Algorytmy Geometryczne

Sprawozdanie – Ćwiczenie 1: Predykaty geometryczne

Krzysztof Chmielewski

Data wykonania: 07.10.2024
Data oddania: 21.10.2024

Spis treści

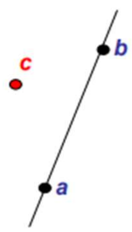
WSTĘP

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 3x3:

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

(2) Wykorzystując wyznacznik macierzy 2x2:

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

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 wykorzystujące 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 odpowiedne 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

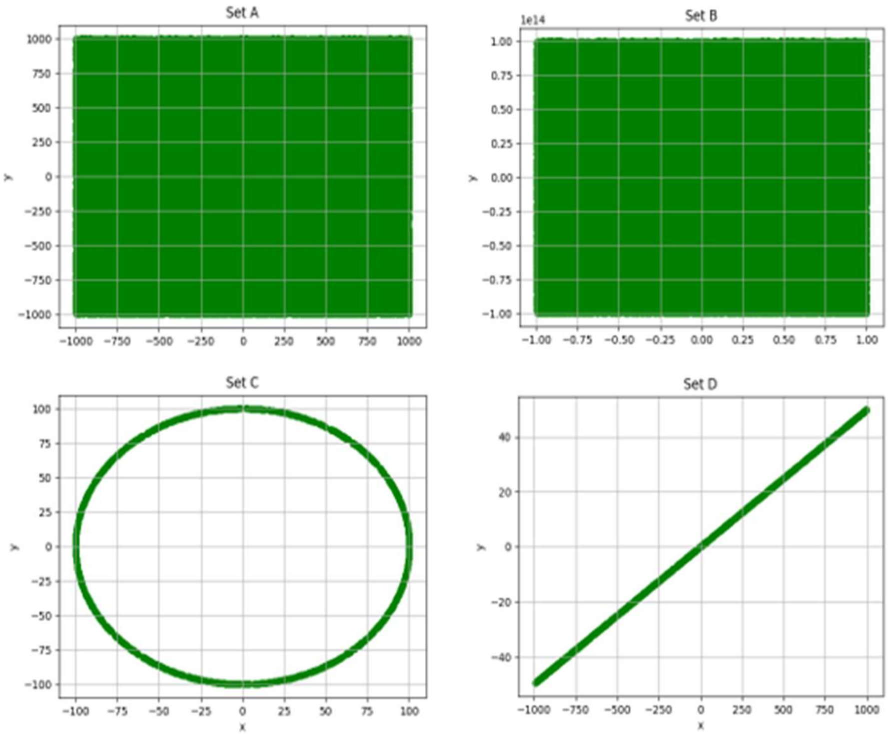
Laptop:

- **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 obliczano 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 wpierw 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 jedne z wielu ilustracji dla każdego zbioru wygenerowane w pliku .ipybn.

ZBIÓR A

Tabela 1 – Rozłożenie punktów dla zbioru A

Rozłożenie punktów	Liczba punktów na lewo od prostej		Liczba punktów na prostej		Liczba punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-8}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-10}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-12}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-14}$	49726	49726	0	0	50274	50274
mat_det_2x2_lib						
$\varepsilon = 0$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-8}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-10}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-12}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-14}$	49726	49726	0	0	50274	50274
mat_det_3x3						
$\varepsilon = 0$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-8}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-10}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-12}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-14}$	49726	49726	0	0	50274	50274
mat_det_3x3_lib						
$\varepsilon = 0$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-8}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-10}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-12}$	49726	49726	0	0	50274	50274
$\varepsilon = 10^{-14}$	49726	49726	0	0	50274	50274

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	Liczba punktów na lewo od prostej		Liczba punktów na prostej		Liczba punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	0	49724	100000	12	0	50264
$\varepsilon = 10^{-8}$	0	49724	100000	12	0	50264
$\varepsilon = 10^{-10}$	0	49724	100000	12	0	50264
$\varepsilon = 10^{-12}$	0	49724	100000	12	0	50264
$\varepsilon = 10^{-14}$	0	49724	100000	12	0	50264
mat_det_2x2_lib						
$\varepsilon = 0$	6606	49723	86707	15	6687	50262
$\varepsilon = 10^{-8}$	6606	49723	86707	15	6687	50262
$\varepsilon = 10^{-10}$	6606	49723	86707	15	6687	50262
$\varepsilon = 10^{-12}$	6606	49723	86707	15	6687	50262
$\varepsilon = 10^{-14}$	6606	49723	86707	15	6687	50262
mat_det_3x3						
$\varepsilon = 0$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-8}$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-10}$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-12}$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-14}$	49728	49728	0	0	50272	50272
mat_det_3x3_lib						
$\varepsilon = 0$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-8}$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-10}$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-12}$	49728	49728	0	0	50272	50272
$\varepsilon = 10^{-14}$	49728	49728	0	0	50272	50272

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.

Pomiędzy funkcjami 2x2 biblioteczną i niebiblioteczną widać różnie zakwalifikowane punkty, jeśli dobrze się nim przyjrzymy to na Rys. 3 na następnej stronie to obserwujemy, że funkcja biblioteczna wybiera 3 punkty więcej w tym przypadku (seed(10)) oraz 2 punkty zostały różnorako zakwalifikowane niż w przypadku funkcji nie bibliotecznej.

```
[(-93645978709503.69, -4656060442924.547), (-83136244038359.05, -4162168861049.8906), (-76303665985317.33, -3821869552245.07
8), (-59483778507533.305, -2980456432453.9688), (24673335684991.97, 1235721452366.6562), (30631769140521.28, 1532915316534.67
19), (77729283103875.38, 3898726486226.4062), (82328980108673.56, 4103439239872.2188), (88511774211733.6, 4406365304678.328),
(92476321324655.03, 4619787802172.9375), (92802254139604.88, 4610008946673.1875), (95162559589823.47, 4739907427797.625)]
Obliczono dla zbioru B z wykorzystaniem funkcji mat_det_2x2
Epsilon: 1e-14
Precyzja: float64
Ilość punktów po lewej od prostej: 49724
Ilość punktów na prostej: 12
Ilość punktów na prawo od prostej: 50264

[(-93645978709503.69, -4656060442924.547), (-83136244038359.05, -4162168861049.8906), (-76303665985317.33, -3821869552245.07
8), (-59483778507533.305, -2980456432453.9688), (-47085553664635.78, -2362627664444.6875), (24673335684991.97, 1235721452366.
6562), (30631769140521.28, 1532915316534.6719), (49584990269393.78, 2488265015923.1562), (77729283103875.38, 3898726486226.40
62), (88511774211733.6, 4406365304678.328), (90940274514662.53, 4513030254606.75), (92476321324655.03, 4619787802172.9375),
(92802254139604.88, 4610008946673.1875), (93082699043317.25, 4620829373685.625), (95162559589823.47, 4739907427797.625)]
Obliczono dla zbioru B z wykorzystaniem funkcji mat_det_2x2_lib
Epsilon: 0
Precyzja: float64
Ilość punktów po lewej od prostej: 49723
Ilość punktów na prostej: 15
Ilość punktów na prawo od prostej: 50262
```

Rys. 3 Porównanie zakwalifikowanych punktów

ZBIÓR C

Tabela 3 – Rozłożenie punktów dla zbioru C

Rozłożenie punktów	Liczba punktów na lewo od prostej		Liczba punktów na prostej		Liczba punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	498	498	0	0	502	502
$\varepsilon = 10^{-8}$	498	498	0	0	502	502
$\varepsilon = 10^{-10}$	498	498	0	0	502	502
$\varepsilon = 10^{-12}$	498	498	0	0	502	502
$\varepsilon = 10^{-14}$	498	498	0	0	502	502
mat_det_2x2_lib						
$\varepsilon = 0$	498	498	0	0	502	502
$\varepsilon = 10^{-8}$	498	498	0	0	502	502
$\varepsilon = 10^{-10}$	498	498	0	0	502	502
$\varepsilon = 10^{-12}$	498	498	0	0	502	502
$\varepsilon = 10^{-14}$	498	498	0	0	502	502
mat_det_3x3						
$\varepsilon = 0$	498	498	0	0	502	502
$\varepsilon = 10^{-8}$	498	498	0	0	502	502
$\varepsilon = 10^{-10}$	498	498	0	0	502	502
$\varepsilon = 10^{-12}$	498	498	0	0	502	502
$\varepsilon = 10^{-14}$	498	498	0	0	502	502
mat_det_3x3_lib						
$\varepsilon = 0$	498	498	0	0	502	502
$\varepsilon = 10^{-8}$	498	498	0	0	502	502
$\varepsilon = 10^{-10}$	498	498	0	0	502	502
$\varepsilon = 10^{-12}$	498	498	0	0	502	502
$\varepsilon = 10^{-14}$	498	498	0	0	502	502

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.

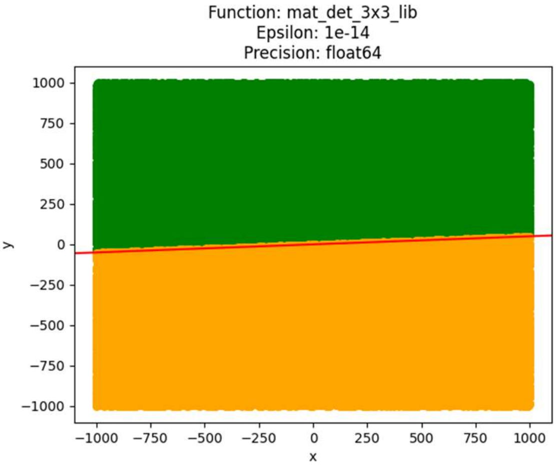
ZBIÓR D

Tabela 4 – Rozłożenie punktów dla zbioru D

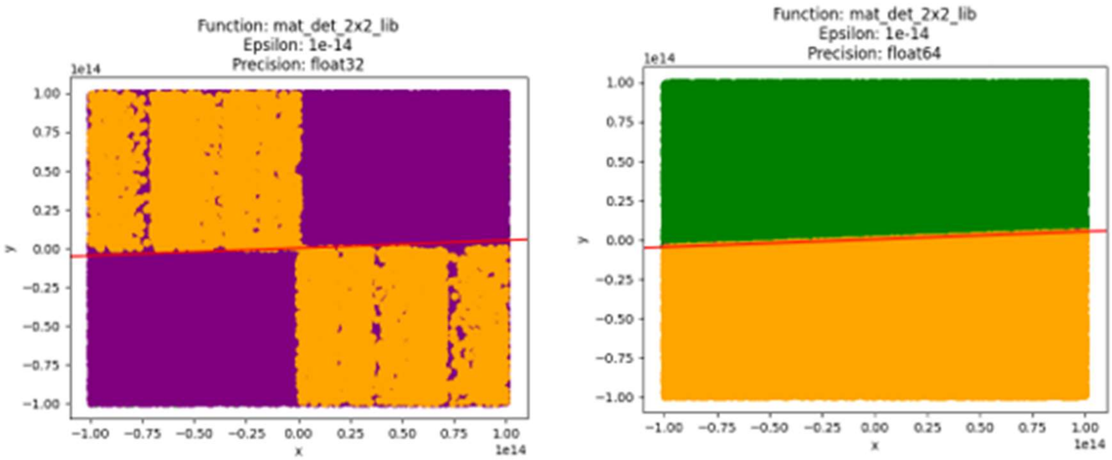
Rozłożenie punktów	Liczba punktów na lewo od prostej		Liczba punktów na prostej		Liczba punktów na prawo od prostej	
Precyzja	float32	float64	float32	float64	float32	float64
mat_det_2x2						
$\varepsilon = 0$	161	159	683	699	156	142
$\varepsilon = 10^{-8}$	160	0	685	1000	155	0
$\varepsilon = 10^{-10}$	161	0	683	1000	156	0
$\varepsilon = 10^{-12}$	161	86	683	831	156	83
$\varepsilon = 10^{-14}$	161	153	683	708	156	83
mat_det_2x2_lib						
$\varepsilon = 0$	506	175	0	676	494	149
$\varepsilon = 10^{-8}$	502	0	5	1000	493	0
$\varepsilon = 10^{-10}$	506	0	0	1000	494	0
$\varepsilon = 10^{-12}$	506	115	0	779	494	106
$\varepsilon = 10^{-14}$	506	165	0	697	494	138
mat_det_3x3						
$\varepsilon = 0$	323	159	377	422	300	419
$\varepsilon = 10^{-8}$	322	0	379	1000	299	0
$\varepsilon = 10^{-10}$	323	0	377	1000	300	0
$\varepsilon = 10^{-12}$	323	0	377	1000	300	0
$\varepsilon = 10^{-14}$	323	0	377	1000	300	0
mat_det_3x3_lib						
$\varepsilon = 0$	480	358	52	295	468	347
$\varepsilon = 10^{-8}$	406	0	180	1000	414	0
$\varepsilon = 10^{-10}$	409	0	177	1000	414	0
$\varepsilon = 10^{-12}$	409	0	177	1000	414	0
$\varepsilon = 10^{-14}$	419	16	154	887	427	97

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.

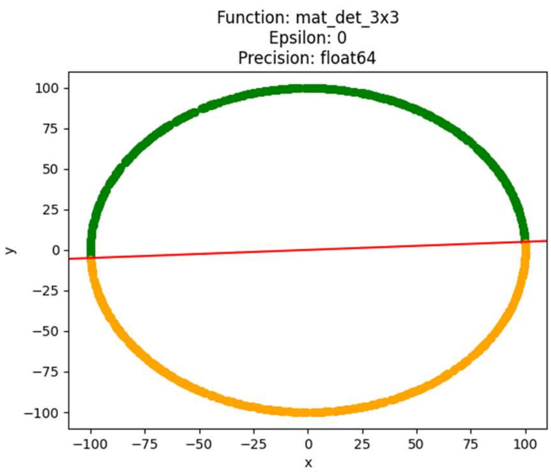
ILUSTRACJE



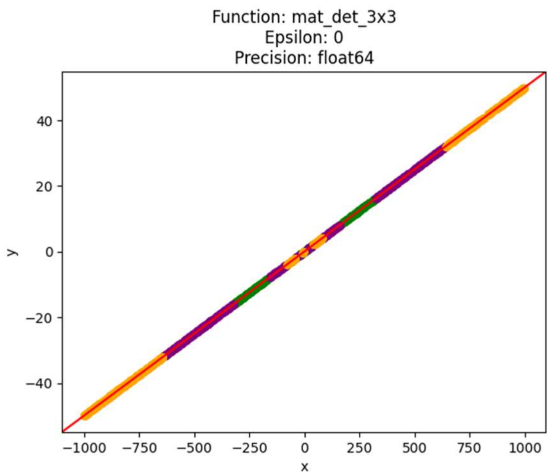
Rys. 4 Ilustracja podziału punktów ze zbioru A



Rys. 5 Porównanie precyzji float32 i float64 z obliczeń na zbiorze B



Rys. 6 Ilustracja podziału punktów ze zbioru C



Rys. 7 Ilustracja podziału punktów ze zbioru D

Uwagi do ilustracji: Punkty zielone – na lewo od prostej, punkty purpurowe – na prostej, punkty pomarańczowe – na prawo od prostej.

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 wynikła 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.