VPython - symulacje fizyczne z grafiką 3D dla każdego

wykład 7

Dr hab. Adam Bzdak

Matplotlib

pixels.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\pixels.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

x = np.random.normal(0, 4, 50000)
y = np.random.normal(0, 4, 50000)

plt.plot(x, y, 'b,')

plt.axis([-15,15,-15,15])
plt.show()
```

Ln: 14 Col: 0

← **plt.style.use('classic')**

50000 punktów z rozkładu Gaussa
średnia = 0, szerokość = 4

niebieskie piksele

Aby dostać takie wykresy jak tutaj należy wpisać **plt.style.use('classic')**

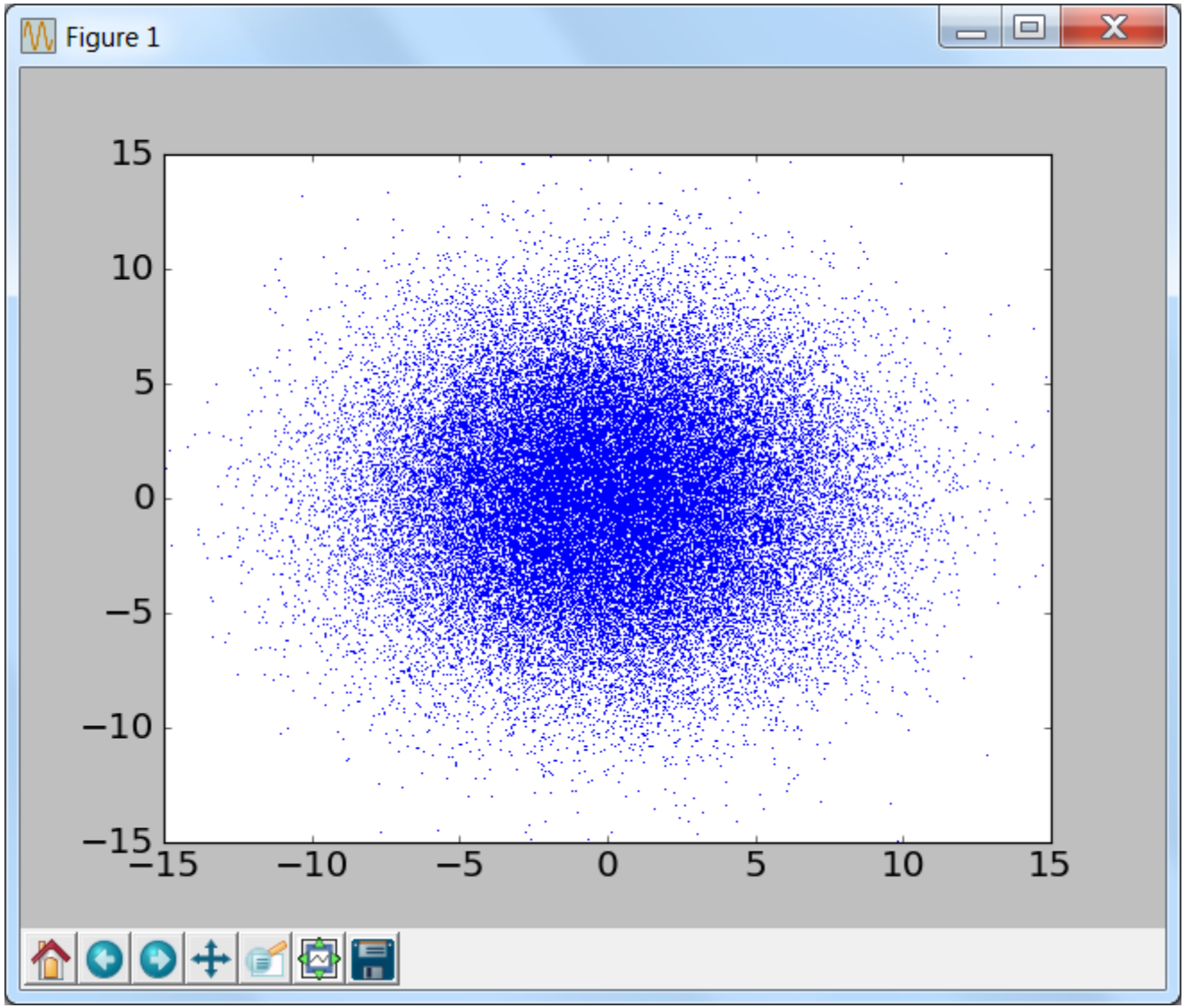


Figure 1
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

Z-order, kolejność rysowania

zorder.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\zorder.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

x = np.random.normal(0, 4, 100)
y = np.random.normal(0, 4, 100)

plt.plot(x, y, 'b*', ms=30, zorder=2)
plt.plot([-10,10], [0,0], 'r-', lw=10, zorder=1)

plt.axis([-15,15,-15,15])
plt.show()
```

Ln: 15 Col: 0

Kolejność rysowania

- zorder=1
- zorder=2
- zorder=3, itd

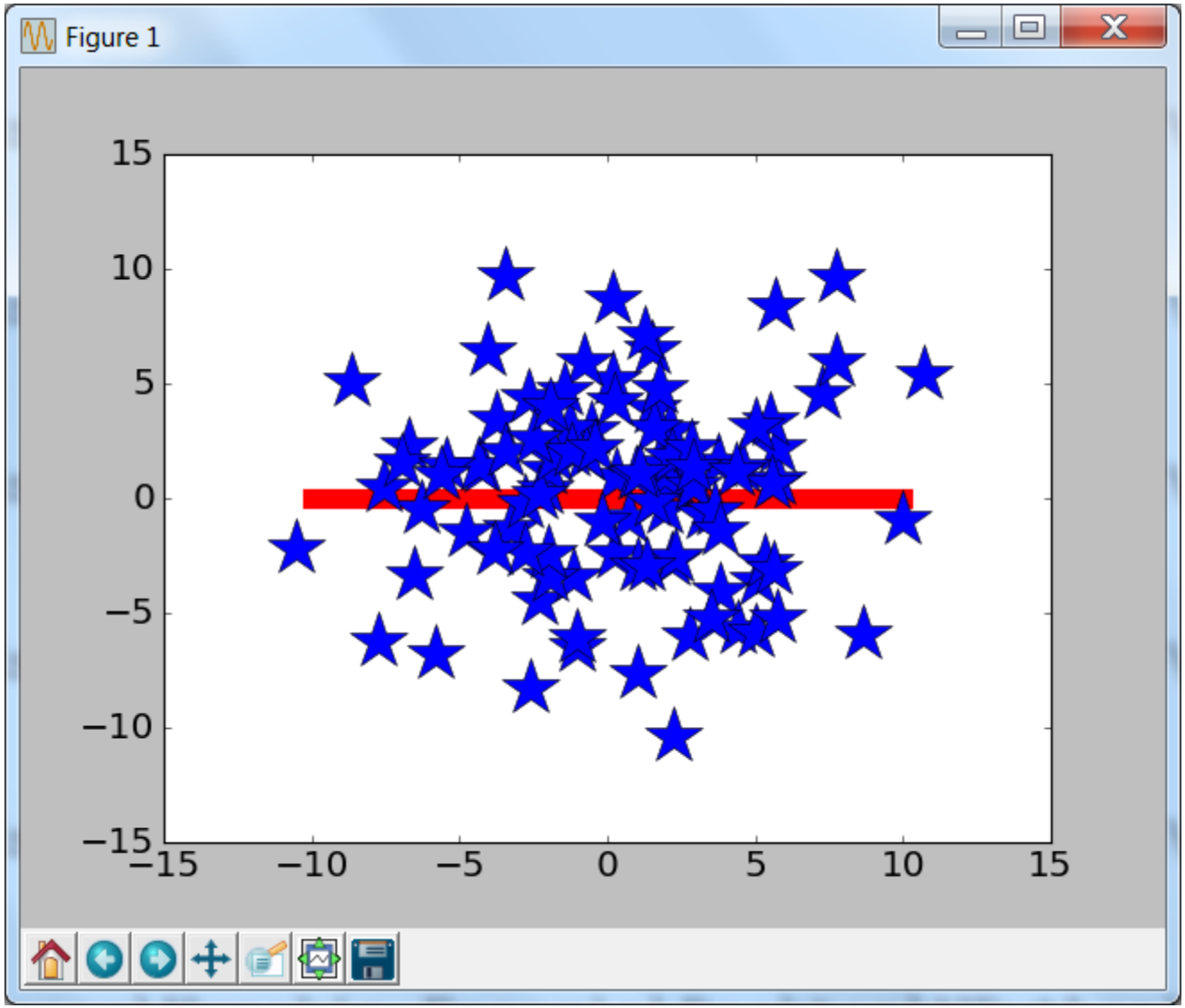


Figure 1
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

Histogram

hist_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\hist_1.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

L = np.random.uniform(0,10,1500)

plt.hist(L)

plt.show()
```

1500 liczb z przedziału [0,10)

histogram

Ln: 12 Col: 0

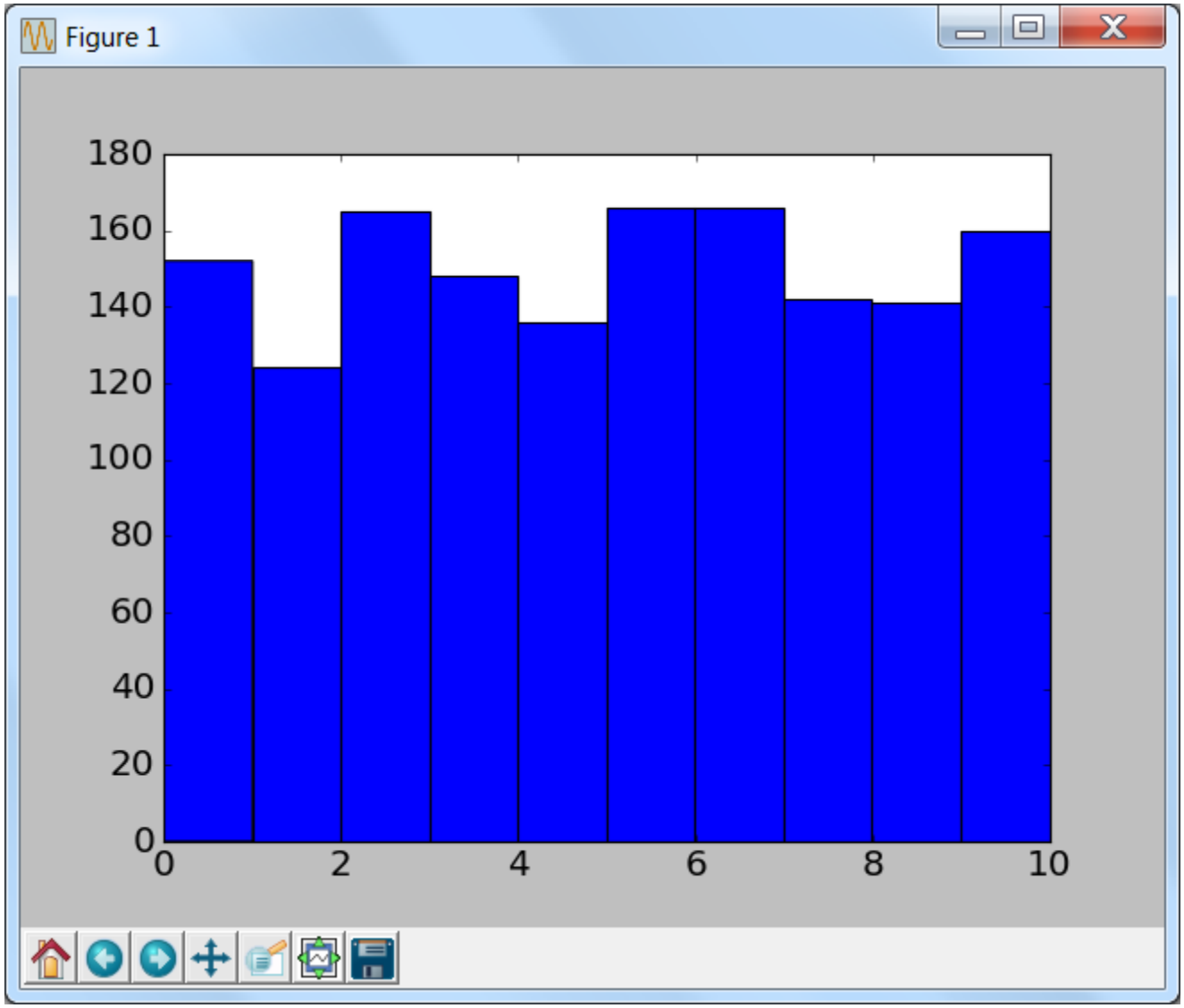


Figure 1
180
160
140
120
100
80
60
40
20
0
0
2
4
6
8
10

Histogram

hist_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\hist_2.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

L = np.random.uniform(0,10,1500)

plt.hist(L, color='green', bins=40, normed=1)

plt.show()
```

Ln: 12 Col: 0

liczba binów, pole histogramu = 1

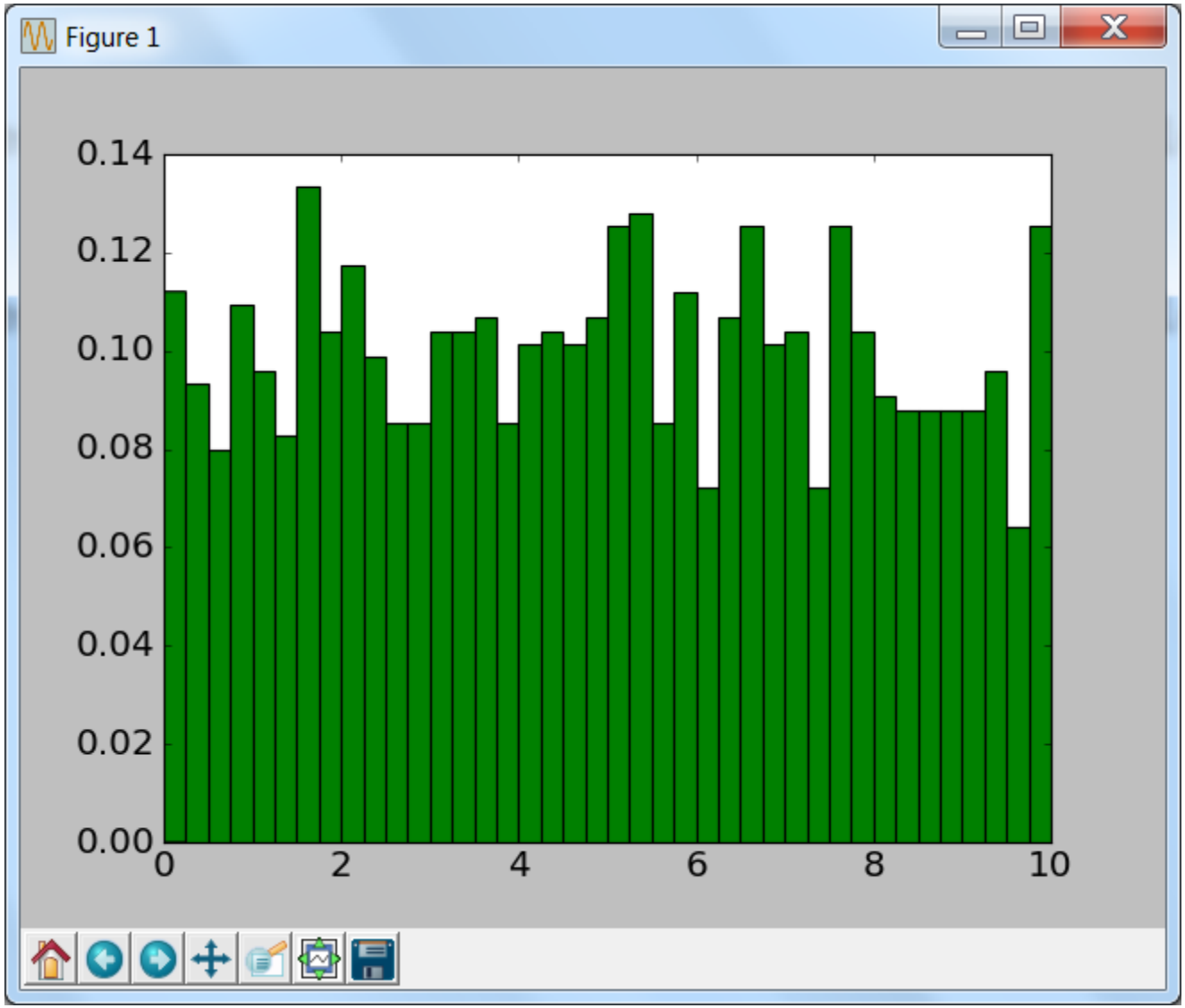


Figure 1
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0
2
4
6
8
10

Histogram

hist_3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\hist_3.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18
plt.rcParams['legend.fontsize'] = 18

L1 = np.random.uniform(0,10,9500)
L2 = np.random.normal(0,3,9500)

plt.hist(L1, color='g', bins=50, normed=1, alpha=0.8,
         label='uniform')
plt.hist(L2, color='b', bins=50, normed=1, alpha=0.5,
         label='normal')

plt.axis([-10, 10, 0, 0.16])
plt.xlabel('x')
plt.ylabel('y')
plt.legend(loc='upper left', frameon=False)
plt.show()
```

Ln: 21 Col: 0

transparentność ↓

Proszę przetestować (ze zmianą liczb):

plt.legend(bbox_to_anchor=(0.5,0.75), frameon=False)

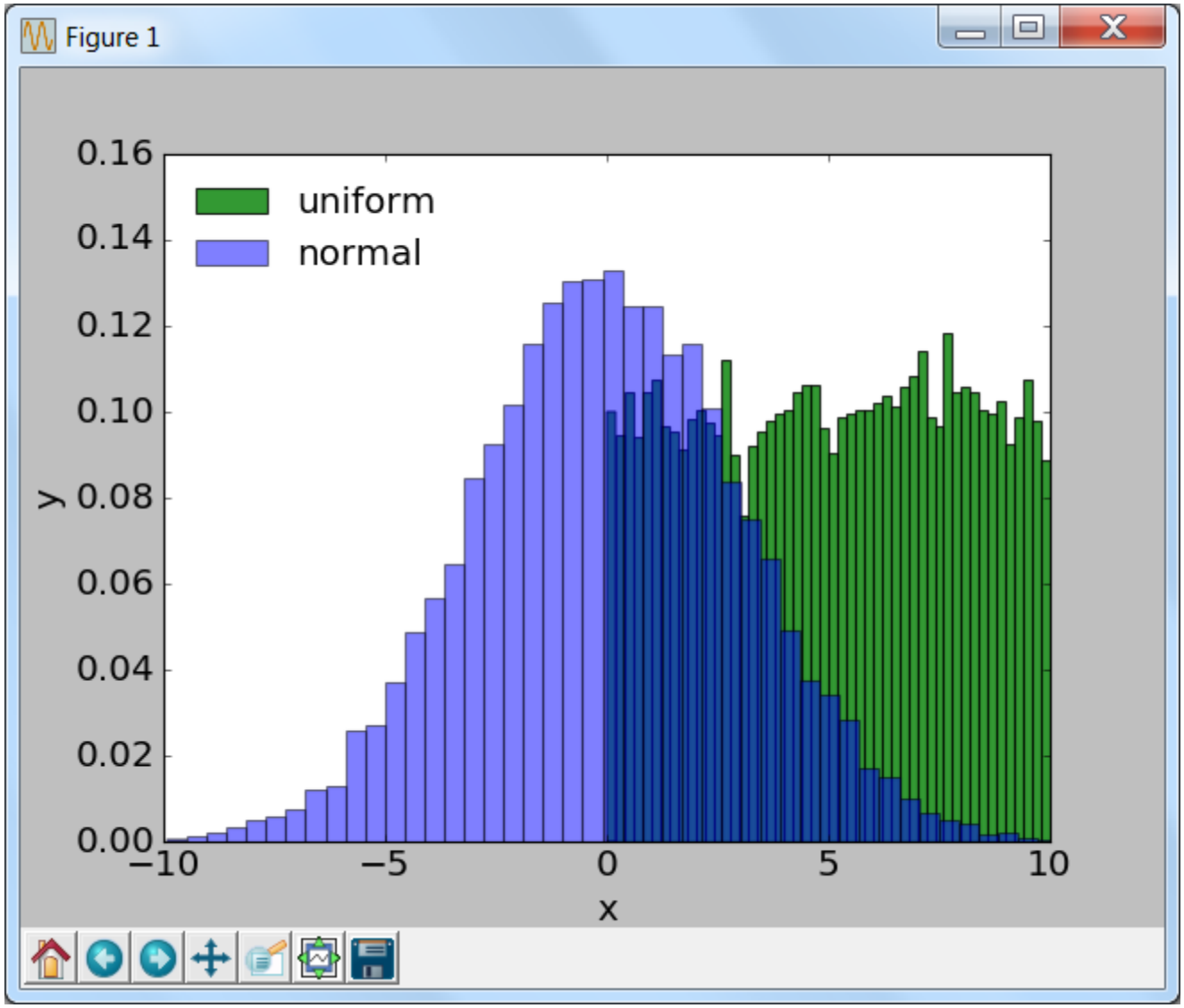


Figure 1
uniform
normal
0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
y
−10
−5
0
5
10
x

Histogram

hist_4.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\hist_4.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

L = np.random.normal(0,3,10**6)

mybins = np.arange(-10,11,2)
# [-10  -8  -6  -4  -2   0   2   4   6   8  10]

plt.hist(L, color='r', bins=mybins, normed=1)

plt.show()
```

Ln: 15 Col: 0

Można definiować biny, tutaj:
bin 1 jest od -10 do -8
bin 2 jest od -8 do -6
bin 3 jest od –6 do -4
itd, i ostatni bin jest od 8 do 10

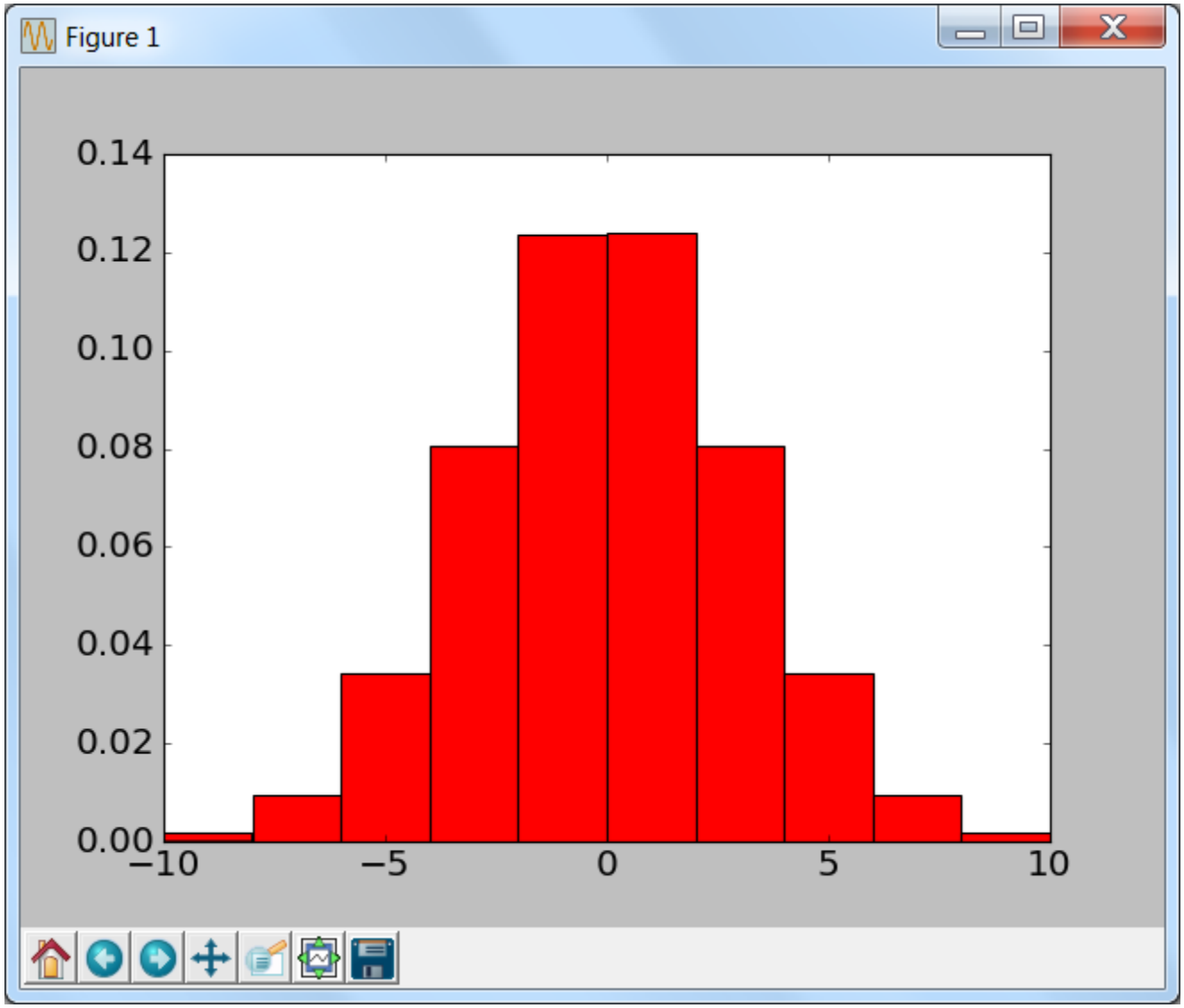


Figure 1
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
−10
−5
0
5
10

Histogram

hist_5.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\hist_5.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

L = np.random.poisson(5, 10**4)    # mean = 5

mybins = np.arange(-0.5,20,1)

plt.hist(L, color='b', bins=mybins, normed=1,
         histtype='step', lw=2)

plt.axis([-1,16,0,0.2])

plt.show()
```

Ln: 17 Col: 0

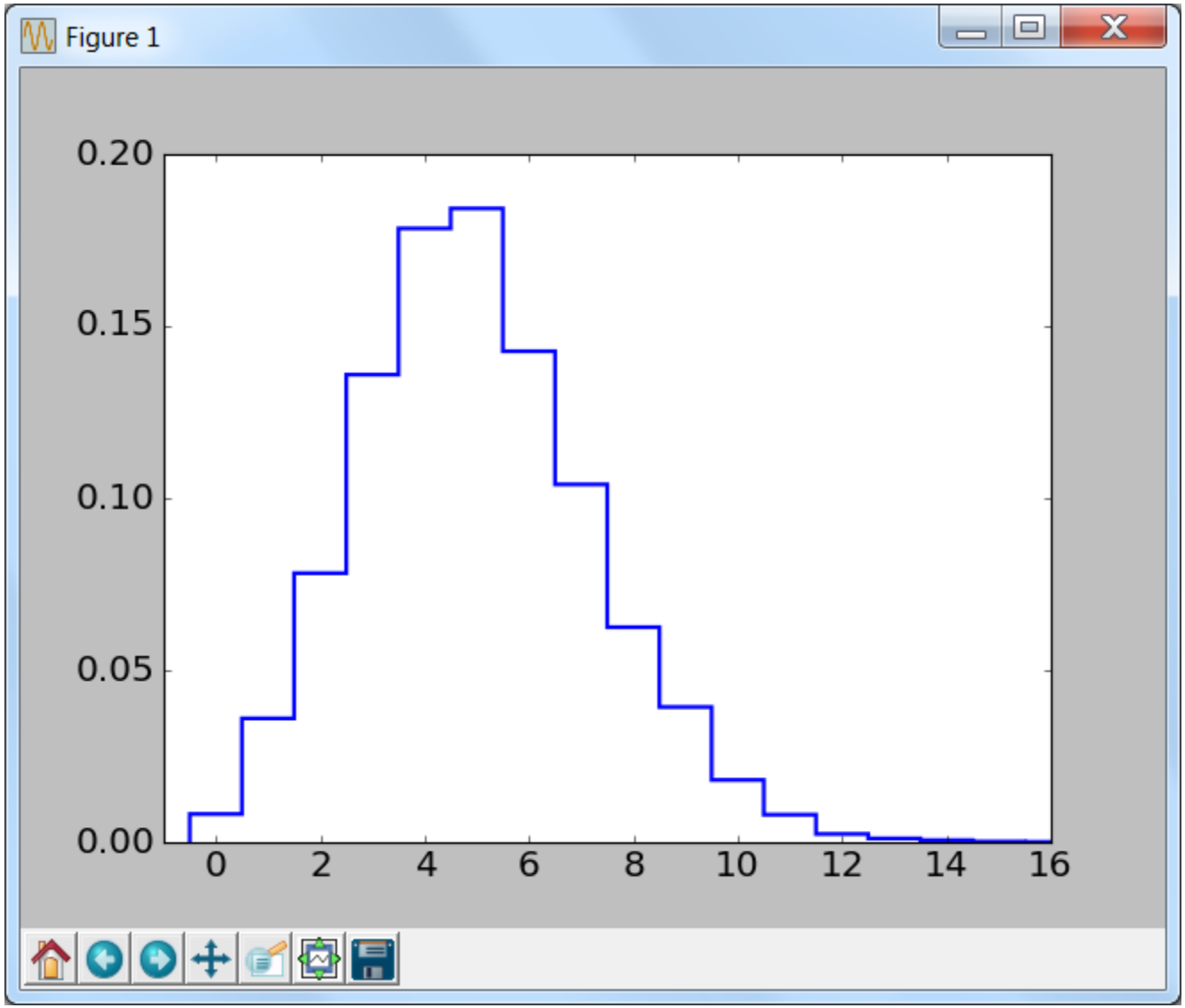


Figure 1
0.20
0.15
0.10
0.05
0.00
0
2
4
6
8
10
12
14
16

Subplots

lec_3_many_a.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\lec_3_many_a.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 15

x = np.arange(0,10,0.1)

plt.subplots_adjust(hspace=0.2, wspace=0.5)

plt.subplot(2,2,1)
plt.plot(x, np.sin(x), 'g-', lw=1.8)

plt.subplot(2,2,2)
plt.semilogy(x, np.exp(x), 'r-', lw=1.8)

plt.subplot(2,2,3)
plt.loglog(np.exp(-x), 'k--', lw=2.5)

plt.subplot(2,2,4)
plt.hist(np.random.normal(0,2,2000), bins=40, normed=1)

plt.show()
```

odstępy między wykresami

2 wiersze, 2 kolumny, wykres nr 1

semilogx dla osi x

2 wiersze, 2 kolumny, wykres nr 4

Ln: 11 Col: 0

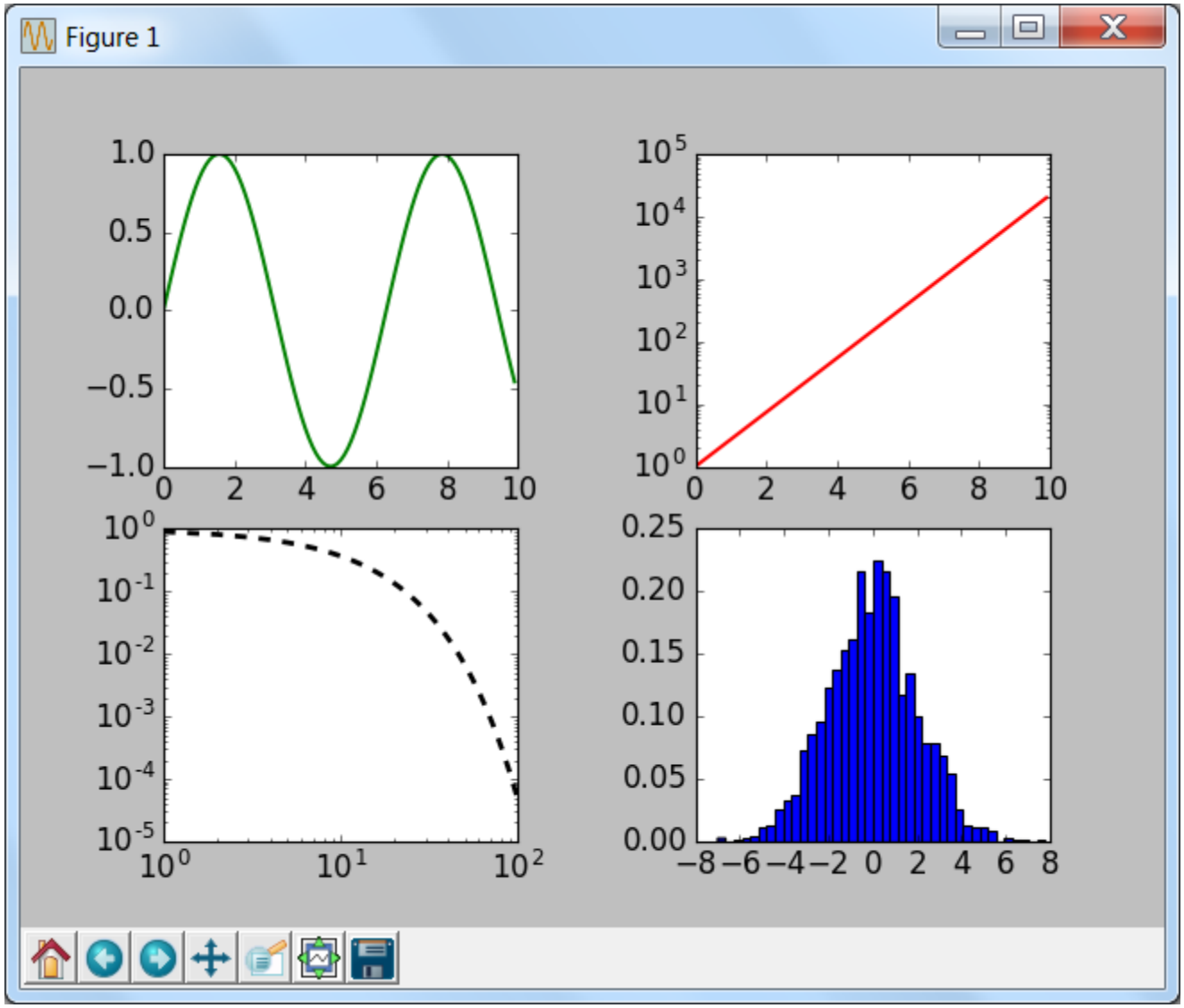


Figure 1
1.0
0.5
0.0
−0.5
−1.0
0 2 4 6 8 10
10^5
10^4
10^3
10^2
10^1
10^0
0 2 4 6 8 10
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^0 10^1 10^2
0.25
0.20
0.15
0.10
0.05
0.00
−8 −6 −4 −2 0 2 4 6 8

Facecolor, axis_bgcolor

facecolor_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\facecolor_2.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

plt.figure(figsize=(8, 6.9), facecolor='pink')

x = np.arange(0,10,0.01)
y = np.sin(x)

plt.plot(x, y, 'Lime', lw=5)

ax = plt.gca()
ax.set_axis_bgcolor('black')

plt.savefig('myfig.pdf', format='pdf', facecolor='pink')
plt.show()
```

Ln: 1 Col: 0

set_axis_bgcolor("black") → set_facecolor("black") w Matplotlib 3

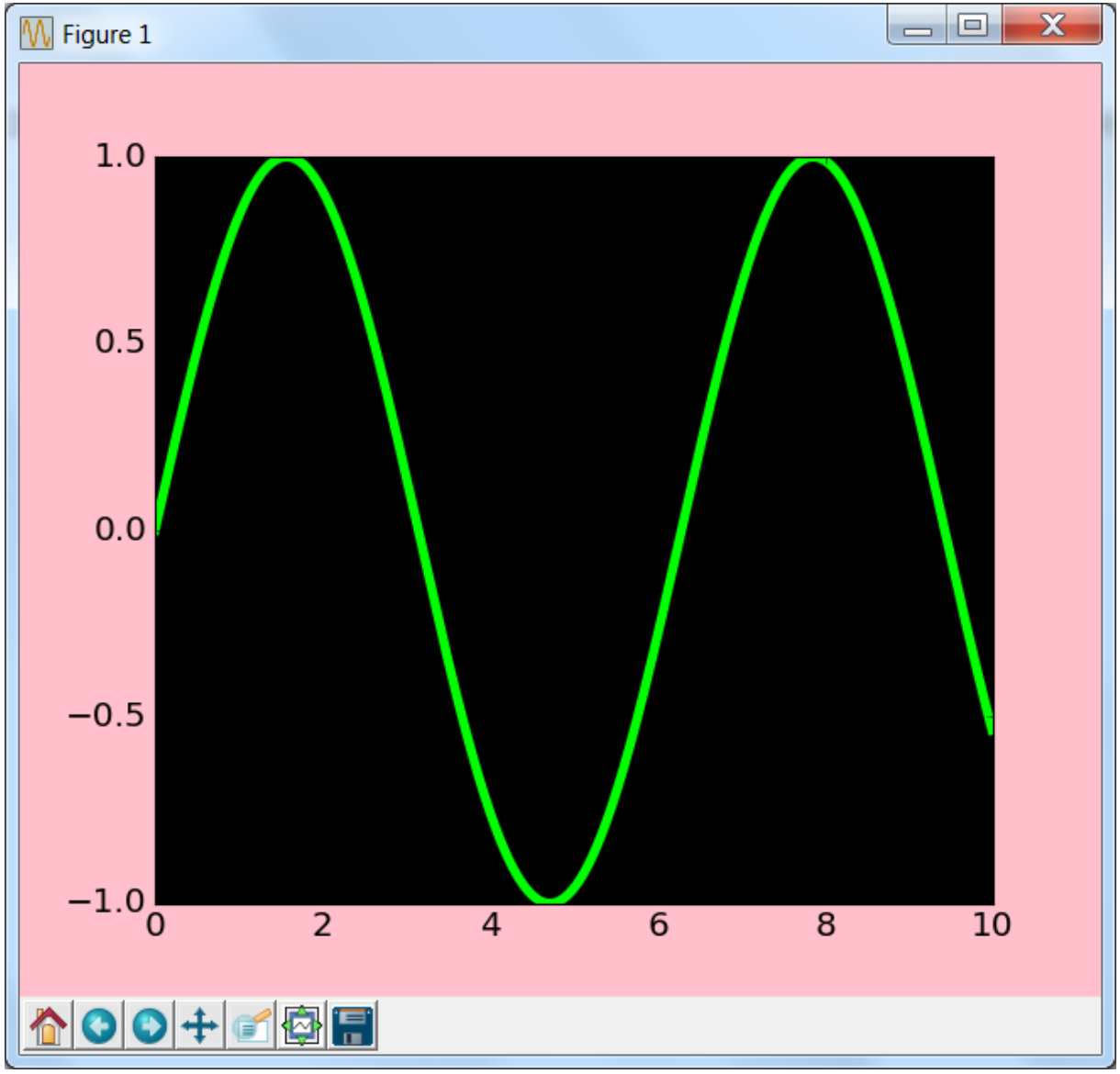


Figure 1
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10

meshgrid

meshgrid.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\meshgrid.py (2.7.12)

File Edit Format Run Options Window Help

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

x = np.array([1,2,3,4,5])
y = np.array([10,20])

(X,Y) = np.meshgrid(x,y)

print X
print Y, '\n'

Z = X + Y
print Z
```

wszystkie pary (x, y)

Ln: 15 Col: 0

```
[[1 2 3 4 5]
 [1 2 3 4 5]]
[[10 10 10 10 10]
 [20 20 20 20 20]]

[[11 12 13 14 15]
 [21 22 23 24 25]]
```

Ln: 12 Col: 4

2D, wykres konturowy

2d_1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\2d_1.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

x = np.linspace(0,2,50)
y = np.linspace(0,2,50)

(X,Y) = np.meshgrid(x,y)
Z = np.exp(-X**2) * np.exp(-Y**4)

levels = [0.03, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99]

a = plt.contour(X, Y, Z, levels, linewidths=1.8)

plt.clabel(a)

plt.xlabel('x')
plt.ylabel('y')
plt.axis([0,2,0,2])
plt.show()
```

kontury

clabel dodaje wartości, tutaj 0.03, 0.1, 0.3, itd..

Ln: 23 Col: 0

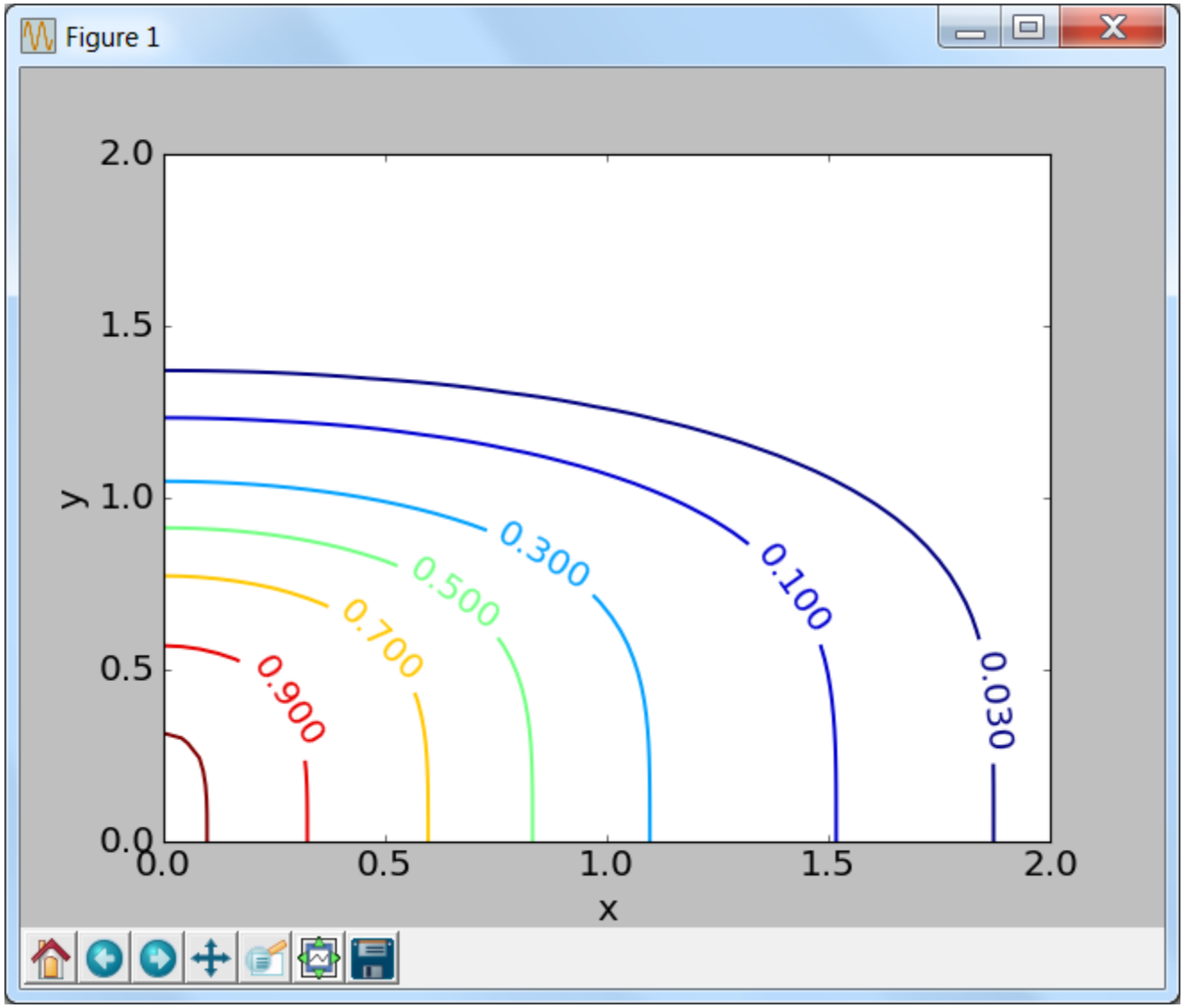


Figure 1
2.0
1.5
1.0
0.5
0.0
y
0.0
0.5
1.0
1.5
2.0
x
0.300
0.100
0.030
0.500
0.700
0.900

2D, wypełniony kontur (filled contour)

2d_2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\2d_2.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

x = np.linspace(0,2,50)
y = np.linspace(0,2,50)

(X,Y) = np.meshgrid(x,y)
Z = np.exp(-X**2) * np.exp(-Y**4)

levels = np.linspace(0,1,11)

plt.contourf(X, Y, Z, levels, cmap='YlGnBu')

plt.colorbar(orientation='vertical')

plt.axis([0,2,0,2])
plt.show()
```

Ln: 21 Col: 0

color map,
tutaj yellow-green-blue

lub orientation='horizontal'

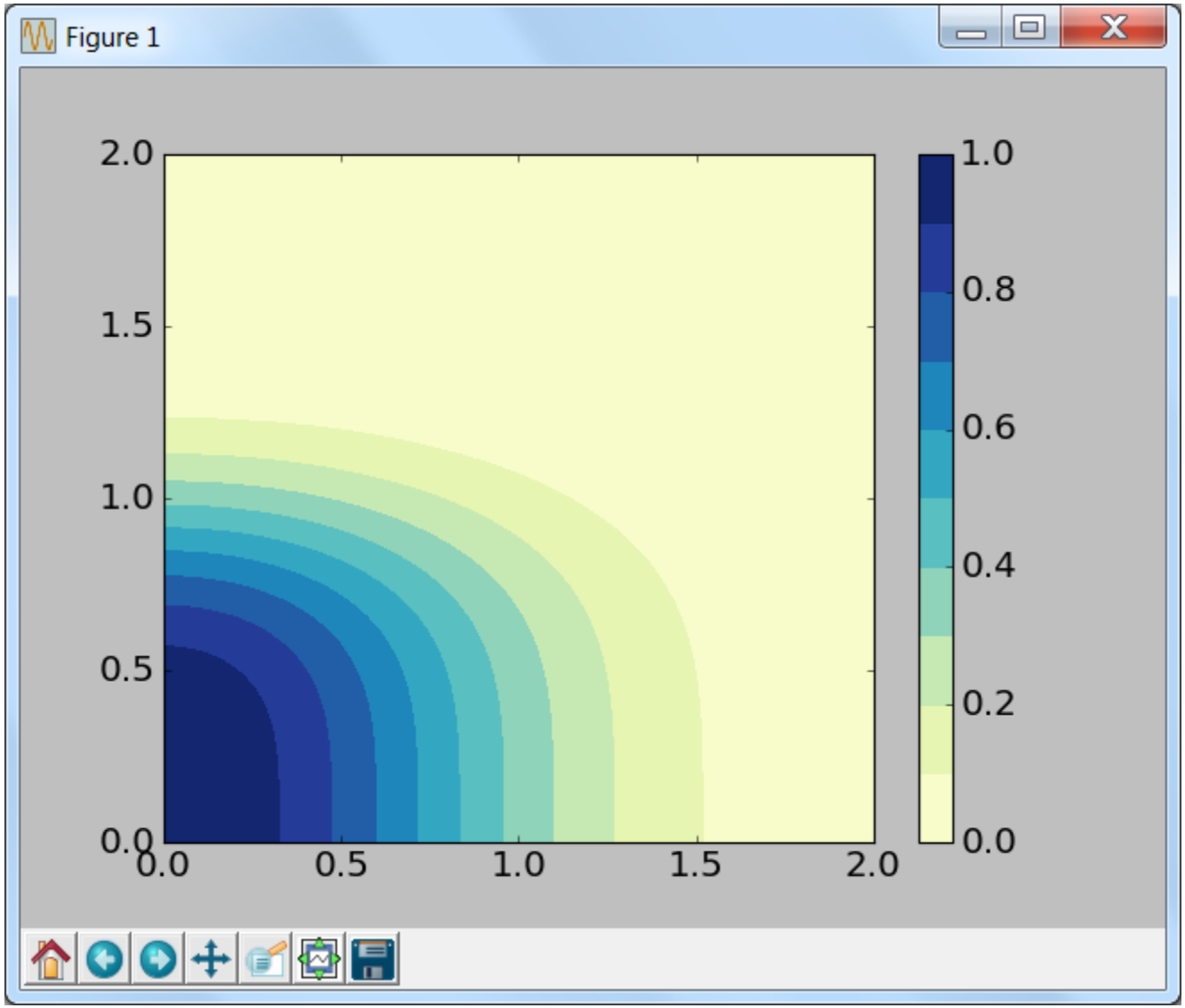
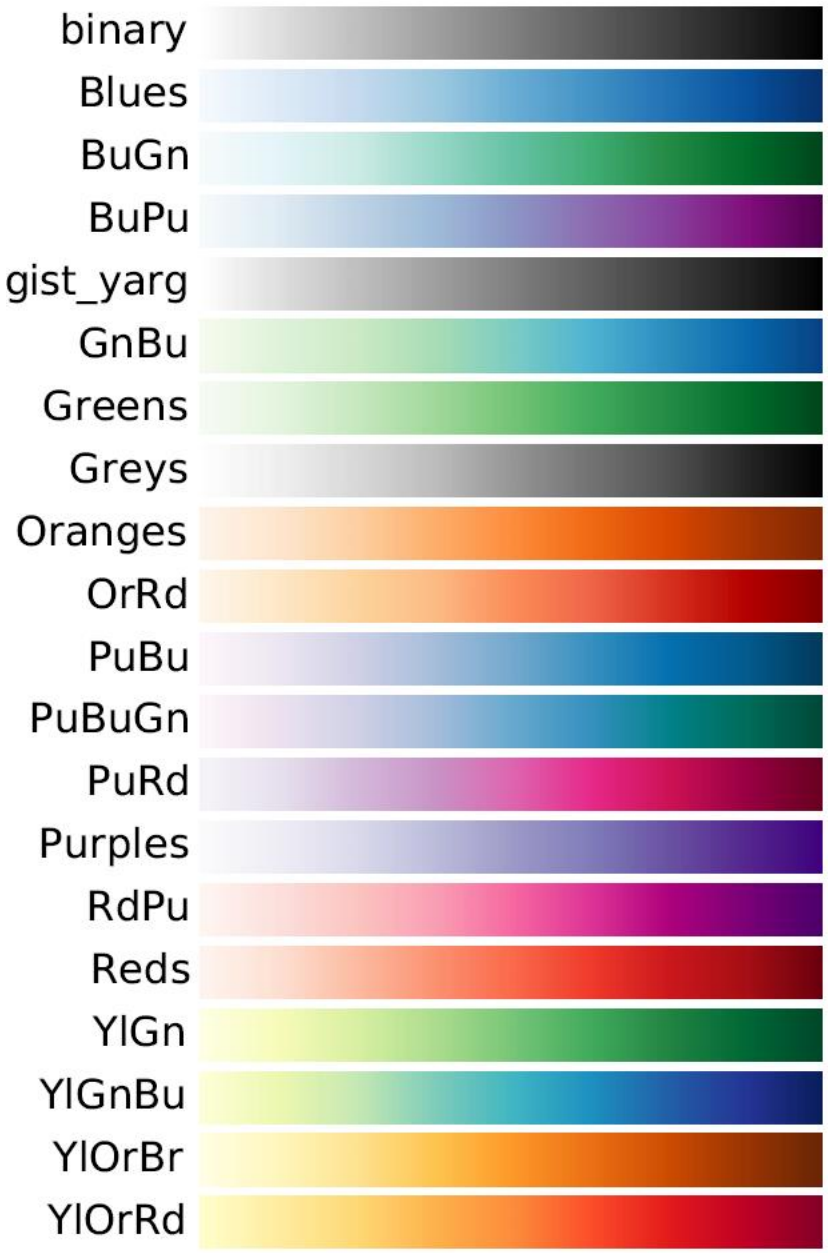
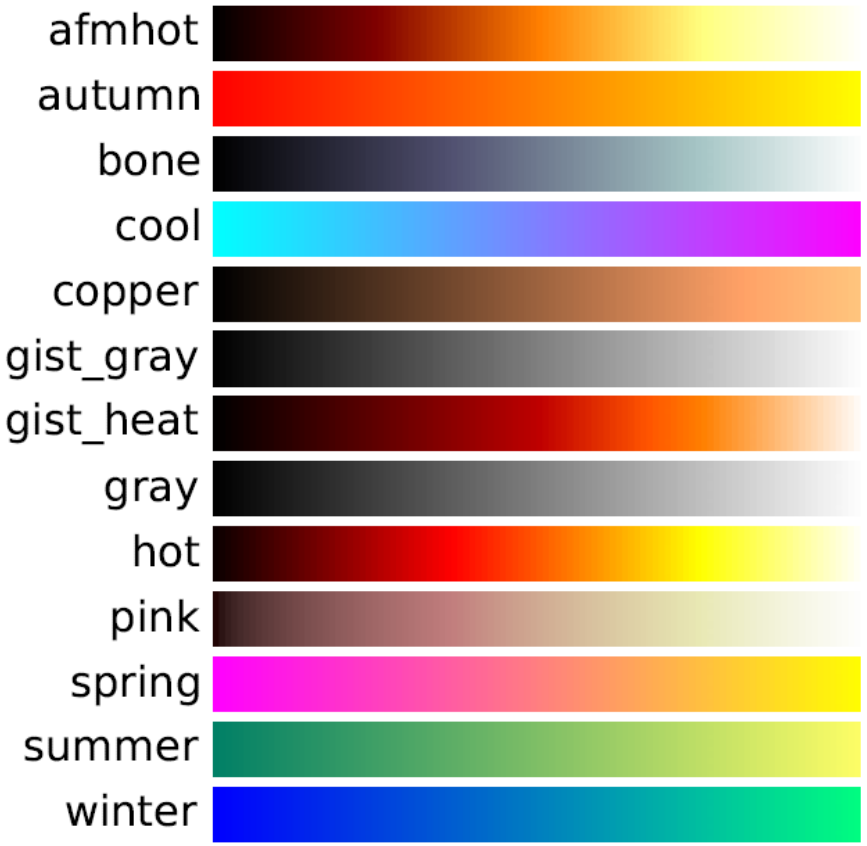
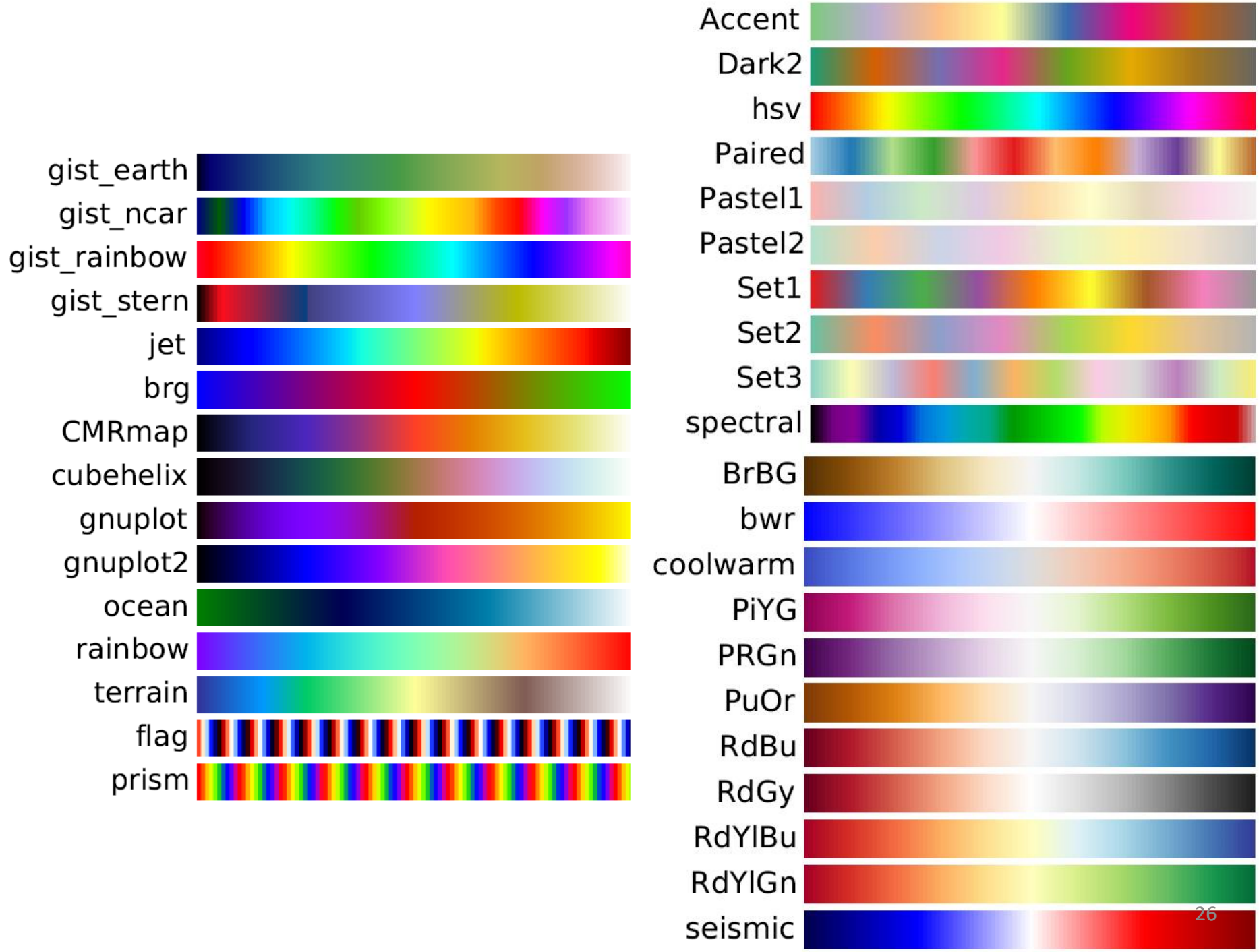


Figure 1
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
1.0
0.8
0.6
0.4
0.2
0.0

Kolory


binary
Blues
BuGn
BuPu
gist_yarg
GnBu
Greens
Greys
Oranges
OrRd
PuBu
PuBuGn
PuRd
Purples
RdPu
Reds
YlGn
YlGnBu
YlOrBr
YlOrRd


afmhot
autumn
bone
cool
copper
gist_gray
gist_heat
gray
hot
pink
spring
summer
winter


gist_earth
gist_ncar
gist_rainbow
gist_stern
jet
brg
CMRmap
cubehelix
gnuplot
gnuplot2
ocean
rainbow
terrain
flag
prism
Accent
Dark2
hsv
Paired
Pastel1
Pastel2
Set1
Set2
Set3
spectral
BrBG
bwr
coolwarm
PiYG
PRGn
PuOr
RdBu
RdGy
RdYlBu
RdYlGn
seismic

wykres 2D

2d_3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\2d_3.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

x = np.linspace(0,2,100)
y = np.linspace(0,2,100)

(X,Y) = np.meshgrid(x,y)
Z = np.exp(-X**2) * np.exp(-Y**4)

plt.pcolor(X, Y, Z)
plt.colorbar(ticks=np.linspace(0,1,5))  # !!!

plt.axis([0,2,0,2])
plt.show()
```

można dodać cmap

Ln: 19 Col: 0

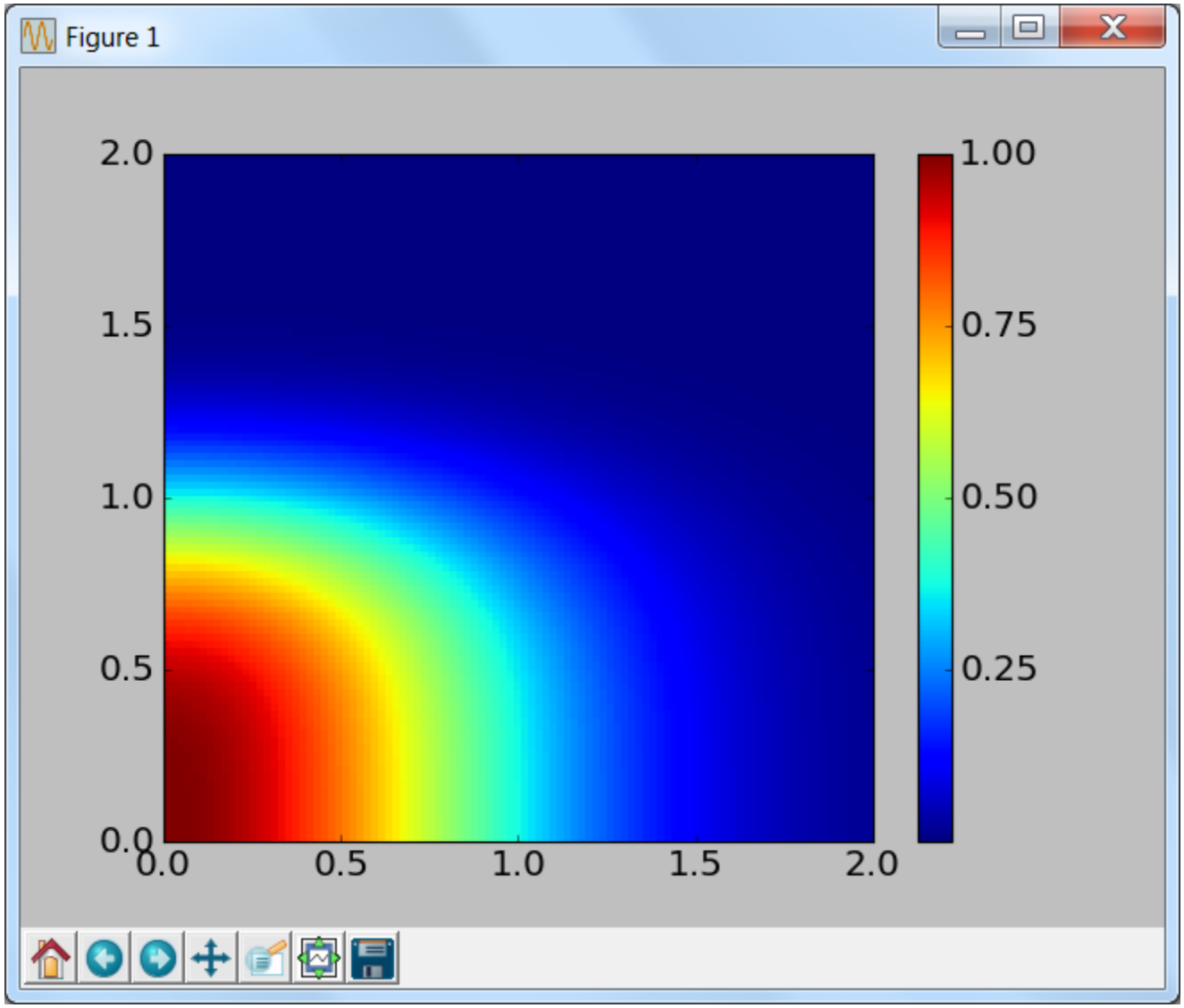


Figure 1
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
1.00
0.75
0.50
0.25

histogram 2D

2d_4.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\2d_4.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18
plt.rcParams['xtick.direction'] = 'out'   # !!!
plt.rcParams['ytick.direction'] = 'out'   # !!!

x = np.random.random(1000)
y = np.random.random(1000)

plt.hist2d(x, y, bins=15, normed=True, cmap='PuBu')

plt.colorbar(ticks=np.linspace(0,2.5,6))

plt.show()
```

Ln: 17 Col: 0

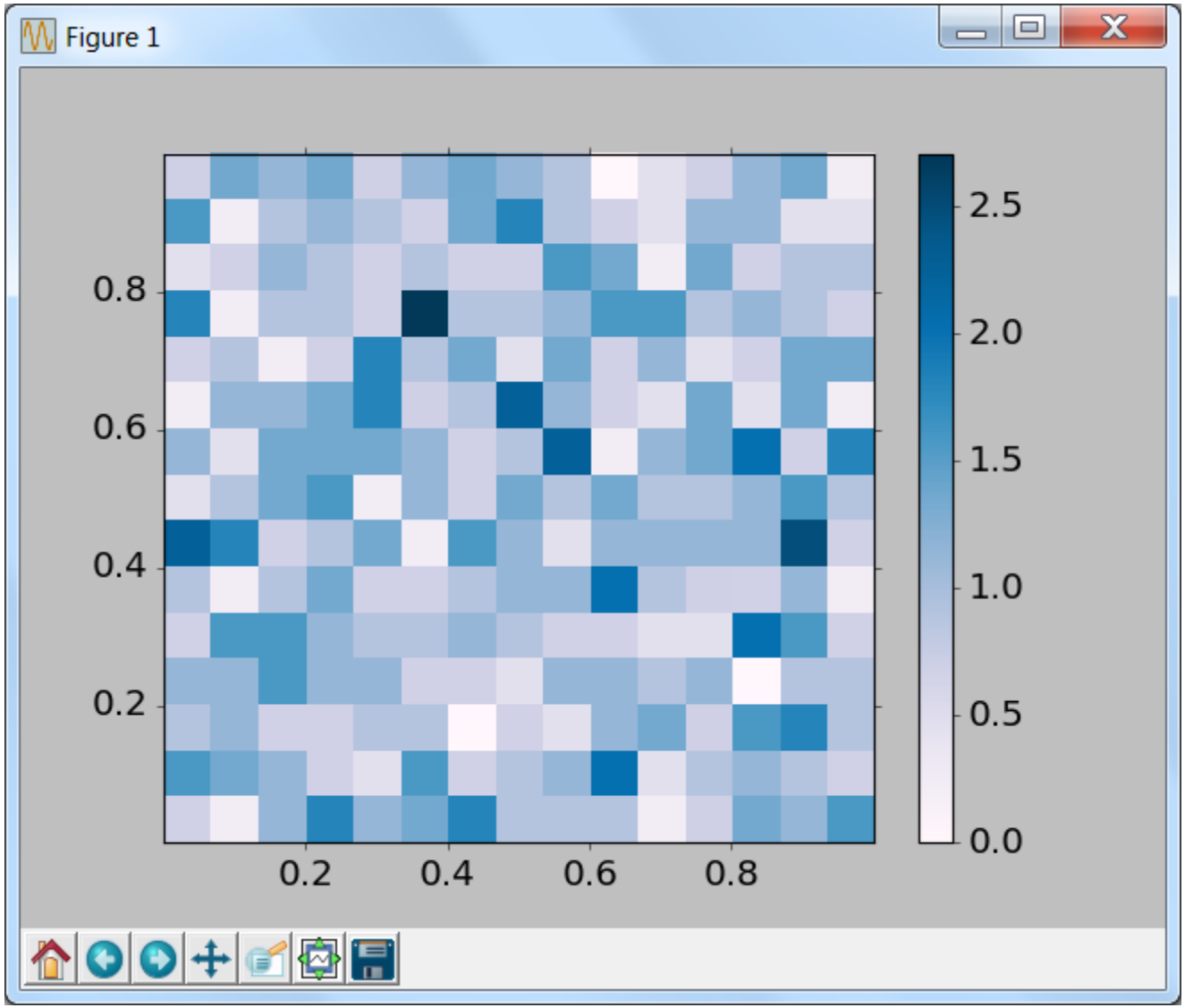


Figure 1
0.8
0.6
0.4
0.2
0.2
0.4
0.6
0.8
2.5
2.0
1.5
1.0
0.5
0.0

Dodatek

2D, wykres konturowy

2d_1b.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_7\2d_1b.py (2.7.12)

File Edit Format Run Options Window Help

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

plt.rcParams['font.size'] = 18

x = np.linspace(0,2,50)
y = np.linspace(0,2,50)

(X,Y) = np.meshgrid(x,y)
Z = np.exp(-X**2) * np.exp(-Y**4)

levels = [0.03, 0.1, 0.3, 0.5, 0.7, 0.9, 0.99]
a = plt.contour(X, Y, Z, levels, linewidths=(10,5,2),
                colors=('k','r','b'))

positions = [(0.2,1.4), (0.2,1.2), (0.2,1.1),
             (0.2,1), (0.2,0.8), (0.2,0.5), (0,0.15)]
plt.clabel(a, fontsize=15, manual=positions)

plt.axis([0,2,0,2])
plt.show()
```

Ln: 23 Col: 0

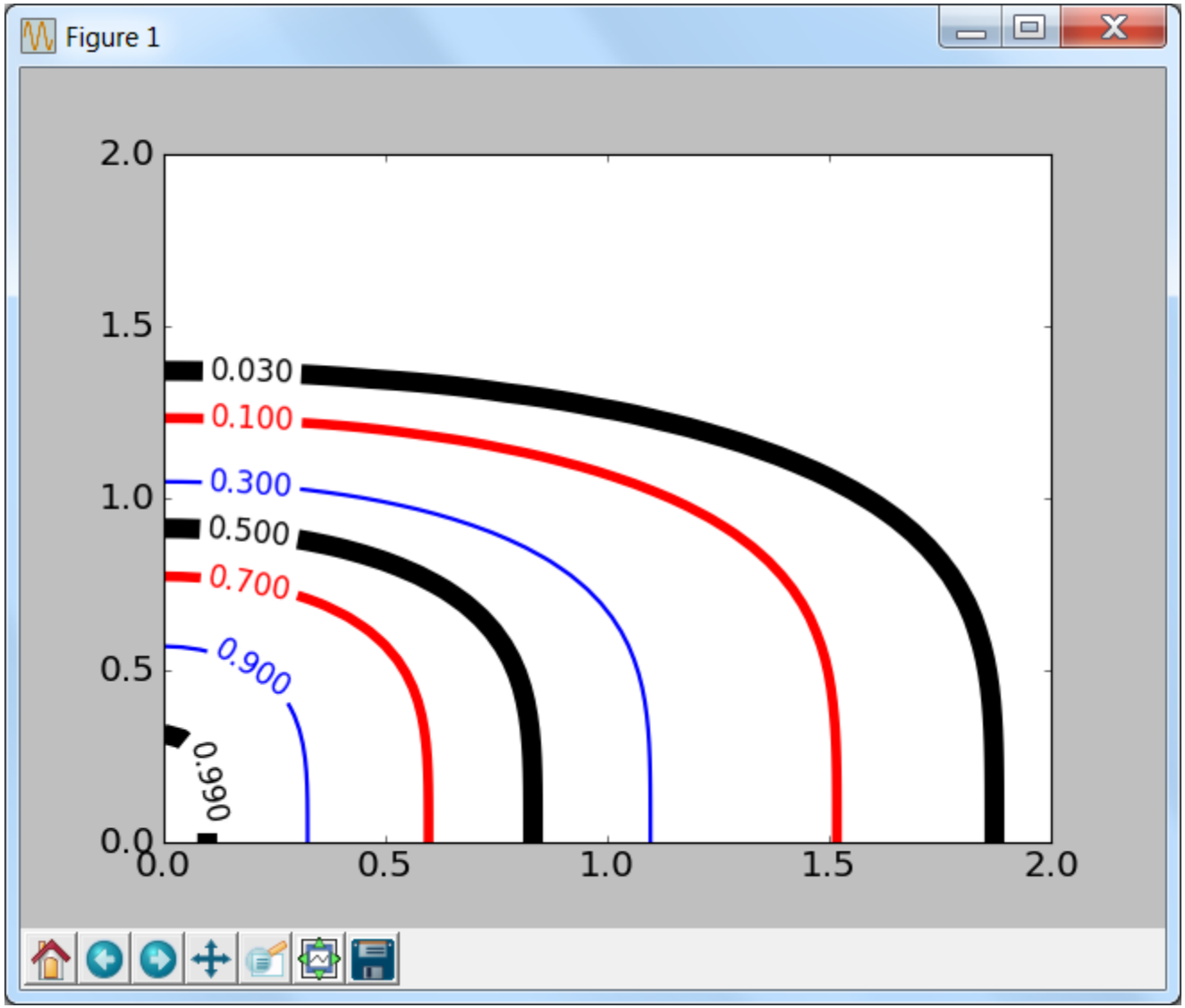


Figure 1
2.0
1.5
1.0
0.5
0.0
0.0
0.5
1.0
1.5
2.0
0.030
0.100
0.300
0.500
0.700
0.900
0.990