

Algorytmy Geometryczne

Sprawozdanie – Ćwiczenie 1: Predykaty geometryczne

Krzysztof Chmielewski

Data wykonania: 07.10.2024

Data oddania: 21.10.2024

Spis treści

WSTĘP	1
CEL ĆWICZENIA	1
TEORIA	1
DANE TECHNICZNE.....	2
REALIZACJA ĆWICZENIA.....	3
WYNIKI I ANALIZA.....	4
ZBIÓR A.....	4
ZBIÓR B.....	5
ZBIÓR C.....	6
ZBIÓR D.....	7
ILUSTRACJE	8
WNIOSKI.....	9

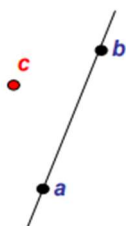
WSTĘP

CEL ĆWICZENIA

Ćwiczenie wprowadzające w zagadnienia geometrii obliczeniowej – implementacja podstawowych predykatów geometrycznych, przeprowadzenie testów, wizualizacja i opracowanie wyników.

TEORIA

Tematem ćwiczenia jest określenie po której stronie prostej (ab) leży punkt c:



Rys. 1 Ilustracja punktów i prostej

Przyjmując oznaczenie współrzędnych a, b, c w następujący sposób $a = (a_x, a_y)$; $b = (b_x, b_y)$; $c = (c_x, c_y)$, możemy wyznaczyć położenie punktu c względem prostej (ab) na dwa sposoby:

- (1) Wykorzystując wyznacznik macierzy 3×3 :

$$\det(a, b, c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}$$

- (2) Wykorzystując wyznacznik macierzy 2×2 :

$$\det(a, b, c) = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$$

Na podstawie znaku wartości wyznacznika jesteśmy w stanie określić położenie punktu c :

- $\det(a, b, c) > 0$; punkt leży na lewo od prostej (a, b)
- $\det(a, b, c) < 0$; punkt leży na prawo od prostej (a, b)
- $\det(a, b, c) = 0$; punkt leży na prostej (a, b)

DANE TECHNICZNE

Ćwiczenie zostało wykonane z użyciem narzędzia graficznego dostarczonego przez Koło Naukowe Bit (<https://github.com/aghbit/Algorytmy-Geometryczne>), które umożliwia wizualizację wykresów i kształtów geometrycznych wykorzystując różne biblioteki języka Python (np. `numpy`, `pandas`, `matplotlib`) oraz `Anacondę` do stworzenia odpowiedniego jądra dla `Jupyter Notebook`. Kod zawarty w pliku **`chmielewski_kod_1.ipynb`** został napisany w języku Python właśnie przy użyciu Jupyter Notebook.

Do utworzenia zbiorów A, B, C, D napisano odpowiednie funkcje wykorzystujące biblioteki `random` oraz zawartą w niej metodę `uniform()`, a także wykorzystano funkcje `sin`, `cos`, `pi` z biblioteki `math`. Przykłady wygenerowano dla różnych dokładności epsilon, oraz czterech różnych funkcji wyliczających położenie punktu względem prostej, z których dwie to funkcje biblioteczne. Obliczenia zostały przeprowadzone dla dwóch precyzji floata: float32 – 32 bitowej precyzji oraz float64 – 64 bitowej. Obliczenia zostały wykonane z wykorzystaniem sprzętu o następujących parametrach:

Komputer -> wirtualna maszyna VirtualBox:

- **Procesor:** AMD Ryzen 5 5600X 6 rdzeniowy, podstawowe taktowanie: 3,70 GHz -> w wirtualnej maszynie wykorzystano jedynie 5 z 12 wątków
- **Karta Graficzna:** NVIDIA GeForce RTX 3060 Ti 8GB GDDR6X
- **RAM:** 32GB DDR4 3000MHz CL16 -> w wirtualnej maszynie wykorzystano jedynie 16GB
- **OS:** Linux Ubuntu 24.04

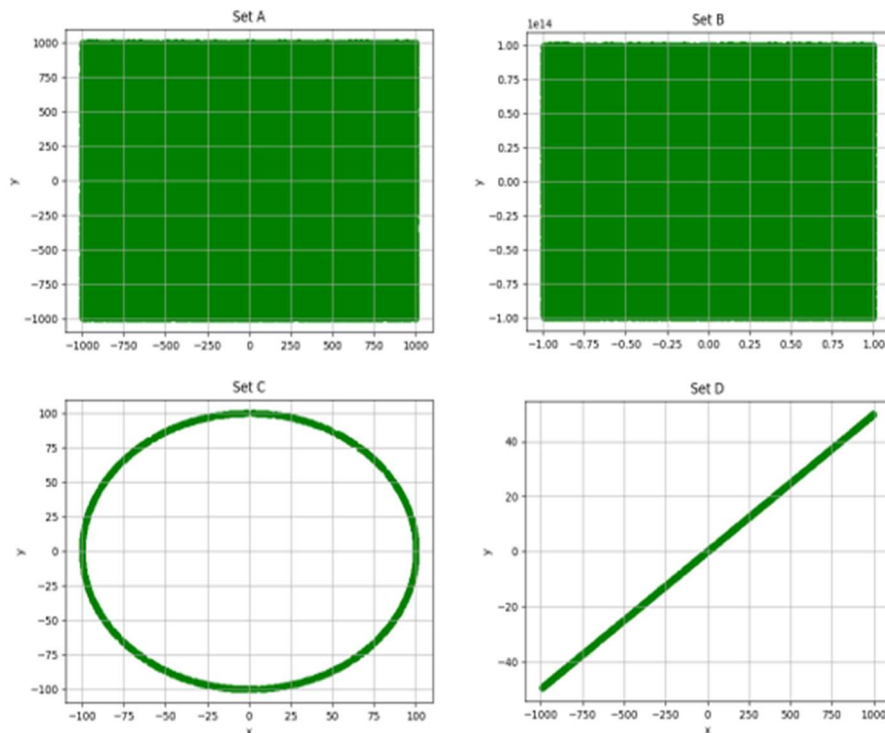
Laptop:

- **Procesor:** Intel Core i5-1235u 10 rdzeniowy, podstawowe taktowanie: 1,30 GHz
- **Karta Graficzna:** zintegrowana z procesorem Intel UHD
- **RAM:** 8GB DDR4 3200MHz
- **OS:** Linux Ubuntu 24.04

REALIZACJA ĆWICZENIA

Obliczenia były wykonywane na czterech różnych zbiorach: A, B, C i D:

- **A:** 10^5 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$
- **B:** 10^5 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$
- **C:** 1000 losowych punktów leżących na okręgu o środku $(0,0)$ i promieniu $R = 100$
- **D:** 1000 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ leżących na prostej wyznaczonej przez wektor (a, b) , przyjęto $a = [-1.0, 0.0]$, $b = [1.0, 0.1]$



Rys. 2 Ilustracja zbiorów A,B,C,D

Wyznaczniki obliczono według czterech funkcji, z których dwie były biblioteczne:

- `mat_det_3x3`
- `mat_det_3x3_lib`
- `mat_det_2x2`
- `mat_det_2x2_lib`

Funkcje z dopiskiem `_lib` wykorzystują biblioteczną funkcję `numpy` oraz zawarte w niej metody `linalg.det` do wyznaczenia wyznacznika macierzy 3x3 lub 2x2. Funkcje bez tego dopiska zostały napisane w oparciu o dowód matematyczny zawarty w pliku ***chmielewski_kod_1.ipynb***. Funkcja `categorize_points` zwraca trzy tablice, w których po kolei zawierają się punkty na lewo od prostej ab , na prostej oraz na prawo od prostej.

Obliczenia wykonano na każdym zbiorze z użyciem każdej z czterech funkcji do obliczania wyznacznika oraz z dokładnością zawartą w tablicy `epsilon`, w której mieści się pięć elementów wyznaczających pięć różnych dokładności. Obliczenia generowano z dokładnością w pierw `float64`, a potem powtórzono dla `float32`. Finalnie zatem wyznaczono $2 \cdot 4 \cdot 4 \cdot 5 = 160$ podziałów punktów. (Wartości dokładności `epsilon`: 0, 10^{-8} , 10^{-10} , 10^{-12} , 10^{-14})

WYNIKI I ANALIZA

Należy pamiętać, że funkcja `random.uniform()` nie ma ustawionego wcześniej `random.seed()`, więc w momencie uruchomienia obliczeń na nowo wyniki mogą się nieznacznie różnić. Na końcu sekcji umieszczono zakładkę ILUSTRACJE, w której pokazano jedno z wielu ilustracji dla każdego zbioru wygenerowane w pliku .ipybn.

ZBIÓR A

Tabela 1 – Rozłożenie punktów dla zbioru A

Rozłożenie punktów	Ilość punktów na lewo od prostej		Ilość punktów na prostej		Ilość punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-8}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-10}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-12}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-14}$	49818	49818	0	0	50182	50182
mat_det_2x2_lib						
$\varepsilon = 0$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-8}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-10}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-12}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-14}$	49818	49818	0	0	50182	50182
mat_det_3x3						
$\varepsilon = 0$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-8}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-10}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-12}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-14}$	49818	49818	0	0	50182	50182
mat_det_3x3_lib						
$\varepsilon = 0$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-8}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-10}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-12}$	49818	49818	0	0	50182	50182
$\varepsilon = 10^{-14}$	49818	49818	0	0	50182	50182

Wyniki dla różnych testów są jednoznaczne i pokazują te same wyniki dla każdej funkcji, również niezależnie od precyzji floata.

ZBIÓR B

Tabela 2 – Rozłożenie punktów dla zbioru B

Rozłożenie punktów	Ilość punktów na lewo od prostej		Ilość punktów na prostej		Ilość punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	0	49753	100000	6	0	50241
$\varepsilon = 10^{-8}$	0	49753	100000	6	0	50241
$\varepsilon = 10^{-10}$	0	49753	100000	6	0	50241
$\varepsilon = 10^{-12}$	0	49753	100000	6	0	50241
$\varepsilon = 10^{-14}$	0	49753	100000	6	0	50241
mat_det_2x2_lib						
$\varepsilon = 0$	6702	49752	86688	10	6610	50238
$\varepsilon = 10^{-8}$	6702	49752	86688	10	6610	50238
$\varepsilon = 10^{-10}$	6702	49752	86688	10	6610	50238
$\varepsilon = 10^{-12}$	6702	49752	86688	10	6610	50238
$\varepsilon = 10^{-14}$	6702	49752	86688	10	6610	50238
mat_det_3x3						
$\varepsilon = 0$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-8}$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-10}$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-12}$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-14}$	49756	49756	0	0	50244	50244
mat_det_3x3_lib						
$\varepsilon = 0$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-8}$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-10}$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-12}$	49756	49756	0	0	50244	50244
$\varepsilon = 10^{-14}$	49756	49756	0	0	50244	50244

W przypadku funkcji obliczających wyznacznik macierzy 2x2 widzimy, że precyzja floata diametralnie zmienia odczytywane wyniki co też można zobaczyć na rys. 4 w sekcji ILUSTRACJE poniżej. Funkcja biblioteczna odróżnia jeszcze pewne punkty na prawo i lewo od prostej, natomiast funkcja niebiblioteczna odczytuje wszystkie punkty tak jakby były one na prostej. Problem błędu precyzji zupełnie znika w przypadku funkcji liczących wyznacznik macierzy 3x3 – dla obu funkcji wyniki są takie same.

ZBIÓR C

Tabela 3 – Rozłożenie punktów dla zbioru C

Rozłożenie punktów	Ilość punktów na lewo od prostej		Ilość punktów na prostej		Ilość punktów na prawo od prostej	
	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	530	530	0	0	470	470
$\varepsilon = 10^{-8}$	530	530	0	0	470	470
$\varepsilon = 10^{-10}$	530	530	0	0	470	470
$\varepsilon = 10^{-12}$	530	530	0	0	470	470
$\varepsilon = 10^{-14}$	530	530	0	0	470	470
mat_det_2x2_lib						
$\varepsilon = 0$	530	530	0	0	470	470
$\varepsilon = 10^{-8}$	530	530	0	0	470	470
$\varepsilon = 10^{-10}$	530	530	0	0	470	470
$\varepsilon = 10^{-12}$	530	530	0	0	470	470
$\varepsilon = 10^{-14}$	530	530	0	0	470	470
mat_det_3x3						
$\varepsilon = 0$	530	530	0	0	470	470
$\varepsilon = 10^{-8}$	530	530	0	0	470	470
$\varepsilon = 10^{-10}$	530	530	0	0	470	470
$\varepsilon = 10^{-12}$	530	530	0	0	470	470
$\varepsilon = 10^{-14}$	530	530	0	0	470	470
mat_det_3x3_lib						
$\varepsilon = 0$	530	530	0	0	470	470
$\varepsilon = 10^{-8}$	530	530	0	0	470	470
$\varepsilon = 10^{-10}$	530	530	0	0	470	470
$\varepsilon = 10^{-12}$	530	530	0	0	470	470
$\varepsilon = 10^{-14}$	530	530	0	0	470	470

Wyniki dla różnych testów są jednoznaczne i pokazują te same wyniki dla każdej funkcji, również niezależnie od precyzji floata, jest to zapewne spowodowane małą ilością punktów w zbiorze.

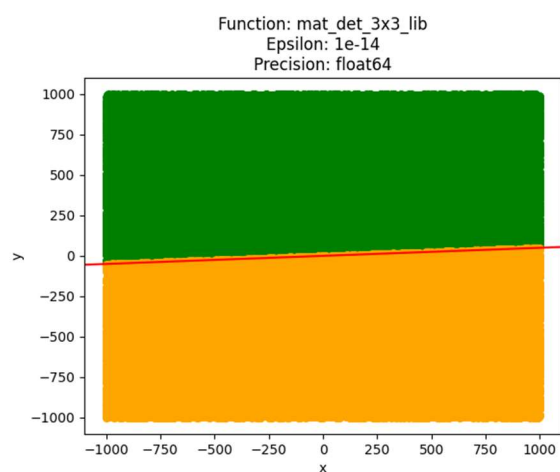
ZBIÓR D

Tabela 4 – Rozłożenie punktów dla zbioru D

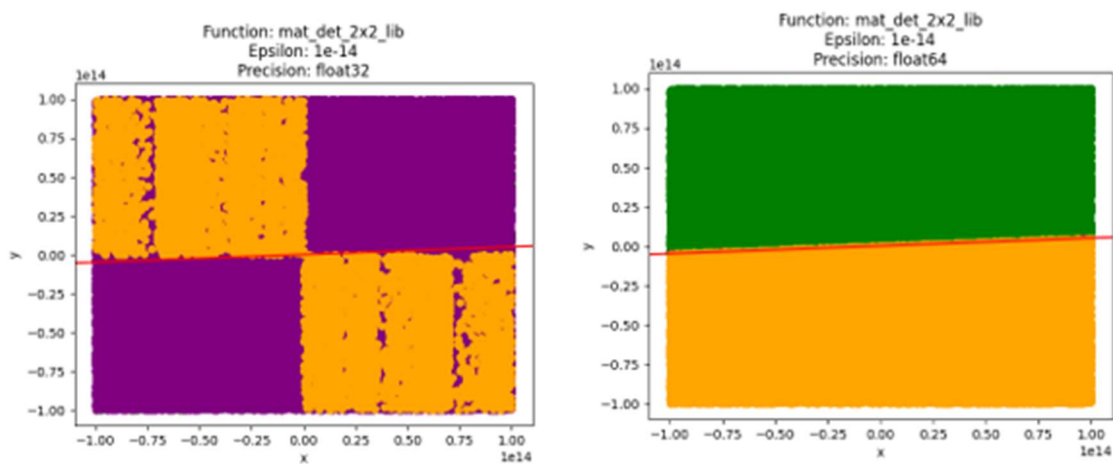
Rozłożenie punktów	Ilość punktów na lewo od prostej		Ilość punktów na prostej		Ilość punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	168	155	698	697	134	148
$\varepsilon = 10^{-8}$	168	0	698	1000	134	0
$\varepsilon = 10^{-10}$	168	0	698	1000	134	0
$\varepsilon = 10^{-12}$	168	90	698	833	134	77
$\varepsilon = 10^{-14}$	168	150	698	704	134	146
mat_det_2x2_lib						
$\varepsilon = 0$	486	165	0	668	514	167
$\varepsilon = 10^{-8}$	486	0	1	1000	513	0
$\varepsilon = 10^{-10}$	486	0	0	1000	514	0
$\varepsilon = 10^{-12}$	486	123	0	759	514	118
$\varepsilon = 10^{-14}$	486	160	0	677	514	163
mat_det_3x3						
$\varepsilon = 0$	305	170	390	442	305	388
$\varepsilon = 10^{-8}$	305	0	391	1000	304	0
$\varepsilon = 10^{-10}$	305	0	390	1000	305	0
$\varepsilon = 10^{-12}$	305	0	390	1000	305	0
$\varepsilon = 10^{-14}$	305	0	390	1000	305	0
mat_det_3x3_lib						
$\varepsilon = 0$	489	379	41	280	470	341
$\varepsilon = 10^{-8}$	419	0	171	1000	410	0
$\varepsilon = 10^{-10}$	419	0	170	1000	411	0
$\varepsilon = 10^{-12}$	419	0	170	1000	411	0
$\varepsilon = 10^{-14}$	425	22	150	886	425	92

W przypadku zbioru punktów D możemy zaobserwować różne rozkłady w ramach poszczególnych precyzji, funkcji i dokładności. Widać zależność dla precyzji float64, gdzie dla $\varepsilon = 10^{-8}$ oraz $\varepsilon = 10^{-12}$ wartość jest ta sama dla różnych funkcji – 1000 punktów na prostej i 0 na lewo i 0 na prawo. Widać także, że dla precyzji float32 zależności są w większości zachowane w przedziałach poszczególnych funkcji jednak różnią się one od pomiarów tą samą funkcją i dokładnością, ale precyzją float64 – na przykład jak pokazano w Tabeli 4 dla funkcji mat_det_3x3_lib, gdzie w ramach precyzji float32 wyniki w miarę się pokrywają, natomiast zupełnie się różnią od wyników precyzji float64.

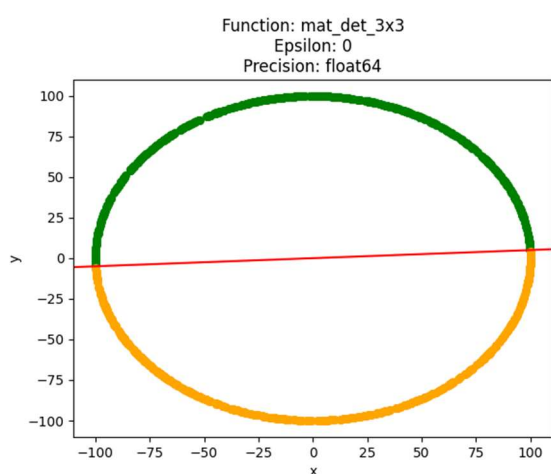
ILUSTRACJE



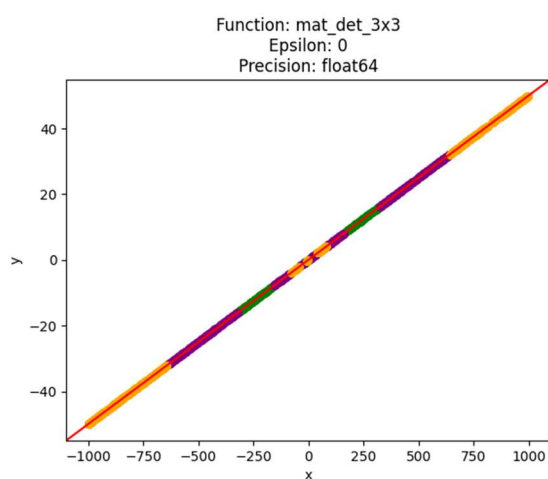
Rys. 3 Ilustracja podziału punktów ze zbioru A



Rys. 4 Porównanie precyzji float32 i float64 z obliczeń na zbiorze B



Rys. 5 Ilustracja podziału punktów ze zbioru C



Rys. 6 Ilustracja podziału punktów ze zbioru D

WNIOSKI

Ćwiczenie pokazuje jak różnie mogą być klasyfikowane punkty względem prostej przy użyciu różnych parametrów dokładności, ale także funkcji kategoryzującej. Patrząc na przykłady podanych zbiorów widać, że badany przedział punktów również ma znaczenie co widać przy porównaniu Tabeli 1 oraz Tabeli 2.

Analizując Tabelę 3 i 4 widzimy jednak, że pomimo tej samej ilości punktów, kategoryzacja wyniku zupełnie w inny sposób ze względu na kształt rysowanej figury. W przypadku zbioru C punkty były tak samo kwalifikowane niezależnie od precyzji floata, czego nie możemy powiedzieć o zbiorze D. W przypadku zbioru D możemy zaobserwować, że precyzja odgrywa znaczącą rolę w tej operacji, co widać w przypadku każdej z użytych funkcji. Patrząc kolumnami w Tabeli 4, gdy porównujemy kolumnę float32 i float64 dla każdej z kategorii (na lewo, na prostej, na prawo), widzimy, że trudno jest odnaleźć bliskie sobie wyniki, w przypadku którejkolwiek funkcji czy też dokładności epsilon.

Ćwiczenie pokazuje nam, że precyzja liczb zmiennoprzecinkowych w komputerze ma znaczenie przy różnych obliczeniach, to czy dana liczba zajmuje 32 czy 64 bity pamięci sprawia, że wyniki badań mogą skutkować różnymi wynikami.