

Aleksander Jóźwik

gr. 2, Pon. godz. 15:00 A

Data wykonania: 07.10.2024

Data oddania: 24.10.2024

Algorytmy Geometryczne - laboratorium 1

Predykaty geometryczne

1. Dane techniczne

- System operacyjny: Fedora Linux (x86-64)
- Procesor: Intel Core i5-8350U (1.70 - 3.60 GHz)
- Pamięć RAM: 16GB (2133 MHz)
- Środowisko: Jupyter Notebook
- Język: Python 3.9.20

W realizacji ćwiczenia wykorzystano biblioteki: *numpy*, *pandas* oraz narzędzie wizualizacji stworzone przez koło naukowe *BIT*.

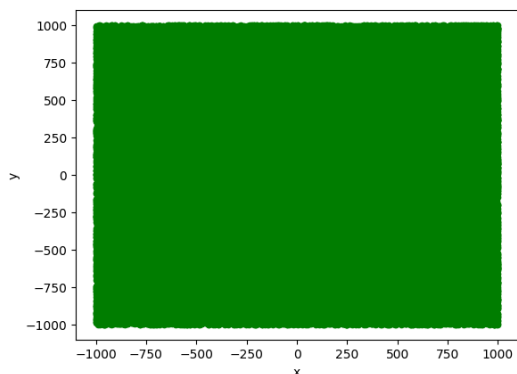
2. Cel ćwiczenia

Celem ćwiczenia jest implementacja podstawowych predykatów geometrycznych, w tym określanie położenia punktu względem prostej. Zadanie obejmuje także przeprowadzenie testów, wizualizację wyników oraz ich analizę i opracowanie.

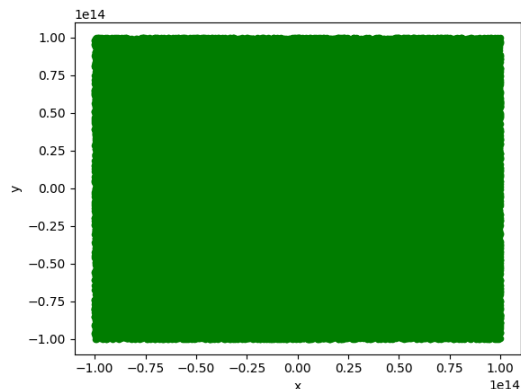
3. Realizacja ćwiczenia

3.1. Generowanie punktów

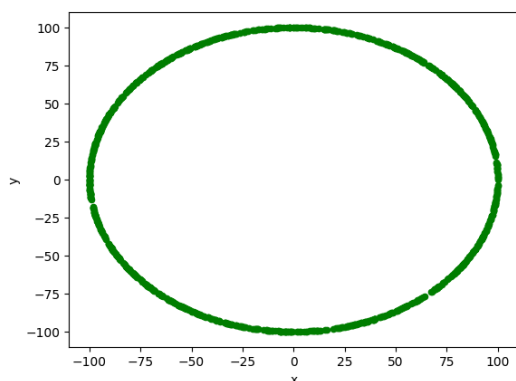
Na początku zostały wygenerowane (przy pomocy *numpy*: `numpy.random.uniform`) 4 zbiory punktów (typu `double`) na dwuwymiarowej przestrzeni euklidesowej.



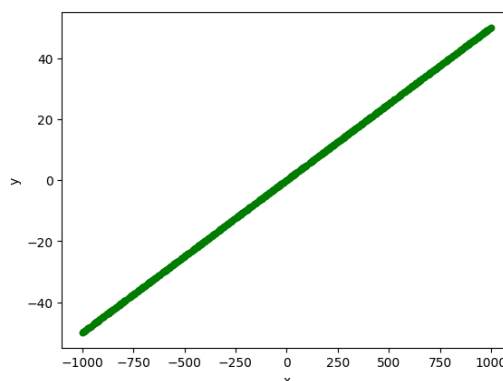
Rysunek 1: Zbiór A: 10^5 losowych punktów o współrzędnych z przedziału $x, y \in [-1000, 1000]^2$



Rysunek 2: Zbiór B: 10^5 losowych punktów o współrzędnych z przedziału $x, y \in [-10^{14}, 10^{14}]^2$



Rysunek 3: Zbiór C: 1000 losowych punktów leżących na okręgu o środku $O = (0, 0)$ i promieniu $R = 100$



Rysunek 4: Zbiór D: 1000 losowych punktów o współrzędnych z przedziału $x \in [-1000, 1000]$ leżących na prostej wyznaczonej przez wektor \vec{ab} , gdzie $a = (-1.0, 0.0)$, $b = (1.0, 0.1)$

3.2. Metody obliczania wyznacznika

Do określenia, po której stronie prostej znajduje się punkt, wykorzystano wartość wyznacznika macierzy:

2x2:

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

lub 3x3:

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

gdzie $a = (a_x, a_y)$, $b = (b_x, b_y)$ są punktami leżącymi na prostej, $c = (c_x, c_y)$ jest punktem, dla którego poszukujemy wiadomości o lokalizacji względem prostej.

Dla:

$\det(a, b, c) < 0 \Rightarrow$ punkt c znajduje się na prawo od prostej ab

$\det(a, b, c) = 0 \Rightarrow$ punkt c znajduje się na prostej ab

$\det(a, b, c) > 0 \Rightarrow$ punkt c znajduje się na lewo od prostej ab

Powyższe wyznaczniki obliczane były przy pomocy własnych implementacji oraz funkcji z biblioteki `numpy: numpy.linalg.det()`.

3.3. Parametry i obserwacje

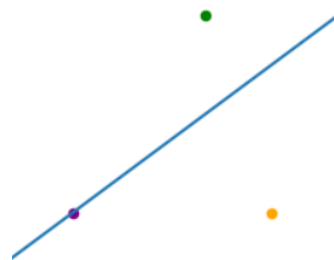
Wygenerowane punkty były klasyfikowane ze względu na ich położenie w stosunku do prostej, z wykorzystaniem różnych metod obliczania wyznacznika. Analiza została przeprowadzona dla dwóch precyzji przechowywania zmiennych oraz różnych wartości tolerancji dla zera, tj.

$$\varepsilon = [0, 10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}]$$

Na podstawie obserwacji wykresów i ramek danych analizowano liczbę punktów zaklasyfikowanych po lewej i prawej stronie prostej, a także różnice w wynikach klasyfikacji między zastosowanymi metodami.

Na wykresach punkty zostały oznaczone kolorami:

- punkty po lewej stronie prostej są zielone,
- na prostej – fioletowe,
- po prawej stronie – pomarańczowe.



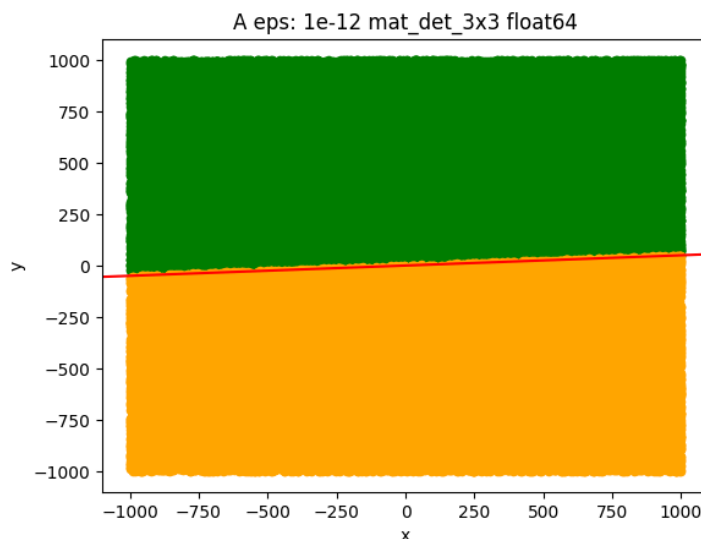
Rysunek 5: Przykładowa klasyfikacja punktów

4. Analiza wyników

4.1. Zbiór A

Precyzja	ϵ	Funkcja liczenia wyznacznika	Punkty na lewo od prostej	Punkty na prostej	Punkty na prawo od prostej
float64	0	mat_det_2x2	49941	0	50059
	1e-14	mat_det_2x2_lib			
float32	1e-12	mat_det_3x3			
	1e-10	mat_det_3x3_lib			
	1e-8				

Tabela 1: Wyniki klasyfikacji punktów ze zbioru A względem ich położenia wobec prostej



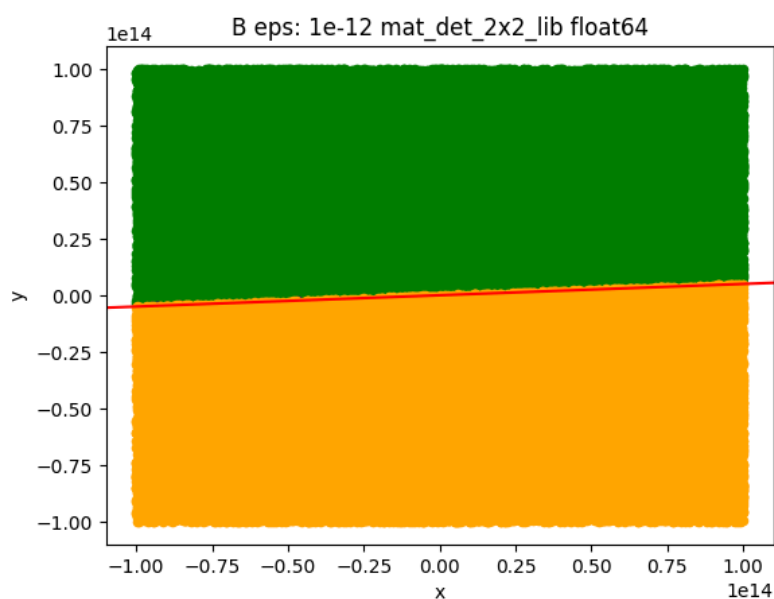
Rysunek 6: Graficzna reprezentacja rozkładu punktów ze zbioru A

Jak można zauważyć w wynikach z Tabeli 1 oraz na Rysunku 6, dla zbioru A wyniki są takie same, niezależnie od przyjętej precyzji, tolerancji dla zera oraz funkcji obliczania wyznacznika.

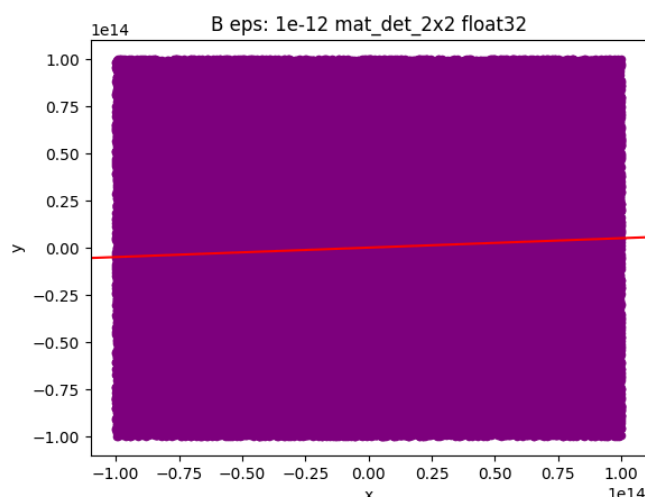
4.2. Zbiór B

Precyzja	ϵ	Funkcja liczenia wyznacznika	Punkty na lewo od prostej	Punkty na prostej	Punkty na prawo od prostej
float64	0 1e-08 1e-10 1e-14 1e-12	mat_det_2x2	50065	6	49929
float64	0 1e-08 1e-10 1e-14 1e-12	mat_det_2x2_lib	50065	7	49928
float64 float32	0 1e-08 1e-10 1e-14 1e-12	mat_det_3x3 mat_det_3x3_lib	50068	0	49932
float32	0 1e-08 1e-10 1e-14 1e-12	mat_det_2x2	0	100000	0
float32	0 1e-08 1e-10 1e-14 1e-12	mat_det_2x2_lib	6648	86767	6585

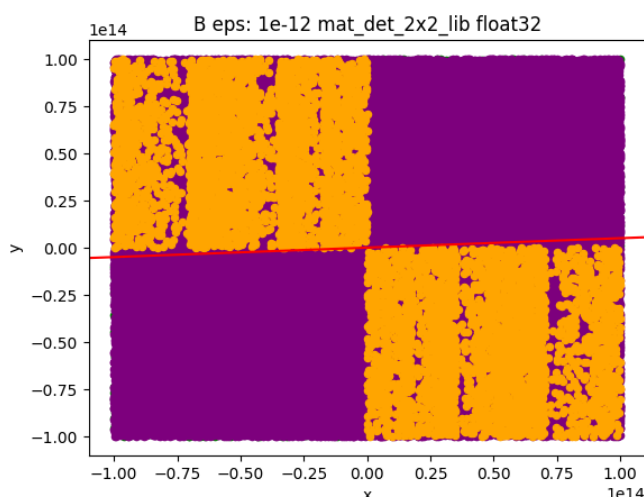
Tabela 2: Wyniki klasyfikacji punktów ze zbioru B względem ich położenia wobec prostej



Rysunek 7: Graficzna reprezentacja rozkładu punktów ze zbioru B (dla większości przypadków)



Rysunek 8: Rozkład punktów dla niższej precyzji zmiennych oraz własnej implementacji wyznacznika 2x2



Rysunek 9: Rozkład punktów dla niższej precyzji zmiennych oraz bibliotecznej implementacji wyznacznika 2x2

Dla drugiego zestawu danych wyniki były odmienne w stosunku do zbioru A. Jak widoczne jest w Tabeli 2, na wynik klasyfikacji znikomy wpływ miała tolerancja dla zera. W większości przypadków sytuacja graficznie prezentuje się jak na Rysunku 7. Drobne odchylenia zaistniały przy określeniu liczby punktów znajdujących się na prostej. W przypadku obu wykorzystanych implementacji obliczania wyznacznika 2x2, kilka punktów zostało tak sklasyfikowanych (warto zaznaczyć, że przy takim zakresie współrzędnych, prawdopodobieństwo, że punkt znajdzie się dokładnie na prostej, jest minimalne).

We własnej implementacji 6 punktami znajdującymi się na prostej były:

$(-46791478047732.56, -2345907731252.1562), (-94812170704360.03, -4720554544727.703),$
 $(-86075737463034.89, -4306096766735.0312), (-79650177844087.97, -3996168677348.625),$
 $(63585526290217.66, 3186187302615.797), (30146152485543.047, 1511276534226.5156).$

Natomiast w przypadku funkcji bibliotecznej takich punktów było 7:

$(-93261521643382.17, -4668383096997.328), (-92907041367145.5, -4606517911029.234),$
 $(-46791478047732.56, -2345907731252.1562), (-94812170704360.03, -4720554544727.703),$
 $(-86075737463034.89, -4306096766735.0312), (-79650177844087.97, -3996168677348.625),$
 $(63585526290217.66, 3186187302615.797).$

Ta różnica wiąże się ze specyfiką reprezentacji liczb zmiennoprzecinkowych. Mimo że liczba bitów mantysy pozostaje stała, przy dużych wartościach liczbowych zwiększona cecha powoduje, że kolejne reprezentowalne liczby są bardziej oddalone od siebie. W konsekwencji, podczas obliczania wyznacznika 2x2, operacja odejmowania podobnych, dużych wartości może prowadzić do utraty precyzji - te same bity mantysy reprezentują bowiem coraz większe różnice bezwzględne między liczbami.

Wyznaczniki 3x3 nie zaklasyfikowały żadnego punktu jako znajdującego się na prostej.

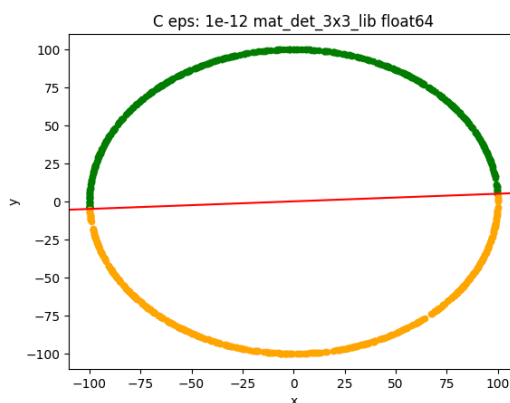
Dla *float32* oraz wyznaczników 2x2 nastąpiły błędy w klasyfikacji (utrata precyzji podobna do przypadku powyżej). Funkcja własnej implementacji określiła, że wszystkie punkty płaszczyzny znajdują się na prostej (Rysunek 8 - wartość wyznacznika wyniosła zawsze 0),

podczas gdy funkcja biblioteczna większość punktów z 2 i 4 ćwiartki układu współrzędnych (Rysunek 9) opisała jako znajdujące się po prawej stronie prostej, a te z 1 i 3 ćwiartki jako znajdujące się na niej. Problem ten nie wystąpił jednak w przypadku wyznaczników 3x3.

4.3. Zbiór C

Precyzja	ϵ	Funkcja liczenia wyznacznika	Punkty na lewo od prostej	Punkty na prostej	Punkty na prawo od prostej
float64	0	mat_det_2x2_lib	488	0	512
	1e-08	mat_det_2x2			
float32	1e-10	mat_det_3x3_lib			
	1e-14	mat_det_3x3			
	1e-12	mat_det_3x3			

Tabela 3: Wyniki klasyfikacji punktów ze zbioru C względem ich położenia wobec prostej



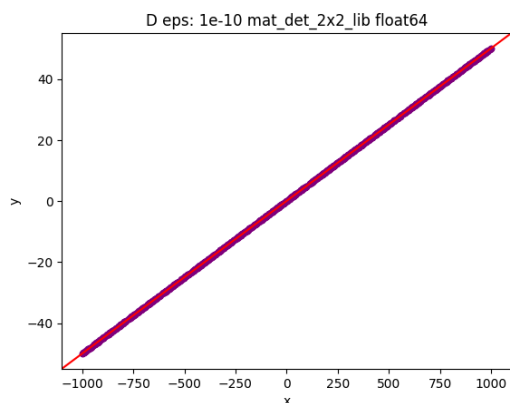
Rysunek 10: Graficzna reprezentacja rozkładu punktów ze zbioru C

Dla trzeciego zestawu danych (Tabela 3 oraz Rysunek 10) wyniki były identyczne we wszystkich analizowanych przypadkach.

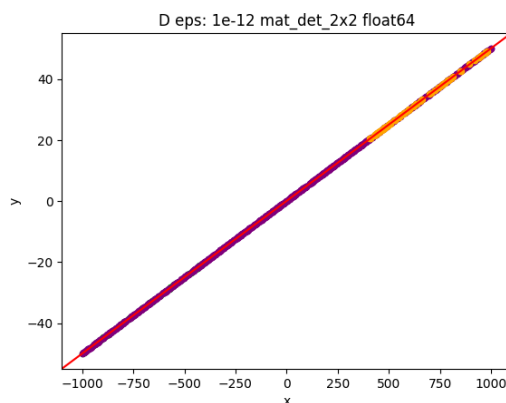
4.4. Zbiór D

Precyzja	ϵ	Klasyfikacja dla różnych funkcji liczenia wyznacznika - wyniki w postaci krotek: (punkty na <i>LEWO</i> od prostej, <i>NA</i> prostej, na <i>PRAWO</i> od prostej)			
		2x2	2x2_lib	3x3	3x3_lib
float64	0	(135, 730, 135)	(164, 671, 165)	(82, 691, 227)	(408, 325, 267)
	1e-14	(129, 744, 127)	(154, 689, 157)	(0, 930, 70)	(47, 908, 45)
	1e-12	(73, 849, 78)	(101, 789, 110)	(0, 1000, 0)	(0, 1000, 0)
	1e-10	(0, 1000, 0)	(0, 1000, 0)	(0, 1000, 0)	(0, 1000, 0)
	1e-08	(0, 1000, 0)	(0, 1000, 0)	(0, 1000, 0)	(0, 1000, 0)
float32	0	(155, 658, 187)	(466, 0, 534)	(246, 571, 183)	(507, 37, 456)
	1e-14	(155, 658, 187)	(466, 0, 534)	(246, 571, 183)	(437, 147, 416)
	1e-12	(155, 658, 187)	(466, 0, 534)	(246, 571, 183)	(430, 163, 407)
	1e-10	(155, 658, 187)	(466, 0, 534)	(246, 571, 183)	(430, 163, 407)
	1e-08	(155, 658, 187)	(465, 1, 534)	(246, 572, 182)	(430, 165, 405)

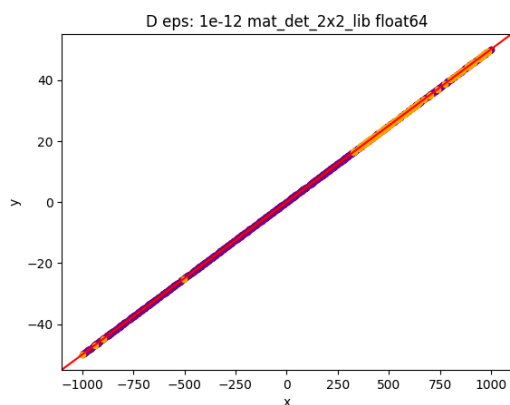
Tabela 4: Wyniki klasyfikacji punktów ze zbioru D względem ich położenia wobec prostej



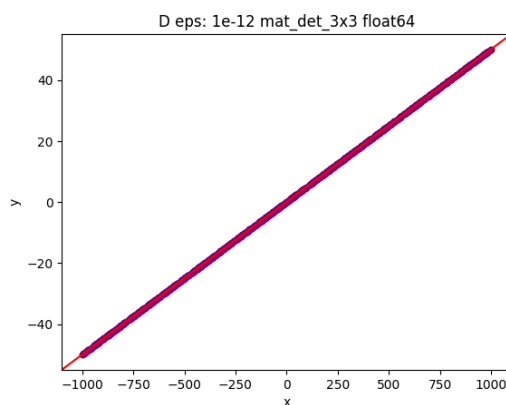
Rysunek 11: $\varepsilon > 10^{-10}$, dowolny wyznacznik



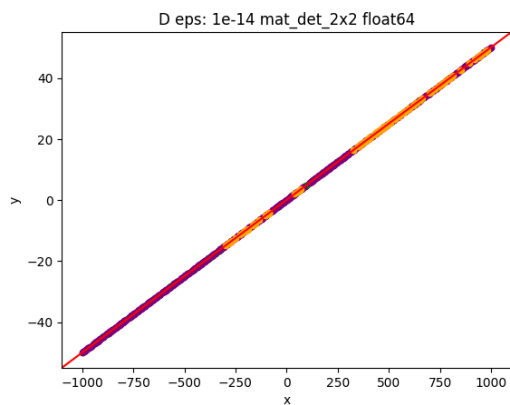
Rysunek 12: $\varepsilon = 10^{-12}$, wyzn. 2x2



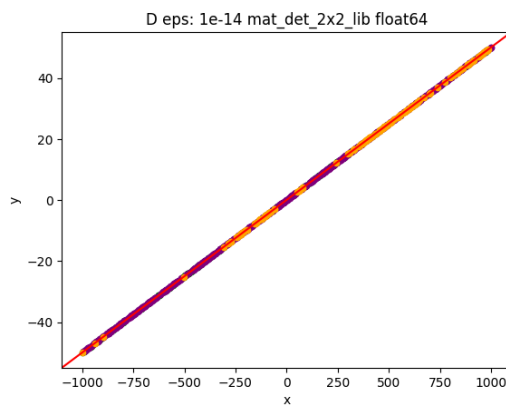
Rysunek 13: $\varepsilon = 10^{-12}$, wyzn. 2x2 biblioteczny



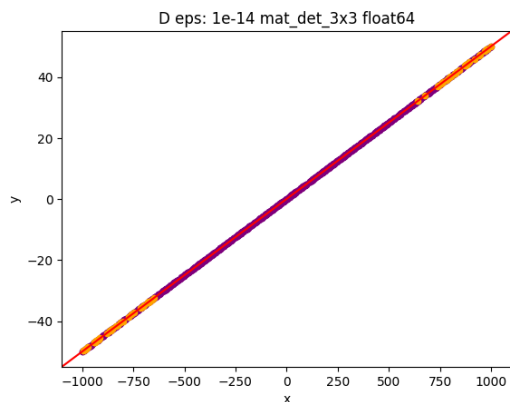
Rysunek 14: $\varepsilon = 10^{-12}$, oba wyznaczniki 3x3



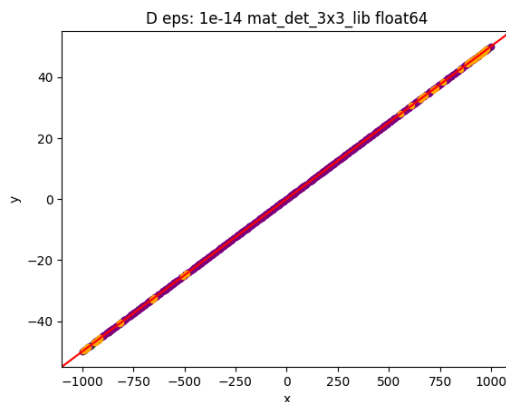
Rysunek 15: $\varepsilon = 10^{-14}$, wyzn. 2x2



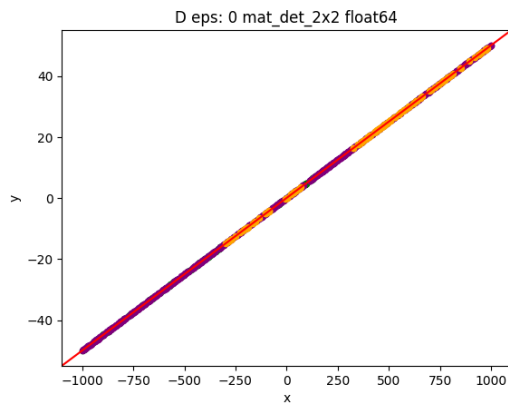
Rysunek 16: $\varepsilon = 10^{-14}$, wyzn. 2x2 biblioteczny



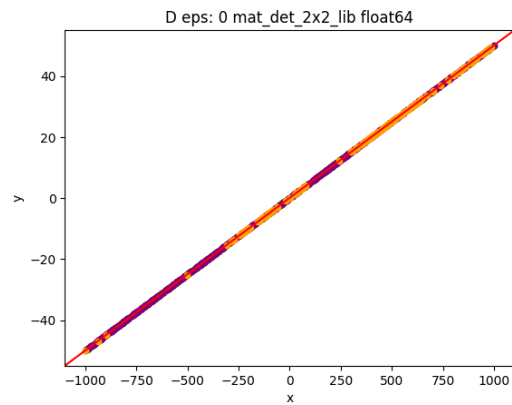
Rysunek 17: $\varepsilon = 10^{-14}$, wyzn. 3x3



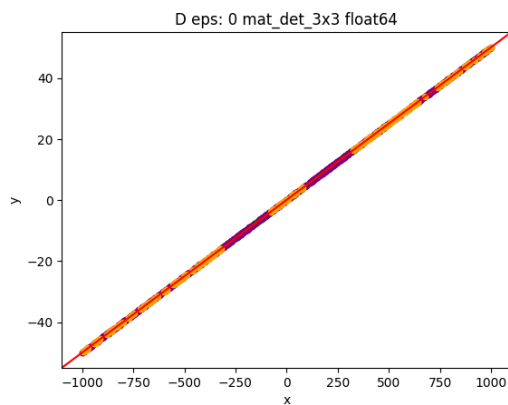
Rysunek 18: $\varepsilon = 10^{-14}$, wyzn. 3x3 biblioteczny



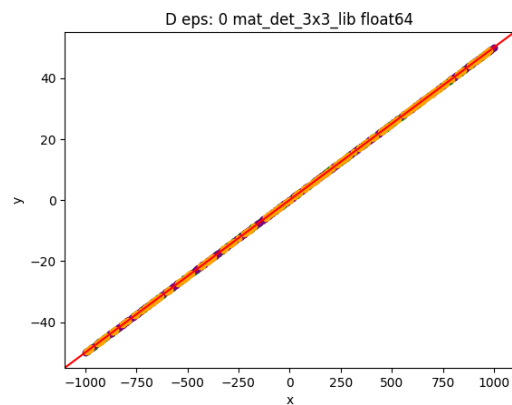
Rysunek 19: $\varepsilon = 0$, wyzn. 2x2



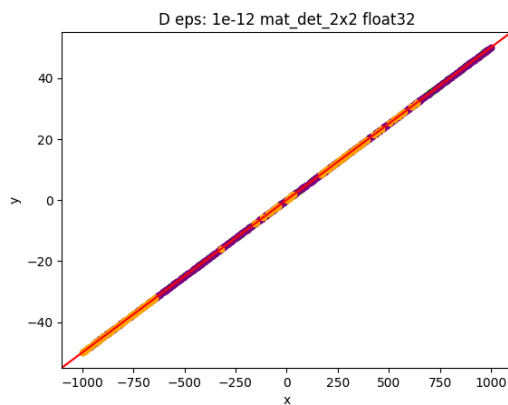
Rysunek 20: $\varepsilon = 0$, wyzn. 2x2 biblioteczny



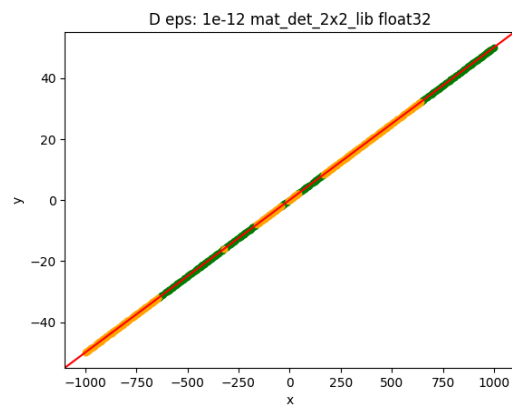
Rysunek 21: $\varepsilon = 0$, wyzn. 3x3



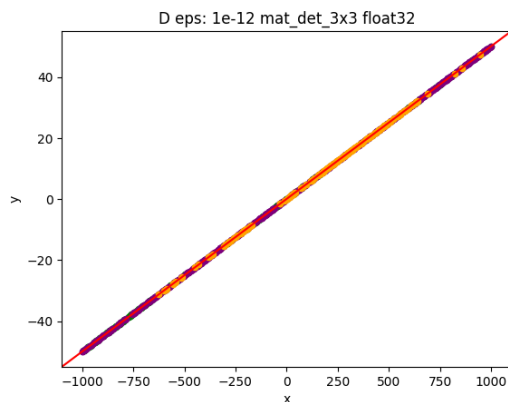
Rysunek 22: $\varepsilon = 0$, wyzn. 3x3 biblioteczny



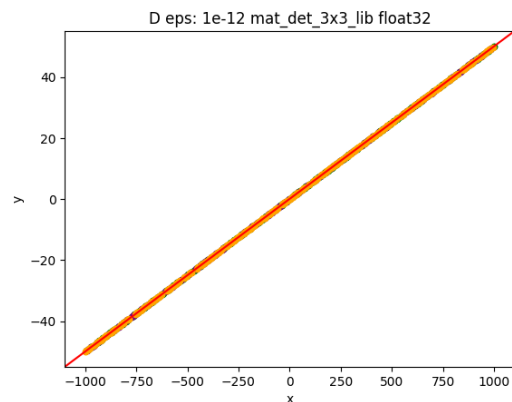
Rysunek 23: $\varepsilon = 10^{-12}$, float32, wyzn. 2x2



Rysunek 24: $\varepsilon = 10^{-12}$, float32, wyzn. 2x2 bibl.



Rysunek 25: $\varepsilon = 10^{-12}$, float32, wyzn. 3x3



Rysunek 26: $\varepsilon = 10^{-12}$, float32, wyzn. 3x3 bibl.

4.4.1. Analiza dla *float64*

Wyniki dla ostatniego zestawu danych okazały się być najbardziej zróżnicowane. Jak można zauważyć w Tabeli 4 oraz na Rysunku 11, dla tolerancji od zera $> 10^{-10}$ znaczenie traci wybór wyznacznika, a klasyfikacja punktów jest dość precyzyjna. W przypadku $\varepsilon = 10^{-12}$ własna implementacja wyznacznika 2x2 (Rysunek 12) zwróciła marginalnie lepszy wynik od implementacji bibliotecznej (Rysunek 13). Z kolei oba wyznaczniki 3x3 (Rysunek 14) dokładnie określiły pozycję punktów. Dla $\varepsilon = 10^{-14}$ na Rysunku 15 oraz 16 wciąż widać, że funkcja biblioteczna jest nieznacznie mniej dokładna. Podobna sytuacja jest w przypadku wyznaczników 3x3, czego nie widać dokładnie na Rysunku 17 oraz 18, lecz zauważyć można w Tabeli 4. Przy braku tolerancji na zbliżenie do zera, wciąż okazują się lepsze funkcje zaimplementowane samodzielnie (Rysunek 19 i 20). Biblioteczny wyznacznik 3x3 (Rysunek 22) klasyfikuje ledwo ponad $\frac{1}{3}$ punktów jako należące na prostą. Jak wynika z danych z Tabeli 4, implementacja wyznacznika 2x2 okazuje się dokładniejsza od wyznacznika 3x3, lecz ten wciąż jest bardziej precyzyjny od bibliotecznego wyznacznika 2x2 (co jednak jest trudne do zauważenia na Rysunku 20).

4.4.2. Analiza dla *float32*

W przypadku skorzystania z mniejszej precyzji zmiennych, wyniki dla poszczególnych ε są bardzo do siebie zbliżone. Funkcje obliczające wartość wyznacznika implementowane samodzielnie okazują się o wiele lepsze od funkcji bibliotecznych, co widać na Rysunkach 23, 24, 25 i 26. Jak można zauważyć w Tabeli 4, w skrajnych przypadkach biblioteczny wyznacznik 3x3 w ogóle nie klasyfikuje punktów na prostą, a jego odpowiednik 2x2 potrafi opisać w ten sposób tylko 4% obiektów.

5. Wnioski

Na podstawie przeprowadzonego ćwiczenia można wyciągnąć następujące wnioski:

1. Precyzja zmiennych ma kluczowe znaczenie dla dokładności obliczeń geometrycznych. Użycie zmiennych typu *float64* generalnie zapewnia lepsze wyniki niż *float32*, szczególnie dla dużych wartości współrzędnych (zbiór B).
2. Tolerancja dla zera (ε) ma istotny wpływ na klasyfikację punktów, zwłaszcza dla zbiorów punktów leżących blisko prostej (zbiór D). Dobór odpowiedniej wartości ε jest kluczowy dla uzyskania wiarygodnych wyników.
3. Własne implementacje wyznaczników często okazywały się bardziej precyzyjne niż funkcje biblioteczne, szczególnie w przypadku mniejszej precyzji zmiennych (*float32*).
4. Dla zbiorów punktów o małym zakresie współrzędnych (zbiór A i C) różnice między metodami są minimalne, co sugeruje, że w takich przypadkach wybór metody jest mniej krytyczny.
5. Przy pracy z bardzo dużymi wartościami współrzędnych (zbiór B) należy zachować ostrożność, gdyż może dojść do utraty precyzji i błędnych klasyfikacji.
6. Mantysa liczby zmiennoprzecinkowej jest skończona, więc w przypadku dużych liczb przechowywana jest tylko określona liczba cyfr. Cyfry na dalszych miejscach dziesiętnych są pomijane, co może prowadzić do utraty precyzji.
7. W praktycznych zastosowaniach należy przeprowadzić testy dla konkretnego zakresu danych i wybrać metodę, która zapewnia najlepszą dokładność przy akceptowalnym czasie obliczeń.

8. Wizualizacja wyników okazała się dobrym (aczkolwiek nie zawsze dokładnym) narzędziem w analizie dokładności różnych metod, pozwalając na szybkie wykrycie anomalii w klasyfikacji punktów.

Podsumowując, wybór odpowiedniej metody obliczania wyznaczników, precyzji zmiennych oraz tolerancji dla zera ma kluczowe znaczenie dla dokładności predykatów geometrycznych. W praktyce należy zawsze dostosować te parametry do specyfiki rozwiązywanego problemu i charakterystyki danych wejściowych. Wyciągając wnioski z analizy zbioru D, można by stwierdzić, że dosyć uniwersalnym podejściem do sprawdzania pozycji punktu wobec prostej byłoby skorzystanie z własnej implementacji wyznacznika 3×3 .