VPython - symulacje fizyczne z grafiką 3D dla każdego

wykład 4

Dr hab. Adam Bzdak

from __future__ import division musi być na samym początku

a.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_4/a.py

File Edit Format Run Options Windows Help

```
from __future__ import division
import math

res = 10/2
print 'res = ' + '\t', res
print 'res = ' + '\t\t\t', res
```

Ln: 6 Col: 30

dla Python 2.7

'\t' = tab

```
>>>
res =    5.0
res =                    5.0
>>>
```

Ln: 1 Col: 0

b.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_4/b.py

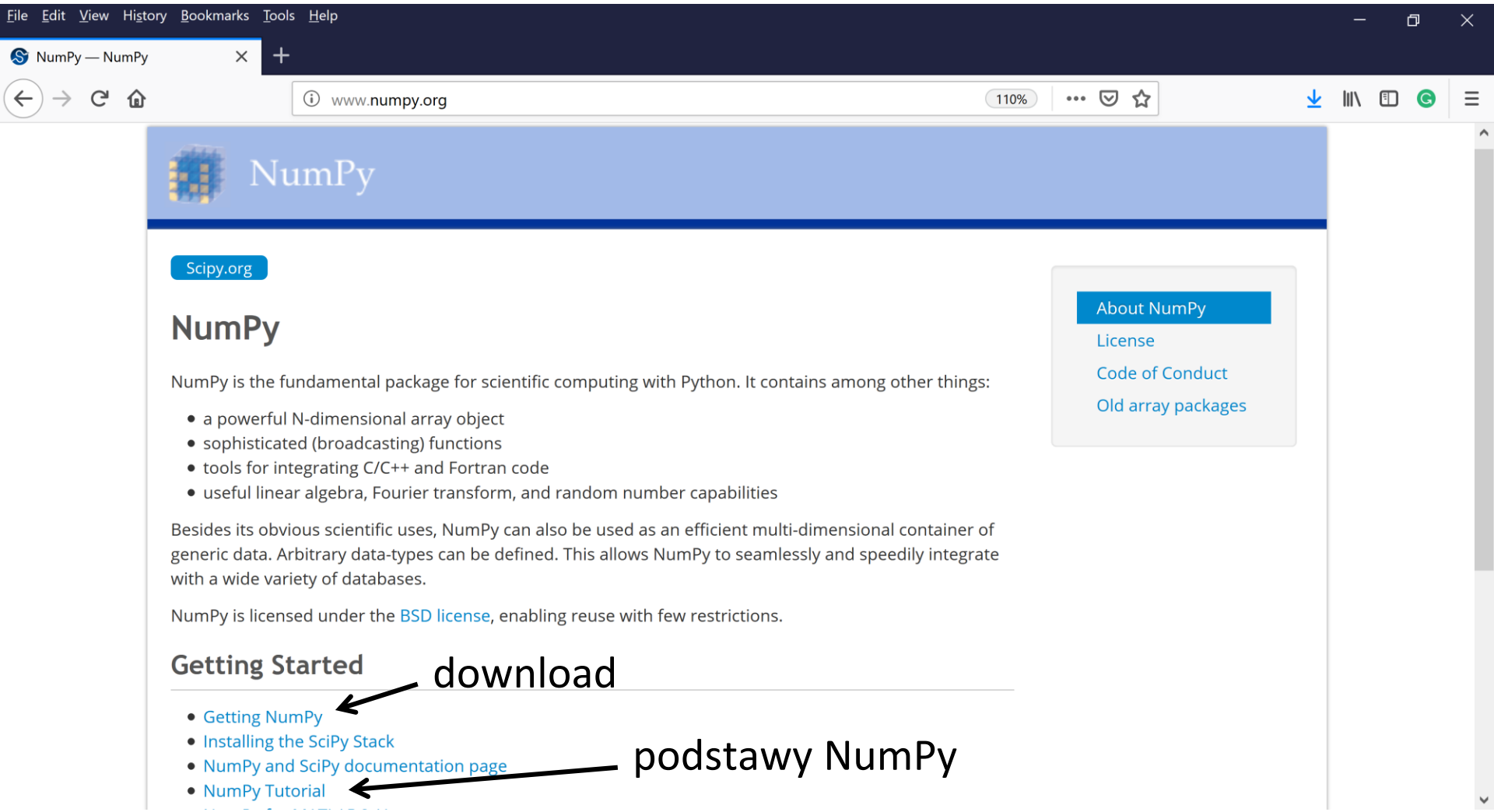
File Edit Format Run Options Windows Help

```
import math
from __future__ import division
```

Ln: 6 Col: 0

NumPy

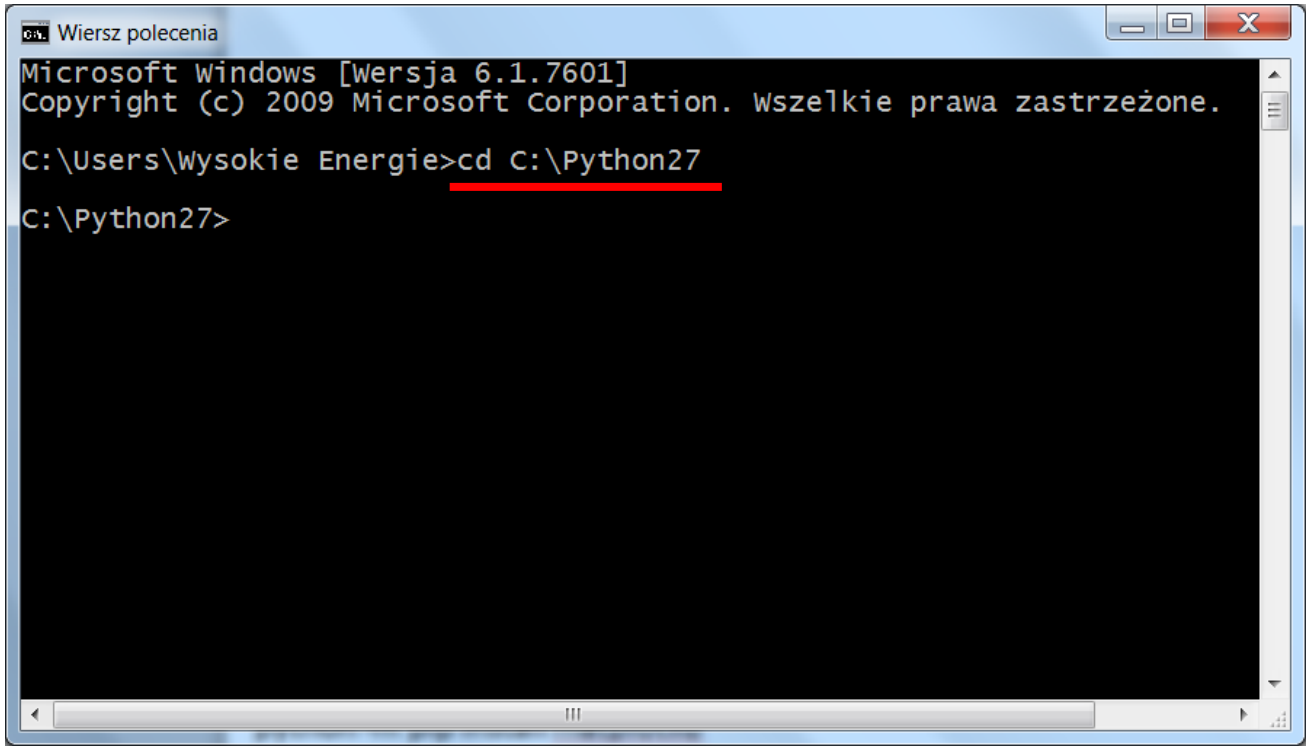
NumPy (Numerical Python for scientific computing)
http://www.numpy.org/



W wierszu poleceń wpisujemy

cd C:\Python27

jeśli tam jest zlokalizowany Python (lub PythonXX, np. Python 34, w zależności od wersji)



Następnie wpisujemy

python -m pip install numpy

lub

python -m pip install --user numpy instaluje NumPy dla local user

pip ?

Python 3.4 (lub nowszy) i Python 2.7.9 (lub nowszy) ma już pip.

Dla Python 2 ≤ 2.7.8 i Python 3 ≤ 3.3 należy zainstalować pip
http://stackoverflow.com/questions/4750806/how-do-i-install-pip-on-windows

NumPy - array

numpy1.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy1.py

File Edit Format Run Options Windows Help

```
import numpy as np

L = [1.0, 2.2, 3.1, 5.5]

a = np.array(L)

print a
print a.shape
print a.ndim
print a.size
print a.dtype.name
```

rozmiar array in każdym wymiarze
liczba wymiarów
całkowita liczba elementów
typ elementów

Ln: 13 Col: 0

```
>>>
[ 1.   2.2  3.1  5.5]
(4,)
1
4
float64
```

Ln: 1 Col: 0

NumPy, array

numpy2.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy2.py

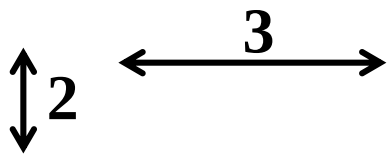
File Edit Format Run Options Windows Help

```
import numpy as np

L = [[1,2,3], [4,5,6]]

a = np.array(L)

print a
print a.shape
print a.ndim
print a.size
print a.dtype.name
```

Ln: 13 Col: 0

dwuwymiarowy array

```
>>>
[[1 2 3]
 [4 5 6]]
(2, 3)
2
6
int32
```



2 rzędy , 3 kolumny

Ln: 1 Col: 0

definiowanie typu

numpy3.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy3.py

File Edit Format Run Options Windows Help

```
import numpy as np

print np.array([1,2,3], dtype=np.float)

print np.array([1.0,2.0,3.0], dtype=np.int)

print np.array([[1,2,3], [4,5,6]], dtype=np.complex)
```

Ln: 9 Col: 0

```
>>>
[ 1.  2.  3.]
[1 2 3]
[[ 1.+0.j  2.+0.j  3.+0.j]
 [ 4.+0.j  5.+0.j  6.+0.j]]
>>>
```

Ln: 1 Col: 0

Dlaczego NumPy ?

numpy9.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy9.py

File Edit Format Run Options Windows Help

```
from __future__ import division
import numpy as np

a = np.array([1,2,3])

print a*10
print a/2
print a**2
print a + 10.5
print np.cos(a)
print a>0
```

Ln: 13 Col: 0

szybkie operacje na każdym elemencie

np.cos(a**)** – używamy funkcji z NumPy

```
>>>
[10 20 30]
[ 0.5  1.   1.5]
[1 4 9]
[ 11.5  12.5  13.5]
[ 0.54030231 -0.41614684 -0.9899925 ]
[ True  True  True]
```

Ln: 1 Col: 0

Szybkie obliczenia

speed.py - C:/Users/Wysokie Energie/Desktop/VPython_17_pl/lec_4/speed.py (2.7.12)

File Edit Format Run Options Window Help

```
import numpy as np
import time

L = range(1,10**7)
L = np.array(L)
print L

start = time.clock()
G1 = []
for i in L:
    G1.append(i + 10)
print time.clock() - start

start = time.clock()
G2 = L + 10
print time.clock() - start
```

Ln: 19 Col: 0

```
[      1       2       3 ..., 9999997 9999998 9999999]
3.47445617899
0.015455537019
>>>
```

Ln: 11 Col: 0

Dlaczego NumPy ?

numpy10.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy10.py

File Edit Format Run Options Windows Help

```
from __future__ import division
import numpy as np

a = np.array([1, 2, 3])

b = np.array([5.0, 4.0, 3.0])

print a+b
print a*b
print b/a
print a>b
print (a-b)==0
```

Ln: 14 Col: 0

```
>>>
[ 6.  6.  6.]
[ 5.  8.  9.]
[ 5.  2.  1.]
[False False False]
[False False  True]
```

Ln: 1 Col: 0

numpy10a.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_4/numpy10a.py

File Edit Format Run Options Windows Help

```
from __future__ import division
import numpy as np

def fun(x, y):
    print "let\'s calculate"
    return np.sqrt(x*y)

a = np.array([4,2,3])
b = np.array([4,6,7.0])

print fun(a,b)
```

Ln: 13 Col: 0

```
>>>
let's calculate
[ 4.          3.46410162  4.58257569]
>>>
```

Ln: 1 Col: 0

zeros, ones

numpy5.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy5.py

File Edit Format Run Options Windows Help

```
import numpy as np

print np.zeros(10)
print np.zeros(10, dtype=np.int)

print np.ones(10)
print np.ones(10, dtype=np.int)
```

Ln: 9 Col: 0

```
>>>
[ 0.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
[0 0 0 0 0 0 0 0 0 0]
[ 1.  1.  1.  1.  1.  1.  1.  1.  1.  1.]
[1 1 1 1 1 1 1 1 1 1]
>>>
```

Ln: 1 Col: 0

float64

zeros, ones

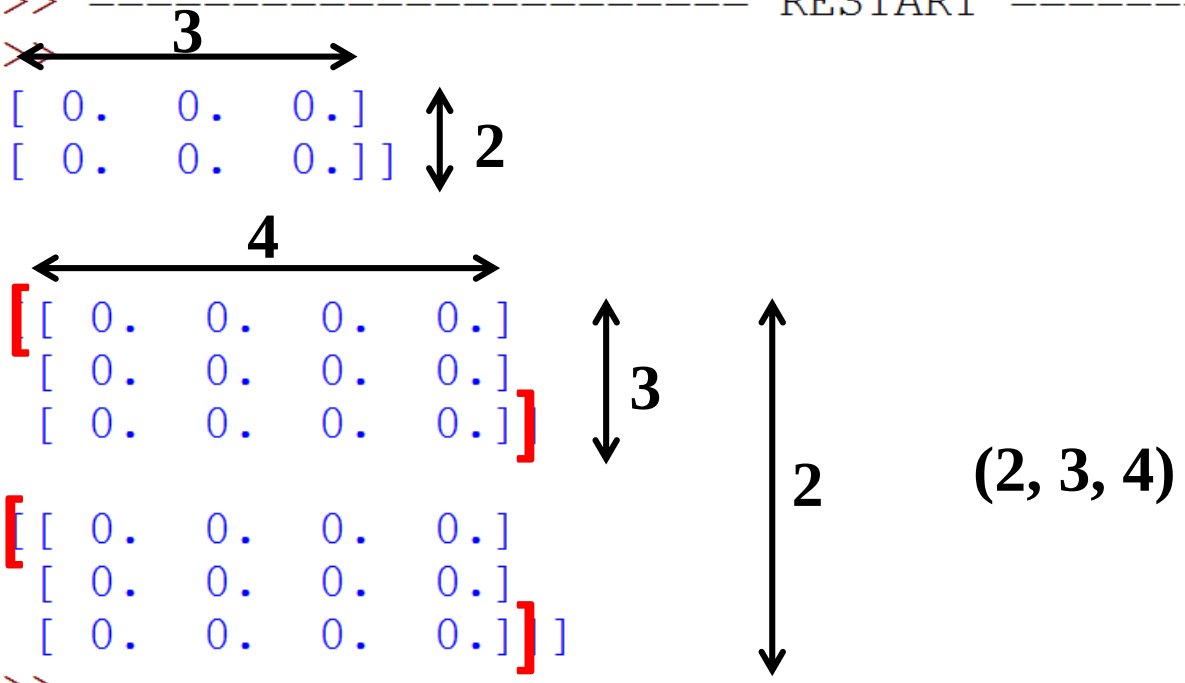
numpy6.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy6.py

File Edit Format Run Options Windows Help

```
import numpy as np

print np.zeros((2,3))
print '\n'
print np.zeros((2,3,4))
```

Ln: 7 Col: 0

```
>>> ========================== RESTART ==========================
>>>
[[ 0.  0.  0.]
 [ 0.  0.  0.]]

[[[ 0.  0.  0.  0.]
  [ 0.  0.  0.  0.]
  [ 0.  0.  0.  0.]]

 [[ 0.  0.  0.  0.]
  [ 0.  0.  0.  0.]
  [ 0.  0.  0.  0.]]]
>>>
```



Ln: 2 Col: 29

arange

numpy4a.py - C:/Users/Wysokie Energie/Desktop/VPython/lec_4/numpy4a.py

File Edit Format Run Options Windows Help

```
import numpy as np

a = np.arange(2, 10, 2)
print a

b = np.arange(0, 1, 0.1)
print b
```

Ln: 9 Col: 0

```
>>>
[2 4 6 8]
[ 0.   0.1  0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9]
>>>
```

Ln: 1 Col: 0

jak **range()** tylko lepsze

b można uzyskać poprzez
[i/10 for i in range(0,10)]

reshape

numpy4.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy4.py

File Edit Format Run Options Windows Help

```
import numpy as np

a = np.arange(0, 10, 1)
print a, '\n'

b = a.reshape(5, 2)
print b
```

5 rzędów, 2 kolumny

Ln: 9 Col: 0

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

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

Ln: 1 Col: 0

sum, min, max

numpy11.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy11.py

File Edit Format Run Options Windows Help

```
import numpy as np

a = np.array([1, 2, 3, 4, 5])

print np.sum(a)
print np.min(a)
print np.max(a)
```

szybsze niż sum() z Pythona

Ln: 9 Col: 0

```
>>>
15
1
5
>>>
```

Ln: 1 Col: 0

inny sposób

a.sum(), a.min(), a.max()

NumPy, liczby przypadkowe

numpy12b.py - C:\Users\Wysokie Energie\Desktop\VPython\lec_4\numpy12b.py (2.7.12)

File Edit Format Run Options Window Help

```
import numpy as np

print np.random.random(3)

print np.random.uniform(0, 10, 5)

print np.random.randint(0, 6, (2,21))
```

Ln: 9 Col: 0

```
>>>
[ 0.4189823   0.77655661  0.06715871]
[ 2.39741697  3.44915269  9.47332416  9.52902428  2.9812941 ]
[[0 2 2 1 2 1 5 3 4 0 5 3 1 2 5 5 5 0 3 2 1]
 [0 2 3 3 5 2 3 4 0 0 5 5 3 1 2 1 2 3 4 5 0]]
>>>
```

Ln: 1 Col: 0

3 liczby float z [0,1)

5 liczb float z przedziału [0,10)

array (2,21) liczb całkowitych z [0,6), bez 6

Proszę sprawdzić
https://docs.scipy.org/doc/numpy/reference/routines.random.html

Wiele generatorów liczb losowych