



Programmeerimise süvendatud algkursus ITI0140

Graafikute ja diagrammide joonistamise pakett Matplotlib

Juhan.ernits@ttu.ee



Matplotlibi installeerimine

Matplotlib võimaldab Pythonis koostada välimuselt kõrge kvaliteediga diagramme ja kuvada funktsioonide graafikuid. Matplotlib on vastupidiselt Sympyle suure hulga sõltuvustega ja seetõttu on selle installeerimine oluliselt keerulisem kui Sympy puhul.

Matplotlibi veebileht: <http://matplotlib.org/>

Paki kaupa installimiseks vt: <http://www.scipy.org/install.html>

„unofficial“ Windows pakid:

<http://www.lfd.uci.edu/~gohlke/pythonlibs/>

Kõige lihtsam on sikutada endale Enthought Canopy või WinPythoni vaba versioon, milles sisaldub Python Matplotlibi ning kõigi vajalike sõltuvustega.

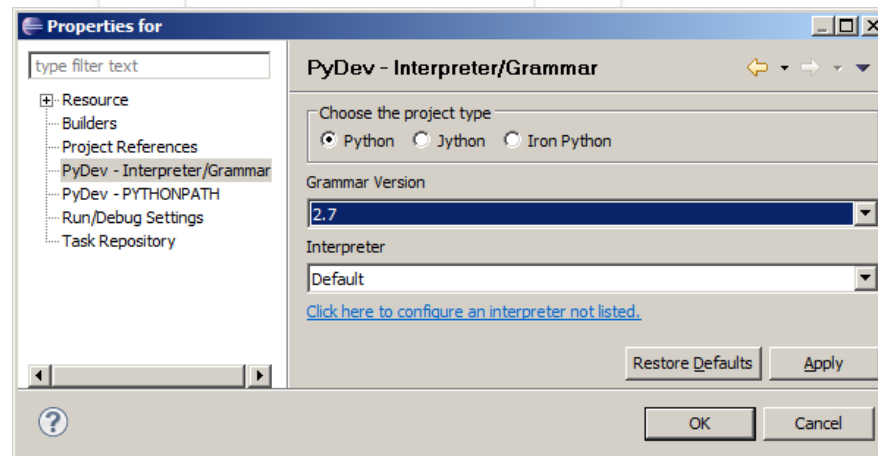
<https://www.enthought.com/products/epd/free/>

Samas võib ka installeerida IPythoni distributsiooni. IPython on Pythoni-põhine pakett, mis omab sarnaseid jooni nt Wolframi Mathematicaga.



Matplotlib ja Eclipse

- Kui installeerisite mõne eraldi Pythoni distributsiooni koos vajalike teekide/moodulitega, siis selleks, et Eclipse-s arendada vastavaid mooduleid pruukivat koodi, tuleb Eclipse-s vastavas PyDev projektis öelda, et kasutatakse vastavat Pythonit.
- Klõpsates Pythoni tüüpi projekti kataloogil hiire paremat klahvi, ning valides "Properties" saab valida uue Pythoni interpretaatori vajutades lingile "Click here to configure an interpreter not listed." Sealt tuleb ette anda siis nt Canopy sees sisalduv Pythoni interpretaator.
- Seejärel käib Eclipse vastava kataloogi läbi ja otsib välja võtmesõnad koodi kirjutamise toetamiseks lähtudes vastavas Pythoni kataloogis installeeritud moodultest.

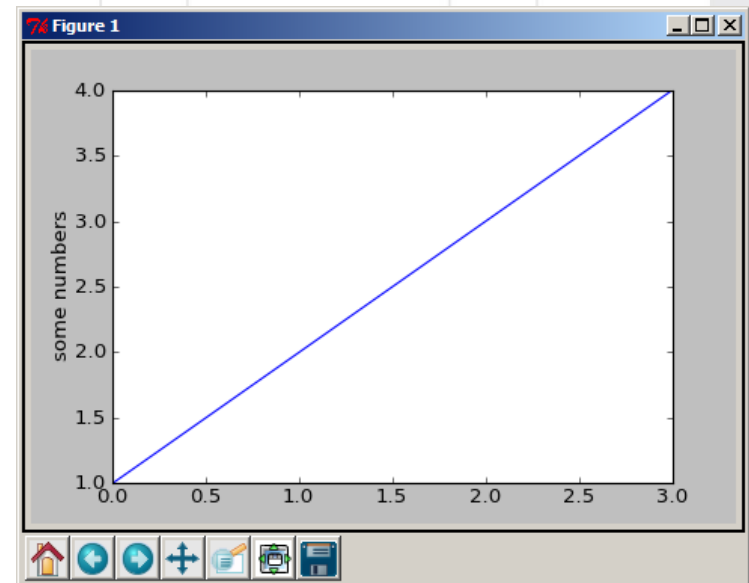




Graafikute joonistamine: arvude vektori kuvamine

```
import matplotlib.pyplot as plt  
plt.plot([1,2,3,4])  
plt.ylabel('some numbers')  
plt.show()
```

```
# Siin joonistatakse graafikule arvud koordinaatidega  
# (0;1), (1;2), (2;3) ja (3;4).  
# X telje väärtused võetakse vaikimisi andmevektori  
# väärtuste indeksite järgi.  
# Punktid ühendatakse pideva  
# joonega.
```

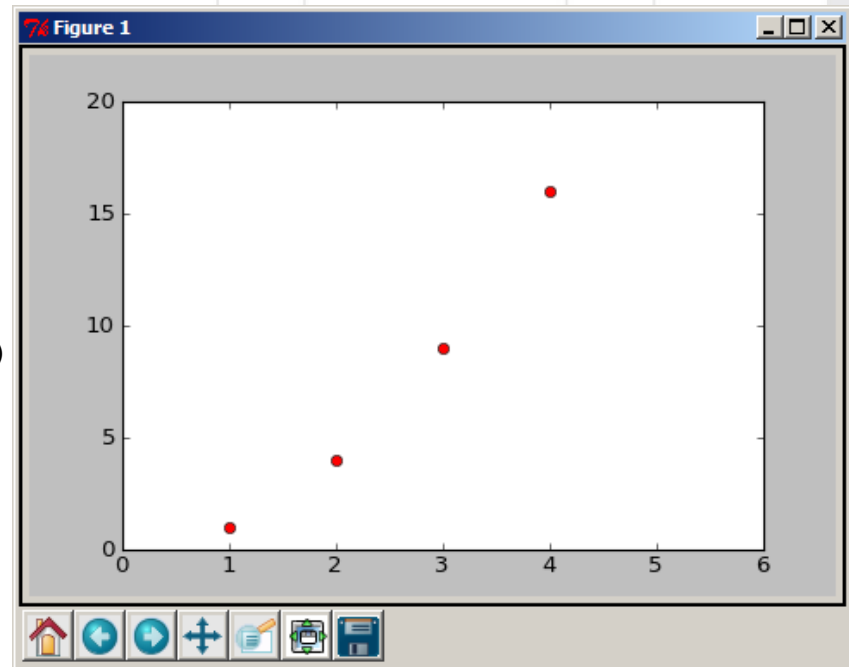




Punktide kandmine graafikule

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4],[1,4,9,16],"ro")
plt.axis([0,6,0,20])
plt.show()
```

```
# Siin on X ja Y koordi-
# naadid eraldi esitatud
# eraldi listidena.
# "ro" tähistab punast
# (red) kujundit 'o'
# (vaikimisi ühendati
# punktid pideva joonega)
```





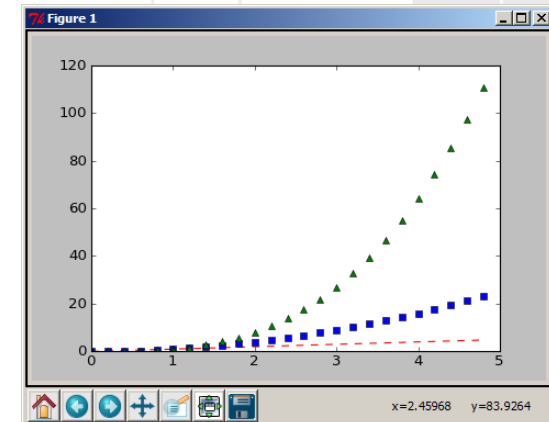
Mitme arvujada kuvamine, kus arvujadad on kuvatud funktsioonina

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

# Numpy on numbrilise arvutuse teek (vastupidiselt sympyle, mis on
# sümbolarvutuse teek)

# Arvuvahemiku defineerimine algpunktist 0 lõpppunkti 5 sammuga 0,2
t = np.arange(0.,5.,0.2)
# print(t) trükiks välja midagi sarnast
# [ 0.  0.2  0.4  0.6  0.8  1.  1.2  1.4  1.6  1.8  2.  2.2  2.4  2.6
#  2.8  3.  3.2  3.4  3.6  3.8  4.  4.2  4.4  4.6  4.8]

plt.plot(t, t, 'r--', t, t**2, "bs", t, t**3,"g^")
plt.show()
```





Mitme graafiku kõrvuti kuvamine

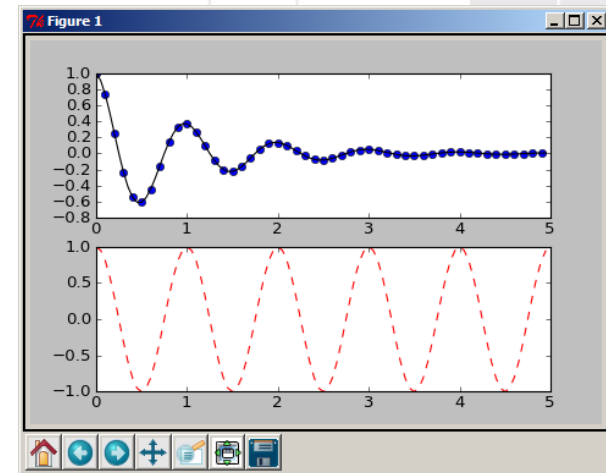
```
import numpy as np
import matplotlib.pyplot as plt

def f(t):
    return np.exp(-t)*np.cos(2*np.pi*t)

t1 = np.arange(0.0, 5.0, 0.1)
t2 = np.arange(0.0, 5.0, 0.05)

plt.figure(1)
# 2 graafikut, ühes tulbas,
# esimene graafik
plt.subplot(211)

plt.plot(t1, f(t1), 'bo', t2, f(t2), 'k')
# 2 graafikut, ühes tulbas, teine graafik.
plt.subplot(212)
plt.plot(t2, np.cos(2*np.pi*t2), 'r--')
plt.show()
```





Sympy ja visualiseerimine

```
# Selleks, et visualiseerida matemaatilisi konstruktsioone võib
# kasutada sympysse sisse-ehitatud visualiseerimist järgnevalt:
>>> from sympy import init_session
>>> init_session()
>>> Python console for SymPy 0.7.5 (Python 3.3.5-64-bit) (ground
    types: python)
```

These commands were executed:

```
>>> from __future__ import division
>>> from sympy import *
>>> x, y, z, t = symbols('x y z t')
>>> k, m, n = symbols('k m n', integer=True)
>>> f, g, h = symbols('f g h', cls=Function)
```

Documentation can be found at <http://www.sympy.org>



Sympy ja visualisointi 2

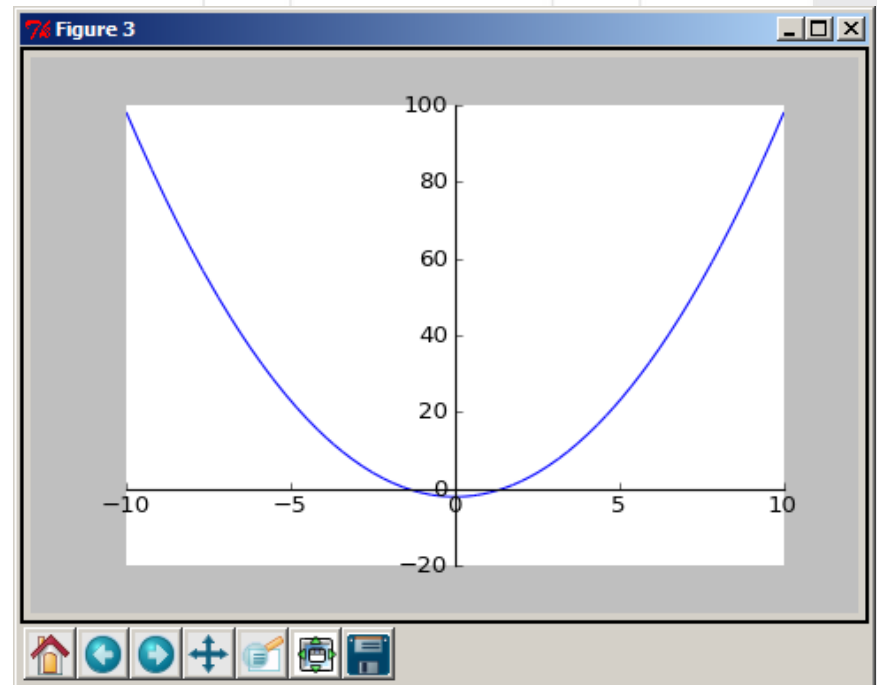
```
>>> f = Integral(x**2,x)
>>> f
      /
      |
      |  2
      | x  dx
      |
      /
>>> g = Derivative(1/x,x)
>>> g
d /1\
--|-|
dx\x/
```



Sympy ja Matplotlib

```
# Sympy võimaldab koostöös Matplotlibiga joonistada  
# funktsioonide graafikuid. Selleks teeb Sympy ise ära töö,  
# mida eelmises näites tegi numpy.arange() funktsioon, ehk  
# arvutab sobivad argumendid etteantud funktsioonile ja  
# neile vastavad väärtused.
```

```
>>> h=x**2 -2  
>>> plot(h)
```



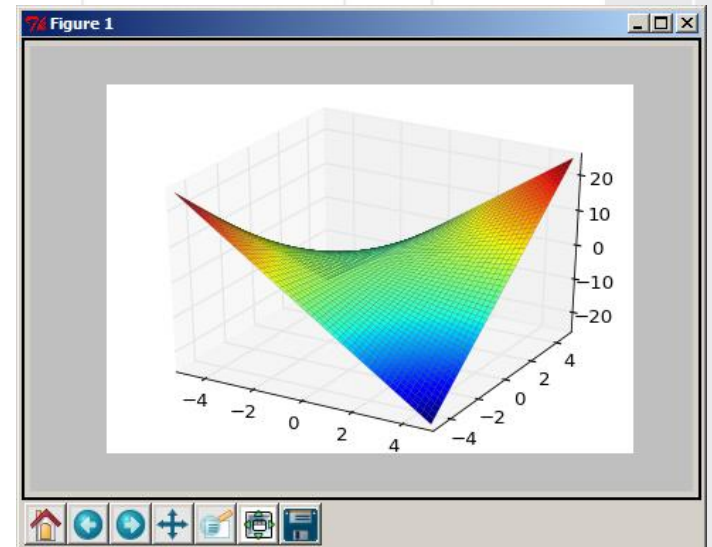


3D graafikud

Sympy võimaldab Matplotlibi abiga graafikule kanda ka pindu:

```
import sympy
import sympy.plotting as plt3d

x, y = sympy.symbols('x y')
plt3d.plot3d(x*y, (x,-5,5), (y,-5,5))
```



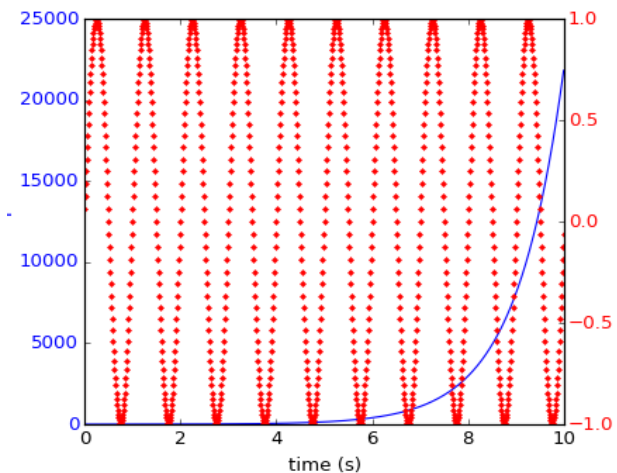


Kahe y telje lisamine graafikule

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

```
fig, ax1 = plt.subplots()
t = np.arange(0.01, 10.0, 0.01)
s1 = np.exp(t)
ax1.plot(t, s1, 'b-')
ax1.set_xlabel('time (s)')
# Make the y-axis label and tick labels match the line color.
ax1.set_ylabel('exp', color='b')
for tl in ax1.get_yticklabels():
    tl.set_color('b')
```

```
ax2 = ax1.twinx()
s2 = np.sin(2*np.pi*t)
ax2.plot(t, s2, 'r.')
ax2.set_ylabel('sin', color='r')
for tl in ax2.get_yticklabels():
    tl.set_color('r')
plt.show()
```





Andmed on keerulised

Andmete mõistmiseks on visualiseerimine väga oluline tööriist.

Samas ei ole alati visuaalselt paistvad seoses põhjuslikud, vt näiteks: <http://www.tylervigen.com>

Samuti on oluline, et andmed pärineks usaldusväärsest allikast. Eestis sobib näiteks Statistikaamet www.stat.ee



Ülesanne

Kirjutada Pythonis Matplotlib moodulitele tuginev programm, mis kuvab lehelt <http://www.stat.ee/rahvastikunaitajad-ja-koosseis>

- > Statistika andmebaas
- > Demograafilised põhinäitajad
- > RV030: SÜNNID, SURMAD JA LOOMULIK IIVE

Võetud andmed tulpades „Elussünnid“, „Surmad“, „Loomulik iive“ graafiliselt joontena. Loomulik iive on soovitatav kuvada teises skaalas.