

Sprawozdanie z ćwiczenia z Algorytmów Geometrycznych

Mikołaj Gawel

13 października 2025

Ćwiczenie nr 1

Dane podstawowe

- **Imię i nazwisko:** Mikołaj Gawel
- **Data:** 13 października 2025
- **Tytuł ćwiczenia:** Algorytmy geometryczne – ćwiczenie 1
- **Numer ćwiczenia:** 1

1 Dane techniczne

- **Język programowania:** Python 3.13.5
- **Środowisko uruchomieniowe:** Jupyter Notebook
- **Biblioteki:** numpy, pandas oraz autorskie narzędzie wizualizacji opracowane przez Koło Naukowe BIT
- **System operacyjny:** Ubuntu 24.04 LTS, jądro Linux 6.14.0-33-generic
- **Architektura komputera:** 64-bitowa
- **Procesor:** AMD Ryzen 7 7735HS with Radeon Graphics (8 rdzeni / 16 wątków, max 4.83 GHz)
- **Tryby pracy CPU:** 32-bit i 64-bit
- **Precyzja obliczeń:** obliczenia wykonywano dla typów `float32` i `float64` (wybrane testy porównawcze dla różnych tolerancji ε)

2 Generowanie zbiorów i wizualizacja

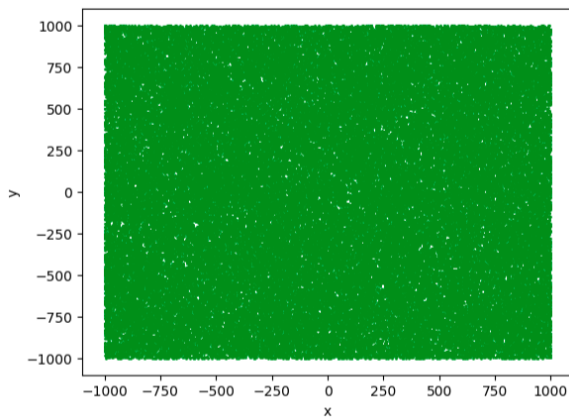
W ćwiczeniu przygotowano cztery zbiory punktów w przestrzeni 2D (typ danych: `double`). Do generowania danych wykorzystano bibliotekę `NumPy`, a dla zachowania powtarzalności wyników ustawiono stałe ziarno generatora (`np.random.seed`). Wszystkie punkty rozmieszczono równomiernie w zadanych przedziałach lub na zadanych krzywych.

1. **Zbiór a:** 10^5 losowych punktów o współrzędnych (x, y) z przedziału $[-1000, 1000]$, wygenerowanych funkcją `np.random.uniform`.
2. **Zbiór b:** 10^5 losowych punktów o współrzędnych (x, y) z przedziału $[-10^{14}, 10^{14}]$, służący do analizy wpływu dużych wartości współrzędnych na precyzję obliczeń.
3. **Zbiór c:** 1000 punktów leżących na okręgu o środku $(0, 0)$ i promieniu $R = 100$, rozmieszczonych równomiernie według parametrycznego równania:

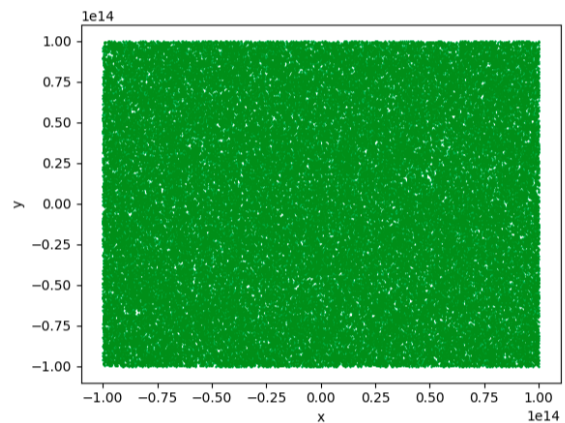
$$x = R \cos(\theta), \quad y = R \sin(\theta), \quad \theta \in [0, 2\pi)$$

4. **Zbiór d:** 1000 punktów leżących na prostej wyznaczonej przez odcinek $a = [-1.0, 0.0]$, $b = [1.0, 0.1]$, generowanych na podstawie równania parametrycznego:

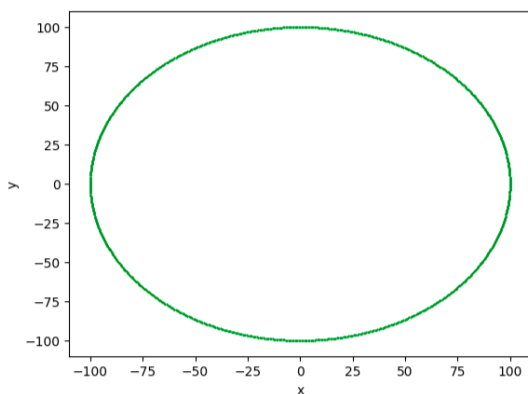
$$x = a_x + t(b_x - a_x), \quad y = a_y + t(b_y - a_y), \quad t \in [0, 1]$$



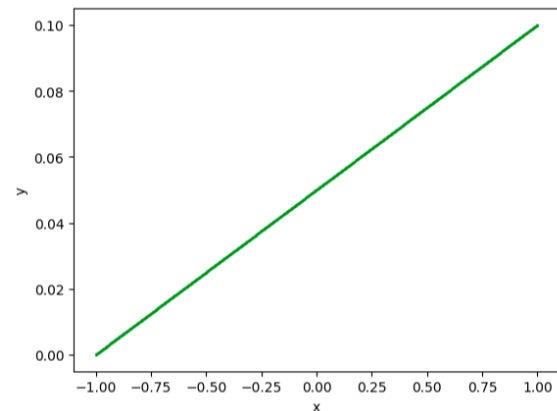
(a) Zbiór a: losowe punkty $[-1000, 1000]$



(b) Zbiór b: losowe punkty $[-10^{14}, 10^{14}]$



(c) Zbiór c: punkty na okręgu $R = 100$



(d) Zbiór d: punkty na prostej ab

Rysunek 1: Wizualizacja czterech zbiorów punktów wygenerowanych w przestrzeni 2D.

3 Klasyfikacja punktów względem prostej z użyciem wyznaczników i tolerancji

3.1 Wyznaczniki do klasyfikacji punktów

Do określenia, po której stronie prostej ab znajduje się punkt c , wykorzystano wartość wyznacznika macierzy w dwóch postaciach:

2×2:

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

3×3:

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

gdzie:

$$a = (a_x, a_y), \quad b = (b_x, b_y), \quad c = (c_x, c_y)$$

Interpretacja znaku wyznacznika:

$$\begin{cases} \det(a, b, c) < 0 & \Rightarrow c \text{ po prawej stronie prostej } ab, \\ \det(a, b, c) = 0 & \Rightarrow c \text{ leży na prostej } ab, \\ \det(a, b, c) > 0 & \Rightarrow c \text{ po lewej stronie prostej } ab. \end{cases}$$

Wyznaczniki obliczono dwiema metodami:

- implementacją własną opartą na definicji macierzy,
- funkcją biblioteczną `numpy.linalg.det`.

3.2 Tolerancja i współliniowość

Dla bardzo małych wartości wyznacznika przyjęto próg tolerancji ε , aby uniknąć problemów z precyzją obliczeń zmiennoprzecinkowych. Poniższa tabela przedstawia wartości ε , które będą analizowane:

Wartość ε
0
10^{-14}
10^{-12}
10^{-9}
10^{-6}

Tabela 1: Przyjęte wartości progu tolerancji ε dla analizy współliniowości punktów.

Interpretacja:

$$|\det(a, b, c)| < \varepsilon \Rightarrow c \text{ traktowany jako współliniowy z } a \text{ i } b.$$

Porównano wpływ precyzji (`float32` i `float64`) oraz wartości tolerancji ε na poprawność klasyfikacji punktów.

4 Analiza klasyfikacji punktów

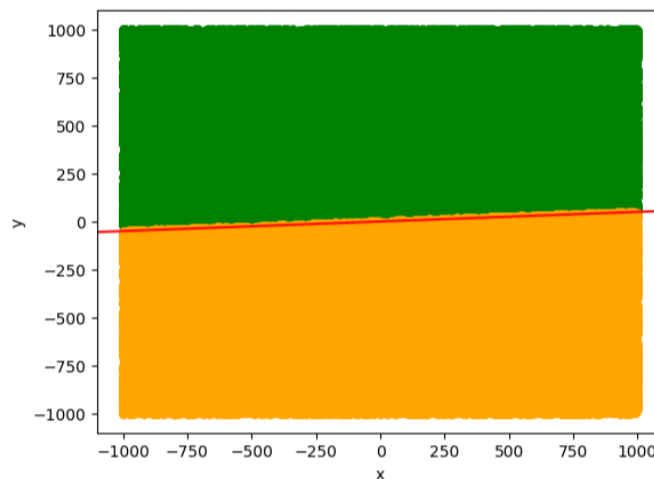
Na rysunkach punkty znajdujące się po lewej stronie prostej ab są zielone, punkty na prostej są fioletowe, a punkty po prawej stronie są pomarańczowe.

4.1 Analiza zbioru A

Dla zbioru A (10^5 losowych punktów w przedziale $[-1000, 1000]$) przeprowadzono analizę klasyfikacji punktów względem prostej ab przy różnych wartościach tolerancji ε i precyzji zmiennoprzecinkowej. Okazało się że niezależnie od sposobu liczenia wyznacznika, tolerancji ε i precyzji zmiennoprzecinkowej wynik rozłożenia punktów był ten sam.

Zbiór	po lewej od prostej	na prostej	po prawej od prostej
A	50207	0	49793

Tabela 2: Klasyfikacja punktów zbioru A względem prostej ab . Wyniki są identyczne dla wszystkich badanych wartości ε , precyzji zmiennoprzecinkowej oraz metod wyznacznika (2x2, 3x3, własna i biblioteczna).



Rysunek 2: Punkty zbioru A względem prostej ab , metoda `mat_det_3x3`, $\varepsilon = 10^{-12}$.

4.2 Analiza zbioru B

Dla zbioru B (10^5 losowych punktów w przedziale $[-10^{14}, 10^{14}]$) przeprowadzono analizę klasyfikacji punktów względem prostej ab przy różnych wartościach tolerancji ε i precyzji zmiennoprzecinkowej.

Analiza klasyfikacji punktów zbioru B pokazuje, że wyniki zależą od typu precyzji zmiennoprzecinkowej oraz metody obliczania wyznacznika. Dla `float64` wszystkie metody (2x2, 2x2_lib, 3x3, 3x3_lib) dają spójne wyniki dla wszystkich wartości ε .

Dla `float32` metoda 2x2 wykazuje silną wrażliwość na precyzję i epsilon – przy najmniejszych epsilonch wiele punktów klasyfikowanych jest niepoprawnie jako współliniowe. Metoda 2x2_lib poprawia dokładność, ale nadal występują pewne przesunięcia. Z kolei metoda 3x3 jest stabilna zarówno dla `float64`, jak i `float32`, zapewniając spójne wyniki niezależnie od epsilon.

Wybór odpowiedniej metody wyznacznika i precyzji jest zatem kluczowy, aby poprawnie identyfikować punkty współliniowe i minimalizować błędną klasyfikację.

Metoda	Float	Epsilon	po lewej	na prostej	po prawej
mat_det_2x2	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	50026	8	49966
mat_det_2x2_lib	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	50026	7	49967
mat_det_3x3	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	50028	0	49972
mat_det_3x3_lib	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	50028	0	49972
mat_det_2x2	float32	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	0	100000	0
mat_det_2x2_lib	float32	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	6707	86608	6685
mat_det_3x3	float32	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	50028	0	49972
mat_det_3x3_lib	float32	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}, 0$	50028	0	49972

Tabela 3. Klasyfikacja punktów zbioru B względem prostej ab dla różnych metod wyznacznika i typów precyzji zmiennoprzecinkowej.

X	Y
-92808447832351.56	-4638961148760.672
-98864619056286.16	-4909449065178.672
58906427054712.25	2943935264015.4688
-86286438458100.44	-4296925450899.4062
-80149683760744.58	-4032111314248.6562
-75061230702931.42	-3763403346938.2812
-47018957292283.914	-2360617191785.9688
-50977113945005.305	-2551653380912.3906

Tabela 4. Punkty klasyfikowane jako „na linii” przez metodę `mat_det_2x2` (float64) dla wszystkich wartości ε .

X	Y
-92808447832351.56	-4638961148760.672
-98864619056286.16	-4909449065178.672
-43980941182258.52	-2206180678847.0312
58906427054712.25	2943935264015.4688
-59717800164392.54	-2988219060924.4375
-75061230702931.42	-3763403346938.2812
-50977113945005.305	-2551653380912.3906

Tabela 5. Punkty klasyfikowane jako „na linii” przez metodę `mat_det_2x2_lib` (float64) dla wszystkich wartości ε .

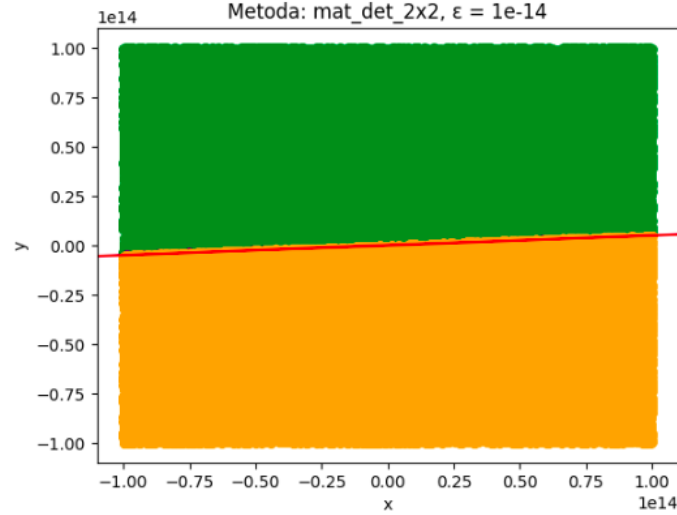
Analiza poprawności współliniowości

Niech prosta przechodzi przez punkty $A(x_a, y_a)$ i $B(x_b, y_b)$. Punkt $P(x, y)$ leży dokładnie na prostej, jeśli:

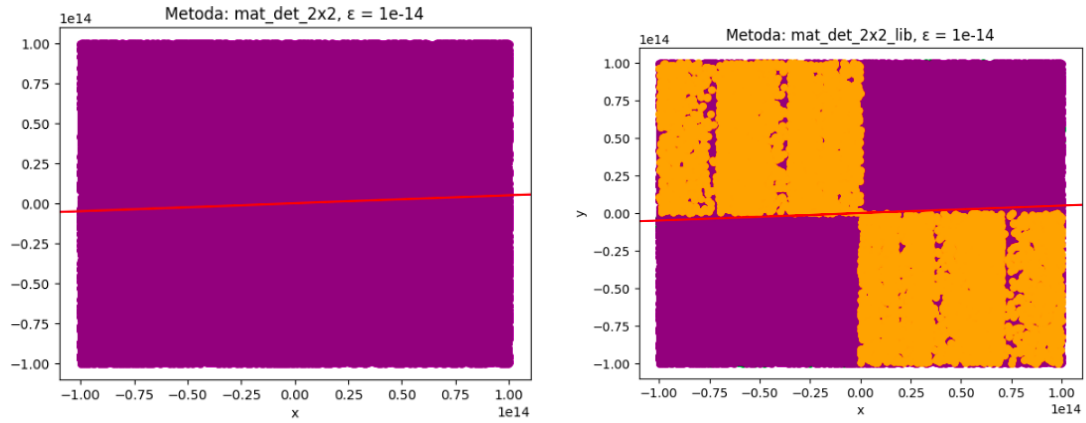
$$\frac{y - y_a}{y_b - y_a} = \frac{x - x_a}{x_b - x_a}.$$

Dla podanych punktów widać, że stosunek $(y - y_a)/(y_b - y_a)$ różni się od $(x - x_a)/(x_b - x_a)$ nawet w najmniejszym odróżnialnym miejscu w `float64`. Oznacza to, że żaden z punktów nie leży dokładnie na prostej AB — metoda `2x2` oraz `2x2_lib` klasyfikuje je jako współliniowe tylko w przybliżeniu, w granicy tolerancji ε .

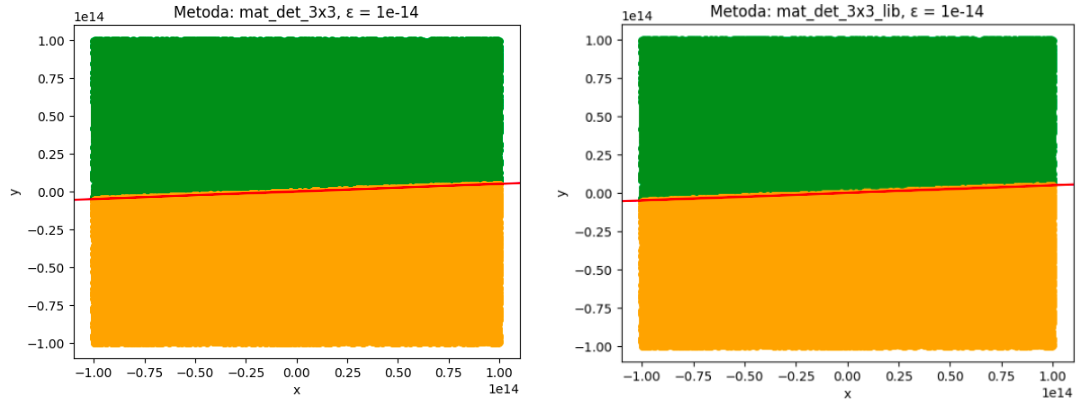
Wnioski: metoda klasyfikuje punkty jako „na linii”, ale ze względu na ograniczoną precyzję zmiennoprzecinkową i ogromne wartości współrzędnych, w rzeczywistości żaden z punktów nie spełnia dokładnie równania prostej.



Rysunek 3: Graficzna reprezentacja punktów zbioru B względem prostej ab dla obliczeń w precyzji `float64`



Rysunek 4: Graficzna reprezentacja punktów zbioru B względem prostej ab dla metod `mat_det_2x2` (po lewej) i `mat_det_2x2_lib` (po prawej) w precyzji `float32`, $\varepsilon = 10^{-14}$.



Rysunek 5: Graficzna reprezentacja punktów zbioru B względem prostej ab dla metod `mat_det_3x3` (po lewej) i `mat_det_3x3_lib` (po prawej) w precyzji `float32`, $\varepsilon = 10^{-14}$.

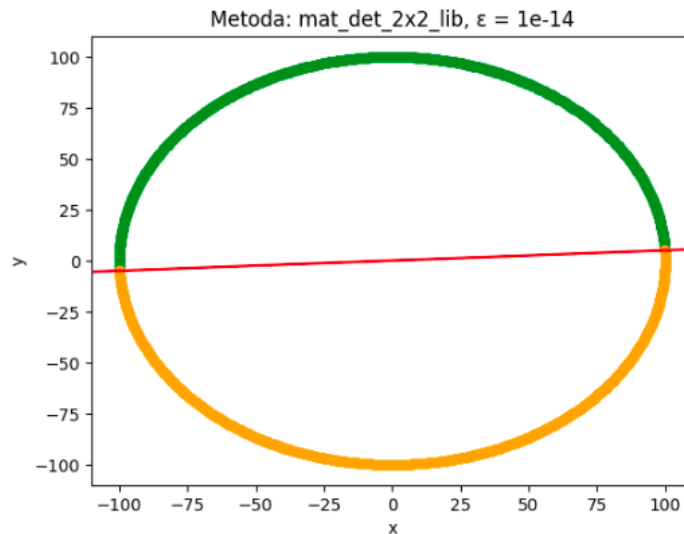
4.3 Analiza zbioru C

Dla zbioru C (1000 punktów rozmieszczonych równomiernie na okręgu o promieniu $R = 100$) przeprowadzono analizę klasyfikacji punktów względem prostej ab przy różnych wartościach tolerancji ε oraz typach precyzji zmiennoprzecinkowej (`float32` i `float64`).

Wyniki pokazały pełną spójność wszystkich metod i precyzji: wszystkie punkty zostały zaklasyfikowane w identyczny sposób niezależnie od wartości ε oraz użytej metody wyznacznika.

Kategoria	Liczba punktów
Po lewej stronie prostej	499
Na linii	0
Po prawej stronie prostej	501

Tabela 6: Podsumowanie klasyfikacji punktów zbioru C względem prostej ab . Wyniki są identyczne dla wszystkich metod wyznacznika, typów precyzji oraz wartości ε .



Rysunek 6: Graficzna reprezentacja punktów zbioru C względem prostej ab .

Wnioski:

- Wszystkie metody wyznacznika (2x2, 2x2_lib, 3x3, 3x3_lib) oraz typy precyzji (float32 i float64) dają identyczną klasyfikację punktów.
- Żaden punkt nie został sklasyfikowany jako współliniowy (na linii), co wynika z faktu, że zbiór C leży na okręgu, a prosta ab nie pokrywa żadnego punktu dokładnie.
- Analiza pokazuje, że przy uporządkowanych i „małych” współrzędnych (tu $x, y \in [-100, 100]$) wszystkie metody są stabilne i spójne niezależnie od wybranej tolerancji ε .

4.4 Analiza zbioru D

Zbiór D został skonstruowany w taki sposób, że wszystkie punkty powinny znajdować się dokładnie na prostej ab . Oznacza to, że idealny wynik klasyfikacji to:

$$\text{po lewej} = 0, \quad \text{na prostej} = 1000, \quad \text{po prawej} = 0.$$

W praktyce jednak uzyskane rezultaty różnią się w zależności od zastosowanej metody obliczania wyznacznika, precyzji typu zmiennoprzecinkowego oraz wartości progu ε . W tabelach 7 i 8 przedstawiono wyniki klasyfikacji punktów dla każdej z metod.

Metoda	Float	Epsilon	po lewej	na prostej	po prawej
mat_det_2x2	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}$ 0	0 340	1000 303	0 357
mat_det_2x2_lib	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}$ 0	0 345	1000 305	0 350
mat_det_3x3	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}$ 0	0 289	1000 431	0 280
mat_det_3x3_lib	float64	$10^{-14}, 10^{-12}, 10^{-9}, 10^{-6}$ 0	0 172	1000 828	0 0

Tabela 7. Klasyfikacja punktów zbioru D względem prostej ab dla metod wyznacznika przy precyzji float64.

Metoda	Float	Epsilon	po lewej	na prostej	po prawej
mat_det_2x2	float32	$10^{-14}, 10^{-12}, 10^{-9}, 0$ 10^{-6}	241 0	281 1000	478 0
mat_det_2x2_lib	float32	$10^{-14}, 10^{-12}, 10^{-9}, 0$ 10^{-6}	372 0	2 1000	626 0
mat_det_3x3	float32	$10^{-14}, 10^{-12}, 10^{-9}, 0$ 10^{-6}	186 0	395 1000	419 0
mat_det_3x3_lib	float32	$10^{-14}, 10^{-12}, 10^{-9}, 0$ 10^{-6}	424 0	156 1000	420 0

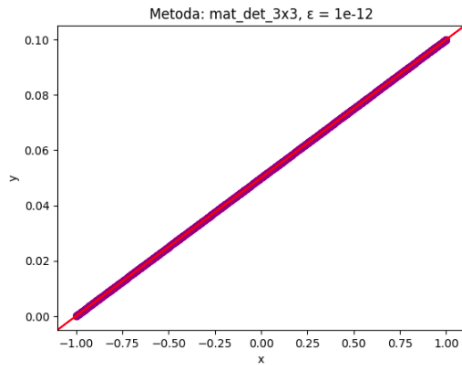
Tabela 8. Klasyfikacja punktów zbioru D względem prostej ab dla metod wyznacznika przy precyzji float32.

Wnioski:

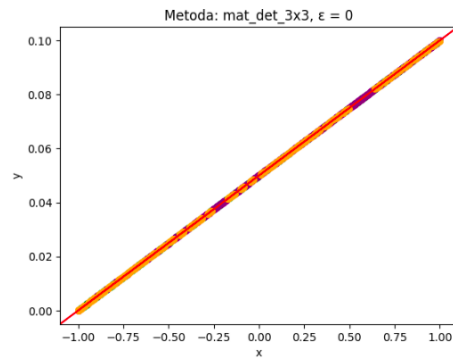
- Dla typu `float64` klasyfikacja jest bardzo stabilna — już dla niewielkich wartości ε (np. 10^{-12}) wszystkie punkty są poprawnie klasyfikowane jako leżące na prostej.
- Przy $\varepsilon = 0$ pojawiają się błędy klasyfikacji spowodowane ograniczoną precyzją reprezentacji zmiennoprzecinkowej — część punktów błędnie przypisywana jest „na lewo” lub „na prawo”.
- W przypadku typu `float32` stabilność obliczeń znacząco spada. Dla małych wartości ε uzyskano duże odchylenia od wyniku idealnego, a dopiero przy $\varepsilon = 10^{-6}$ klasyfikacja jest w pełni poprawna.
- Różnice między metodami `mat_det_2x2`, `mat_det_3x3` oraz ich wersjami `_lib` są niewielkie dla `float64`, lecz przy `float32` biblioteczne implementacje wykazują większą niestabilność, co sugeruje większą wrażliwość na błędy zaokrągleń.

Wizualizacja wyników klasyfikacji:

Precyzja float64

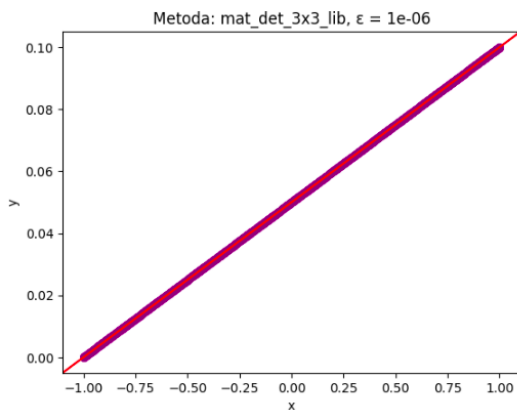


(a) Metoda `mat_det_3x3`, $\varepsilon = 10^{-12}$, `float64`.

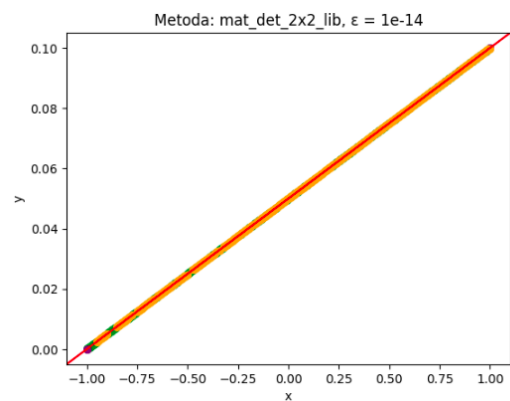


(b) Metoda `mat_det_3x3`, $\varepsilon = 0$, `float64`.

Precyzja float32



(c) Metoda `mat_det_3x3_lib`, $\varepsilon = 10^{-6}$, `float32`.



(d) Metoda `mat_det_2x2_lib`, $\varepsilon = 10^{-14}$, `float32`.

Rysunek 7: Graficzne przedstawienie klasyfikacji punktów zbioru D względem prostej ab dla wybranych metod, wartości ε oraz precyzji numerycznej.

5 Wnioski

1. Tolerancja ε wpływa na klasyfikację współliniowości. Optymalne wartości: 10^{-6} (`float32`), 10^{-12} (`float64`).
2. Precyzja `float64` zapewnia stabilne wyniki nawet dla małych ε , `float32` wymaga większej tolerancji.
3. Metoda 3x3 (własna i biblioteczna) jest najbardziej odporna na błędy numeryczne, zwłaszcza przy dużych współrzędnych.
4. Zbiory o małych, uporządkowanych współrzędnych klasyfikowane są poprawnie przez wszystkie metody, natomiast duże wartości współrzędnych wymagają starannego doboru ε i precyzji.

Podsumowując, stabilna klasyfikacja punktów w 2D wymaga stosowania `float64` oraz odpowiednio dobranej metody wyznacznika i progu tolerancji.