

Python

<https://www.anaconda.com/products/individual>



[Products](#) ▾

[Pricing](#)

[Solutions](#) ▾

[Resources](#) ▾

[Blog](#)

[Company](#) ▾

[Get Started](#)



Individual Edition

Your data science toolkit

With over 25 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Anaconda Individual Edition

[Download](#) 

For Windows

Python 3.8 • 64-Bit Graphical Installer • 477 MB

Get Additional Installers



Python

<https://www.anaconda.com/products/individual>



[Products](#) ▾

[Pricing](#)

[Solutions](#) ▾


[Resources](#) ▾

[Blog](#)

[Company](#) ▾

[Get Started](#)

Anaconda Installers

Windows 

Python 3.8

64-Bit Graphical Installer (477 MB)


32-Bit Graphical Installer (409 MB)

MacOS 

Python 3.8

64-Bit Graphical Installer (440 MB)

64-Bit Command Line Installer (433 MB)

Linux 

Python 3.8

64-Bit (x86) Installer (544 MB)

64-Bit (Power8 and Power9) Installer (285 MB)

64-Bit (AWS Graviton2 / ARM64) Installer (413 M)

64-bit (Linux on IBM Z & LinuxONE) Installer (292 M)

Python



CMD.exe Prompt

0.1.1

Run a cmd.exe terminal with your current environment from Navigator activated

Launch



Datalore

Online Data Analysis Tool with smart coding assistance by JetBrains. Edit and run your Python notebooks in the cloud and share them with your team.

Launch



IBM Watson Studio Cloud

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train machine learning models. Prepare data and build models, using open source data science tools or visual modeling.

Launch



JupyterLab

3.1.7

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Launch



Jupyter Notebook

6.4.3

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Launch



Powershell Prompt

0.0.1

Run a Powershell terminal with your current environment from Navigator activated

Launch



Qt Console

5.1.0

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Launch



Spyder

5.0.5

Scientific PYTHON Development Environment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Launch



Glueviz

1.0.0

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install



Orange 3

3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install



PyCharm Professional

A full-fledged IDE by JetBrains for both Scientific and Web Python development. Supports HTML, JS, and SQL.

Install



RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

Spyder

Python - Spyder

Spyder (Python 3.8)

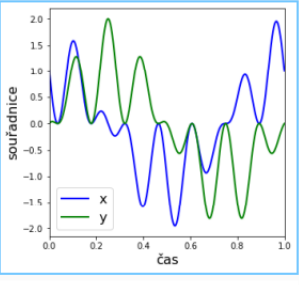
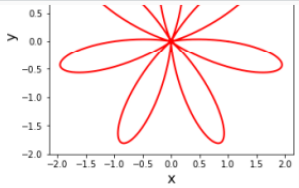
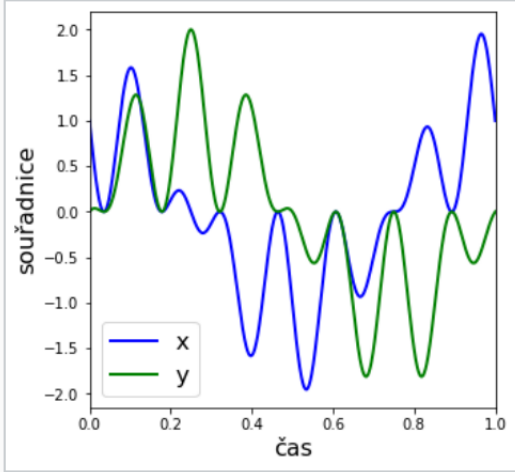
File Edit Search Source Run Debug Consoles Projects Tools View Help

F:\text\teach\FyzikaI\2021\prednaska\prednaska2

F:\text\teach\FyzikaI\2021\prednaska\prednaska2\sroubovnice.py

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 r=1
5 T=1
6 omega=2*np.pi/T
7 vz=T/10
8 t=np.linspace(0,10*T,1000)
9 x=r*np.cos(omega*t)
10 y=r*np.sin(omega*t)
11 z=vz*t
12
13 fig=plt.figure(figsize=(8,8))
14 ax=fig.add_subplot(projection='3d')
15 ax.plot(x,y,z,lw=2,c='red')
16 ax.set_xlabel("x",fontsize=16)
17 ax.set_ylabel("y",fontsize=16)
18 ax.set_zlabel("z",fontsize=16)
19
```

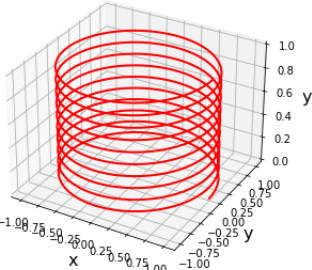
HEA_sim.py - bulk\sc2 x HEA_sim.py - bulk\sc2b x HEA_sim.py - bulk\sc6 x DPL_spectra.py x DPL_view.py x dsc_bcg.py x sroubovnice.py x



Help Variable Explorer Plots Files

Console 1/A x

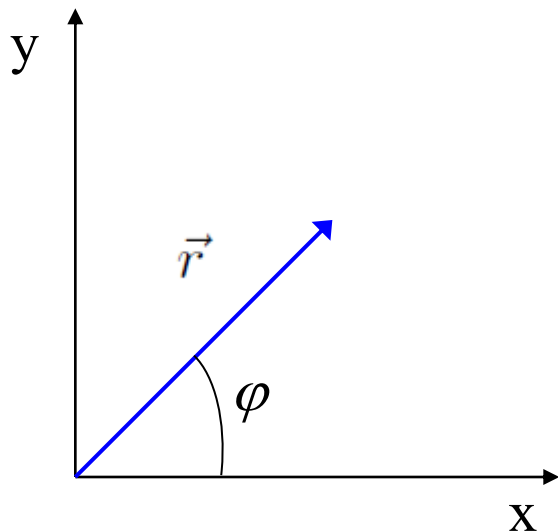
In [32]: runfile('F:/text/teach/FyzikaI/2021/prednaska/prednaska2/sroubovnice.py', wdir='F:/text/teach/FyzikaI/2021/prednaska/prednaska2')



In [33]: runfile('F:/text/teach/FyzikaI/2021/prednaska/prednaska2/sroubovnice.py', wdir='F:/text/teach/FyzikaI/2021/prednaska/prednaska2')

IPython console History

Kruhový pohyb



polární souřadnice

$$r(t) = r$$

$$\varphi(t) = \omega t$$

ω - úhlová rychlost

$$T = \frac{2\pi}{\omega} \quad \text{- perioda}$$

kartézské souřadnice

$$x(t) = r \cos \varphi = r \cos(\omega t)$$

$$y(t) = r \sin \varphi = r \sin(\omega t)$$

Kruhový pohyb

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

uloha1a.py

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt
```

```
r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic
```

```
#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda
```

```
#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

importování knihoven

numpy – knihovna pro numerické výpočty

matplotlib – knihovna pro kreslení grafů

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt
```

```
r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic
```

definování konstant

```
#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda
```

```
#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```


Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

vytvoření pole časů – rozdělení intervalu $0, T+dt$ na úseky o velikosti dt

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic
```

```
#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda
```

```
#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

$$x(t) = r \cos \varphi = r \cos(\omega t)$$

$$y(t) = r \sin \varphi = r \sin(\omega t)$$

výpočet souřadnic – s polem t je možné pracovat najednou jako s celkem

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

vytvoření obrázku – velikost obrázku 5” x 5”

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

název grafu

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časova zavislost souradnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

barva bodů grafu

tloušťka čáry spojující body grafu

typ bodů grafu

popis do legendy

vykreslení grafu
závislosti x na čase,
tj. bodů o souřadnicích $[t, x]$

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

vykreslení grafu
závislosti y na čase,
tj. bodů o souřadnicích $[t, y]$

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda

#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```

popisky osy x a y

Python – příklad rovnoměrný pohyb po kružnici

```
import numpy as np
import matplotlib.pyplot as plt

r=1 #polomer (1 m)
T=1 #perioda otaceni (1 s)
omega=2*np.pi/T #uhlova rychlost otaceni
dt=T/100 #casovy krok
t=np.arange(0,T+dt,dt) #pole casu
x=r*np.cos(omega*t) #pole x-ovych souradnic
y=r*np.sin(omega*t) #pole y-ovych souradnic

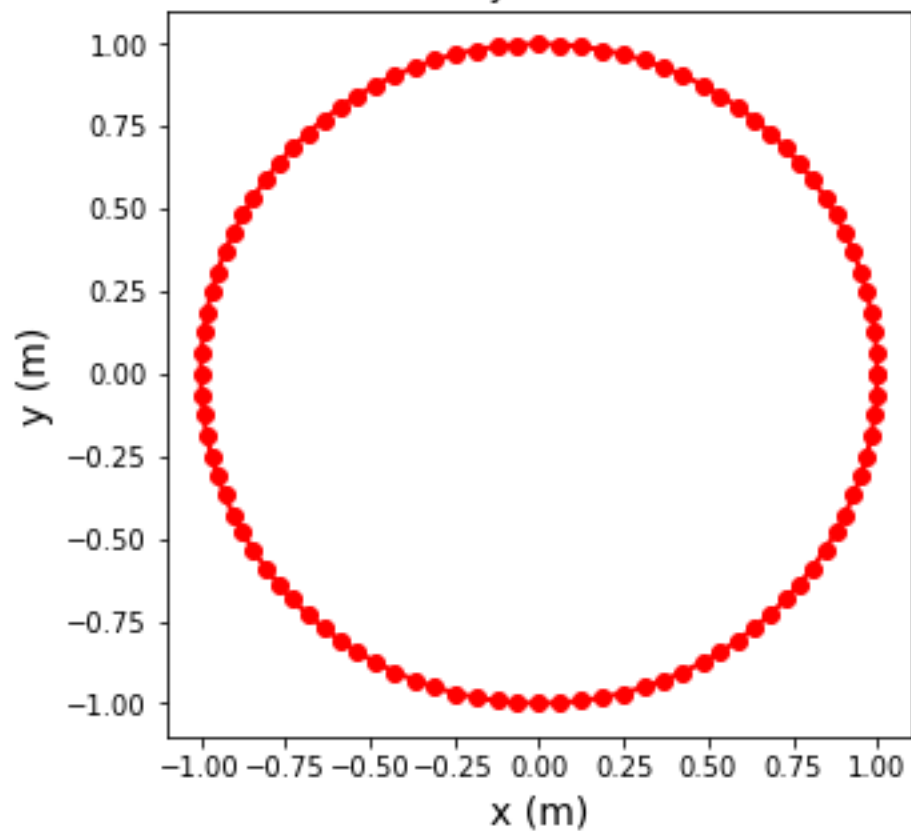
#casova zavislost spouradnic
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('časová závislost souřadnic',fontsize=14) #nazev grafu
ax.plot(t,x,c='green',lw = 2, marker='o',label='x(t)') #x versus t
ax.plot(t,y,c='blue',lw = 2, marker='o',label='y(t)') #y versus t
ax.set_xlabel('t (s)',fontsize=14) #popis osy x
ax.set_ylabel('x (m)',fontsize=14) #popis osy y
plt.legend(fontsize=14) #legenda
```

vypsání legendy do grafu

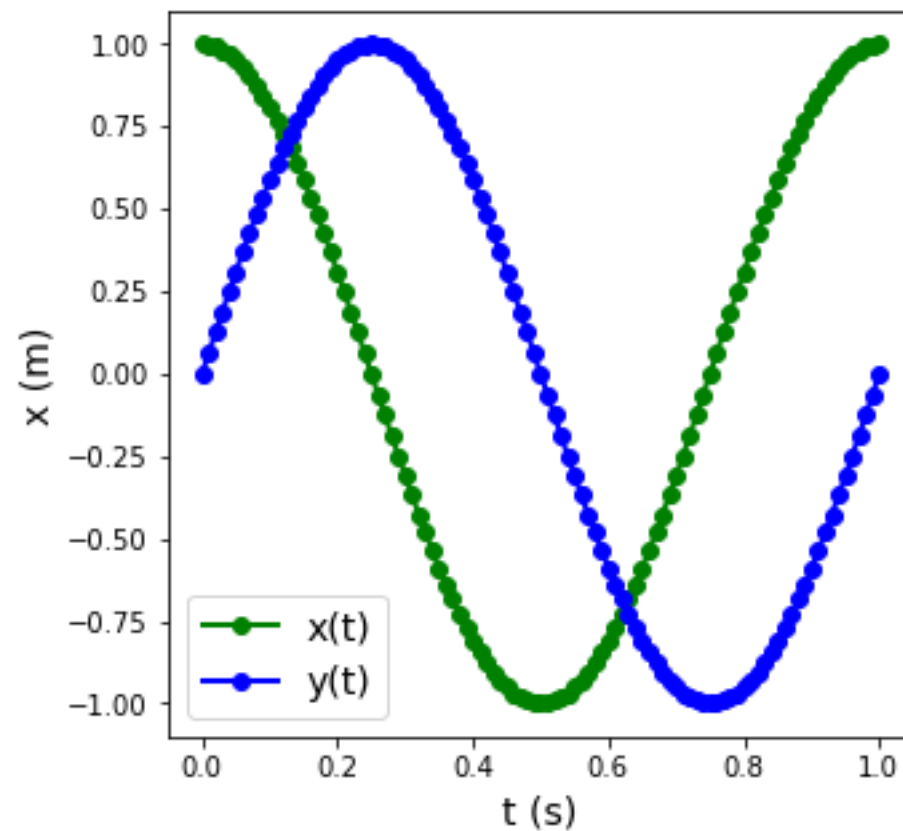
```
#trajektorie
fig,ax=plt.subplots(figsize=(5,5)) #def. obrazku
plt.title('trajektorie',fontsize=14) #nazev grafu
ax.plot(x,y,c='red',lw = 2, marker='o') #y versus x
ax.set_xlabel('x (m)',fontsize=14) #popis osy x
ax.set_ylabel('y (m)',fontsize=14) #popis osy y
```


Python – příklad rovnoměrný pohyb po kružnici

trajektorie



časová závislost souřadnic



Python - pole

definice pole: `x = np.empty(10, dtype = float)`

i-tý prvek pole: `x[i]`

index začíná od 0

počet prvků pole

typ prvků pole (implicitní je np.float64)

`np.shape(x) = 10`

jednorozměrné pole
10 prvků
float single precision

definice pole: `a = np.empty([10, 3], dtype = int)`

i-tý prvek pole: `a[i, j]`

indexy začínají od 0

počet řádků

počet sloupců

typ prvků pole (implicitní je np.float64)

`np.shape(x) = (10, 3)`

dvourozměrné pole
10 x 3 prvků integer
tj. matice typu 10x3
obsahující celá čísla

Python - cyklus

cyklus for

```
import numpy as np

x=np.empty(10,dtype=float)

for i in range(np.size(x)):
    x[i]=np.sqrt(i)

print(x)
```

```
[0.  1.  1.41421356 1.73205081 2.  2.23606798 2.44948974 2.64575131 2.82842712 3. ]
```

cyklus for pro 2D pole

```
import numpy as np

a=np.empty([10,3],dtype=int)

for i in range(np.size(a,0)):
    for j in range(np.size(a,1)):
        a[i,j]=i+j

print(a)
```

```
[[ 0  1  2]
 [ 1  2  3]
 [ 2  3  4]
 [ 3  4  5]
 [ 4  5  6]
 [ 5  6  7]
 [ 6  7  8]
 [ 7  8  9]
 [ 8  9 10]
 [ 9 10 11]]
```

Python - funkce

definice funkce

```
import numpy as np

def abs_hod(x):
    if x > 0:
        return(x)
    else:
        return(-x)

x=-5
print('x = {0:4.1f}, |x| = {1:4.1f}'.format(x,abs_hod(x)))
```

```
x = -5.0, |x| = 5.0
```