34.800    36.000
  7.00    47.600    49.000
  8.00    62.400    64.000
  9.00    79.200    81.000
 10.00    98.000   100.000
 11.00   118.800   121.000
 12.00   141.600   144.000
 13.00   166.400   169.000
 14.00   193.200   196.000
 15.00   222.000   225.000
 16.00   252.800   256.000
 17.00   285.600   289.000
 18.00   320.400   324.000
 19.00   357.200   361.000
 20.00   396.000   400.000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

h = 0.1

```
Dla rownania dy/dx=2x

x0  = 0
xkon= 20
y0  = 0
krok oblicz. h = 0.1
krok wydruku jako N*h, N = 10

   x    |   y(x)  |   y=x^2
--------------------------

  0.00     0.000     0.000
  1.00     0.900     1.000
  2.00     3.800     4.000
  3.00     8.700     9.000
  4.00    15.600    16.000
  5.00    24.500    25.000
  6.00    35.400    36.000
  7.00    48.300    49.000
  8.00    63.200    64.000
  9.00    80.100    81.000
 10.00    99.000   100.000
 11.00   119.900   121.000
 12.00   142.800   144.000
 13.00   167.700   169.000
 14.00   194.600   196.000
 15.00   223.500   225.000
 16.00   254.400   256.000
 17.00   287.300   289.000
 18.00   322.200   324.000
 19.00   359.100   361.000
 20.00   398.000   400.000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

h = 0.01

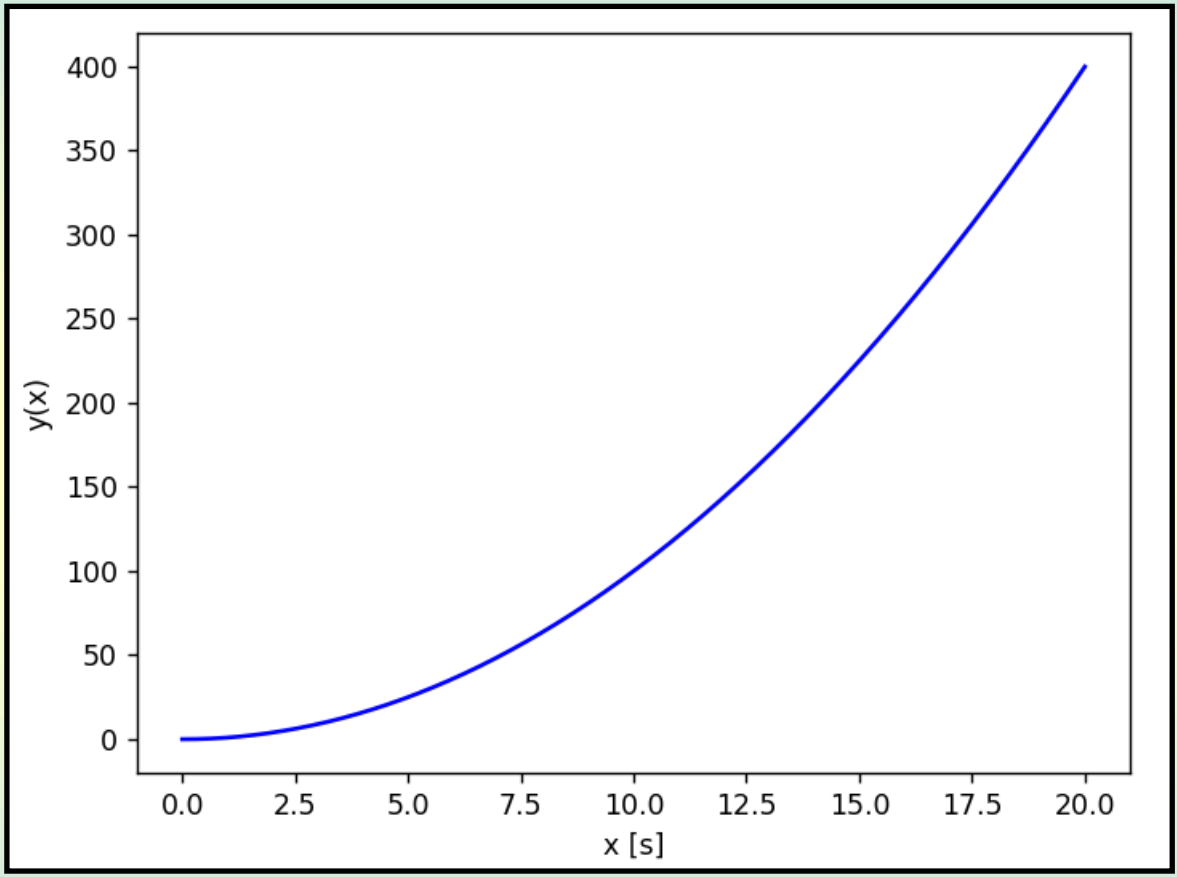
```
Dla rownania dy/dx=2x

x0  = 0
xkon= 20
y0  = 0
krok oblicz. h = 0.01
krok wydruku jako N*h, N = 100

   x    |   y(x)  |   y=x^2
--------------------------

  0.00     0.000     0.000
  1.00     0.990     1.000
  2.00     3.980     4.000
  3.00     8.970     9.000
  4.00    15.960    16.000
  5.00    24.950    25.000
  6.00    35.940    36.000
  7.00    48.930    49.000
  8.00    63.920    64.000
  9.00    80.910    81.000
 10.00    99.900   100.000
 11.00   120.890   121.000
 12.00   143.880   144.000
 13.00   168.870   169.000
 14.00   195.860   196.000
 15.00   224.850   225.000
 16.00   255.840   256.000
 17.00   288.830   289.000
 18.00   323.820   324.000
 19.00   360.810   361.000
 20.00   399.800   400.000

--------------------------------
Process exited with return value 0
Press any key to continue . . .
```

Python

03a_met_Eulera_z2.py - C:\Users\makuc\Documents

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as plt
import math

#definicja i warunki początkowe
h=0.001;  xkon=20;  x0=0;  y0=0

#definicje równań
def poch_y(x,y):
    return 2*x

#zmienne pomocnicze
n=0;   N=20;  x=x0;  y=y0;

#zmienne na wyniki
x_dru=[x]
y_dru=[y]

#główna pętla
while x<xkon:
    #obliczanie y(x) met. Eulera
    y=y+h*poch_y(x, y);
    n=n+1
    if n==N:
        n=0
        x_dru.append(x)
        y_dru.append(y)
    x=x+h

#wyswietlanie wykresu y(x)
plt.plot(x_dru, y_dru, color='blue')
plt.xlabel('x [s]')
plt.ylabel('y(x)')
plt.show()
```



04_met_Eulera_z3.cpp

```cpp
/* m. Eulera zadanie 3*/

#include <stdlib.h>
#include <iostream>
using namespace std;

double fpoch(double x, double y)
{ return 3*x*x-2*y*y; }

int main()
{
    double x, x0, xkon, y, y0, h ;
    int k, i, N;
    printf("\n   dy/dx=3*x*x-2*y*y\n");
    cout << "\nx0  = ";  cin >> x0;
    cout << "xkon= ";    cin >> xkon;
    cout << "y0  = ";    cin >> y0;
    cout << "krok oblicz. h = ";    cin >> h;
    cout << "krok wydruku jako N*h, N = ";   cin >> N;
    printf("\n   x  |  y - m.Eulera");
    printf("\n----------------------\n\n");
    x=x0;   y=y0;
    printf(" %6.2f %10.3f\n", x, y);
    k=0;
    i=0;
    while(x<xkon)
     { y=y+h*fpoch(x,y);
       i++;
       if (i==N) {printf(" %6.3f %10.3f\n",x+h,y);  i=0;}
       k++;
       x=x0+k*h;}
 return 0;
}
```

$$\frac{dy}{dx} = y' = f(x,y) = 3x^2 - 2y^2$$

oraz $\mathbf{x(0) = x0 = 0}$ **i** $\mathbf{y(0) = y0 = 1}$

metoda Eulera

h = 0.2

```
   dy/dx=3*x*x-2*y*y

x0  = 0
xkon= 3
y0  = 1
krok oblicz. h = 0.2
krok wydruku jako N*h, N = 1

   x  |  y - m.Eulera
----------------------

  0.00      1.000
 0.200      0.600
 0.400      0.480
 0.600      0.484
 0.800      0.606
 1.000      0.843
 1.200      1.159
 1.400      1.486
 1.600      1.779
 1.800      2.049
 2.000      2.314
 2.200      2.573
 2.400      2.829
 2.600      3.083
 2.800      3.337
 3.000      3.587
```

h = 0.05

```
   dy/dx=3*x*x-2*y*y

x0  = 0
xkon= 3
y0  = 1
krok oblicz. h = 0.05
krok wydruku jako N*h, N = 4

   x  |  y - m.Eulera
----------------------

  0.00      1.000
 0.200      0.700
 0.400      0.581
 0.600      0.586
 0.800      0.706
 1.000      0.922
 1.200      1.197
 1.400      1.489
 1.600      1.774
 1.800      2.047
 2.000      2.312
 2.200      2.572
 2.400      2.829
 2.600      3.083
 2.800      3.336
 3.000      3.588
```

h = 0.01

```
   dy/dx=3*x*x-2*y*y

x0  = 0
xkon= 3
y0  = 1
krok oblicz. h = 0.01
krok wydruku jako N*h, N = 20

   x  |  y - m.Eulera
----------------------

  0.00      1.000
 0.200      0.717
 0.400      0.601
 0.600      0.609
 0.800      0.728
 1.000      0.939
 1.200      1.206
 1.400      1.492
 1.600      1.774
 1.800      2.047
 2.000      2.312
 2.200      2.572
 2.400      2.829
 2.600      3.083
 2.800      3.336
 3.000      3.588
```

05_met_Eulera_u_z3.cpp

```
/* m. Eulera ulepszona zadanie 3*/

#include <stdlib.h>
#include <iostream>
using namespace std;

double fpoch(double x, double y)
{ return 3*x*x-2*y*y; }

int main()
{
    double x, x0, xkon, y, y0, k1, h ;
    int k, i, N;
    printf("\n   dy/dx=3*x*x-2*y*y\n");
    cout << "\nx0  = ";  cin >> x0;
    cout << "xkon= ";    cin >> xkon;
    cout << "y0  = ";    cin >> y0;
    cout << "krok oblicz. h = ";    cin >> h;
    cout << "krok wydruku jako N*h, N = ";   cin >> N;
    printf("\n   x  |  y - m.Eulera u.");
    printf("\n-----------------------\n\n");
    x=x0;   y=y0;
    printf(" %6.2f %10.3f\n", x, y);
    k=0;
    i=0;
    while(x<xkon)
     { k1=h/2*fpoch(x, y);
       y=y+h*fpoch(x+h/2, y+k1);
       i++;
       if (i==N) {printf(" %6.3f %10.3f\n",x+h,y);  i=0;}
       k++;
       x=x0+k*h;}
 return 0;
}
```

$$\frac{dy}{dx} = y' = f(x,y) = 3x^2 - 2y^2$$

oraz $\mathbf{x(0) = x0 = 0}$ **i** $\mathbf{y(0) = y0 = 1}$

ulepszona metoda Eulera

h = 0.2

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.2
krok wydruku jako N*h, N = 1

   x   |   y - m.Eulera u.
-----------------------

  0.00       1.000
 0.200       0.750
 0.400       0.635
 0.600       0.640
 0.800       0.757
 1.000       0.964
 1.200       1.225
 1.400       1.503
 1.600       1.778
 1.800       2.047
 2.000       2.310
 2.200       2.569
 2.400       2.825
 2.600       3.078
 2.800       3.327
 3.000       3.571
```

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.2
krok wydruku jako N*h, N = 1

   x   |   y - m.Eulera
-----------------------

  0.00       1.000
 0.200       0.600
 0.400       0.480
 0.600       0.484
 0.800       0.606
 1.000       0.843
 1.200       1.159
 1.400       1.486
 1.600       1.779
 1.800       2.049
 2.000       2.314
 2.200       2.573
 2.400       2.829
 2.600       3.083
 2.800       3.337
 3.000       3.587
```

h = 0.01

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.01
krok wydruku jako N*h, N = 20

   x   |   y - m.Eulera u.
-----------------------

  0.00       1.000
 0.200       0.721
 0.400       0.606
 0.600       0.614
 0.800       0.733
 1.000       0.943
 1.200       1.208
 1.400       1.493
 1.600       1.774
 1.800       2.047
 2.000       2.312
 2.200       2.572
 2.400       2.829
 2.600       3.083
 2.800       3.336
 3.000       3.588
```

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.01
krok wydruku jako N*h, N = 20

   x   |   y - m.Eulera
-----------------------

  0.00       1.000
 0.200       0.717
 0.400       0.601
 0.600       0.609
 0.800       0.728
 1.000       0.939
 1.200       1.206
 1.400       1.492
 1.600       1.774
 1.800       2.047
 2.000       2.312
 2.200       2.572
 2.400       2.829
 2.600       3.083
 2.800       3.336
 3.000       3.588
```

06_met_RK4_z3.cpp

```
/* m. m. RK4 zadanie 3*/

#include <stdlib.h>
#include <iostream>
using namespace std;

double fpoch(double x, double y)
{ return 3*x*x-2*y*y; }

int main()
{
    double x, x0, xkon, y, y0, k1, k2, k3, k4, h;
    int k, i, N;
    printf("\n   dy/dx=3*x*x-2*y*y\n");
    cout << "\nx0  = ";  cin >> x0;
    cout << "xkon= ";    cin >> xkon;
    cout << "y0  = ";    cin >> y0;
    cout << "krok oblicz. h = ";    cin >> h;
    cout << "krok wydruku jako N*h, N = ";   cin >> N;
    printf("\n   x  |  y -  m. RK4");
    printf("\n----------------------\n\n");
    x=x0;   y=y0;
    printf(" %6.2f %10.3f\n", x, y);
    k=0;
    i=0;
    while(x<xkon)
     { k1=h*fpoch(x, y);
       k2=h*fpoch(x+h/2, y+k1/2);
       k3=h*fpoch(x+h/2, y+k2/2);
       k4=h*fpoch(x+h, y+k3);
       y=y+(k1+2*k2+2*k3+k4)/6;

       if (++i == N) {printf(" %6.2f %10.3f\n",x+h,y); i=0;}
       x=x0 + (++k)*h;
     }
   return 0;
}
```

$$\frac{dy}{dx} = y' = f(x,y) = 3x^2 - 2y^2$$

oraz $\mathbf{x(0) = x0 = 0}$ i $\mathbf{y(0) = y0 = 1}$

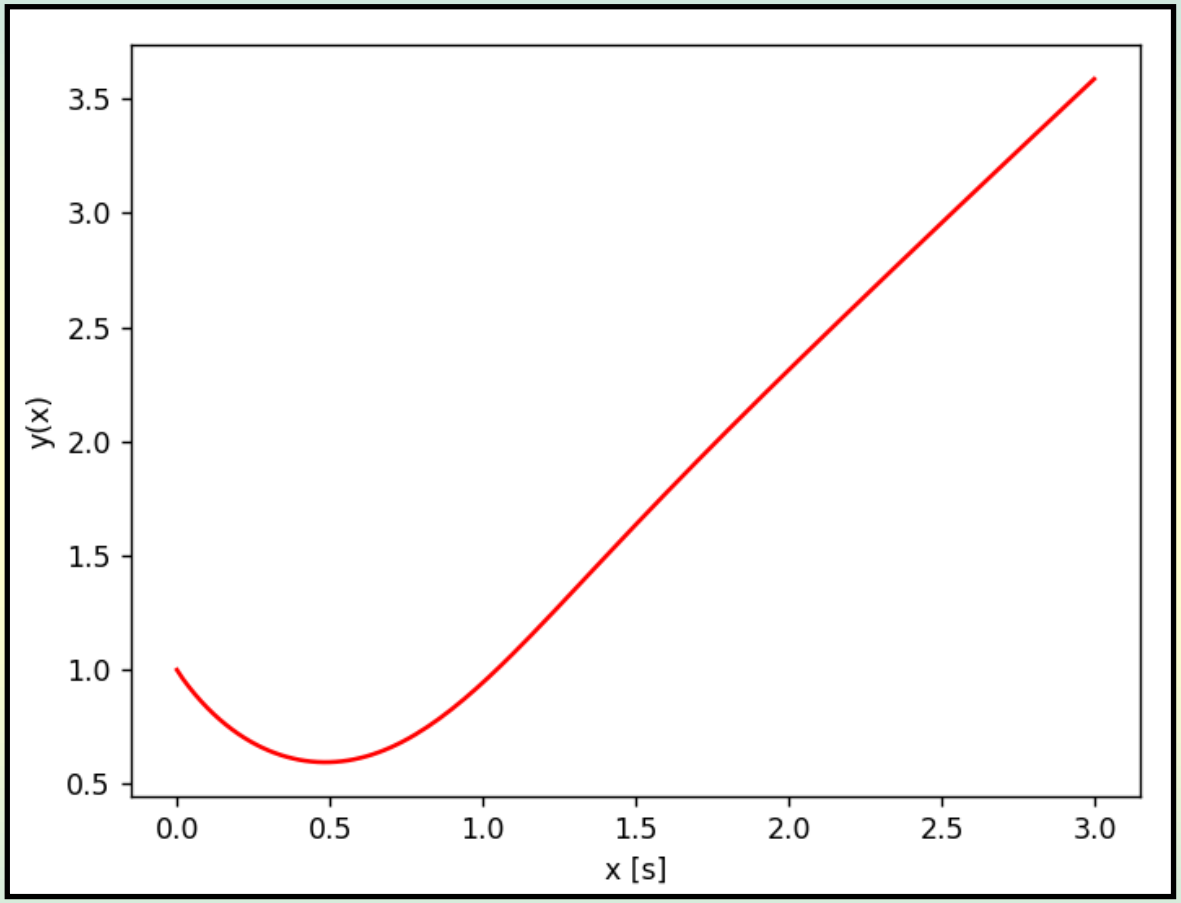
metoda RK4

RK4
Python

06_met_RK4_z3.py - C:\Users\makuc\Documents\01_

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as plt
import math

#definicja i warunki początkowe
h=0.001;  xkon=10;  x0=0;  y0=1;

#definicje równań
def poch_y(x,y):
    return 3*x*x-2*y*y

#zmienne pomocnicze
n=0;   N=20;  x=x0;  y=y0;

#zmienne na wyniki
x_dru=[x]
y_dru=[y]

#główna pętla
#oblicz. y(x) met. RK4
while x<xkon:
    k1=h*poch_y(x, y);
    k2=h*poch_y(x+h/2, y+k1/2);
    k3=h*poch_y(x+h/2, y+k2/2);
    k4=h*poch_y(x, y+k3);
    y=y+(k1+2*k2+2*k3+k4)/6;
    n=n+1
    if n==N:
        n=0
        x_dru.append(x)
        y_dru.append(y)
    x=x+h

#wyswietlanie wykresu y(x)
plt.plot(x_dru, y_dru, color='green')
plt.xlabel('x [s]')
plt.ylabel('y(x)')
plt.show()
```



Porównanie wyników uzyskanych ulepszoną metodą Eulera i RK-4

$$\frac{dy}{dx} = 3 * x^2 - 2 * y^2$$

h = 0.2

m. Eulera ulepszona

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.2
krok wydruku jako N*h, N = 1

   x   |   y - m.Eulera u.
-----------------------

   0.00       1.000
  0.200       0.750
  0.400       0.635
  0.600       0.640
  0.800       0.757
  1.000       0.964
  1.200       1.225
  1.400       1.503
  1.600       1.778
  1.800       2.047
  2.000       2.310
  2.200       2.569
  2.400       2.825
  2.600       3.078
  2.800       3.327
  3.000       3.571
```

m. RK-4

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.2
krok wydruku jako N*h, N = 1

   x   |   y -   m. RK4
-----------------------

   0.00       1.000
   0.20       0.721
   0.40       0.606
   0.60       0.614
   0.80       0.733
   1.00       0.943
   1.20       1.209
   1.40       1.493
   1.60       1.774
   1.80       2.046
   2.00       2.311
   2.20       2.571
   2.40       2.828
   2.60       3.083
   2.80       3.335
   3.00       3.587
```

$$\frac{dy}{dx} = 3 * x^2 - 2 * y^2$$

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.01
krok wydruku jako N*h, N = 20

   x   |   y - m.Eulera u.
-----------------------

   0.00       1.000
  0.200       0.721
  0.400       0.606
  0.600       0.614
  0.800       0.733
  1.000       0.943
  1.200       1.208
  1.400       1.493
  1.600       1.774
  1.800       2.047
  2.000       2.312
  2.200       2.572
  2.400       2.829
  2.600       3.083
  2.800       3.336
  3.000       3.588
```

m. Eulera ulepszona

h = 0.01

```
   dy/dx=3*x*x-2*y*y

x0   = 0
xkon= 3
y0   = 1
krok oblicz. h = 0.01
krok wydruku jako N*h, N = 20

   x   |   y -   m. RK4
----------------------

   0.00       1.000
   0.20       0.721
   0.40       0.606
   0.60       0.614
   0.80       0.733
   1.00       0.943
   1.20       1.208
   1.40       1.493
   1.60       1.774
   1.80       2.047
   2.00       2.312
   2.20       2.572
   2.40       2.829
   2.60       3.083
   2.80       3.336
   3.00       3.588
```

m. RK-4

Koniec Spr. 1