

Matematyka dla informatyki stosowanej i systemów pomiarowych

dr Remigiusz Durka

Instytut Fizyki Teoretycznej

*Zakład Teorii Grawitacji i Oddziaływań
Fundamentalnych*



Uniwersytet
Wrocławski

Wrocław

200 lat pod niemieckim,
400 lat pod polskim,
500 lat pod czeskim panowaniem



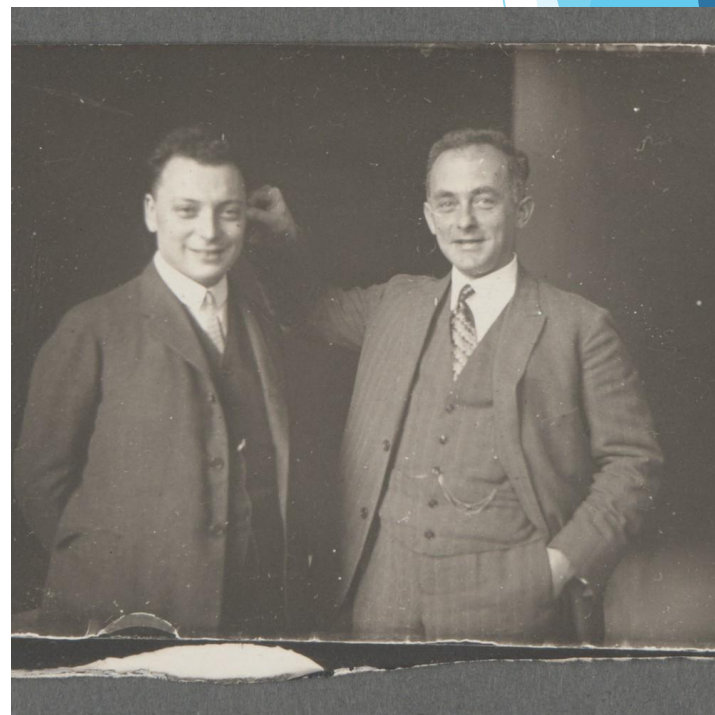
- W czasach antycznych w miejscu lub okolicach istniała miejscowość o nazwie **Budorgis**.
- Została ona odwzorowana na antycznej **mapie Ptolemeusza** z lat 142-147 n.e.
- W roku 985 na Ostrowie Tumskim powstał pierwszy gród wybudowany przez **Mieszka I**.
- W średniowiecznej Kronice Polskiej Galla Anonima spisanej w latach 1112-1116 Wrocław obok Krakowa oraz Sandomierza zaliczony został do **jednej z trzech głównych stolic Królestwa Polskiego**.
- **Mikołaj Kopernik** od 1503 r. do 1538 r. był kanonikiem kapituły wrocławskiej.

Wrocławscy nobliści

- Stąd pochodzi (czyli urodziło się, pobierało nauki albo pracowało) aż **jedenastu** laureatów Nagrody Nobla!
- Dwie nagrody literackie, cztery z fizyki, 3 z chemii oraz po jednej z medycyny i ekonomii

FIZYKA:

- **Phillip Lénárd** (1862-1947)
- **Otto Stern** (1888-1969)
- **Erwin Schrödinger** (1887-1961)
- **Max Born** (1882-1970)



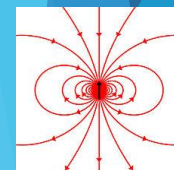
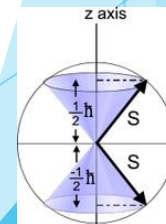
Wrocławscy nobliści

$$\frac{1}{\sqrt{2}}|\uparrow\rangle + \frac{1}{\sqrt{2}}|\downarrow\rangle$$

- **Erwin Schrödinger** (tak, ten od jednocześnie żywego i martwego kota i jeszcze bardziej **sławnego równania** będącego fundamentem fizyki kwantowej) był wykładowcą w 1921 roku.

$$[\hat{x}, \hat{p}_x] = i\hbar$$

- Tutaj studentem fizyki a później asystentem był **Maks Born**, któremu zawdzięczamy interpretację **modułu funkcji falowej** jako gęstości prawdopodobieństwa znalezienia cząstki (z czym wiążą się te wszystkie orbitale, o których uczyliście się na chemii w szkole średniej), a **także komutator operatorów położenia i pędu** oraz stworzenie nazwy "**mechanika kwantowa**"!
- Tu też wychowywał się, studiował i zrobił doktorat **Otto Stern**, który później wykazał w eksperymencie skwantowanie wewnętrznego momentu pędu, czyli **spinu**.



Wrocław

- ▶ Swoje centra mają tu Google, Nokia, IBM, Siemens, Opera, Atos, Dolby,
- ▶ Działają też światowi liderzy z branż takich jak motoryzacja, elektronika i AGD, inżynieria mechaniczna, chemia i farmaceutyka. To firmy, które mają mocno rozbudowane działy IT. Mocny jest też sektor finansowy.
- ▶ Ale w mieście intensywnie rozwijają się też rodzime firmy IT, podbijające europejskie i światowe rynki i wiele start-upów.
- ▶ Wiele spółek realizuje model „born global”, co oznacza, że od pierwszego dnia istnienia są w stanie działać globalnie i pracować dla międzynarodowych klientów.

Globalne korporacje z centrami w Wrocławiu:

1. Google – posiada centrum badawczo-rozwojowe.
2. Nokia – specjalizuje się w telekomunikacji i technologiach sieciowych.
3. IBM – dostarcza rozwiązania IT oraz usługi technologiczne.
4. Siemens – działa w obszarze automatyzacji przemysłu i energii.
5. Opera Software – rozwija przeglądarkę internetową Opera.
6. Atos – globalna firma IT i doradcza.
7. Dolby – rozwija technologie dźwięku i obrazu.
8. 3M – amerykański konglomerat, który produkuje szeroką gamę produktów.
9. Capgemini – międzynarodowa firma świadcząca usługi IT i konsultingowe.
10. Credit Suisse – duża szwajcarska firma świadcząca usługi finansowe.

Znane polskie firmy i start-upy z Wrocławia:

1. Techland – jeden z największych polskich producentów gier komputerowych, znany z takich tytułów jak *Dying Light* i *Call of Juarez*.
2. LiveChat Software – firma dostarczająca oprogramowanie do obsługi klienta i komunikacji online.
3. PWR Racing Team – zespół inżynierski z Politechniki Wrocławskiej, który projektuje i buduje bolidy wyścigowe, uczestniczący w międzynarodowych zawodach *Formula Student*.
4. Objectivity – firma IT zajmująca się tworzeniem oprogramowania na zamówienie.
5. Synerise – start-up specjalizujący się w sztucznej inteligencji i analizie danych.
6. Ten Square Games – producent gier mobilnych, specjalizujący się w grach typu free-to-play.
7. Tooploox – firma technologiczna, zajmująca się m.in. sztuczną inteligencją, uczeniem maszynowym oraz produktami cyfrowymi.
8. Unit4 Polska – firma specjalizująca się w oprogramowaniu ERP dla sektora publicznego i edukacji.

Zakres zajęć

- ▶ Realizacja w ciągu zaledwie dwóch semestrów w wymiarze 75 godzin na semestr, z czego
- ▶ 45 godzin (3 tygodniowo) zarezerwowano na zajęcia w pracowni komputerowej
- ▶ 30 godzin (2 tygodniowo) przeznaczone są na wykłady

Zaliczenie

- ▶ Ćwiczenia/Laboratorium: 2 kolokwia
- ▶ Wykład: egzamin pisemny

Kurs dostępny online

<http://users.ift.uni.wroc.pl/~zkoza/matematyka/>

- ▶ autor: prof. Zbigniew Koza

<http://ift.uni.wroc.pl/~rdurka/matissp/>

- ▶ autor: dr Remigiusz Durka

Zagadnienia

- ▶ Narzędzia informatyczne do obliczeń inżynierskich (numerycznych i symbolicznych)
- ▶ Granice funkcji, ciągi, szeregi
- ▶ Funkcje i ich wykresy
- ▶ Narzędzia do wizualizacji funkcji/danych
- ▶ Pochodne i całki funkcji jednej i dwóch zmiennych
- ▶ Równania różniczkowe zwyczajne
- ▶ Równania różniczkowe cząstkowe

- ▶ Liczby zespolone
- ▶ Wektory i macierze
- ▶ Układy równań liniowych
- ▶ Zagadnienie na wartości i wektory własne
- ▶ Równania nieliniowe i algebraiczne
- ▶ Algebra

Idea stojąca za kursem

- ▶ Podstawową ideą leżącą u podstaw jego konstrukcji jest przekonanie, że licencjacki kurs matematyki powinien dać studentom **ogólne rozeznanie** w matematyce wyższej traktowanej jako ***narzędzie*** do rozwiązywania konkretnych problemów a sposób nauczania powinien nadążać za najnowszymi osiągnięciami techniki.
- ▶ Kurs oparty jest na **narzędziach informatycznych**:
(*Octave, Gnuplot, Wolfram Alpha*)

Idea stojąca za kursem

- ▶ Kurs **ten nie ma na celu** kształcenia zawodowych matematyków i w bardzo niewielkim stopniu porusza zagadnienie dla matematyków kluczowe: dowodzenie twierdzeń w oparciu o rygorystyczny formalizm.
- ▶ Nauka matematyki **jako narzędzia** pozwala szybko pokazać jej użyteczność bez wrzucania studentów na zbyt dla nich głębokie wody matematycznej abstrakcji...
- ▶ Wadą takiego podejścia jest oczywiście **groźba powierzchownego potraktowania** matematyki, bo siłą matematyki jest jej zdolność do uogólnień, które z kolei wymagają pewnego poziomu abstrakcji.

Jednoosobowa armia do rozwiązywania problemów



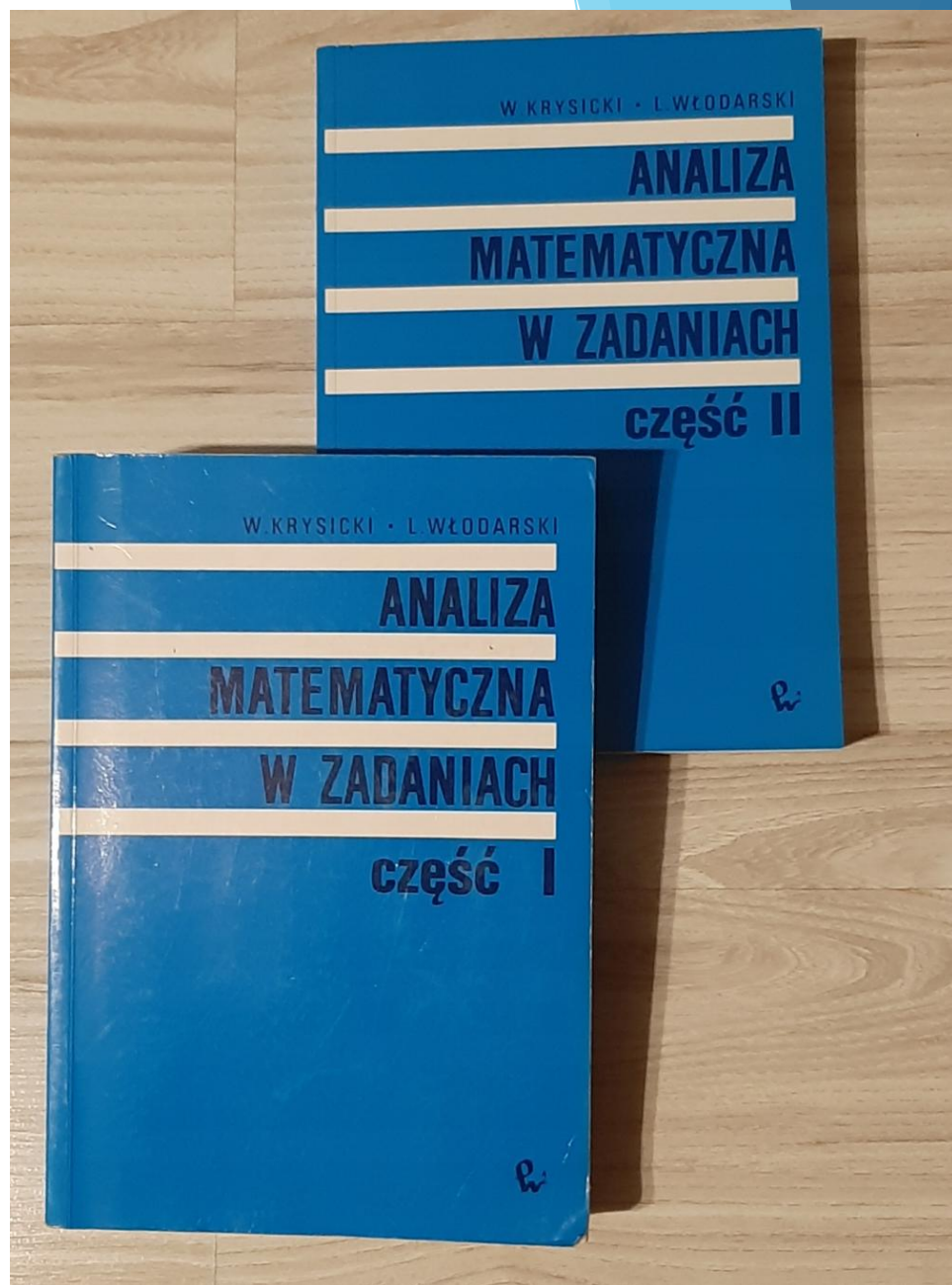
(Math)gyver

Analiza matematyczna w zadaniach

Krysicki, Włodarski

tom 1

tom 2



EVERYTHING

I SAY

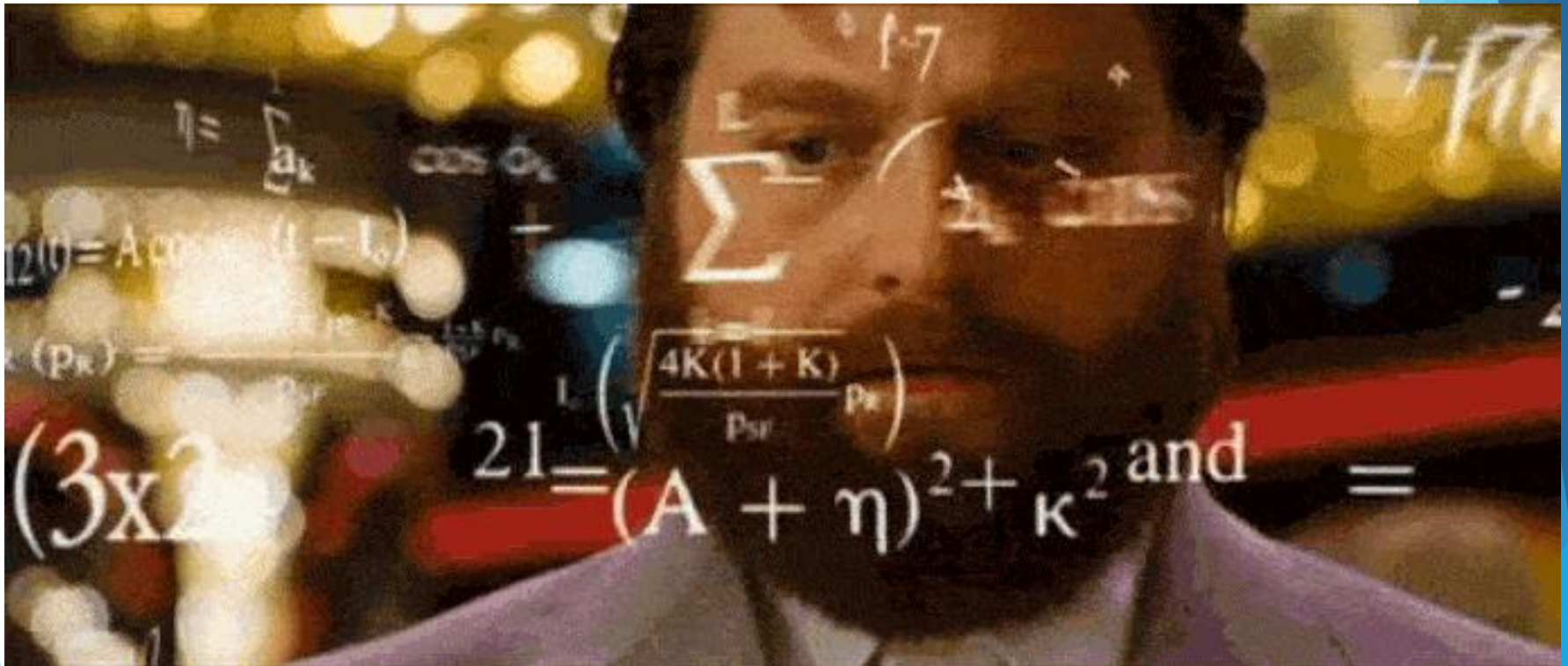
WILL BE ON

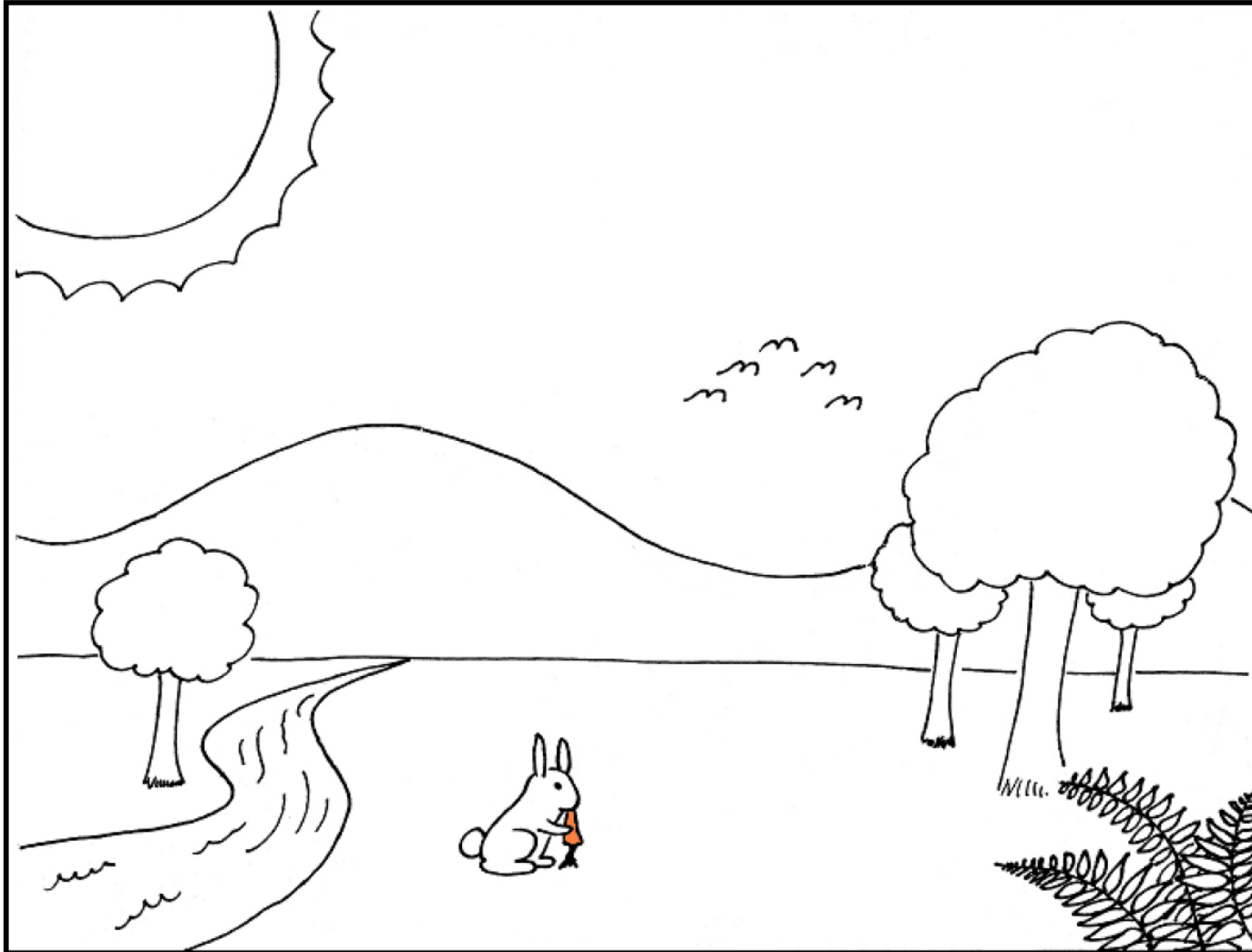
THE EXAM



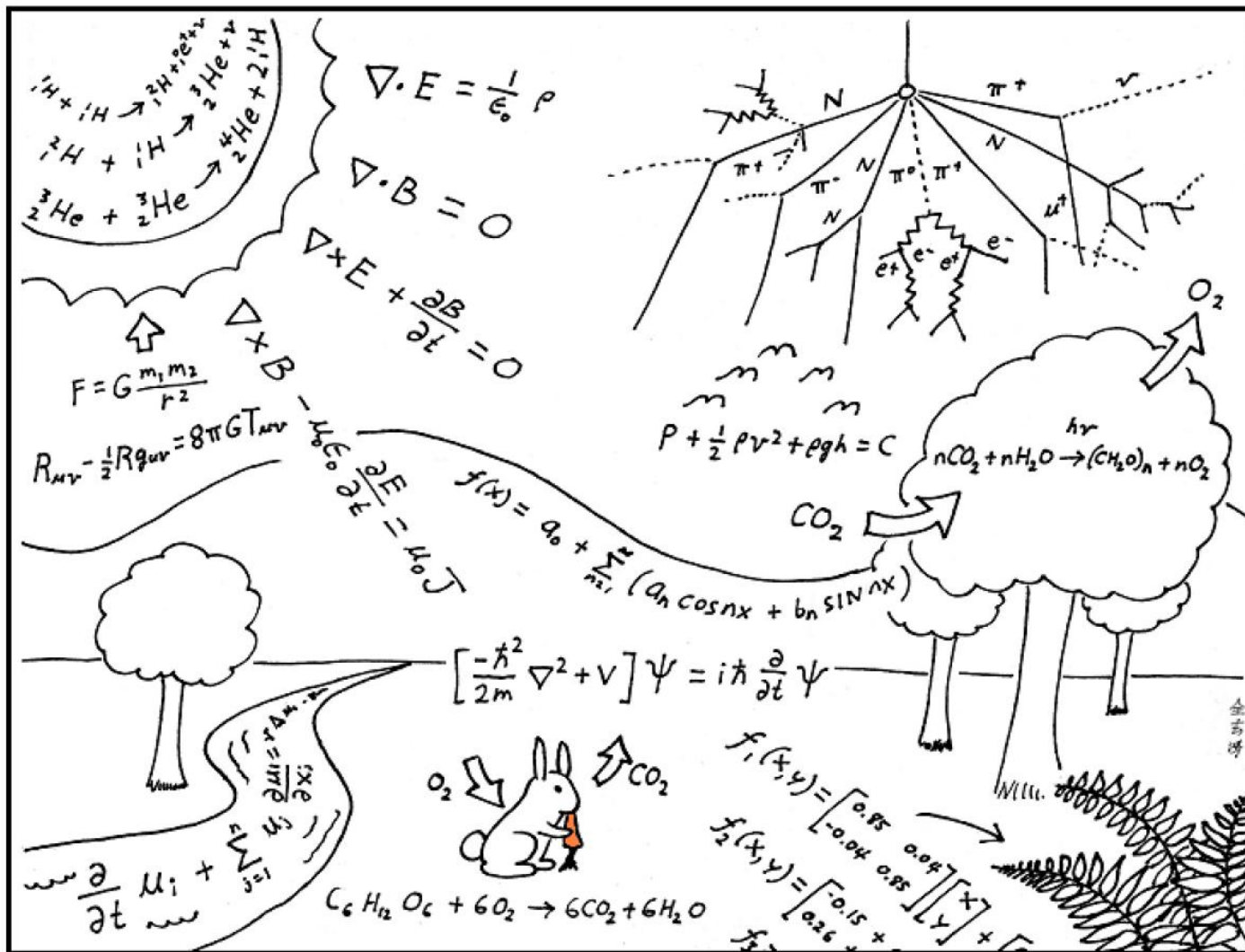


- Matematyka nie sprowadza się do “dowodzenia twierdzeń” lub “realizacji obliczeń na kalkulatorze”



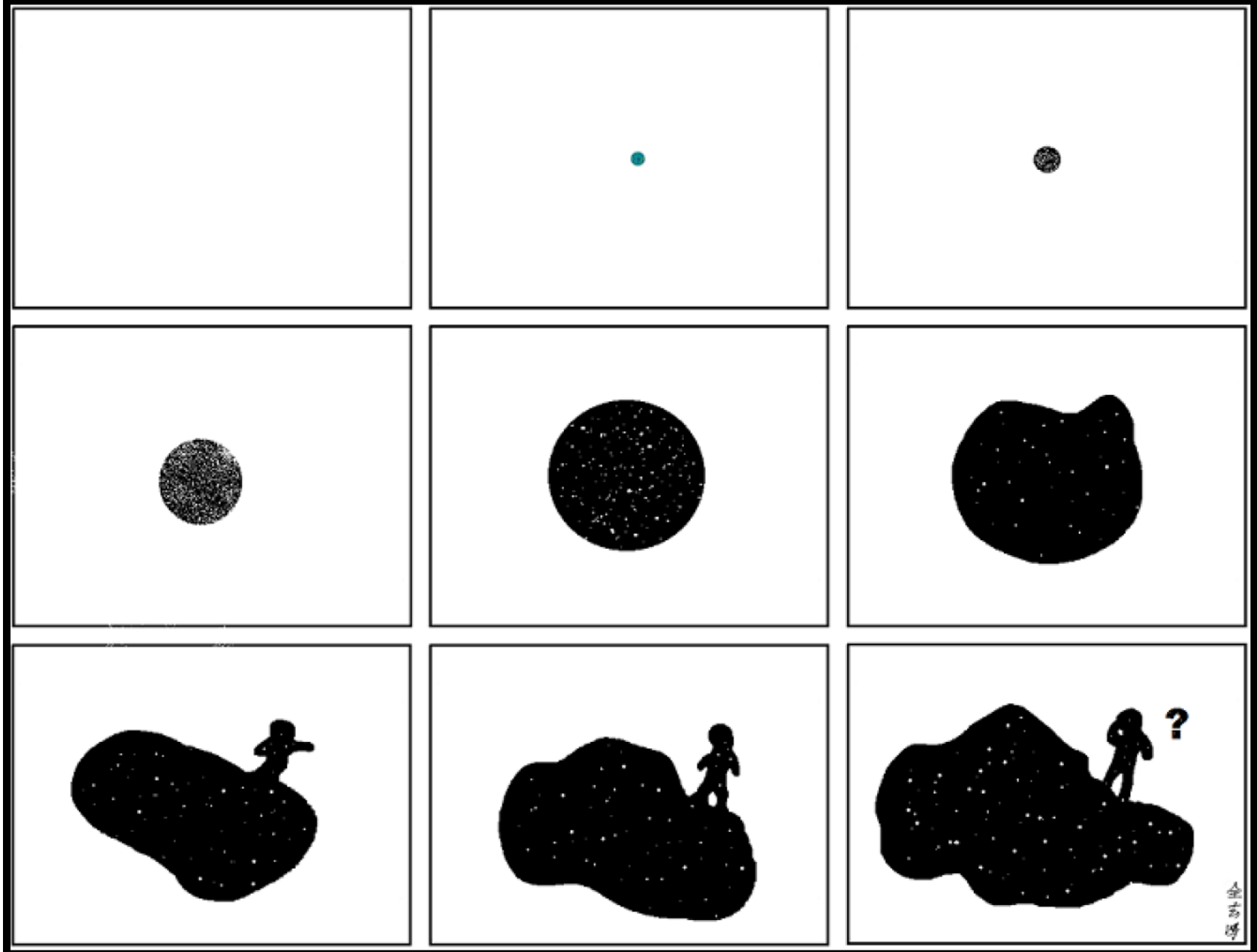


Abstroose Goose

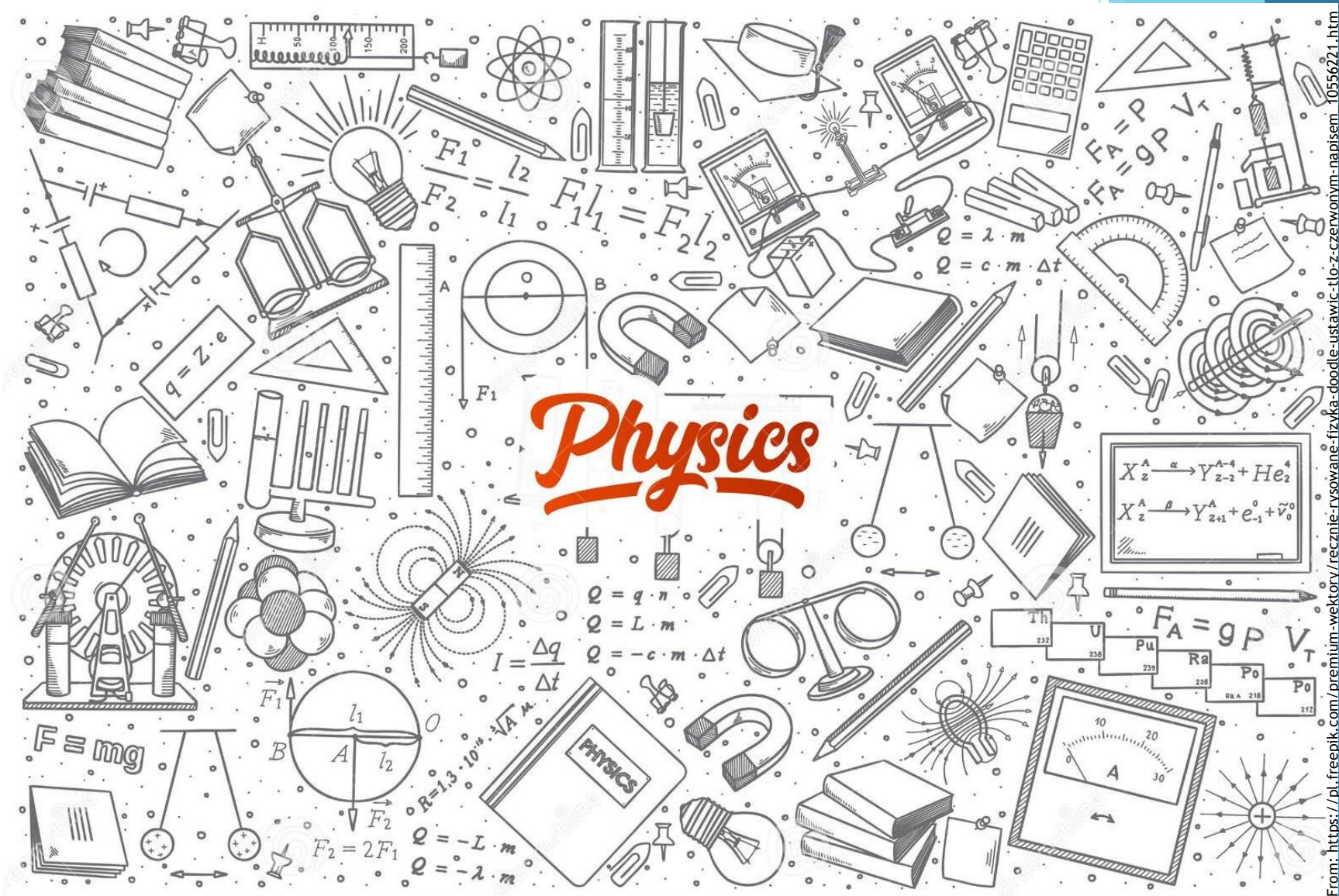


This is how scientists see the world.

Wszechświat starający się sam siebie zrozumieć

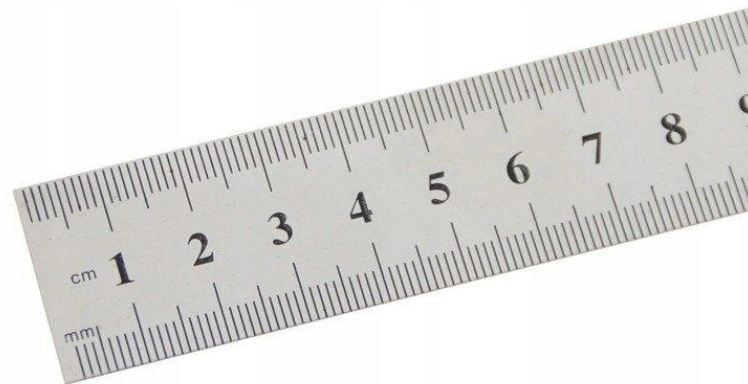


Eksploracja **jak rzeczy działają** i **dla czego!**



Od skalarów do tensorów

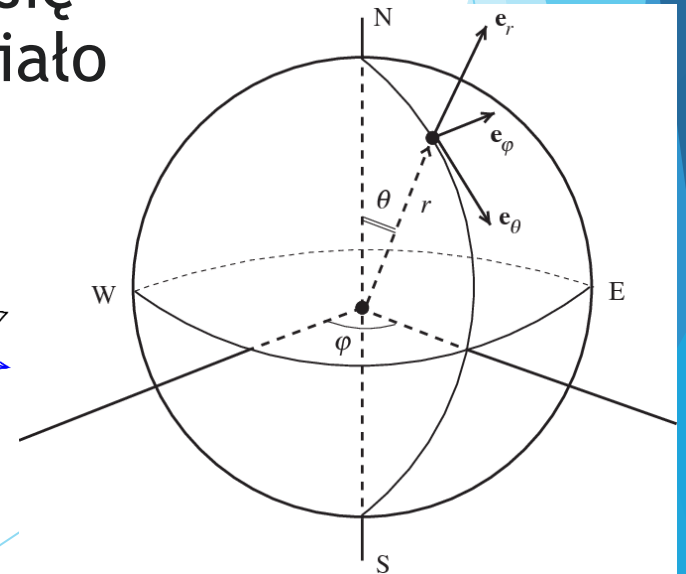
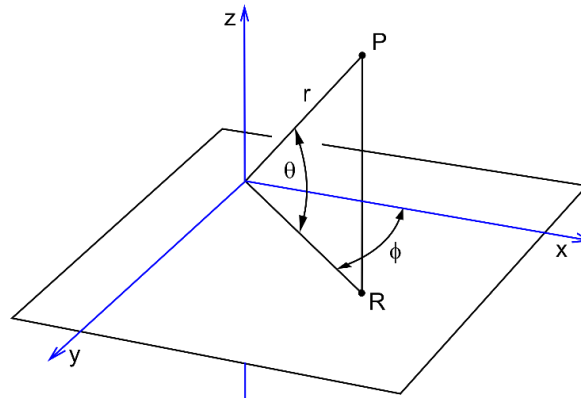
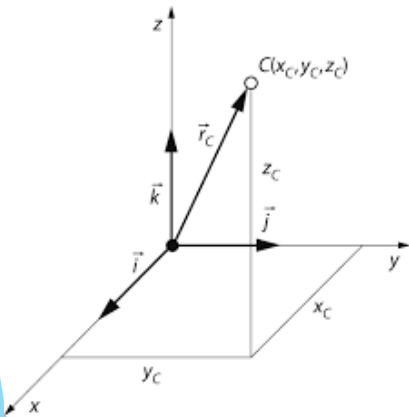
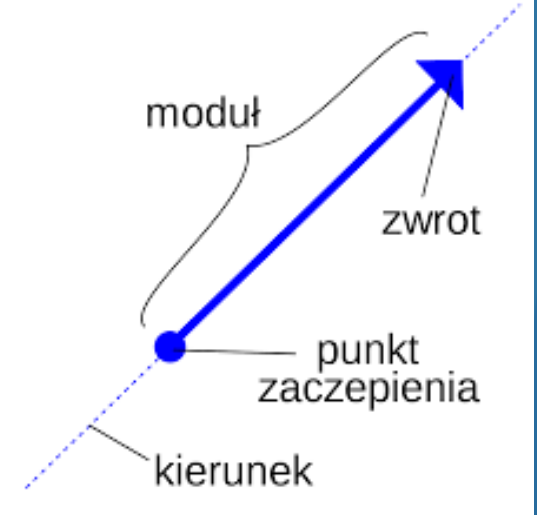
- ▶ **Skalar** - wielkość, do której określenia wystarczy jedna liczba rzeczywista (wraz z wymiarem wielkości fizycznej lub bezwymiarowa)
- ▶ masa M ,
- ▶ długość L ,
- ▶ pole powierzchni A ,
- ▶ objętość V ,
- ▶ temperatura T ,
- ▶ gęstość,
- ▶ praca W ,
- ▶ potencjał pola elektrostatycznego lub grawitacyjnego.



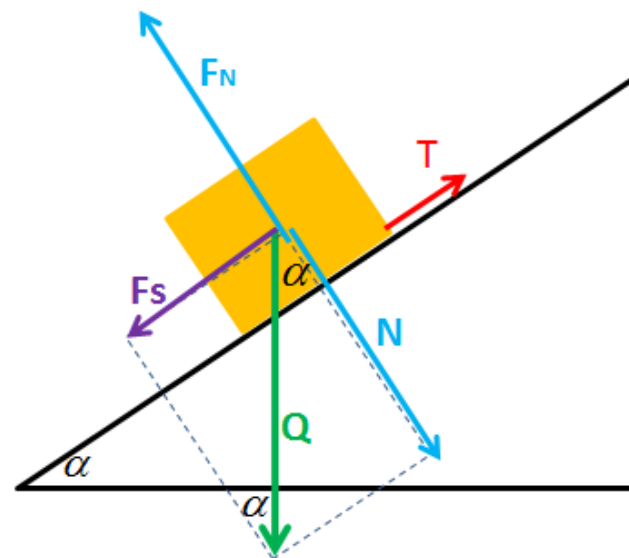
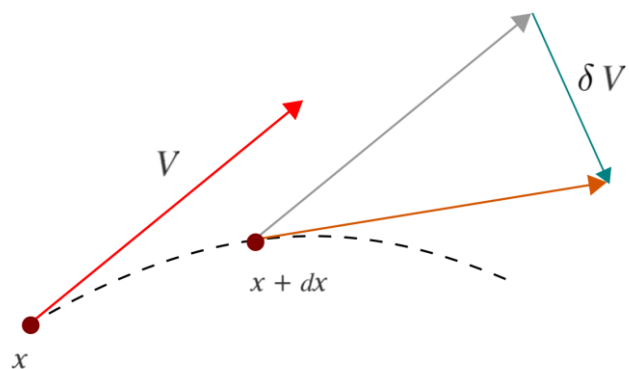
$$\phi = -G \frac{M}{r}$$

Od skalarów do tensorów

- ▶ **Wektor** jest to obiekt matematyczny opisywany za pomocą modułu (długością) oraz kierunku wraz ze zwrotem.
- ▶ Wektory odgrywają ważną rolę w **fizyce** opisując m.in. położenie, **prędkość** i **przyspieszenie** poruszającego się obiektu oraz **siły** działająca na ciało



Od skalarów do tensorów



Pole wektorowe

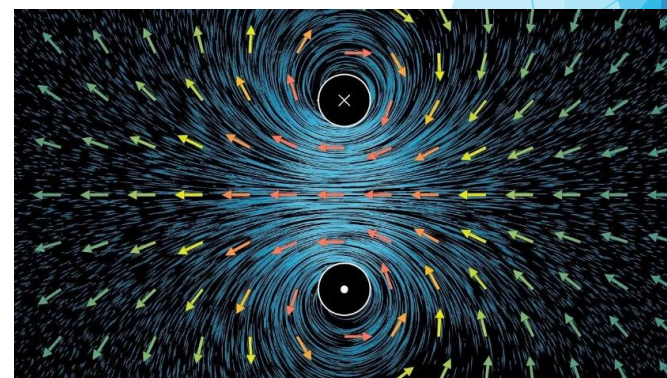
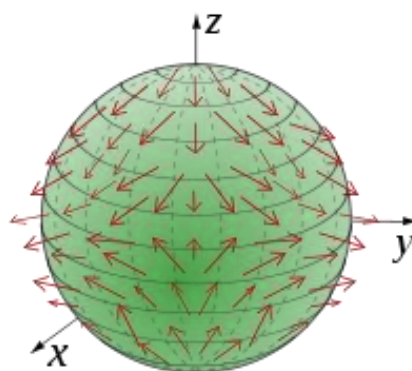
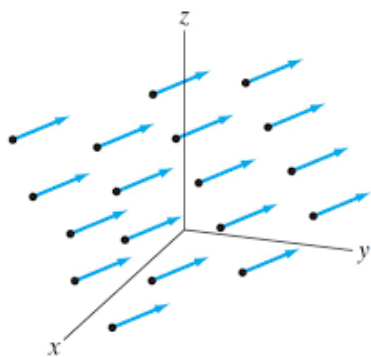


Figure 3.40 The constant vector field $F = a$.

Równania Maxwella

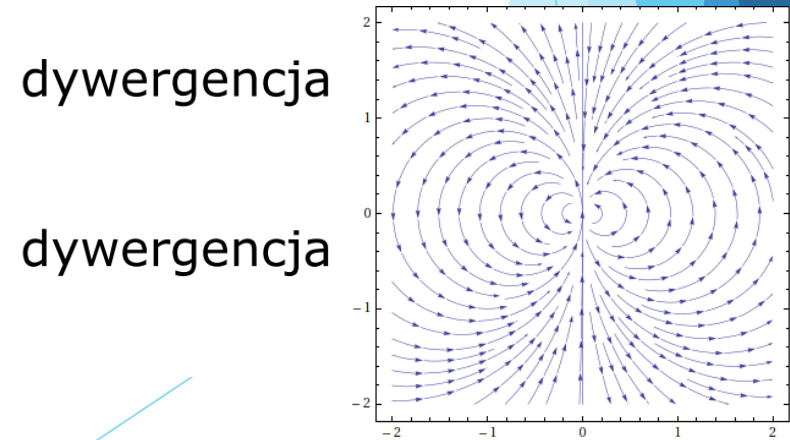
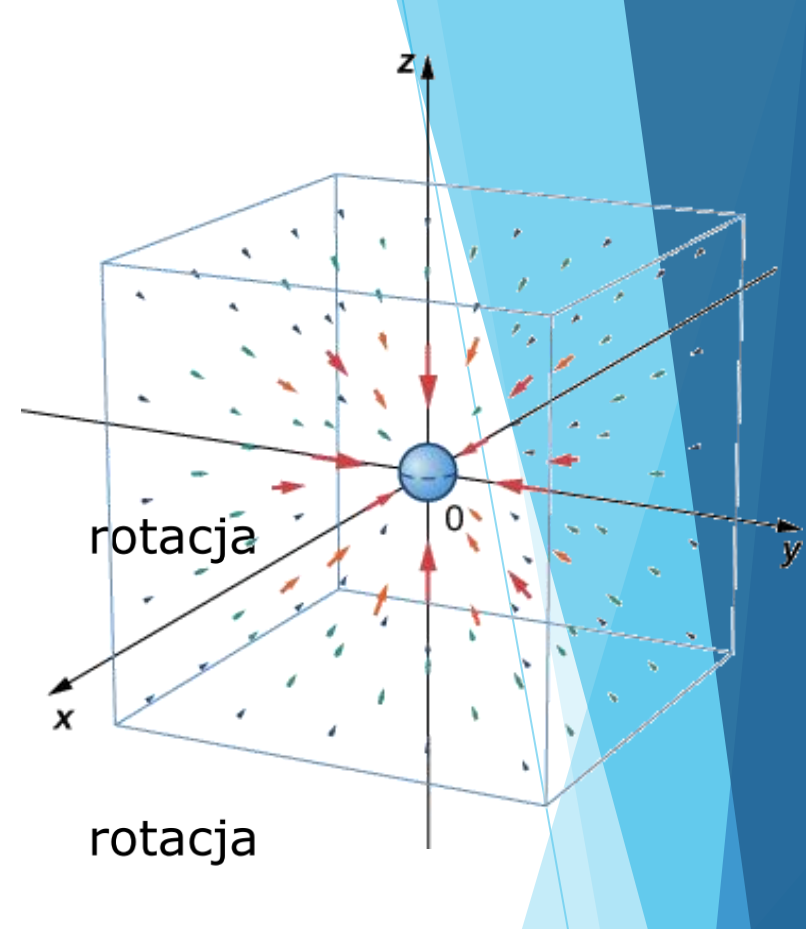
$$\vec{\nabla} \times \vec{E} = -\frac{\partial \vec{B}}{\partial t}$$

$$\vec{\nabla} \times \vec{B} = \mu\epsilon\vec{j} + \frac{\partial \vec{E}}{\partial t}$$

$$\epsilon\vec{\nabla} \cdot \vec{E} = \rho$$

$$\vec{\nabla} \cdot \vec{B} = 0$$

Elektryczność + Magnetyzm



Od skalarów do tensorów

Macierz/Tablica

Kolumna/Wiersz

Bloki i hiperbloki

Scalar Vector Matrix Tensor

1

$\begin{bmatrix} 1 \\ 2 \end{bmatrix}$

$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$

$\begin{bmatrix} \begin{bmatrix} 1 & 2 \end{bmatrix} & \begin{bmatrix} 3 & 2 \end{bmatrix} \\ \begin{bmatrix} 1 & 7 \end{bmatrix} & \begin{bmatrix} 5 & 4 \end{bmatrix} \end{bmatrix}$

▪ Tensors are n dimensional arrays

- Scalar is 0 D tensor
- Vector is 1 D tensor
- Matrix is 2 D tensor

1.5
Scalar

1.5
1.1
1.3
Vector

1.5, 1.6, 1.7
1.1, 1.2, 1.3
1.3, 1.6, 1.7
Matrix

$T = \begin{bmatrix} X_{111} & X_{112} & X_{113} & \dots & X_{11N} \\ X_{211} & X_{212} & X_{213} & \dots & X_{21N} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{M11} & X_{M12} & X_{M13} & \dots & X_{M1N} \end{bmatrix}$

Tensor

rank	object
0	scalar
1	vector
2	$N \times N$ matrix
≥ 3	tensor



Od równania na jedną zmienną do równań różniczkowych

► $x-1=2 \Rightarrow x=3$ (szukamy LICZBY)

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 1 \\ x_1 - x_2 + x_3 = 2 \\ 3x_1 + x_2 - 2x_3 = 3 \end{cases} \quad (\text{szukamy LICZB})$$

Jest to układ równań postaci $AX = B$, gdzie

$$A = \begin{bmatrix} 2 & 3 & -1 \\ 1 & -1 & 1 \\ 3 & 1 & -2 \end{bmatrix}, X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Do rozwiązania układu zastosujemy wzory Cramera:

$$x_1 = \frac{\det A_{x_1}}{\det A}, x_2 = \frac{\det A_{x_2}}{\det A}, x_3 = \frac{\det A_{x_3}}{\det A}$$

gdzie $A_{x_1}, A_{x_2}, A_{x_3}$ to macierze powstałe przez zastąpienie odpowiednich kolumn macierzy A kolumną wyrazów wolnych B :

$$A_{x_1} = \begin{bmatrix} 1 & 3 & -1 \\ 2 & -1 & 1 \\ 3 & 1 & -2 \end{bmatrix}, A_{x_2} = \begin{bmatrix} 2 & 1 & -1 \\ 1 & 2 & 1 \\ 3 & 3 & -2 \end{bmatrix}, A_{x_3} = \begin{bmatrix} 2 & 3 & 1 \\ 1 & -1 & 2 \\ 3 & 1 & 3 \end{bmatrix}$$

$$\text{🍌} + \text{🍌} = 16$$

$$\text{🍏} + \text{🍏} + \text{🍐} = 22$$

$$\text{🍐} + \text{🍌} = \text{🍏}$$

$$\text{🍐} = \text{🕒}$$

$$\text{🍌} = \text{🕒}$$

$$\text{🍏} = \text{🕒}$$

Równania różniczkowe

- ▶ Izaak Newton - “dajcie mi konfiguracje sił a powiem Wam co się stanie dalej”

$$ma = F$$

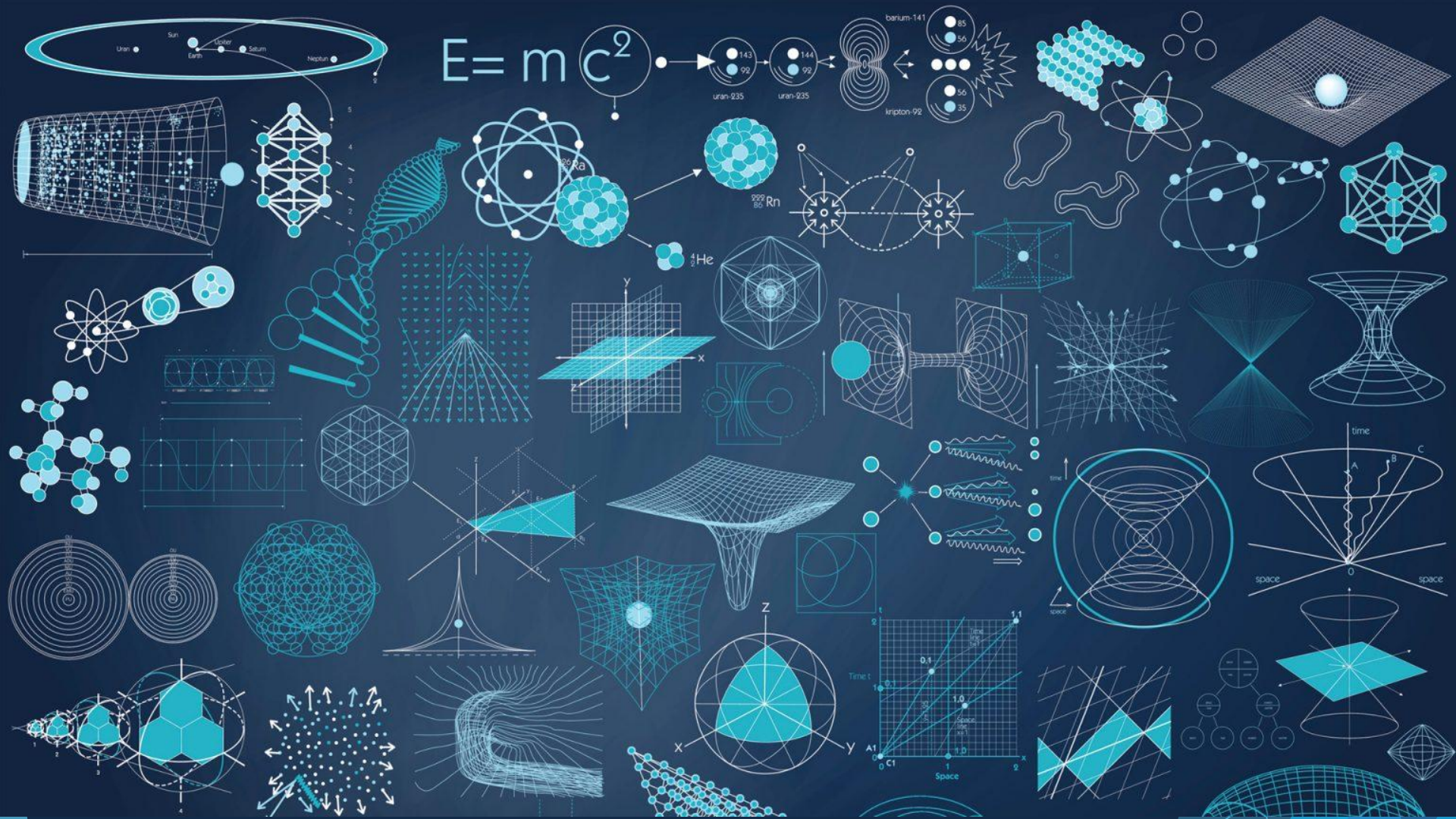
$$m \frac{d^2}{dt^2} x(t) = F$$

szukamy funkcji $x(t)$



Np. nie ma sił $F=0 \Rightarrow x(t) = C_1 \cdot t + C_2$
 $x(t) = V_0 \cdot t + x_0$

Siła constans $F=mg \Rightarrow x(t) = \frac{g \cdot t^2}{2} + C_1 \cdot t + C_2$



Wszędzie obliczenia

ChatGPT



ChatGPT

A mathematical pendulum, also known as a simple pendulum, is an idealized model used in physics to describe the motion of a pendulum under the influence of gravity. It consists of a weightless, inextensible cord of fixed length l , with a mass m (referred to as a bob) attached to one end, while the other end is fixed at a pivot point. When displaced from its equilibrium position and released, the mass swings back and forth due to the gravitational force, undergoing periodic motion.

The motion of a mathematical pendulum is described by the differential equation:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \sin(\theta) = 0$$

where:

- θ is the angular displacement from the vertical,
- g is the acceleration due to gravity,
- l is the length of the pendulum.

For small angles (θ), the $\sin(\theta)$ term can be approximated by θ (in radians), leading to a simpler form of the equation that describes simple harmonic motion:

$$\frac{d^2\theta}{dt^2} + \frac{g}{l} \theta = 0$$

The period T of the pendulum, which is the time it takes for the pendulum to complete one full swing back and forth, is given by:

$$T = 2\pi \sqrt{\frac{l}{g}}$$

This formula shows that the period T is independent of the mass m and depends only on the length l of the pendulum and the gravitational acceleration g .

Gemini



A mathematical pendulum is an idealized model of a real pendulum. It aims to simplify the real-world behavior by making certain assumptions, making it easier to analyze and understand its motions. Here are the key characteristics of a mathematical pendulum:

Idealizations:

- **Point mass:** The bob (the swinging object) is treated as a point with all its mass concentrated at a single location.
- **Massless string/rod:** The string or rod suspending the bob has no mass, so it doesn't contribute to the movement.
- **Frictionless:** There is no friction or air resistance acting on the bob, so its swings wouldn't slow down over time.
- **Small angles:** The pendulum swings back and forth only through small angles compared to its length. This allows us to use sine and cosine functions for calculations, which simplifies the equations.
- **Planar motion:** The bob moves only in a vertical plane determined by its initial swing.

Motion:

- **Oscillation:** Due to gravity, the bob oscillates back and forth about its equilibrium position (hanging straight down).
- **Period:** The time it takes to complete one full swing (back and forth) is called the period and is independent of the bob's mass, depending only on the length of the pendulum and the acceleration due to gravity (g).
- **Frequency:** This is the number of swings per second and is inversely proportional to the period.

ChatGPT

A mathematical pendulum, or simple pendulum, is a model consisting of a mass (the "bob") attached to a string or rod of fixed length, which swings freely under the influence of gravity. The motion is described by a second-order differential equation derived from Newton's second law and assuming small angles (linear approximation):

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \sin(\theta) = 0$$

where:

- θ is the angular displacement from the vertical,
- g is the acceleration due to gravity (9.81 m/s^2 on the surface of the Earth),
- L is the length of the pendulum.

For small angles ($\theta \ll 1$ radian), $\sin(\theta) \approx \theta$, simplifying the equation to:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \theta = 0$$

This simplification leads to simple harmonic motion, which can be solved analytically. However, to solve the non-linear equation for any angle, we typically use numerical methods. Below is a Python code snippet using `scipy.integrate.solve_ivp` to solve the differential equation for a mathematical pendulum with an initial angle:

python

Copy code

```
import numpy as np
from scipy.integrate import solve_ivp
import matplotlib.pyplot as plt

# Constants
g = 9.81 # acceleration due to gravity, in m/s^2
L = 1.0 # length of pendulum, in meters

# Differential equation for the pendulum (theta'' + (g/L)*sin(theta) = 0)
def pendulum(t, y):
    theta, theta_dot = y
    dtheta_dt = theta_dot
    dtheta_dot_dt = -(g/L) * np.sin(theta)
    return [dtheta_dt, dtheta_dot_dt]

# Initial conditions: theta = 0.2 radians, theta_dot = 0.0 radians/s
y0 = [0.2, 0.0]

# Time span for the solution
t_span = (0, 10) # 10 seconds
t_eval = np.linspace(t_span[0], t_span[1], 1000)

# Solve the differential equation
sol = solve_ivp(pendulum, t_span, y0, t_eval=t_eval, method='RK45')

# Plot the solution
plt.plot(sol.t, sol.y[0])
plt.xlabel('Time (s)')
plt.ylabel('Angular displacement (radians)')
plt.title('Pendulum Motion')
plt.grid(True)
plt.show()
```

This code defines the differential equation for the pendulum's motion, solves it over a time span of 10 seconds for a pendulum of length 1 meter with an initial displacement of 0.2 radians, and plots the angular displacement over time.

GPT + Colab



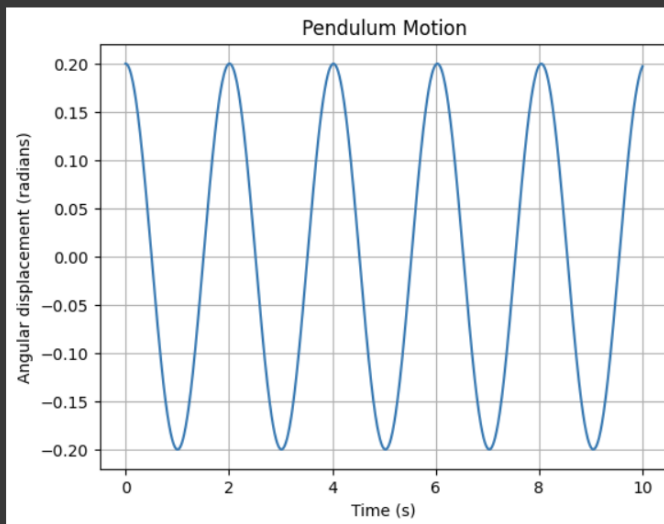
MP.ipynb

File Edit View Insert Runtime Tools Help

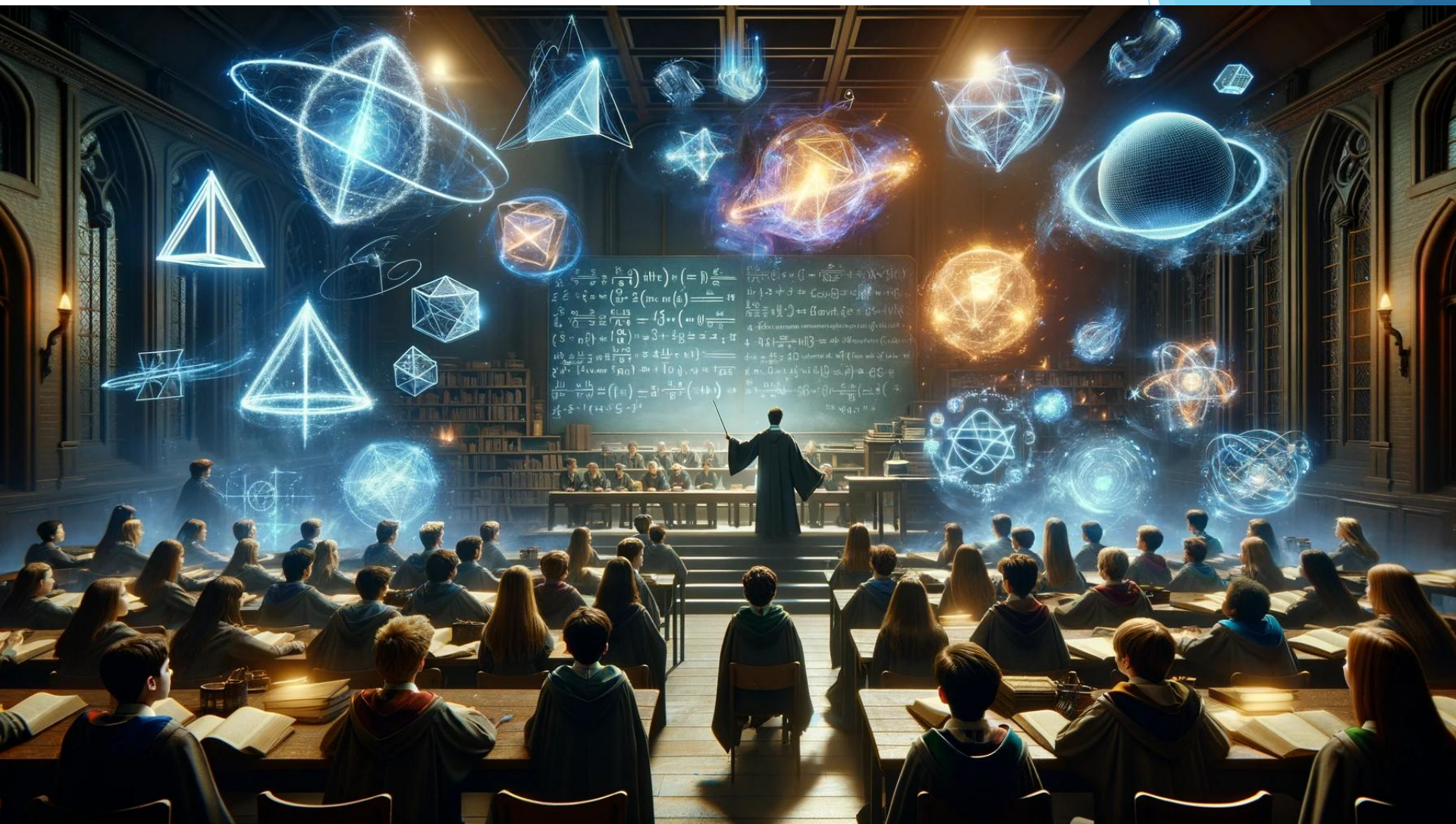
+ Code + Text



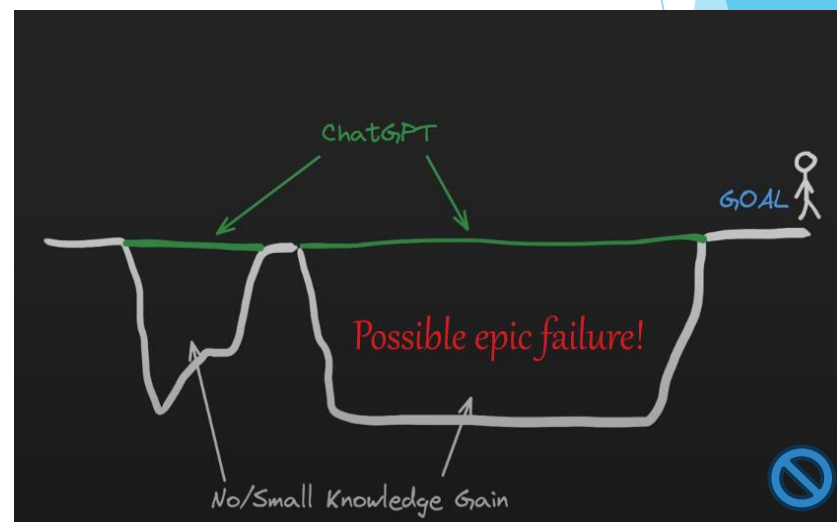
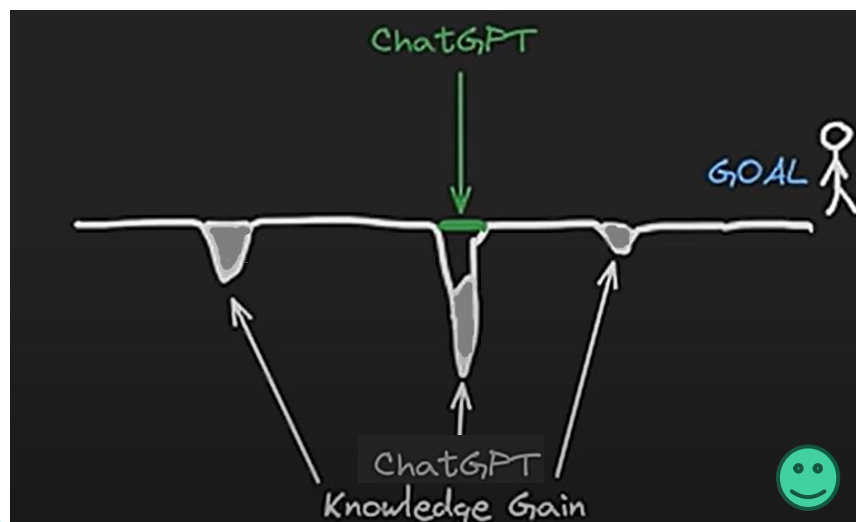
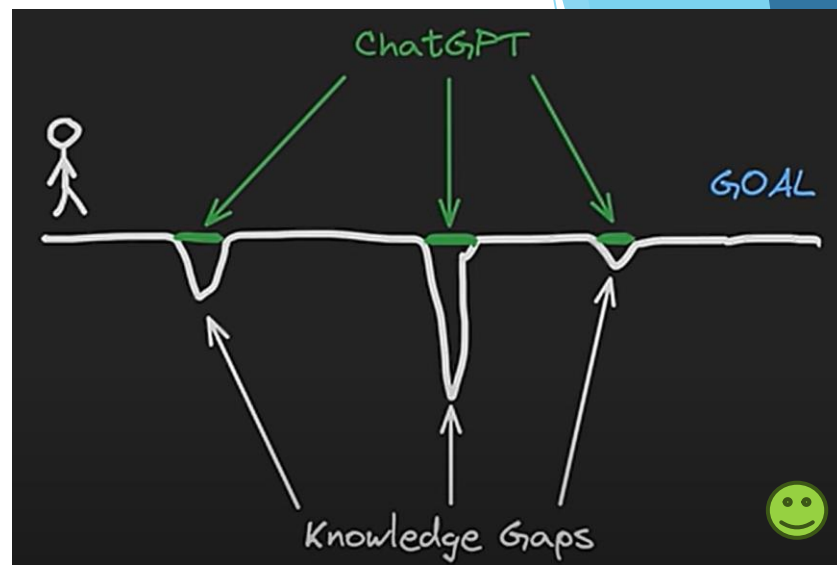
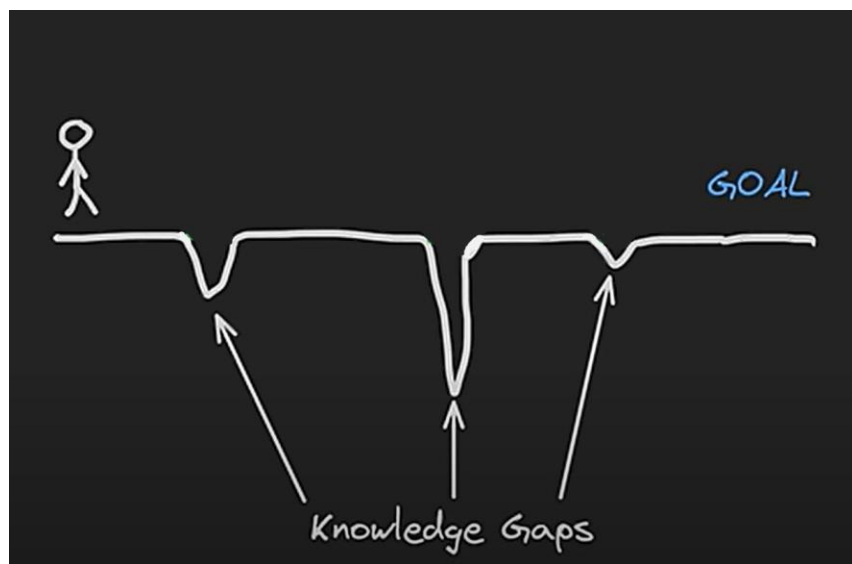
```
[1] 1 import numpy as np
2 from scipy.integrate import solve_ivp
3 import matplotlib.pyplot as plt
4
5 # Constants
6 g = 9.81 # acceleration due to gravity, in m/s^2
7 L = 1.0 # length of pendulum, in meters
8
9 # Differential equation for the pendulum (theta'' + (g/L)*sin(theta) = 0)
10 def pendulum(t, y):
11     theta, theta_dot = y
12     dtheta_dt = theta_dot
13     dtheta_dot_dt = -(g/L) * np.sin(theta)
14     return [dtheta_dt, dtheta_dot_dt]
15
16 # Initial conditions: theta = 0.2 radians, theta_dot = 0.0 radians/s
17 y0 = [0.2, 0.0]
18
19 # Time span for the solution
20 t_span = (0, 10) # 10 seconds
21 t_eval = np.linspace(t_span[0], t_span[1], 1000)
22
23 # Solve the differential equation
24 sol = solve_ivp(pendulum, t_span, y0, t_eval=t_eval, method='RK45')
25
26 # Plot the solution
27 plt.plot(sol.t, sol.y[0])
28 plt.xlabel('Time (s)')
29 plt.ylabel('Angular displacement (radians)')
30 plt.title('Pendulum Motion')
31 plt.grid(True)
32 plt.show()
33
```



Dostównie rzucamy zaklęcia!



Jak powinno się używać i nieużywać ChataGPT



From https://www.youtube.com/watch?v=6CGtwF_5kzY

ChatGPT jako rower dla umysłu!

Inne narzędzia

- Geogebra - <https://www.geogebra.org/>
- Colab - <https://colab.research.google.com/>
- ChatGPT - <https://chat.openai.com/>
- Gemini Google - <https://gemini.google.com/app>
- Claude - <https://claude.ai/>

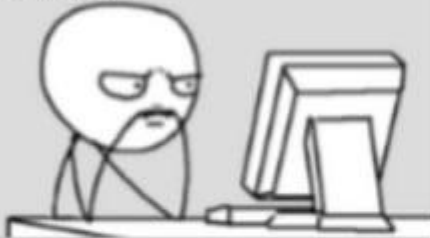
Ty musisz być
za kierownicą
tych narzędzi!



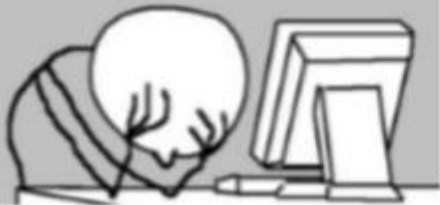
Workflow

Days before OpenAI

Developer coding
- 2 hours



Developer debugging
- 6 hours

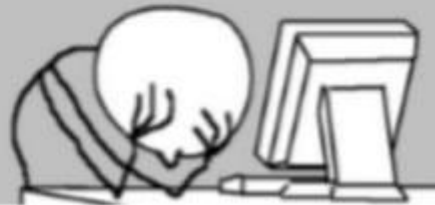


Days after OpenAI

ChatGPT generates
Codes - 5 min



Developer debugging
- 24 hours



Przykład dostępnych narzędzi

► <https://oeis.org/>



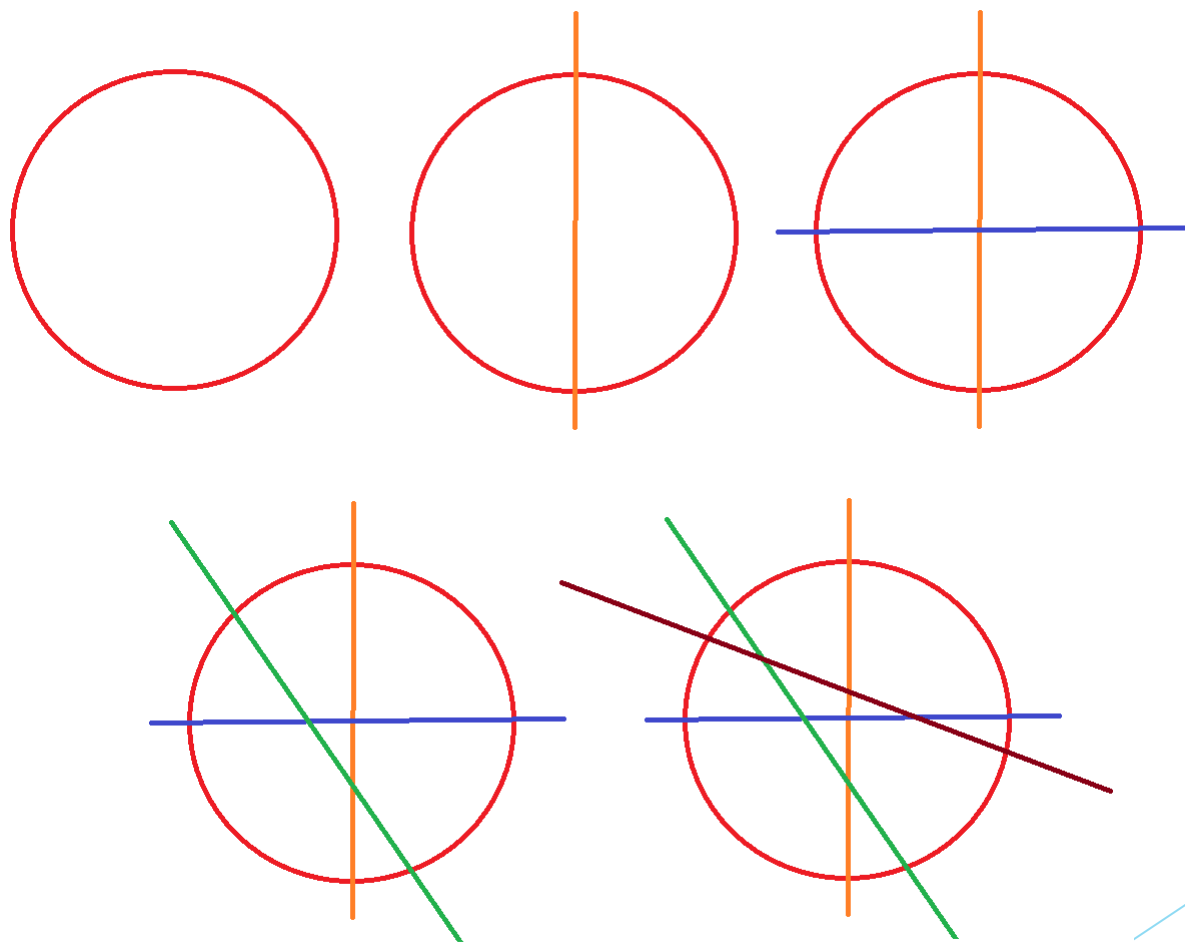
Encyklopedia ciągów liczb całkowitych w wersji on-line

Proszę podać ciąg, słowo (po angielsku), lub numer ciągu

[Wskazówki](#) [Welcome](#) [Video](#)

Kolejne strony są prezentowane w języku angielskim.

Używając strony: <https://oeis.org/> określ na ile maksymalnie kawałków będzie podzielony tort po 10 cięciach.



Używając strony: <https://oeis.org/> określ na ile maksymalnie kawałków będzie podzielony tort po 10 cięciach.

1,2,4,7,11 Search Query Search [Hints](#)
 (Greetings from [The On-Line Encyclopedia of Integer Sequences!](#))

Search: **seq:1,2,4,7,11**

Displaying 1-10 of 185 results found. page 1 [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) ... [19](#)

Sort: relevance | [references](#) | [number](#) | [modified](#) | [created](#) Format: long | [short](#) | [data](#)

[A000124](#) Central polygonal numbers (the Lazy Caterer's sequence): $n(n+1)/2 + 1$; or, maximal number of pieces formed when slicing a pancake with n cuts. +20
333
 (Formerly M1041 N0391)

1, 2, 4, 7, 11, 16, 22, 29, 37, 46, 56, 67, 79, 92, 106, 121, 137, 154, 172, 191, 211, 232, 254, 277, 301, 326, 352, 379, 407, 436, 466, 497, 529, 562, 596, 631, 667, 704, 742, 781, 821, 862, 904, 947, 991, 1036, 1082, 1129, 1177, 1226, 1276, 1327, 1379 ([list](#); [graph](#); [refs](#); [listen](#); [history](#); [text](#); [internal format](#))

OFFSET 0,2

COMMENTS These are Hogben's central polygonal numbers with the (two-dimensional) symbol
 2
 .P
 1 n
 The first line cuts the pancake into 2 pieces. For $n > 1$, the n -th line crosses every earlier line (avoids parallelism) and also avoids every previous line intersection, thus increasing the number of pieces by n . For 16 lines, for example, the number of pieces is $2 + 2 + 3 + 4 + 5 + \dots + 16 = 137$. These are the triangular numbers plus 1 (cf. [A000217](#)).
 $m = (n-1)(n-2)/2 + 1$ is also the smallest number of edges such that all graphs with n nodes and m edges are connected. - [Keith Briggs](#), May 14 2004
 Also maximal number of grandchildren of a binary vector of length $n+2$. E.g., a binary vector of length 6 can produce at most 11 different vectors when 2 bits are deleted.
 This is also the order dimension of the (strong) Bruhat order on the finite Coxeter group B_{n+1} . - Nathan Reading (reading(AT)math.umn.edu), Mar 07 2002
 Number of 132- and 321-avoiding permutations of $\{1, 2, \dots, n+1\}$. - [Emeric Deutsch](#), Mar 14 2002

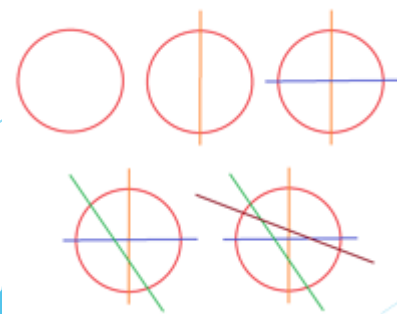
Bliżej końca mamy formułę:

MAPLE

[A000124](#) := n -> n*(n+1)/2+1;

Sprawdzamy czy działa np. dla $n=4$, czyli $4*(4+1)/2+1=11$

Sprawdzamy dla $n=10$, czyli $10*(10+1)/2+1=56$

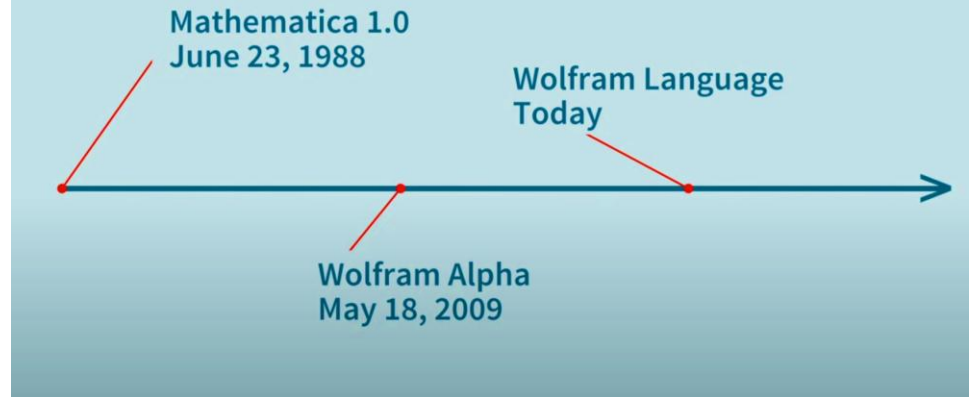


Mathematica

[Stephen Wolfram]



What is Mathematica?



- ▶ <https://www.youtube.com/watch?v=QMAjAQg2Grc>
- ▶ <https://www.youtube.com/watch?v=ysVdWiBVKHc>

Symbolic Language

$f[x]$

Mathematical Computation

$$\sum_{k=0}^{\infty} \frac{(a_1)_k}{(b_1)_k}$$

Numerics



Visualization



Algebraic Manipulation

$A=B$

Number Theory

Data Analysis



Graph Computation

Interactive Computation

Image Computation

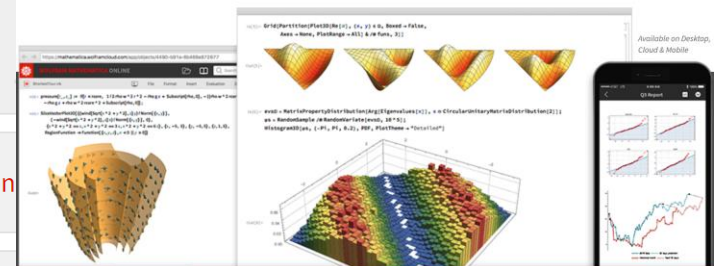


Geometric Computation

Importing & Exporting

WOLFRAM MATHEMATICA

The world's definitive system for modern technical computing



<http://www.wolframalpha.com/>



Enter what you want to calculate or know about



Browse Examples Surprise Me

Compute expert-level answers using Wolfram's breakthrough algorithms, knowledgebase and AI technology

Mathematics ›

Step-by-Step Solutions

Elementary Math

x^2-1 Algebra

Plotting & Graphics

$\int f(x) dx$ Calculus & Analysis

$\frac{x}{12}$ Geometry

$y''(x)$ Differential Equations

Statistics

... More »

Science & Technology ›

Units & Measures

Physics

Chemistry

Engineering

Computational Sciences

Earth Sciences

Materials

Transportation

... More »

Society & Culture ›

People

Arts & Media

Dates & Times

Words & Linguistics

Money & Finance

Food & Nutrition

Political Geography

History

... More »

Everyday Life ›

Personal Health

Personal Finance

Surprises

Entertainment

Household Science

Household Math

Hobbies

Today's World

... More »

Octave Command Line and GUI

- ▶ Octave jest to program komputerowy oraz skryptowy język programowania przeznaczony do wykonywania obliczeń numerycznych, wolny odpowiednik komercyjnego programu Matlab.

- ▶ Program jest aktywnie rozwijany od 1992 roku.

The screenshot displays the GNU Octave 4.4.1 GUI. The interface is divided into several panes:

- File Browser:** Located on the left, it shows the file system structure. The current directory is `C:\Users\USER`. The workspace contains variables: `ans` (double, 1x1, value 1.4285...).
- Command Window:** The central pane showing the Octave prompt and user input/output. It displays the Octave startup message, including the version (4.4.1), copyright (© 2018 John W. Eaton and others), and a disclaimer. It also shows the user input `>> 1+2+3+4` and the output `ans = 10`, followed by `>> 1/ans` and the output `ans = 0.10000`, and finally `>> 1/7` and the output `ans = 0.14286`.
- Variable Editor:** Located on the right, it shows the current workspace variables. The variable `ans` is listed with its class (double), dimensions (1x1), and value (1.4285...).
- Command History:** Located at the bottom left, it shows a list of commands entered in the Command Window, including `1+2+3+4`, `1/ans`, `1/7`, `format long`, and `1/7`.

A separate window titled `C:\OCTAVE-1.1\bin\octave-gui.exe` shows the Octave command line interface (CLI) running the same commands as the GUI. The CLI output is identical to the Command Window output in the GUI.

Octave służy do rozwiązywania problemów numerycznych, m.in.:

- ▶ obliczanie wartości wyrażeń (w tym wyrażeń zawierających zaawansowane funkcje matematyczne, np. funkcje zespolone)
- ▶ znajdowania wartości sum i iloczynów ciągów liczb o bardzo dużej liczbie elementów
- ▶ znajdowania rozwiązań układów równań liniowych (mogących mieć nawet tysiące niewiadomych)
- ▶ rozwiązywania równań i układów równań nieliniowych
- ▶ znajdowania wartości całek (oznaczonych)
- ▶ rozwiązywania układów równań różniczkowych zwyczajnych
- ▶ rozwiązywania układów równań różniczkowych cząstkowych
- ▶ rozwiązywanie standardowych problemów algebry liniowej, m.in. wyznaczania wartości i wektorów własnych
- ▶ prezentacji rozwiązań w postaci wykresów
- ▶ Octave to także [skryptowy język programowania](#) posiadający mechanizmy włączania do obliczeń wysokowydajnych funkcji napisanych w kompilowanych językach programowania, np. C/C++.

Obliczenia numeryczne i symboliczne

- ▶ Są dwa różne podejścia do rozwiązywania problemów matematycznych i fizycznych, często wykorzystywane w naukach ścisłych i inżynierii.

1. Obliczenia symboliczne:

1. Polegają na manipulacji wyrażeń matematycznych w formie symbolicznej, np. znajdowaniu dokładnych rozwiązań równań.
2. Wykorzystują reguły algebry, rachunku różniczkowego i całkowego do przekształcania wyrażeń w formie dokładnej.
3. Typowe narzędzia: Mathematica, Maple, SymPy (Python).
4. Przykłady zastosowań: dowody matematyczne, analiza funkcji, symboliczne rozwiązywanie równań.

2. Obliczenia numeryczne:

1. Polegają na przybliżonym rozwiązywaniu równań matematycznych za pomocą algorytmów numerycznych.
2. Używane, gdy analityczne (symboliczne) rozwiązania są trudne lub niemożliwe do uzyskania.
3. Wymagają zastosowania metod takich jak różniczkowanie numeryczne, całkowanie, metody iteracyjne (np. metoda Newtona).
4. Typowe narzędzia: MATLAB, NumPy (Python), Fortran, C++.
5. Przykłady zastosowań: symulacje fizyczne, analiza danych, rozwiązywanie równań różniczkowych.

Obliczenia symboliczne

$$2^x = 8$$

$$2^x = 2^3$$

$$x = 3$$

$$e^x = 10$$

$$x = \ln 10$$

$$x = 2,3025\dots$$

$$\sin x = \cos x$$

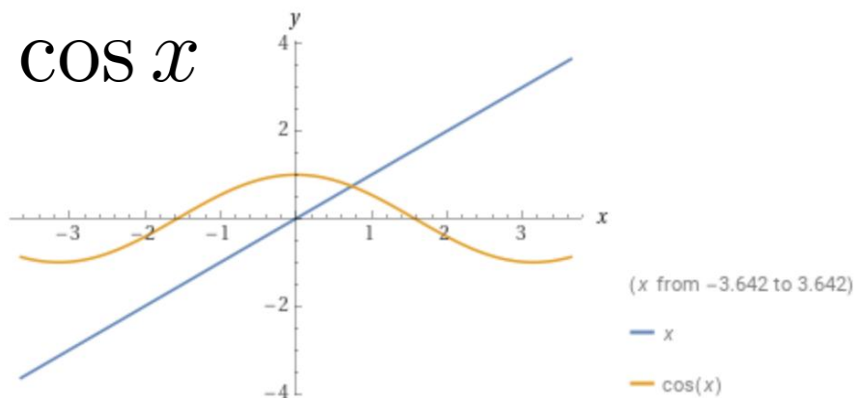
$$\tan x = 1$$

$$x = \arctan(1) = \frac{\pi}{4}$$

Obliczenia numeryczne

$$x = \cos x$$

$$x \approx 0.739085\dots$$



Octave
(Oktawa)

Octave umożliwia wykonywanie wszystkich podstawowych operacji matematycznych. Wykaz najważniejszych operatorów rozpoznawanych przez Octave znajduje się w Tabeli 1.

Tabela 1. Podstawowe operatory arytmetyczne Octave

Operacja	Operator	Przykłady
potęgowanie	** lub ^	2**3, 2^3
mnożenie	*	2 * 3
dzielenie	/ lub \	1/7 lub 7\1
dodawanie	+	2 + 2
odejmowanie	-	3 - 2

Tabela 2. Najważniejsze stałe w Octave

Stała	Zapis w Octave	Znaczenie
π	pi	iloraz obwodu koła do jego średnicy
e	e	podstawa logarytmów naturalnych
i	i lub j	jednostka urojona

Dokładne znaczenie dwóch ostatnich stałych zostanie omówione w dalszej części kursu. Teraz wystarczy zapamiętać, że e jest pewną liczbą,

```
>> e
ans = 2.7183
```

natomiast i jest „magicznym obiektem”, który podniesiony do kwadratu daje -1:

```
>> i*i
ans = -1
```

Liczenie w Octave

```
>> pi^2
ans = 9.8696
>> log10(1000)
ans = 3
>> log10(1e3)
ans = 3
>> sin(cos(sin(cos(1))))
ans = 0.76471
>> factorial(6)
ans = 720
>> factorial(50)
ans = 3.0414e+064
>> log10(factorial(50))
ans = 64.483
```

```
> cos([0, 0.1, 0.2])
ans =
    1.00000    0.99500    0.98007
```

```
>> sin(30 * pi / 180) # sinus 30 stopni
ans = 0.50000
```

help and doc

```
>> help factorial
'factorial' is a function from the file C:\OCTAVE~1.1\share\octave\4.4.1\m\specfun\factorial.m

-- factorial (N)
  Return the factorial of N where N is a real non-negative integer.

  If N is a scalar, this is equivalent to 'prod (1:N)'. For vector
  or matrix arguments, return the factorial of each element in the
  array.

  For non-integers see the generalized factorial function 'gamma'.
  Note that the factorial function grows large quite quickly, and
  even with double precision values overflow will occur if N > 171.
  For such cases consider 'gammaln'.

  See also: prod, gamma, gammaln.

Additional help for built-in functions and operators is
available in the online version of the manual. Use the command
'doc <topic>' to search the manual index.
```

```
>> doc factorial
>> |
```

factorial (*n*)

Return the factorial of *n* where *n* is a real non-negative integer.

If *n* is a scalar, this is equivalent to `prod (1:n)`. For vector or matrix arguments, return the factorial of each element in the array.

For non-integers see the generalized factorial function `gamma`. Note that the factorial function grows large quite quickly, and even with double precision values overflow will occur if *n* > 171. For such cases consider `gammaln`.

See also: [prod](#), [gamma](#), [gammaln](#).

Skrypty i funkcje

Założmy, że skrypt nazywa się `programik.m`, a jego zawartość wygląda następująco:

```
# W pliku programik.m znajduje się trywialny program testowy Octave
1/7
2/7
3/7
```

Jeżeli wydamy polecenie `programik`:

```
> programik
```

Program odpowie następująco:

```
ans = 0.14286
ans = 0.28571
ans = 0.42857
```

co świadczy o wykonaniu wszystkich instrukcji z pliku `programik.m`.

Założmy, że naszym celem jest zdefiniowanie funkcji o nazwie `sin5`, która dla argumentu x obliczać będzie wartość wyrażenia $x - x^3/6 + x^5/120$. Aby zdefiniować taką funkcję:

1. Tworzymy (w katalogu bieżącym) plik o nazwie `sin5.m`
2. W pliku tym wpisujemy:

```
function y = sin5(x)
    y = x - x**3/6 + x**5/120;
endfunction
```

Uwaga! Powyższa definicja nie jest optymalna i wkrótce ją poprawimy.

Funkcję tę wywołujemy w Octave w naturalny sposób:

```
# Funkcja sin5 przybliża wartość funkcji sin(x) za pomocą wielomianu stopnia 5.
#
# Przybliżenie to jest całkiem dobre dla małych wartości x,
# np. dla |x| < 1.
#
```

```
function y = sin5(x)
    y = x - x.**3/6 + x.**5/120;
endfunction
```

```
>> sin5 (0.1)
ans = 0.099833
```


Wielomiany [\[edytuj\]](#)

definicja wielomianu

```
octave:3> c=[1 1 1]
c = 1 1 1
```

wyświetlenie wielomianu

```
octave:4> polyout(c, 'z')
1*z^2 + 1*z^1 + 1
```

Miejsca zerowe wielomianu czyli rozwiązania równania $1 * z^2 + 1 * z^1 + 1 = 0$

```
octave:5> roots(c)
ans =
-0.50000 + 0.86603i
-0.50000 - 0.86603i
```

Obliczyć wartość wielomianu $7x^3 - x^2 + 5$ w punkcie 11.

```
P=[7 -1 0 5];
polyval(P, 11)
```

Wynikiem jest

```
9201
```

Rozwiązanie układu równań liniowych [

Rozwiązać układ równań $\begin{cases} 2x - y = 1 \\ x + 3y = 11 \end{cases}$

```
A = [2 -1; 1 3];  
f = [1 11];  
u = A\f'
```

lub alternatywnie:

```
u=inv(A)*f'
```

Wynikiem jest:

```
u = [2 3]
```

Wykresy w Oktawie

Octave

File Edit Debug Window Help News

Current Directory: C:\Users\USER

File Browser

C:/Users/USER

Name

- .config
- .gimp-2.8
- .oracle_jre_usage
- .thumbnails
- .VirtualBox
- 3D Objects
- Contacts
- Desktop
- Documents
- Downloads

Workspace

Filter

Var	Class	Dimensi	Value	Attribute
ans	double	1x1	0	

Command History

Filter

- 1+2+3+4
- 1/ans
- 1/7
- 1/ans
- format long
- 1/7
- help factorial
- doc factorial
- help sin
- sombbrero
- help sombrero

Command Window

'sin' is a built-in function from the file libinterp/corefcn/mappers.cc

```
-- sin (X)
    Compute the sine for each element of X in radians.

    See also: asin, sind, sinh.

Additional help for built-in functions and operators is
available in the online version of the manual. Use the command
'doc <topic>' to search the manual index.

Help and information about Octave is also available on the WWW
at https://www.octave.org and via the help@octave.org
mailing list.
>> sombrero
>> help sombrero
'sombrero' is a function from the file C:\OCTAVE-1.1\share\octave\
libinterp\octave\src\functions\liboctinterp\src\functions\sombrero.cc

-- sombrero ()
-- sombrero (N)
-- Z = sombrero (...)
-- [X, Y, Z] = sombrero (...)
    Plot the familiar 3-D sombrero function.

The function plotted is

    z = sin (sqrt (x^2 + y^2)) / (sqrt (x^2 + y^2))

Called without a return argument, 'sombrero' plots the surface
of the above function over the meshgrid [-8,8] using 'surf'.

If N is a scalar the plot is made with N grid lines. The
value for N is 41.

When called with output arguments, return the data for the
function evaluated over the meshgrid. This can subsequently be plotted
using 'surf (X, Y, Z)'.

See also: peaks, meshgrid, mesh, surf.

Additional help for built-in functions and operators is
available in the online version of the manual. Use the command
'doc <topic>' to search the manual index.

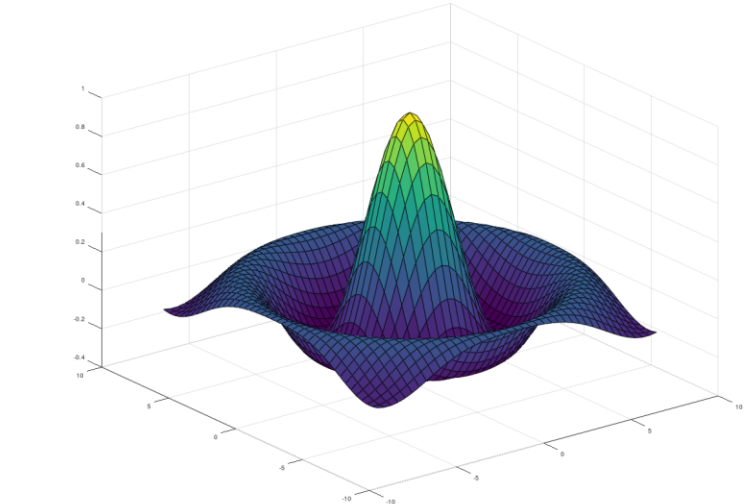
Help and information about Octave is also available on the WWW
at https://www.octave.org and via the help@octave.org
mailing list.
>>
```

Variable Editor

Figure 1

File Edit Help

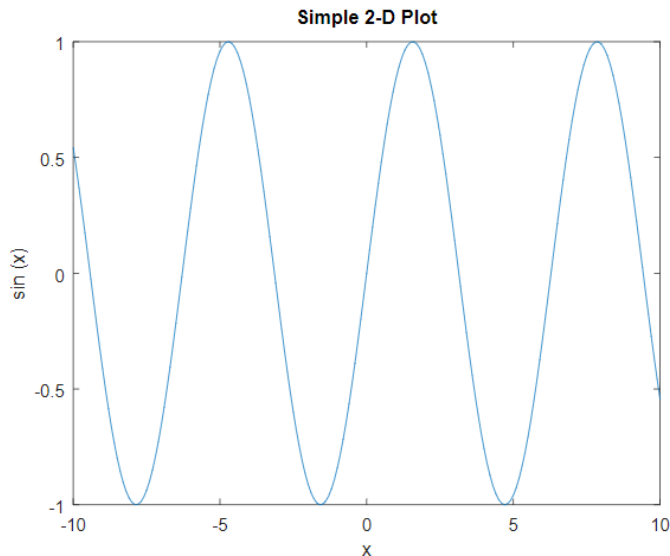
Zoom In Zoom Out Insert Text Axes Grid Autoscale



(9.4825, -6.133)

Wykresy 2D

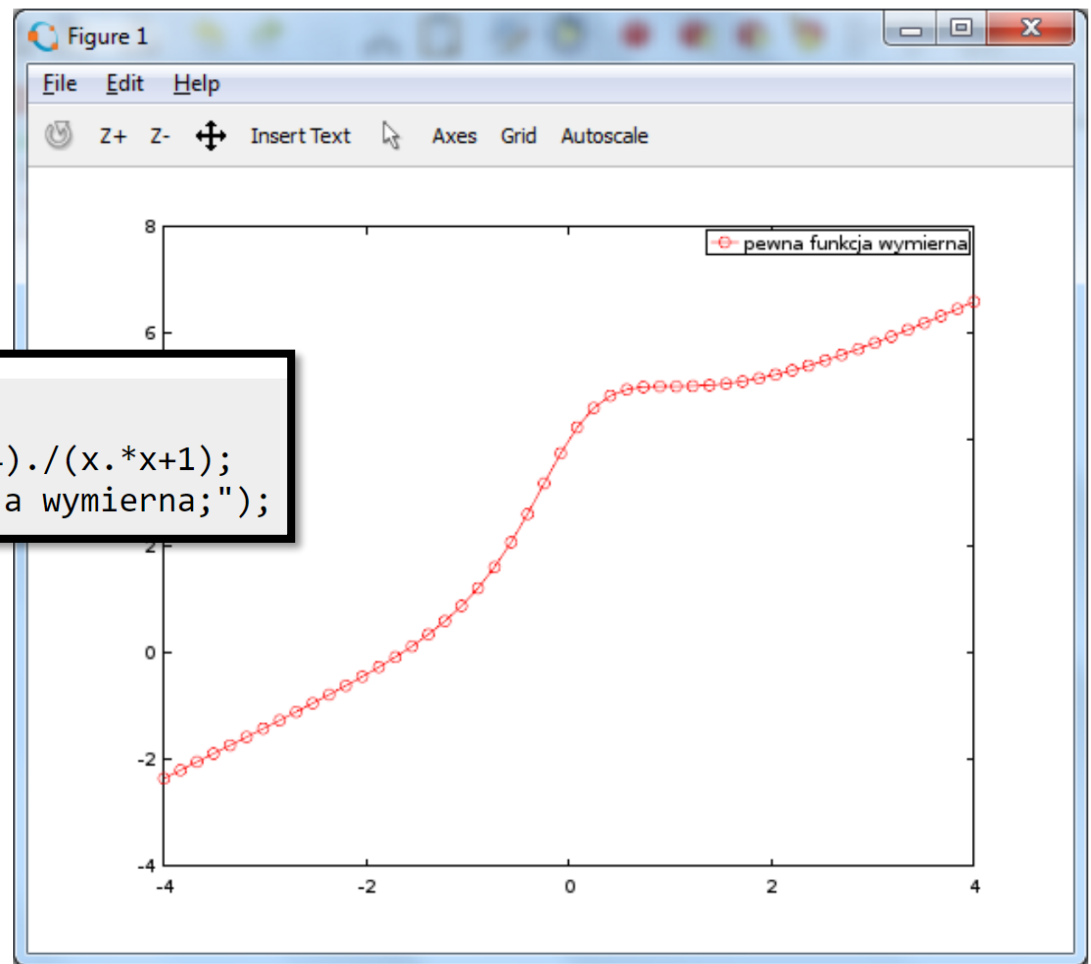
```
x = -10:0.1:10; # Create an evenly-spaced vector from -10..10
y = sin (x);     # y is also a vector
plot (x, y);
title ("Simple 2-D Plot");
xlabel ("x");
ylabel ("sin (x)");
```



Octave to tylko i aż
kalkulator na sterydach

Wykresy 2D

```
x = linspace(-4,4,50);  
y = (x.^3 + 2*x.^2 + 3*x + 4)./(x.*x+1);  
plot(x, y, "ro-;pewna funkcja wymierna;");
```



Zmianie uległa definicja wektora x . Funkcja **linspace** (**a,b, N**) generuje ciąg N równoodległych punktów, z których pierwszy równy jest a , a ostatni – b . Zaletą tego podejścia jest to, że od razu widać, z ilu elementów składa się x .

Druga zmiana zaszła w wywołaniu funkcji `plot`: doszedł trzeci argument, napis **"ro-;pewna funkcja wymierna;"**.

Ten trzeci argument definiuje format wykresu i składa się z liter i symboli umieszczonych w napisie w dowolnej kolejności

Opcje stylu

■ Styl linii:

- - Linia ciągła (opcja domyślna)
- -- Linia przerywana
- : Linia kropkowana
- - . Linia przerywana z kropkami

■ Styl symbolu reprezentującego punkt danych:

- + plusik
- o kółko
- * gwiazdka
- . punkt
- x krzyżyk
- s kwadrat
- d romb (ang. *diamond*)
- ^ trójkąt z ostrzem w górę
- v trójkąt w dół
- > trójkąt z ostrzem w prawo
- < trójkąt z ostrzem w lewo
- p pięciokąt
- h sześciokąt (ang. *hexagram*)

■ Kolor linii i/lub symbolu:

- k czarny (ang. *black*)
- r czerwony
- g zielony
- b niebieski
- m *magenta*
- c *cyan*
- w biały

■ Opis wykresu (legenda)

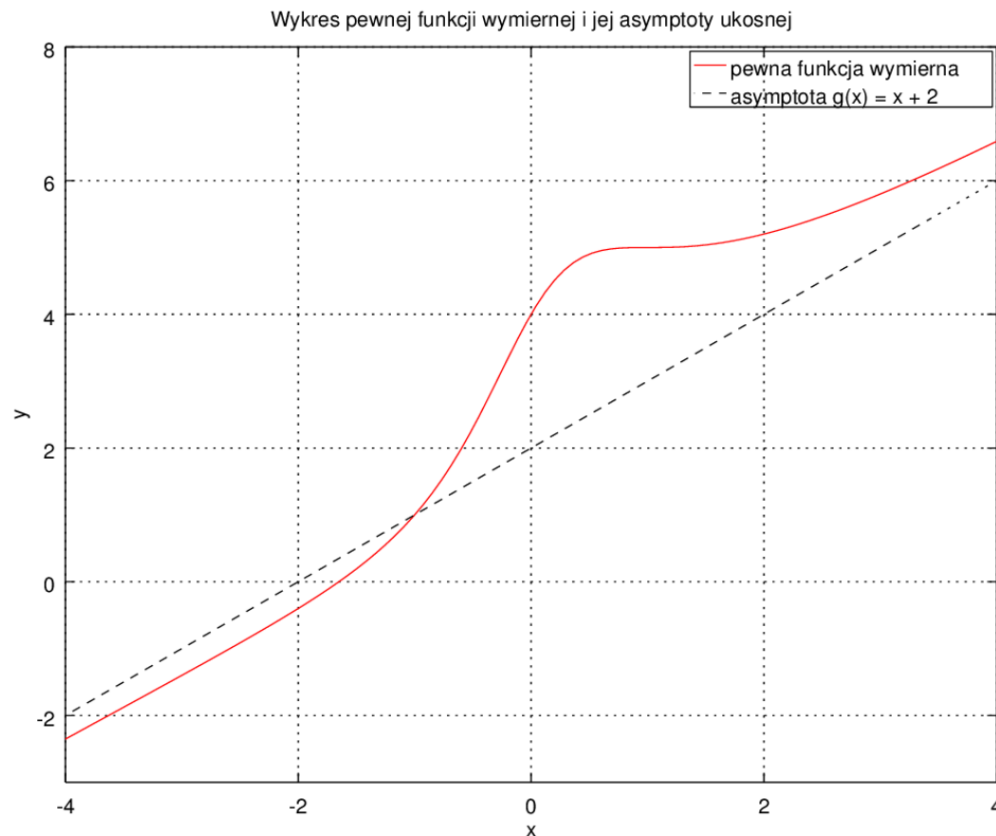
- ;legenda;

Dodajmy jeszcze kilka wodotrysków, jak podpisy osi, tytuł całego rysunku (niestety, bez polskich liter), siatkę, ograniczenie zakresu osi „y” do $[-3:8]$ i zapiszmy wykres na dysku w formacie eps i png:

```
x = linspace(-4,4,100);  
y = (x.^3 + 2*x.^2 + 3*x + 4)./(x.*x+1);  
y2 = x + 2;  
plot(x, y, "r-;pewna funkcja wymierna;", x, y2, "k--;asymptota g(x) = x + 2;");  
xlabel("x");  
ylabel("y");  
title("Wykres pewnej funkcji wymiernej i jej asymptoty ukosnej");  
ylim([-3,8]);  
grid on;  
print "rys.eps"  
print "rys.png"
```

Po paru
wodotryskach...

Oto wynik działania powyższego skryptu:



Operacje matematyczne

Dwumian Newtona

Dwumianem Newtona nazywamy wzór:

$$(x + y)^n = \binom{n}{0} x^n y^0 + \binom{n}{1} x^{n-1} y^1 + \binom{n}{2} x^{n-2} y^2 + \dots + \binom{n}{n-1} x^1 y^{n-1} + \binom{n}{n} x^0 y^n$$

gdzie symbol $\binom{n}{k}$ oznacza współczynnik dwumianowy i jest obliczany ze wzoru:

$$\binom{n}{k} = \frac{n!}{k! \cdot (n - k)!}$$

Silnia

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n - 1) \cdot n$$

Potęgi i silnia

$$\boxed{a^{-1} = \frac{1}{a}}$$

$$a^0 = 1$$

$$a^1 = a \quad \div a$$

$$a^2 = a \cdot a \quad \div a$$

$$a^3 = a \cdot a \cdot a \quad \div a$$

$(-1)!$ is undefined.

$$0! = 1 \quad (\text{by definition})$$

$$1! = 1$$

$$2! = 2$$

$$3! = 6$$

$$4! = 24$$

$$5! = 120$$

$$\left(\frac{1}{2}\right)! = \frac{\sqrt{\pi}}{2}$$

Wzory do zapamiętania

$$a^m \cdot a^n = a^{m+n}$$

$$(a^m)^n = a^{m \cdot n}$$

$$a^m \cdot b^m = (a \cdot b)^m$$

$$a^{-n} = \frac{1}{a^n}$$

$$a^{\frac{1}{n}} = \sqrt[n]{a}$$

Mnożenie jako szybkie dodawanie

$$\underbrace{3 + 3 + 3 + 3}_4 = \text{'cztery razy trzy'} = 4 \cdot 3 = 12$$

Potęgowanie jako szybkie mnożenie

$$\underbrace{3 \cdot 3 \cdot 3 \cdot 3}_4 = \text{'trzy do potęgi czwartej'} = 3^4 = 81$$

Co następne?

$$\begin{cases} x^a & \text{funkcja potęgowa} \\ a^x & \text{funkcja wykładnicza} \end{cases}$$

$$x^2 \quad \text{versus} \quad 2^x$$

Potęga vs wykładnik

Co szybciej rośnie, x^2 vs 2^x ?

$$x = 10$$

$$x^2 = 10^2 = 100$$

$$2^{10} = 2^5 \cdot 2^5 = 32 \cdot 32 = 1024$$

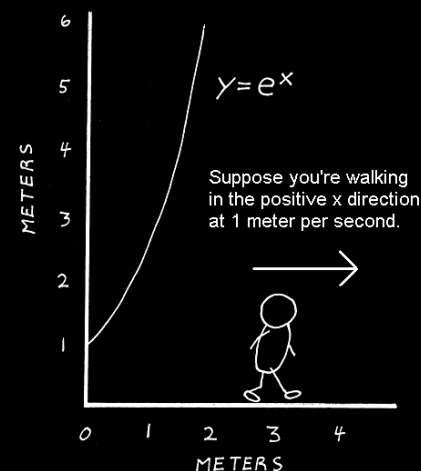
$$x = 100$$

$$x^2 = 100^2 = 10000$$

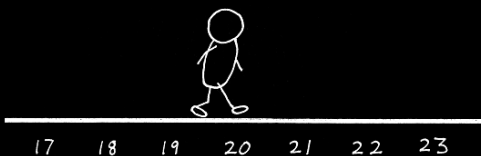
$$2^{100} = ?$$

Ile cyfr ma liczba 2^{100} , wiedząc, że $\log_{10} 2 = 0.3$?

$$2^{100} = (10^{\log_{10} 2})^{100} = 10^{100 \cdot \log_{10} 2} \approx 10^{30}$$

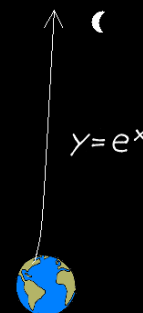


Then, by the time you're about 20 meters to the right of zero...



...the y position of the graph would be flying past the moon at about the speed of light.

That's all I wanted to say.



Logarytmy

$$\log_a b = c \iff a^c = b$$

$$a^{\log_a b} = b$$

$$\log_a (a^b) = b$$

$$\log_a (b \cdot c) = \log_a b + \log_a c$$

$$\log_a \left(\frac{b}{c} \right) = \log_a b - \log_a c$$

$$\log_a (b^c) = c \log_a b$$

$$\log_a b = \frac{\log_c b}{\log_c a}$$

$$\log_a (\sqrt[n]{b}) = \frac{\log_a b}{n}$$

$$\log_a (b^c) = c \log_a b$$

$$\log_a (\log_b c) = \log_b (\log_a c)$$

Logarithm Properties

$$\log_a xy = \log_a x + \log_a y$$

$$\log_a \frac{x}{y} = \log_a x - \log_a y$$

$$\log_a x^n = n \log_a x$$

$$\log_a b = \frac{\log_c b}{\log_c a}$$

$$\log_a b = \frac{1}{\log_b a}$$

The following can be derived from the above

$$\log_a 1 = 0$$

$$\log_a a = 1$$

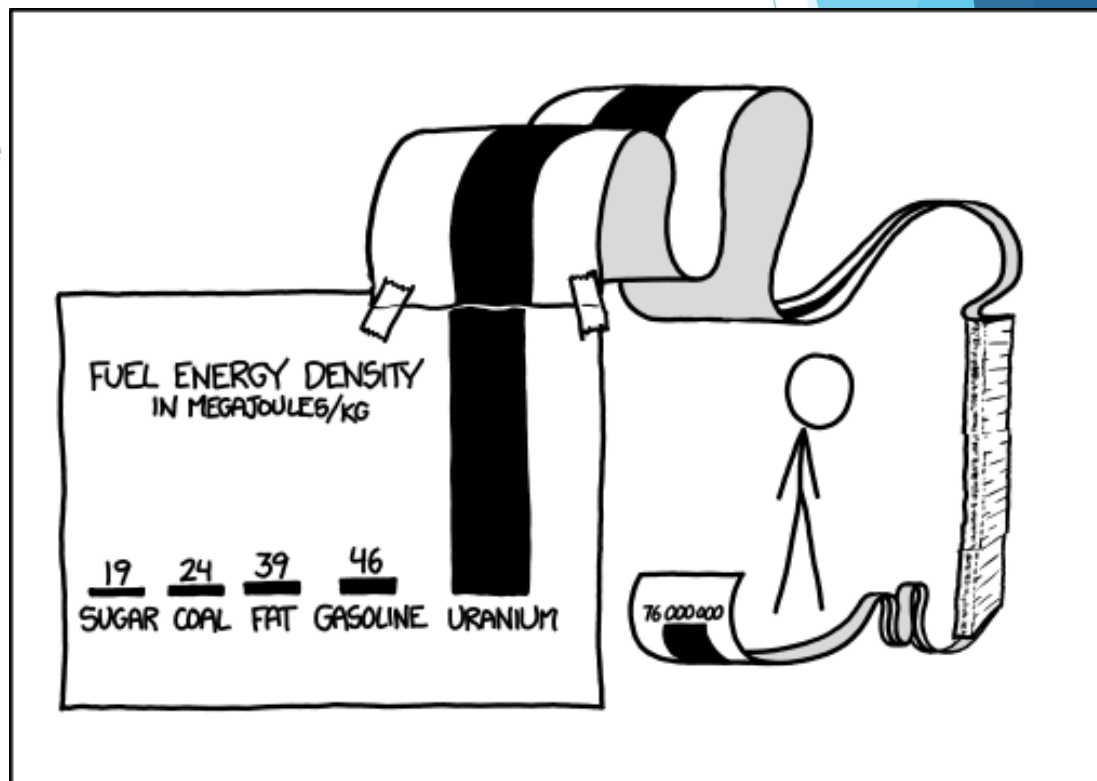
$$\log_a a^r = r$$

$$\log_a \frac{1}{b} = -\log_a b$$

$$\log_{\frac{1}{a}} b = -\log_a b$$

$$\log_a b \log_b c = \log_a c$$

$$\log_{a^m} a^n = \frac{n}{m}, m \neq 0$$



SCIENCE TIP: LOG SCALES ARE FOR QUITTERS WHO CAN'T FIND ENOUGH PAPER TO MAKE THEIR POINT *PROPERLY*.

Thanos problem



log_2 (people on Earth/1 person)

Extended Keyboard

Upload

Input interpretation:

$$\log_2 \left(\frac{\text{world population}}{1 \text{ person}} \right)$$

$\log_b(x)$ is the base- b logarithm

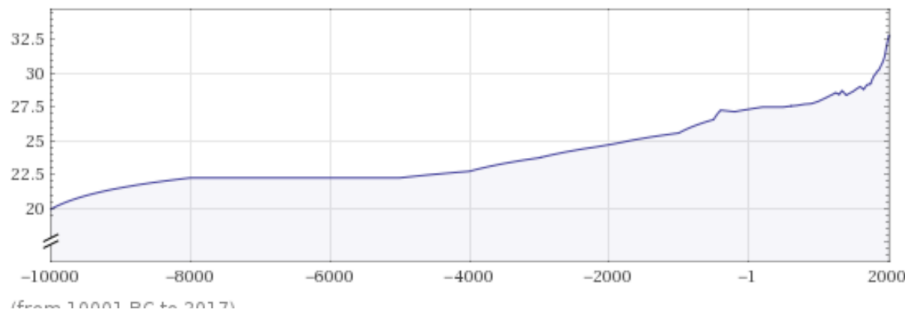
[Open code](#)

Result:

[Show details](#)

32.8 (2017 estimate)

History:



Odruch Pawłowa

- ▶ Kupić lampkę i używać ją tylko do nauki.

Trenować intuicję

- ▶ **Pada pytanie?** Starać się dać w głowie odpowiedź.
- ▶ Dobra odpowiedź: +10 do samopoczucia!
- ▶ Zła odpowiedź: zastanowić się czemu polegliśmy.

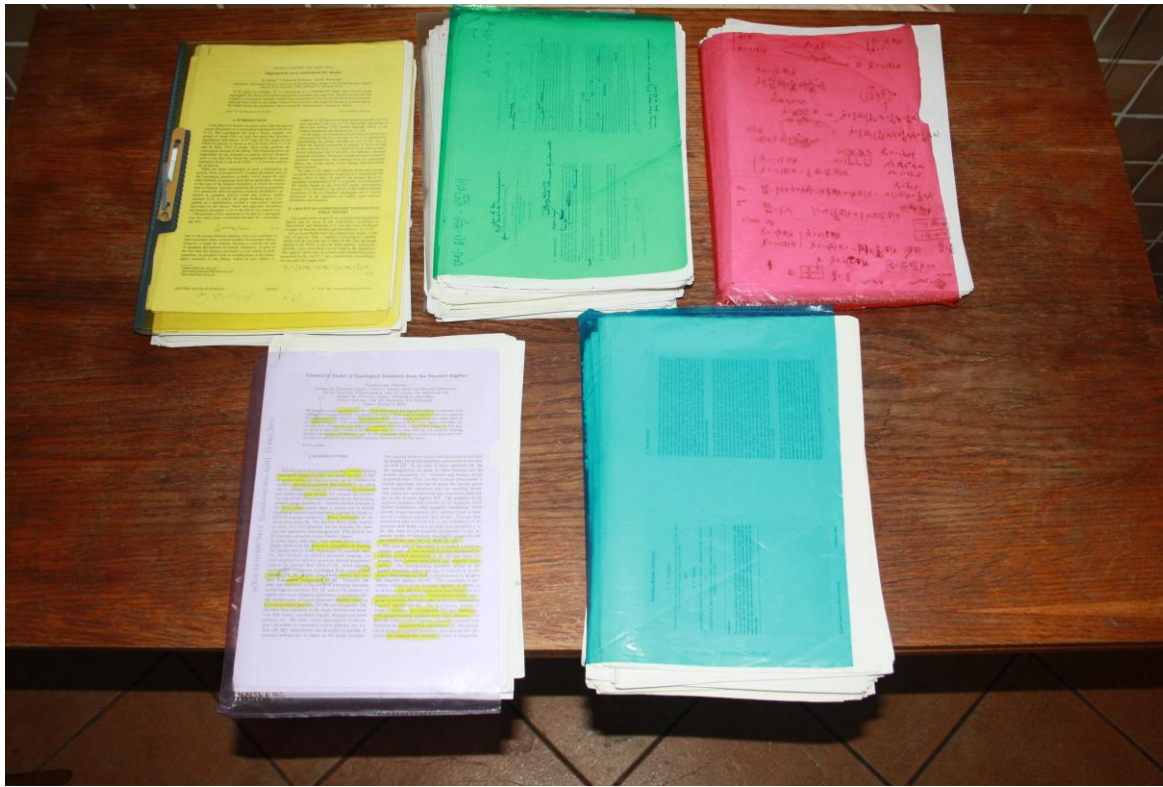
Egzaminy i sprawdziany

- ▶ Jeśli zarywać noce, to robić to **poza ostatnią nocą przed egzaminem!**
- ▶ 15% lepsze wyniki dzięki picu wody przed sprawdzianem.

*Tak jak od sportowców oczekujemy naturalnej sportowej rywalizacji,
tak od studentów na kierunkach ścisłych oczekujemy odruchowego:*

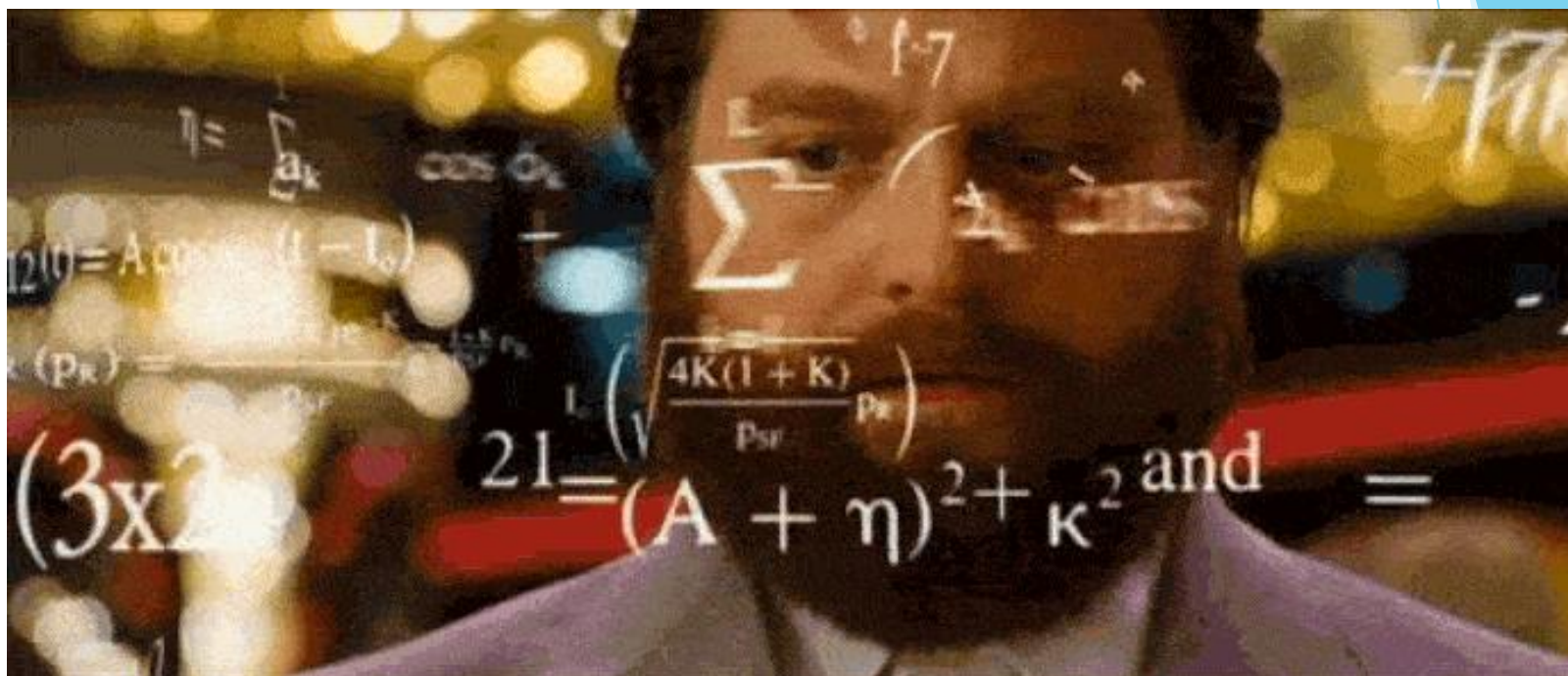
ROZWIĄZYWANIA PROBLEMÓW!





- ▶ Trzymaj **porządek** w papierach!
- ▶ Zakup całą **ryzę papieru!** (zwolnienie z ekologii na 3 lata)
- ▶ Dzięki Uwr - masz MS Teams, Worda, PowerPointa, Excella oraz folder **OneDrive (1 TB)**
- ▶ Zrób sobie podfoldery z: Wykład, Ćwiczenia
- ▶ Używaj GptChatów do pisania notatek.
- ▶ Współpracuj z ludźmi (ale nie odpisu!j)
- ▶ Nabywaj wartościowe umiejętności
(język obcy, jak się uczyć, jak prezentować rzeczy, etc.)

Prawidłowe rozwiązywanie zadań
powinno wyglądać tak...



ZADANIE: Dany jest ciąg (a_n) geometryczny $a_1=2, a_2=6, a_3, a_4, \dots$
Liczby $a_3, c, c/2$ w podanej kolejności tworzą ciąg arytmetyczny.
Oblicz c .

Ciąg geometryczny: $a_2 = a_1 q$
czyli $q = 6/2 = 3$

Zaś to daje automatycznie
 $a_3 = a_2 q = 6 * 3 = 18$

Ciąg arytmetyczny? $b_2 - b_1 = r$ oraz $b_3 - b_2 = r$

$c - 18 = r = c/2 - c$ (czyli możemy odpuścić r)
 $c - 18 = c/2 - c$

$c + c/2 = 18$
 $3/2 c = 18$
 $c = 12$

Sprawdzenie!!! Czy: $\{2, 6, 18\}$ jest geometryczny? Czy $\{18, 12, 6\}$ jest arytmetyczny?