Algorytmy geometryczne
Sprawozdanie z laboratorium 1.

Michał Nożkiewicz
gr. Czw_11:20_B

1. Użyte narzędzia i oprogramowanie

Ćwiczenie realizowałem na komputerze z systemem Windows 10 x64**,**
Procesor: Intel Core i5-4460 @3.2 GHz
Pamięc RAM: 8GB

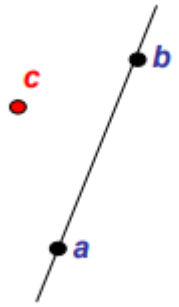
Kod programu pisałem w środowisku Jupyter notebook, w języku Python, z wykorzystaniem bibliotek *numpy, math, random, time* oraz *matplotlib.*

2. Wstęp teoretyczny

Zadaniem tego laboratorium było zaimplementowanie funkcji, która dla różnych zbiorów punktów, sprawdzała po której stronie danej prostej znajdują się punkty tego zbioru.
Do stwierdzenia po której stronie prostej znajduje się dany punkt może posłużyć jedna z dwóch metod:

- wyznacznik macierzy 3x3:

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



- wyznacznik macierzy 2x2:

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

W obu przypadkach zależności są takie same:

$$\det(a,b,c) = \begin{cases} < 0 - c\ znajdule\ się\ po\ lewej\ stronie\ prostej\ ab \\ > 0 - c\ znajduje\ się\ po\ prawej\ stronie\ prostej \\ = 0 - c\ leży\ na\ prostej \end{cases}$$

3. Realizacja ćwiczenia

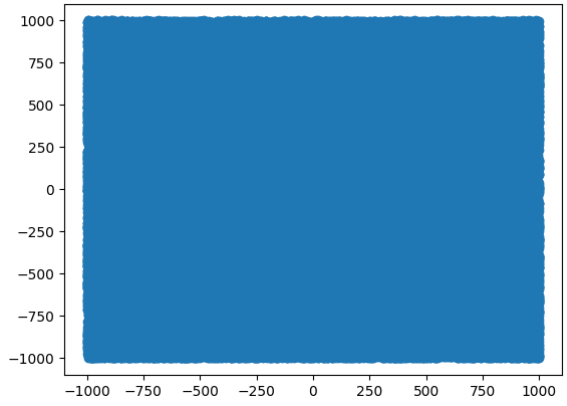
W ćwiczeniu wykorzystałem 4 zbiory punktów, wszystkie były to punkty 2D o współrzędnych typu double wylosowane z rozkładu jednostajnego.

- Zbiór A
  100 000 punktów o współrzędnych z przedziału [-1000, 1000],
- Zbiór B
  100 000 punktów o współrzędncyh z przedziału [-10^{14}, 10^{14}],
- Zbiór C
  1000 punktów leżących na okręgu o środku (0, 0) i promieniu R = 100,
- Ziór D
  1000 punktów o współrzędnych z przedziału [-1000, 1000], lężących na prostej przechodzącej przez punkty a = (-1.0, 0.0) i b = (1.0, 0.1)

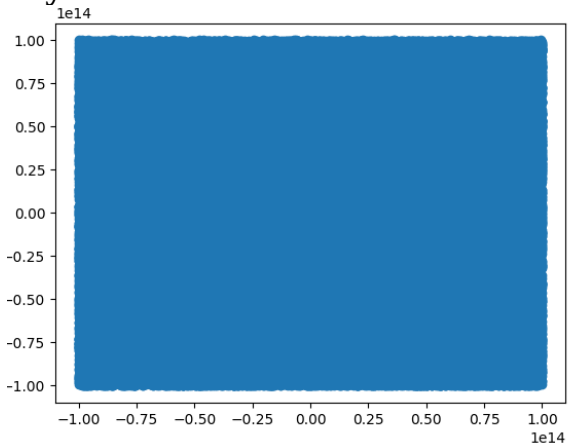
Losowanie punktów zrealizowałem korzystając z funkcji biblioteki random random.uniform, która przyjmuje dwa argumenty będące końcami przedziału z którego mamy losować i zwraca liczbę z rozkładu jednostajnego.

Do generowania punktów ze zbioru C, wykorzystałem parametryczne równanie okręgu i skorzystałem dodatkowo z funkcji math.sin i math.cos z biblioteki math. Do generowania punktów ze zbioru D wyznaczyłem równanie kierunkowe prostej przechodzącej przez dwa podane punkty i po wylosowaniu współrzędnej x, można było wyznaczyć współrzędną y.

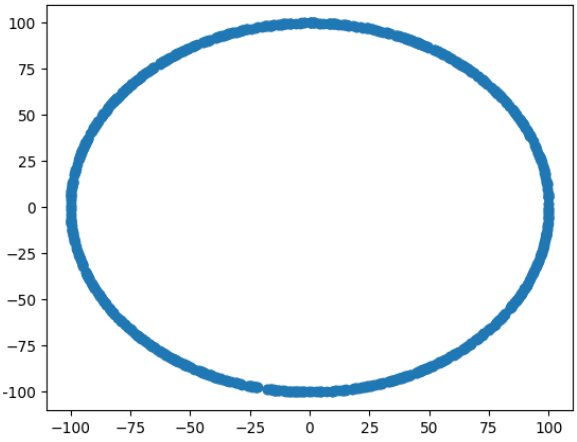
Wykres 3.1 Zbiór A



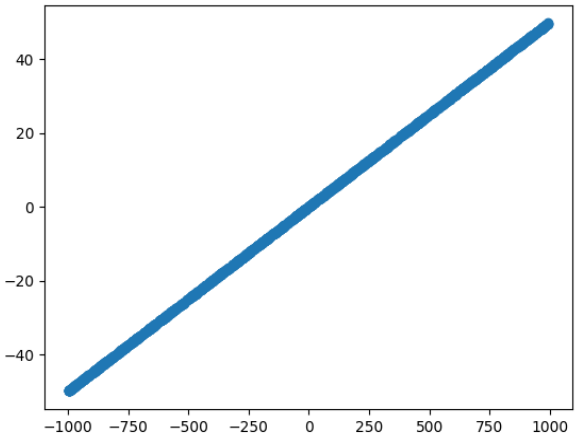
Wykres 3.2 Zbiór B



Wykres 3.3 Zbiór C



Wykres 3.4 Zbiór D



Następnie zaimplementowałem funkcje do obliczania wyznaczników:

- det_1, samodzielnie zaimplementowana funkcja obliczająca wyznacznik z macierzy 2x2,
- det_2, samodzielnie zaimplementowana funkcja obliczająca wyznacznik z macierzy 3x3,
- det_3, funkcja obliczająca wyznacznik macierzy 2x2 z wykorzystaniem funkcji bibliotecznej z numpy,
- det_4, funkcja obliczająca wyznacznik macierzy 3x3 z wykorzystaniem funkcji bibliotecznej z numpy.

Kolejnym krokiem było zaimplementowanie funkcji, która dla danego zbioru punktów, podzieliła by punkty według położenia względem prostej(ta sama prosta,

według której generowałem punkty w zbiorze D). Punkty położone po lewej stronie osi koloruje na zielone, na prawo na niebiesko, a leżące na osi na czerwone. Ponadto należy przyjąć odpowiednią tolerancję dla zera, na podstawie której będzie można testować różne rodzaje obliczania wyznaczników. Przyjąłem następujące tolerancje: $\varepsilon_1 = 10^{-12}$, $\varepsilon_2 = 10^{-14}$, $\varepsilon_3 = 10^{-16}$.

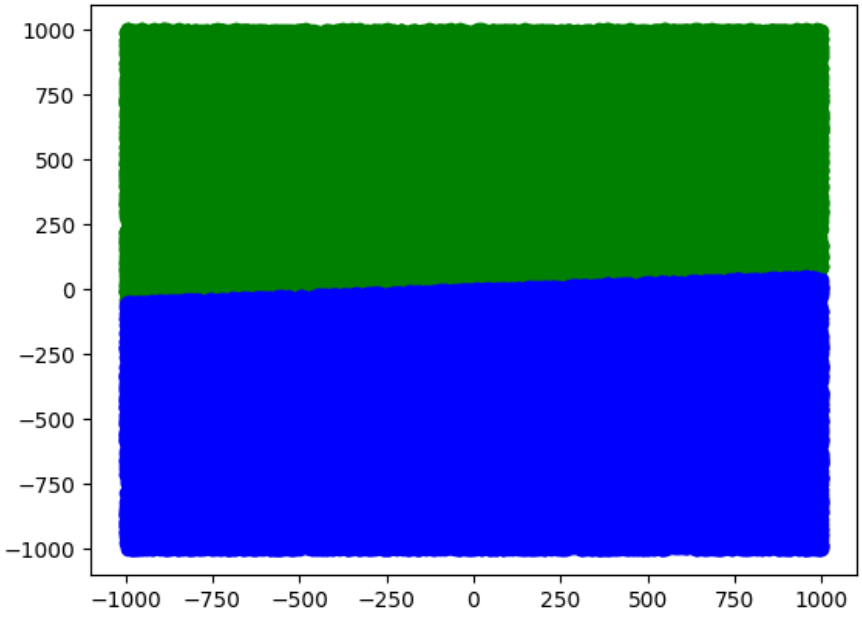
4. Wizualizacja funkcji categorize

Tu przedstawię przykładowe wykresy dla różnych tolerancji i wyznaczników, szczegółowe informacje będą opisane w tabelce w rozdziale 5.

4.1 Wizualizacja zbioru A

Dla zbioru A, każdy wykres, niezależnie od przyjętej tolerancji i funkcji obliczającej wyznacznik, wyglądał dokładnie tak samo.

Wykres 4.1.1 Zbiór A (dowolny ε i wyznacznik)



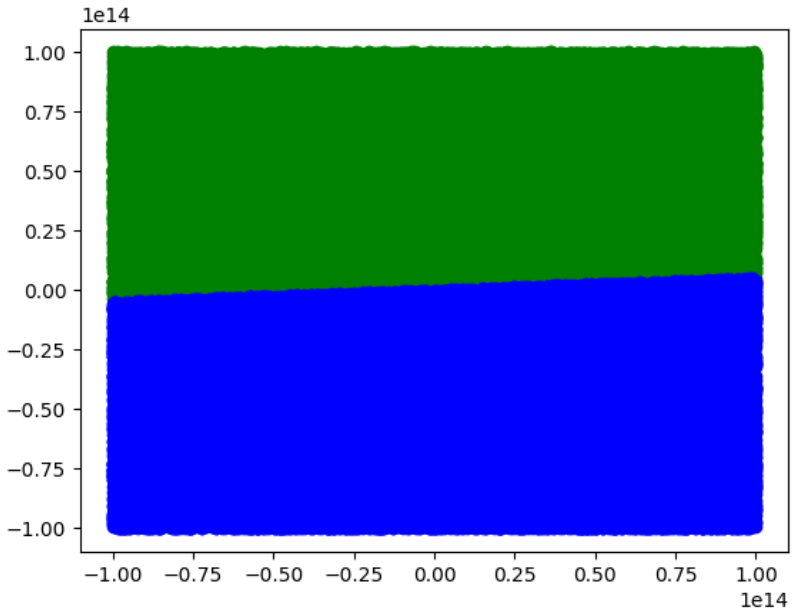
Licznik punktów:

Po lewej stronie: 50149 *Po prawej stronie: 49851* *Na linii: 0*

4.2 Wizualizacja zbioru B

Różne wyniki można było jednak dostrzec dla zbioru B.

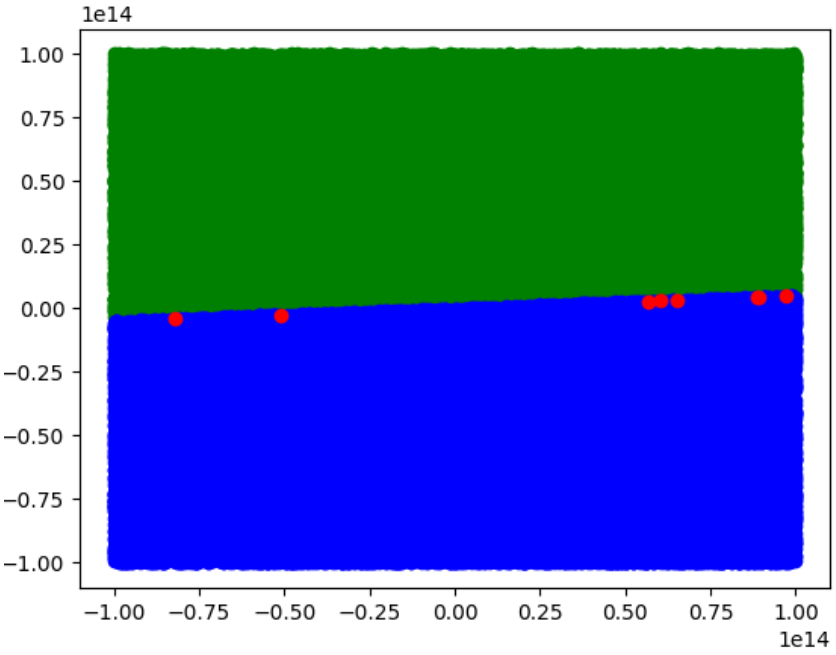
Wykres 4.2.1 Zbiór B (ε_1 i wyznaczniki 3x3)



Licznik punktów:

Po lewej stronie: 49908 *Po prawej stronie: 50092* *Na linii: 0*

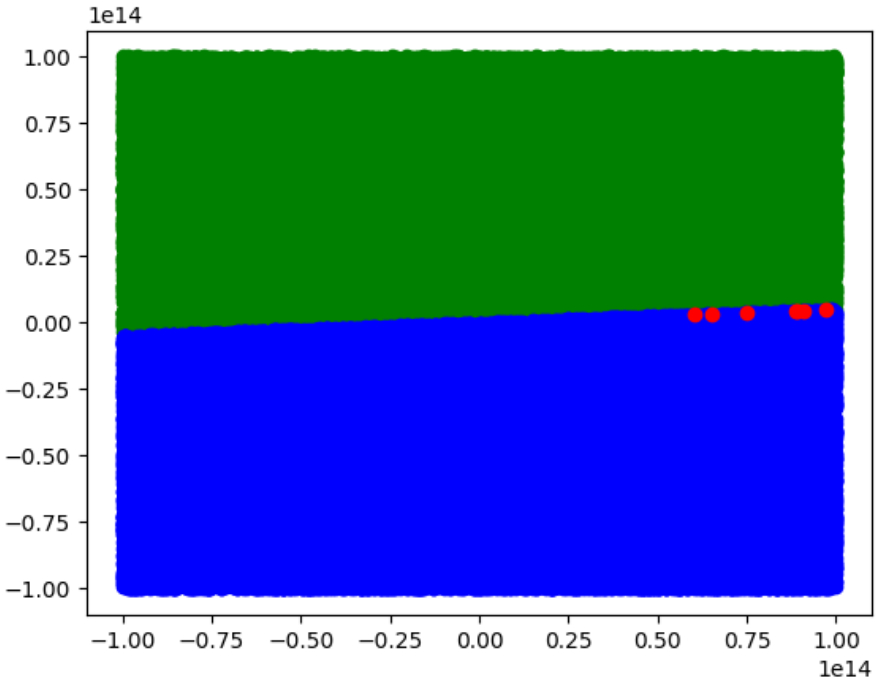
Wykres 4.2.2 Zbiór B (ε_1 i wyznacznik 2x2 własny)



Licznik punktów:

Po lewej stronie: 49905 *Po prawej stronie: 50087* *Na linii: 8*

Wykres 4.2.3 Zbiór B (ε_1 i wyznacznik 2x2 numpy)



Licznik punktów:

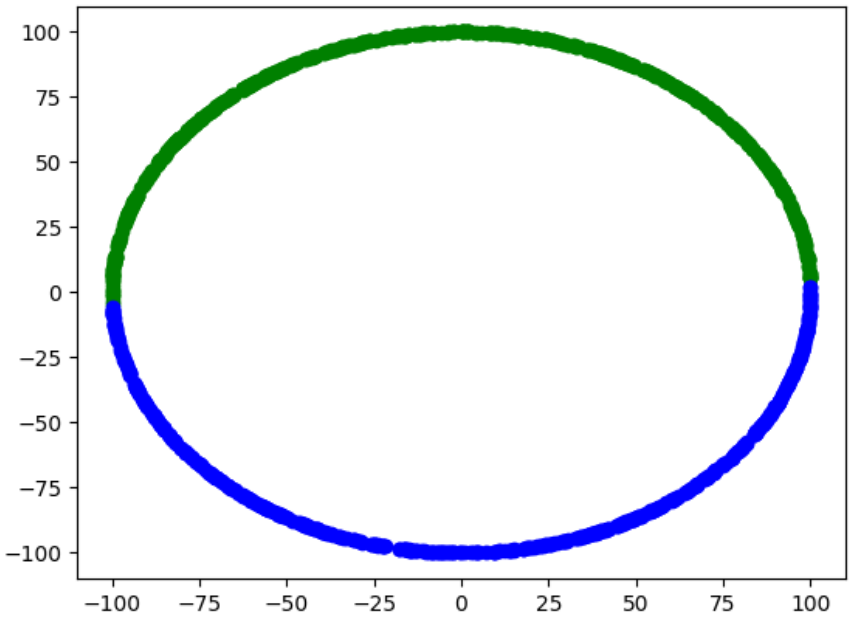
Po lewej stronie: 49906 *Po prawej stronie: 50087* *Na linii: 7*

Wyniki dla mniejszych dokładności były dokładnie takie same dla każdego wykresu. Można wnioskować, że przy obliczeniach na tak dużych liczbach, wyniki blisko zera zaokrągliły się do zera przez co tolerancja nie miała już znaczenia. Jednak zauważalne są różnice dla różnych wyznaczników, 3x3 są w tym przypadku dokładniejsze(lub faktycznie aż osiem punktów wylosowało się na prostej)

4.3 Wizualizacja zbioru C

Analogicznie jak w przypadku zbioru A, który także zawierał punkty z małego przedziału, nie zaobserwowano różnic dla różnych wyznaczników i tolerancji.

Wykres 4.3.1 Zbiór C (dowolny ε i wyznacznik)



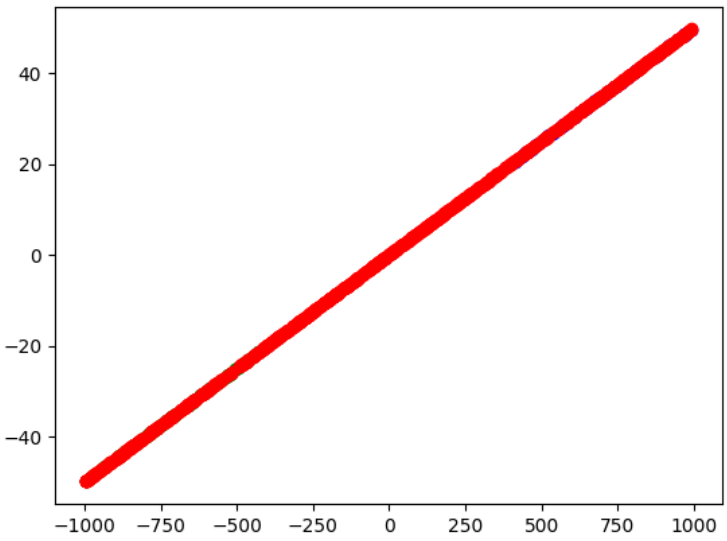
Licznik punktów:

Po lewej stronie: 494 *Po prawej stronie: 506* *Na linii: 0*

4.4 Wizualizacja zbioru D

Największe różnice były widoczne dla zbioru D. Wykresy znacznie się różniły, dla każdej tolerancji i wyznacznika.

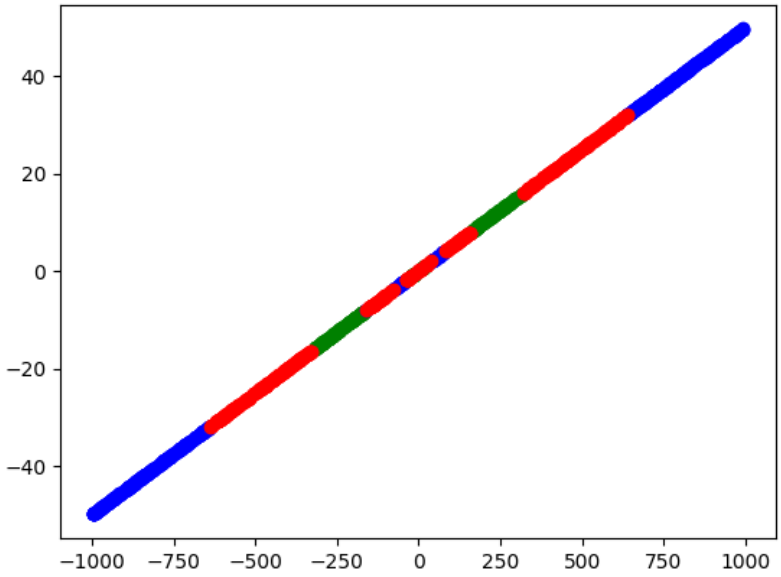
Wykres 4.4.1 Zbiór D (ε_1(oba 3x3) i ε_2 (3x3 własny))



Licznik punktów:

Po lewej stronie: 0 *Po prawej stronie: 0* *Na linii: 1000*

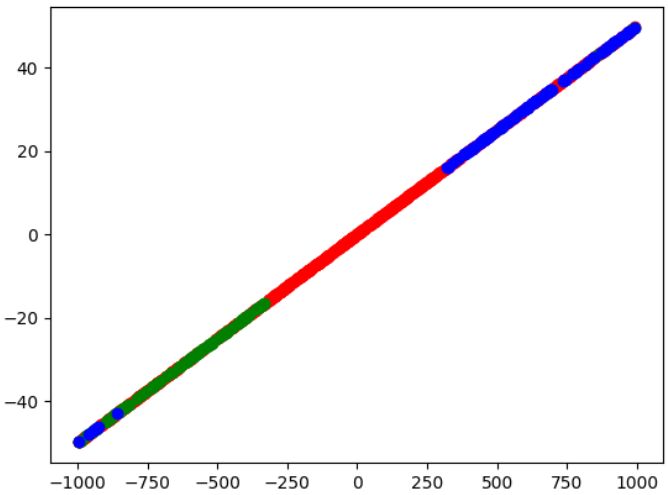
Wykres 4.4.2 Zbiór D (ε_3(3x3 własny))



Licznik punktów:

Po lewej stronie: 190 *Po prawej stronie: 420* *Na linii: 390*

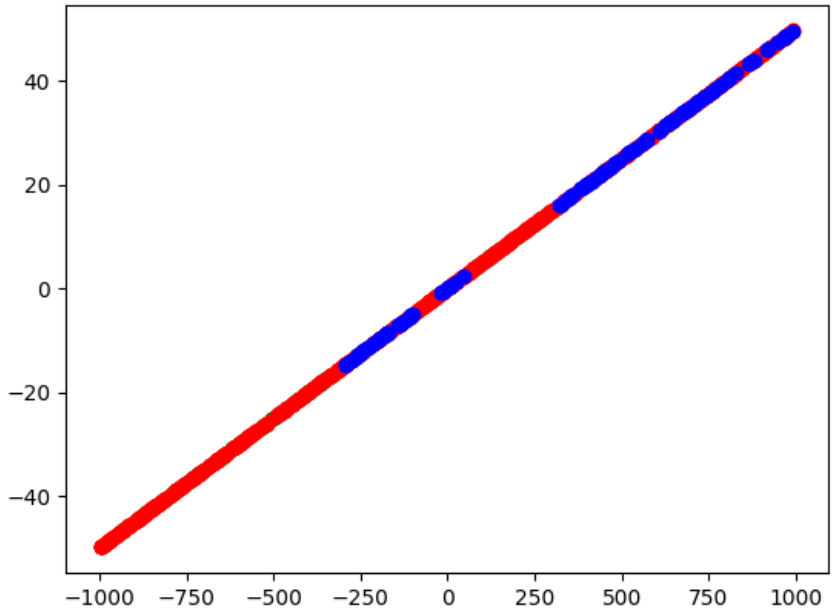
Wykres 4.4.3 Zbiór D (ε_1(2x2 numpy))



Licznik punktów:

Po lewej stronie: 100 *Po prawej stronie: 112* *Na linii: 788*

Wykres 4.4.4 Zbiór D (ε_3(2x2 własny))



Licznik punktów:

Po lewej stronie: 134 *Po prawej stronie: 139* *Na linii: 727*

5. Szcegółowe tabele pomiarów

Tabele dla każdej tolerancji i wyznacznika określają liczbe punktów sklasyfikowanych po konkretnych stronach prostej lub na niej.

$$\varepsilon_1 = 10^{-12}, \varepsilon_2 = 10^{-14}, \varepsilon_3 = 10^{-16}$$

Tabela 5.1 Zbiór A

Wyznacznik	Tolerancja	Po lewej	Po prawej	Na linii
Wszystkie wyznaczniki	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	50149	49851	0

Tabela 5.2 Zbiór B

Wyznacznik	Tolerancja	Po lewej	Po prawej	Na linii
2x2(własny)	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	49905	50087	8
2x2 (numpy)	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	49906	50087	7
3x3 (własny)	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	49908	50092	0
3x3 (numpy)	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	49908	50092	0

Tabela 5.3 Zbiór C

Wyznacznik	Tolerancja	Po lewej	Po prawej	Na linii
Wszystkie wyznaczniki	$\varepsilon_1, \varepsilon_2, \varepsilon_3$	494	506	0

Tabela 5.4 Zbiór D

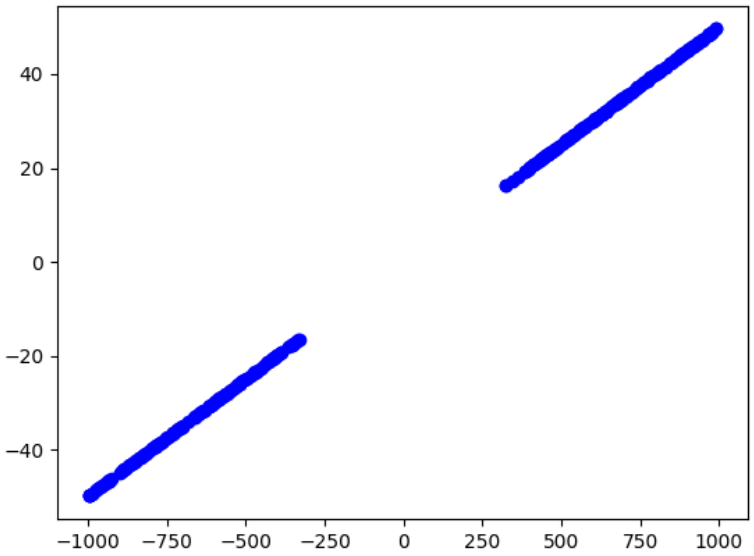
Wyznacznik	Tolerancja	Po lewej	Po prawej	Na linii
Wszystkie wyznaczniki	$\varepsilon_0 = 10^{-10}$	0	0	1000
2x2(własny)	$\varepsilon 1$	79	72	849
2x2 (numpy)		100	112	788
3x3 (własny)		0	0	1000
3x3 (numpy)		0	0	1000
2x2 (własny)		130	131	739

2x2 (numpy)	ε_2	141	167	692
3x3 (własny)		0	0	1000
3x3 (numpy)		19	94	887
2x2 (własny)	ε_3	134	139	727
2x2 (numpy)		148	174	678
3x3 (własny)		190	420	390
3x3 (numpy)		378	329	293

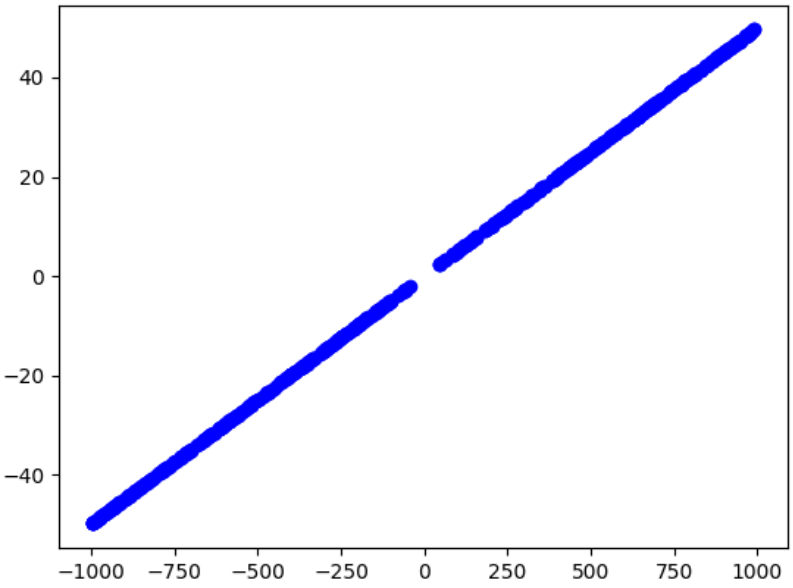
6. Porównanie różnic w wykresach

Ostatnim krokiem ćwiczenia jest napisanie funkcji, która porównuje wykresy. Funkcja przyjmuje listę wyznaczników, które ma ze sobą porównać oraz niepewność. Punkt będzie należał do wykresu wtedy i tylko wtedy, gdy nie został sklasyfikowany tak samo dla wszystkich wyznaczników podanych w argumencie. Jak widać po wynikach przedstawionych w tabelach, jedynym wartym rozpatrzenia pod tym kątem jest zbiór D.

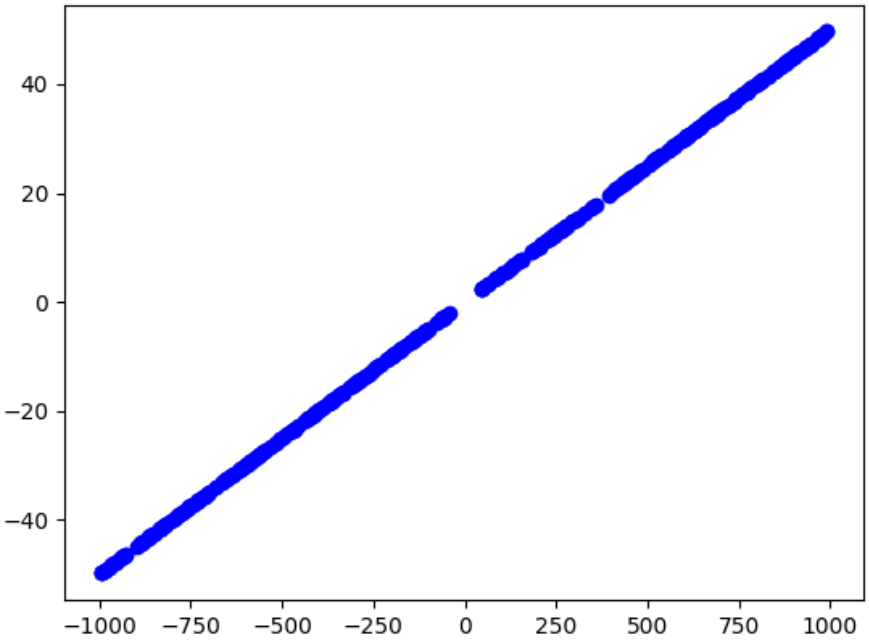
Wykres 6.1 Porówanie wszystkich wyznaczników (ε_1)



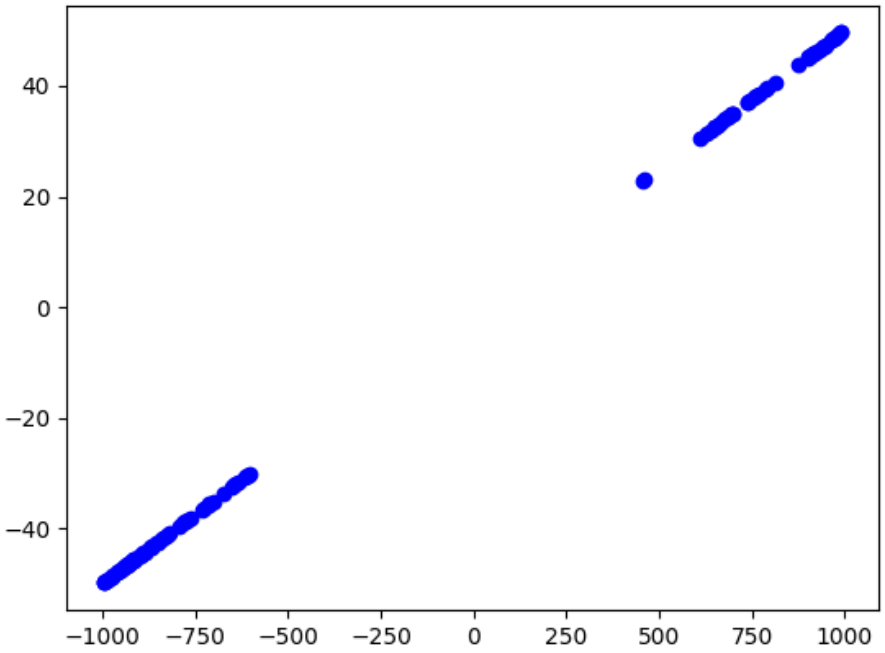
Wykres 6.2 Porównanie wszystkich wyznaczników (ε_2)



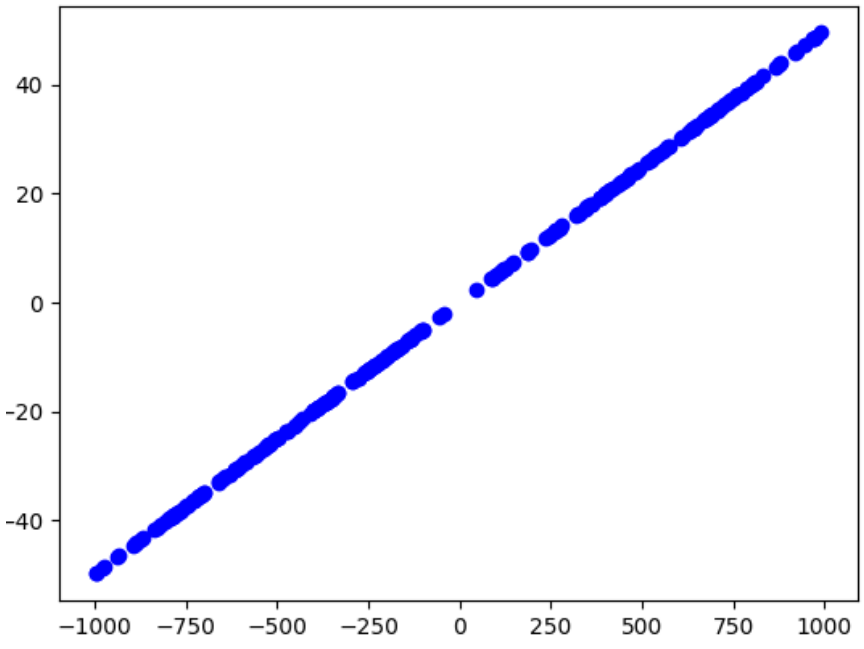
Wykres 6.3 Porównanie 2x2 (ε_2)



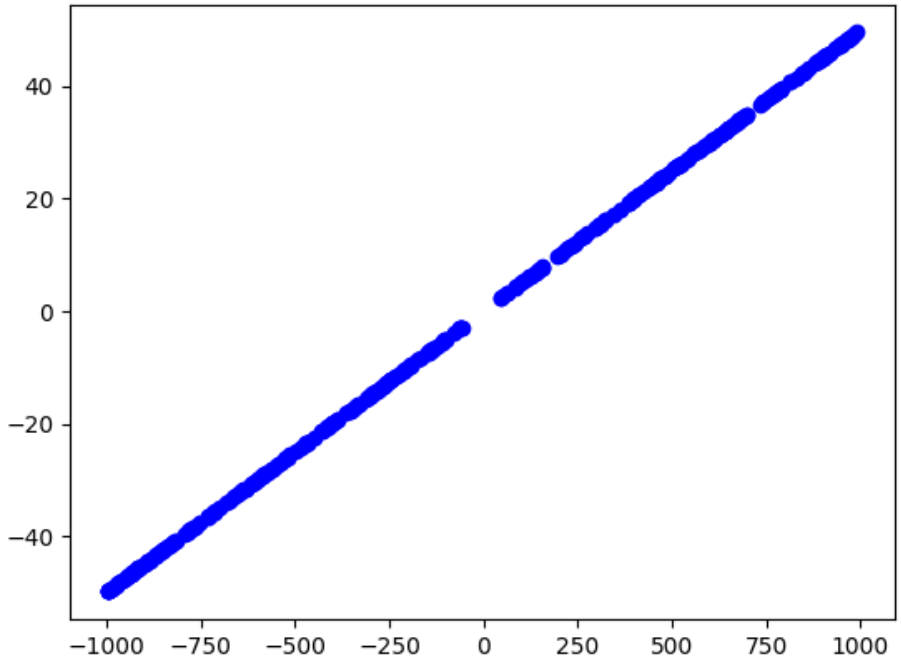
Wykres 6.4 Porównanie 3x3 (ε_2)



Wykres 6.5 Porównanie wyznaczników własnych (ε_2)



Wykres 6.6 Porównanie wyznaczników numpy (ε_2)



7. Porównanie czasu działania

Testy szybkości przeprowadziłem korzystając z trzech zestawów danych, rozmiaru odpowiednio 100 000, 500 000 i 1 000 000 elementów. Do pomiaru czasu użyłem funkcji time z biblioteki time. Testowałem je na funkcji categorize. Przyjęta tolerancja oczywiście nie miała znaczenia, więc mogła być dowolna. Dla każdego zestawu przeprowadziłem po 3 testy, wyniki w tabeli to ich średnia arytmetyczna.

Tabela 7.1 Porównanie czasu działania

	Czas działania[s] dla danej liczby elementów		
Wyznacznik	100 000	500 000	1 000 000
2x2(własny)	0.0825	0.3577	0.7199
2x2 (numpy)	0.7667	3.9098	7.77
3x3 (własny)	0.0943	0.461	0.9529
3x3 (numpy)	0.845	4.2906	8.4307

8. Wnioski

Jak widać w zbiorach A oraz C różne metody obliczania wyznacznika nie dawały znacząco różnych wyników, stosując rozkład jednostajny przy generowaniu punktów, prawdopodobieństwo wylosowania punktów blisko linii, było małe.

W zbiorze B dla danego wyznaczniki były zawsze takie same niezależnie od tolerancji, co można argumentować tym, że punkty które wylosowały się blisko zera, ze względu na dokładność obliczeń pythona stawały się równe zero, mogło na to wpłynąć to, że liczby wygenerowane w zbiorze B, były bardzo duże w porównaniu z innymi zestawami.

Największe różnice zaobserwowano w zbiorze D, dla tolerancji rzędu 10^{-10}, jeszcze wszystkie wyznaczniki ze 100% dokładnością kwalifikowały wszystkie punkty na prostej. Jednak przy tolerancji 10^{-12} skuteczniejsze okazywały się wyznaczniki 3x3, które wciąż miały 100% dokładność, a wyznaczniki 2x2 miały skuteczność w granicach 78 – 85 %.

Jednak, gdy tolerancja zaczęła się zmniejszać, dokładność wyznaczników 3x3 znacznie się obniżyła, w przypadku wyznacznika 3x3 numpy nawet do niespełna 30%. Podczas, gdy dokładność wyznaczników 2x2 także się zmniejszyła, nie była ona tak drastyczna jak drugich, dlatego dla mniejszych dokładności są one znacznie lepsze.

Podsumowując, według mnie najlepszym wyborem jest wyznacznik 3x3 własnej implementacji, zarówno ze względu na czas działania jak i precyzje obliczeń.