

Matematyczne aspekty analizy danych

semestr zimowy 2026/2027

Dr Anna Muranova
UWM w Olsztynie

Wykład 2

Rozdział 2. Podstawy matematyki oraz pakiet sympy

Instalowanie pakietów

Wpisz

```
%pip install sympy
```

SymPy jest biblioteką matematyczną (systemem algebry komputerowej lub komputerowym systemem obliczeń symbolicznych), która została napisana od podstaw w języku Python i używa go jako narzędzia interakcji (nie trzeba uczyć się nowego języka programowania, jak w przypadku większości innych systemów tego typu).

Co to jest matematyka symboliczna?

Python operuje na liczbach zmiennoprzecinkowych:

```
import math
a = math.sqrt(3)
print(a)
print(a**2)
```

W matematyce symbolicznej operujemy na liczbach i symbolach

```
import sympy
a = sympy.sqrt(3)
print(a)
print(a.evalf())
print(a.evalf(50))
print(a**2)
```

Jeszcze przykład

```
import math
import sympy

print(math.sin(math.pi))
print(math.pi)
print(sympy.sin(sympy.pi))
print(sympy.pi)

print(math.log(math.e))
print(math.e)
print(sympy.log(sympy.E))
print(sympy.E)
```

Uwaga: inny sposób podłączania pakietu

```
from sympy import *  
from math import *  
print(E)  
print(e)  
print(E==e)  
print(E-e)  
print(E.evalf()-e)  
print(E.evalf()==e)  
print(E.evalf())
```

```
print(pi)#uwaga!
```

Porównaj:

```
from math import *  
from sympy import *  
  
print(pi)
```

Sumowanie w Sympy

```
from sympy import *

i, n = symbols('i n')

# iterujemy po elementach i od 1 do n
# nastepne mnozimy i sumujemy
suma = Sum(2*i, (i, 1, n))

#ustawiamy n na 5
#iterujemy po liczbach od 1 do 5
suma_do_5 = suma.subs(n, 5)
print(suma_do_5)
print(suma_do_5.doit())
```

Porównaj

```
from sympy import *

print(sum(pow(i,0.5)**2 for i in range(0,11)))

i, n = symbols('i n')
suma = Sum(sqrt(i)**2, (i,1,n))

suma_do_10 = suma.subs(n,10)
print(suma_do_10)
print(suma_do_10.doit())
```

doit() vs evalf

```
from sympy import *

print(sum(pow(i,0.5) for i in range(0,11)))

i, n = symbols('i n')
suma = Sum(sqrt(i), (i,1,n))

suma_do_10 = suma.subs(n,10)
print(suma_do_10)
print(suma_do_10.doit())
print(suma_do_10.evalf())
```

Potęgowanie

$$x^n = \underbrace{x \cdot x \cdot \dots \cdot x}_{n \text{ razy}}$$

Przykład: $x^2 x^3 = x^{2+3} = 5$

$$x^{-n} = 1/x^n$$

Przykład: $x^2/x^5 = x^{2-5} = -3$

$$x^{\frac{p}{q}} = \sqrt[q]{x^p}, x > 0$$

Uwaga:

$$(-4)^{\frac{1}{2}} = \sqrt[2]{-4} = \pm 2i$$

$$(-4)^{\frac{1}{2}} = (-4)^{\frac{2}{4}} = 16^{\frac{1}{4}} = \sqrt[4]{16} = 2$$

Uwaga: $x^0 = 1$, ponieważ

$$1 = x^3/x^3 = x^{3-3} = x^0 \text{ dla } x \neq 0.$$

Zakładamy, że $0^0 = 1$ też.

Potęgi rzeczywiste: $8^\pi \approx 8^{\frac{31415926535}{10000000000}} \approx 687.2913$

Potęgowanie w sympy

```
x = symbols('x')  
wyrazenie1 = x**2/x**5  
print(wyrazenie1)
```

```
wyrazenie2 = x**(1/2)/x**(3/2)  
print(wyrazenie2)
```

```
wyrazenie3 = x**(Rational(1,2))/x**(Rational(3,2))  
print(wyrazenie3)
```

Potęgowanie w math

```
import math

print(math.pow(2,100))
print(pow(2,100))
print(2**100)

#print(math.pow(-2,0.5))
print(pow((-2),0.5))
print((-2)**0.5)
```

Logarytmy

$$a^x = b$$

$$x = \log_a b$$

Do jakiej potęgi podnieść a , żeby otrzymać b ?

Uwaga: $a, b > 0, a \neq 1$.

Przykład: $x = \log_2 8 \Rightarrow 2^x = 8 \Rightarrow x = 3$.

Właściwości:

Działanie	Właściwości potęgi	Właściwości logarytmy
Mnożenie	$c^x \cdot c^y = c^{x+y}$	$\log_c(ab) = \log_c(a) + \log_c(b)$
Dzielenie	$\frac{c^x}{c^y} = c^{x-y}$	$\log_c\left(\frac{a}{b}\right) = \log_c(a) - \log_c(b)$
Potęgowanie	$(c^x)^z = c^{xz}$	$\log_c(a^z) = z \log_c a$
Wykładnik zerowy	$c^0 = 1$	$\log_c 1 = 0$

Wskazówka: $x = \log_c a, y = \log_c b$

Logarytmy w math i w sympy

```
import math
import sympy
```

```
x = math.log(8,2)
print(x)
```

```
x = sympy.log(8,2)
print(x)
```

```
x = math.log(10,2)
print(x)
print(2**x)
```

```
x = sympy.log(10,2)
print(x)
print(x.evalf())
print(2**x)
print((2**x).evalf())
```

Liczba Eulera

Podstawa logarytmu naturalnego, liczba e , liczba Eulera – stała matematyczna wykorzystywana w wielu dziedzinach matematyki i fizyki.

W przybliżeniu wynosi

$$e \approx 2,718281828459$$

```
import math
import sympy

print(math.e)
print(sympy.E)
print(sympy.E.evalf())

print(math.log(math.e**2))
print(sympy.log(sympy.E**2))
```

Liczba Eulera

Podstawa logarytmu naturalnego, liczba e , liczba Eulera – stała matematyczna wykorzystywana w wielu dziedzinach matematyki i fizyki.

W przybliżeniu wynosi

$$e \approx 2,718281828459$$

```
import math
import sympy

print(math.e)
print(sympy.E)
print(sympy.E.evalf())

print(math.log(math.e**2))
print(sympy.log(sympy.E**2))
```

Definicja 1

Liczba e może być zdefiniowana na kilka równoważnych sposobów.

- ▶ Jako granica ciągu, e jest określana przez

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n.$$

```
from sympy import *  
  
n = symbols('n')  
print(Limit((1+1/n)**n, n, +oo).doit())  
  
print([(1+1/(10*n))**(10*n) for n in range(1,200)])
```

Definicja 2

Liczba e może być zdefiniowana na kilka równoważnych sposobów.

- ▶ Jako suma szeregu e jest określana przez

$$e = \sum_{k=0}^{\infty} \frac{1}{k!} = \lim_{n \rightarrow \infty} \sum_{k=0}^n \frac{1}{k!}$$

```
from sympy import *

k, n = symbols('k n')

suma = Sum(1/factorial(k), (k, 0, n))
print(suma)

print(Limit(suma, n, +oo).doit())

suma_do_10 = suma.subs(n, 10) #ustawiamy n na 10
print(suma_do_10)
print(suma_do_10.doit())
print(suma_do_10.evalf())
print(suma.subs(n, +oo).doit())
```

Definicja 3

- ▶ Za pomocą całki: liczbę e można także zdefiniować jako jedyną liczbę rzeczywistą taką że

$$\int_1^e \frac{1}{t} dt = 1$$

(to znaczy, że liczba e to taka, że pole powierzchni pod hiperbolą $f(t) = 1/t$ od 1 do e jest równe 1).

- ▶ Za pomocą funkcji: liczbę e można również zdefiniować jako taki argument funkcji

$$f(x) = x^{1/x}, \quad x > 0$$

dla którego jej wartość jest największa.

- ▶ Za pomocą stycznej: e jest podstawą takiej funkcji wykładniczej, że styczna do jej wykresu w punkcie $(0, 1)$ ma współczynnik kierunkowy równy 1.

Własności

- ▶ pochodna funkcji $(e^x)' = e^x$
- ▶ całka funkcji $\int e^x dx = e^x + C$, gdzie C jest dowolną stałą całkowania.
- ▶ Jest jednym z elementów wzoru Eulera (zwanego też „najpiękniejszym wzorem matematyki”), wiążącej e z innymi słynnymi liczbami: jednostką urojoną i , π , jednością i zerem:

$$e^{i\pi} = -1$$

(Definicja) e jest podstawą logarytmu naturalnego:

$$\ln x := \log_e x.$$

Funkcje

Funkcje – to wyrażenia, które definiują relacje między dwiema lub wieloma zmiennymi: mówiąc ściślej, funkcja przyjmuje *zmienny wejściowe*, umieszcza je w wyrażeniu i generuje *zmienny wyjściową*.

Na przykład,

$$y = 2x + 1 \text{ lub } f(x) = 2x + 1$$

x	y = f(x)
0	1
0.5	2
1.25	3.5
3	7

```
def f(x):  
    return 2 * x + 1
```

```
x_values = [0, 0.5, 1.25, 3]  
for x in x_values:  
    print(f(x))
```

Definiowanie funkcje matematycznych w sympy

```
from sympy import *
```

```
x = symbols('x')
```

```
f = 2*x+1
```

```
print(f.subs(x, 0.5))
```

```
print([f.subs(x, i) for i in [0, 0.5, 1.25, 3]])
```

Oprócz tego pozwala na działania na funkcjach:

```
from sympy import *
```

```
x = symbols('x')
```

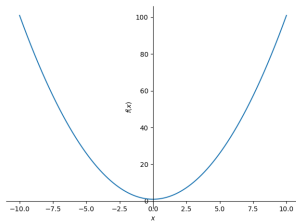
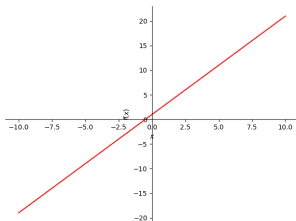
```
f = x**2-1
```

```
print(f.factor())
```

```
print(diff(f,x))
```

```
print(Limit((1+1/x)**x, x, +oo).doit())
```

Wykresy



Kody:

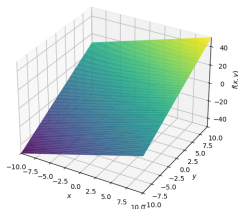
```
from sympy import *  
x = symbols('x')
```

```
f = 2*x+1  
plot(f, line_color='red')
```

```
plot (x**2 +1)
```

Uwaga: możliwe, że trzeba zainstalować matplotlib

Wykresy 3d



Kod:

```
from sympy import *  
from sympy.plotting import plot3d
```

```
x, y = symbols('x y')
```

```
f = 2*x + 3*y  
plot3d(f)
```

Uwaga: możliwe, że trzeba zainstalować matplotlib

Podstawowe funkcje elementarne

Podstawowe funkcje elementarne obejmują funkcje:

- ▶ stałe $f(x) = \text{const}$,
- ▶ potęgowe $f(x) = x^r$,
- ▶ wykładnicze $f(x) = a^x$,
- ▶ logarytmiczne $f(x) = \log_a x$
- ▶ trygonometryczne $f(x) = \sin(x)$ i td,
- ▶ cyklometryczne (funkcje odwrotne do funkcji trygonometrycznych)
 $f(x) = \arccos(x)$ td

Rodzaje funkcje elementarnych

Funkcje, które można otrzymać z podstawowych funkcji elementarnych za pomocą skończonej ilości działań arytmetycznych oraz operacji złożenia funkcji nazywamy **funkcjami elementarnymi**.

- ▶ wielomianowa: $f(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0$
 - ▶ liniowa $f(x) = ax + b, a \neq 0$
 - ▶ kwadratowa $f(x) = ax^2 + bx + c, a \neq 0$
- ▶ wymierna: $f(x) = \frac{a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0}{b_m x^m + b_{m-1} x^{m-1} + b_{m-2} x^{m-2} + \dots + b_1 x + b_0}$
- ▶ hiperboliczne, nprz: $\sinh x = \frac{e^x - e^{-x}}{2}, \cosh x = \frac{e^x + e^{-x}}{2}$
https://pl.wikipedia.org/wiki/Funkcje_hiperboliczne
- ▶ polowe (odwrotne do hiperbolicznych), nprz. $\operatorname{arcsinh} = \ln(x + \sqrt{x^2 + 1})$
https://pl.wikipedia.org/wiki/Funkcje_hiperboliczne_odwrotne
- ▶ funkcja Gaussa

$$f_{\mu, \sigma}(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(x - \mu)^2}{2\sigma^2}\right).$$

Funkcje nieelementarne

- ▶ funkcja znaku (signum)
- ▶ podłoga
- ▶ sufit,
- ▶ część ułamkowa (mantysa)
- ▶ funkcja Dirichleta

$$1_{\mathbb{Q}}(x) = \begin{cases} 1 & \text{dla } x \in \mathbb{Q}, \\ 0 & \text{dla } x \notin \mathbb{Q}. \end{cases}$$

- ▶ Funkcja Γ (zwana też funkcją gamma Eulera) – funkcja specjalna, która rozszerza pojęcie silni na zbiór liczb rzeczywistych i zespolonych. Jeżeli x część rzeczywista liczby zespolonej $z = x + iy$ jest dodatnia, to

$$\Gamma(z) = \int_0^{+\infty} t^{z-1} e^{-t} dt, \quad x > 0$$

Funkcja wykładnicza e^x

Szczególnym przypadkiem funkcji wykładniczej jest ta o podstawie równej e – podstawie logarytmu naturalnego. Innym oznaczeniem takiej funkcji jest $\exp(x)$ nazywane krótko eksponensem.

Inna definicja eksponensa:

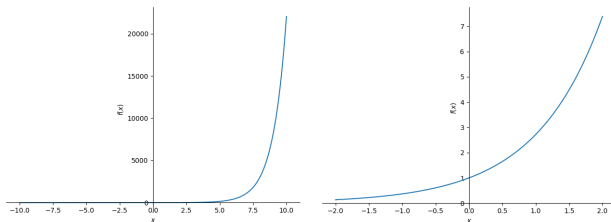
$$e^x = \exp(x) = \sum_{k=0}^{\infty} \frac{x^k}{k!}.$$

```
import math
```

```
print(math.exp(1e-5) - 1) # gives result accurate to 11 places
print(math.expm1(1e-5) )
```

Oprócz tego `math.exp(x)` dokładniej niż `math.e ** x` oraz `pow(math.e, x)`.

Funkcja wykładnicza e^x w sympy



Kody:

```
from sympy import *
```

```
x = symbols('x')
```

```
f = exp(x)
```

```
plot(f)
```

```
plot(f, (x, -2, 2))
```

Granica funkcji – wartość, do której obrazy danej funkcji zbliżają się nieograniczenie dla argumentów dostatecznie bliskich wybranemu punktowi. Na przykład,

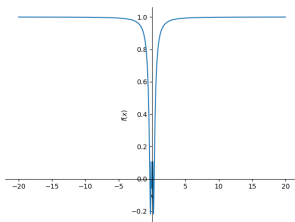
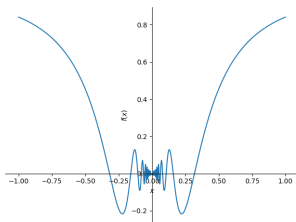
$$\lim_{x \rightarrow \infty} \frac{1}{x} = 0$$

$$\lim_{x \rightarrow \infty} \left(1 + \frac{1}{x}\right)^x = e$$

$$\lim_{x \rightarrow \infty} x \sin\left(\frac{1}{x}\right) = 1$$

$$\lim_{x \rightarrow 0} x \sin\left(\frac{1}{x}\right) = 0$$

Granice w sympy



Kody:

```
from sympy import *
```

```
x = symbols('x')
```

```
f = x*sin(1/x)
```

```
plot(f, (x, -1, 1))
```

```
plot(f, (x, -20, 20))
```

```
print(Limit(f, x, 0).doit())
```

```
print(Limit(f, x, +oo).doit())
```

Pochodne

Pochodne są bardzo ważne w analizie danych. Często używa się ich w uczeniu maszynowym i innych algorytmach matematycznych, zwłaszcza w metodzie gradientu prostego. Kiedy nachylenie jest równe 0, to oznacza, że osiągnęliśmy minimum lub maksimum zmiennej wyjściowej. Koncepcja ta jest przydatna w regresji liniowej, regresji logistycznej i sieciach neuronowych.

Pochodna funkcji – nieformalnie: miara szybkości funkcji, czyli tempa zmian jej wartości względem zmian jej argumentów.

$$\lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Obliczenie pochodnej, korzystając z definicji 1

Oblicz pochodną funkcji $f(x) = x^2$ w punkcie $x_0 = 2$, korzystając z definicji.

Rozwiązanie:

Liczymy wartość pochodnej w punkcie x_0 , korzystając z definicji:

$$\begin{aligned} f'(2) &= \lim_{h \rightarrow 0} \frac{f(2+h) - f(2)}{h} = \lim_{h \rightarrow 0} \frac{(2+h)^2 - 2^2}{h} \\ &= \lim_{h \rightarrow 0} \frac{4 + 4h + h^2 - 4}{h} = \lim_{h \rightarrow 0} \frac{4h + h^2}{h} = \lim_{h \rightarrow 0} (4 + h) = 4 \end{aligned}$$

Obliczenie pochodnej, korzystając z definicji 2

Możemy również policzyć z definicji wzór ogólny pochodnej dla tej funkcji ($f(x) = x^2$):

$$\begin{aligned} f'(x) &= \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} = \lim_{h \rightarrow 0} \frac{(x+h)^2 - x^2}{h} \\ &= \lim_{h \rightarrow 0} \frac{x^2 + 2xh + h^2 - x^2}{h} = \lim_{h \rightarrow 0} \frac{2xh + h^2}{h} = \lim_{h \rightarrow 0} (2x + h) = 2x \end{aligned}$$

Czyli ostatecznie:

$$f'(x) = 2x$$

Można też napisać równoważnie:

$$(x^2)' = 2x$$

Korzystając z tak wyliczonego wzoru, możemy teraz obliczyć wartość pochodnej w dowolnym punkcie, np.:

$$f'(2) = 2 \cdot 2 = 4, f'(0) = 2 \cdot 0 = 0, f'(-5) = 2 \cdot (-5) = -10$$

Praktycznie zawsze opłaca się najpierw policzyć pochodną funkcji (zwłaszcza, że mamy do dyspozycji gotowe wzory na liczenie pochodnych), a dopiero potem wyznaczyć jej wartość w konkretnym punkcie.

Kalkulator pochodnych napisany w Pythonie

```
def pochodna_x(f, x, rozmiar_kroku):  
    m = (f(x + rozmiar_kroku) - f(x)) / rozmiar_kroku  
    return m  
  
def moja_funkcja(x):  
    return x**2  
  
nachylenie_w_punkcie_2 = pochodna_x (moja_funkcja, 2, 0.00001)  
print(nachylenie_w_punkcie_2)
```

Wynik: 4.000010000027032

Tabela pochodnych ważniejszych funkcji elementarnych

Nazwa funkcji	Wzór funkcji i wzór jej pochodnej
funkcja stała	$(c)' = 0$
funkcja potęgowa, $\alpha \neq 0$	$(x^\alpha)' = \alpha x^{\alpha-1}$
pierwiastek kwadratowy, tzn. $\alpha = \frac{1}{2}$	$(\sqrt{x})' = \frac{1}{2\sqrt{x}}$
funkcja wykładnicza, $a > 0, a \neq 1$	$(a^x)' = a^x \ln a$
funkcja wykładnicza o podstawie e	$(e^x)' = e^x$
funkcja logarytmiczna, $a > 0, a \neq 1$	$(\log_a x)' = \frac{1}{x \ln a}$
logarytm naturalny, tzn. $a = e$	$(\ln x)' = \frac{1}{x}$
sinus	$(\sin x)' = \cos x$
cosinus	$(\cos x)' = -\sin x$
tangens	$(\tan x)' = \frac{1}{\cos^2 x} = 1 + \tan^2 x$
cotangens	$(\cot x)' = -\frac{1}{\sin^2 x} = -(1 + \cot^2 x)$
arkus sinus	$(\arcsin x)' = \frac{1}{\sqrt{1-x^2}}$
arkus cosinus	$(\arccos x)' = -\frac{1}{\sqrt{1-x^2}}$
arkus tangens	$(\arctan x)' = \frac{1}{1+x^2}$
arkus cotangens	$(\operatorname{arccot} x)' = -\frac{1}{1+x^2}$
sinus hiperboliczny	$(\sinh x)' = \cosh x$
cosinus hiperboliczny	$(\cosh x)' = \sinh x$

Ważniejsze reguły obliczenia pochodnych

$$(f(x) + g(x))' = f'(x) + g'(x)$$

$$(f(x) - g(x))' = f'(x) - g'(x)$$

$$(f(x)g(x))' = f'(x)g(x) + f(x)g'(x)$$

$$\left(\frac{f(x)}{g(x)}\right)' = \frac{f'(x)g(x) - f(x)g'(x)}{g^2(x)}$$

$$(f(g(x)))' = f'(g(x))g'(x)$$