

Algorytmy geometryczne – sprawozdanie z ćw. 1

1. Cel ćwiczenia:

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

2. Dane techniczne:

Język implementacji: Python

Środowisko programistyczne: Jupyter Notebook

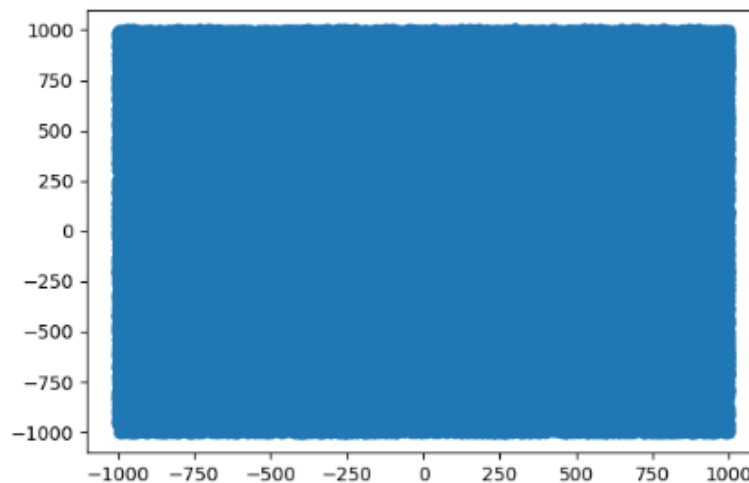
System operacyjny: Microsoft Windows 10 Pro x64

Procesor: Intel(R) Core(TM) i5-10400F CPU @ 2.90GHz, 2904 MHz

3. Zestawy danych i ich wizualizacja:

Na początku zaimplementowałem kod, w którym tworzę cztery zbiory punktów oraz wizualizuje je:

- 10^5 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ (**Wykres 1**),



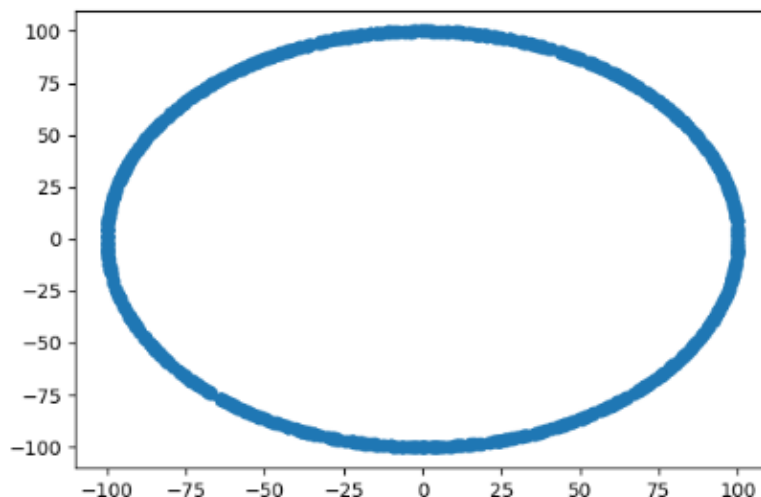
Wykres 1 Zestaw danych 1

- 10^5 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$ (**Wykres 2**),



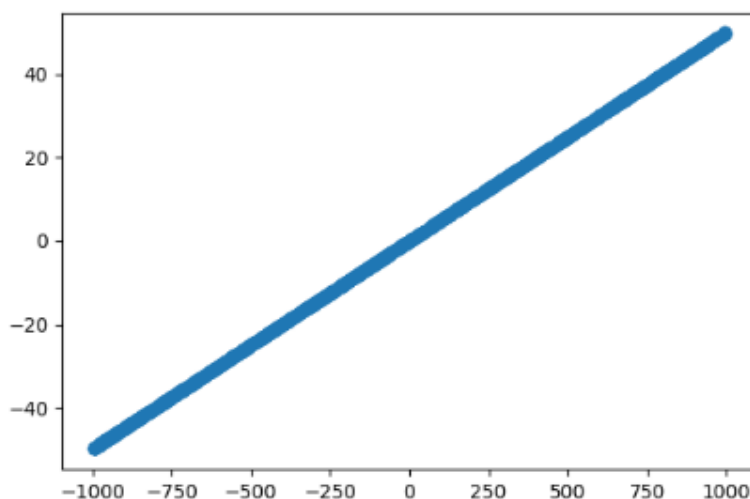
Wykres 2 Zestaw danych 2

- 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu $R=100$ (**Wykres 3**),



Wykres 3 Zestaw danych 3

- 1000 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ leżących na prostej wyznaczonej przez wektor (a, b) , gdzie $a = [-1.0, 0.0]$, $b = [1.0, 0.1]$ (**Wykres 4**).



Wykres 4 Zestaw danych 4

4. Implementacja wyznaczników:

Zaimplementowałem własne funkcje liczące wyznaczniki macierzy 2×2 , 3×3 (odpowiednio $det_{2 \times 2}$, $det_{3 \times 3}$) oraz korzystając z biblioteki Numpy $det_{N \times N_lib}$. Przetestowałem również szybkość działania tych funkcji na zbiorze danych o liczebności 10^5 .

Funkcja	$det_{2 \times 2}$	$det_{N \times N_lib}$ dla 2×2	$det_{3 \times 3}$	$det_{N \times N_lib}$ dla 3×3
Czas obliczeń	0,0409s	0,5585s	0,0838s	0,6443s

Tabela 1 Porównanie czasów wykonywania danych funkcji

5. Kategoryzacja punktów:

Zaimplementowałem funkcję `categorize_points`, która dla danego zbioru punktów X , punktów a, b oraz podanych trzech dokładności $eps=(10^{-6}, 10^{-12}, 10^{-18})$ przyporządkowuje położenie każdego punktu z X względem prostej wytyczonej przez punkty a, b . Ze względu na ograniczoną dokładność sprzętu, dane dokładności eps służą do umożliwiania określenia danego punktu jako leżącego na podanej prostej, a użycie różnych funkcji liczących wyznacznik macierzy możliwość ich porównania.

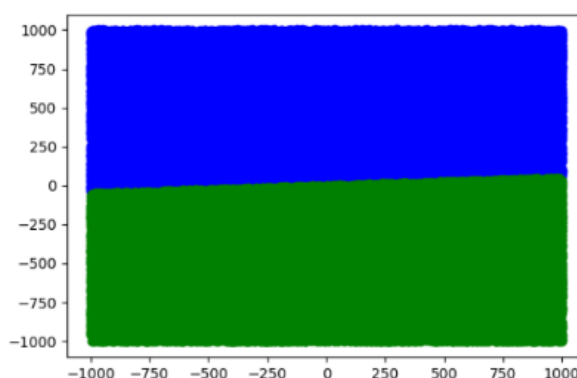
6. Porównywanie wyników dla danych funkcji liczących wyznaczniki przy podanym ϵ :

W tabelach poniżej zamieściłem w zależności od użytej funkcji liczącej wyznacznik liczbę punktów na każdym możliwym położeniu prostej. Za wielkość tolerancji dla zera przyjąłem $\epsilon=10^{-12}$. Poniżej każdej tabeli zamieściłem wykresy odnoszące się bezpośrednio do wyników z tabel.

- Zestaw danych 1:

Położenie	<i>detNxN_lib</i> dla 2x2	<i>det2x2</i>	<i>detNxN_lib</i> dla 3x3	<i>det3x3</i>
Po lewej	50291	50291	50291	50291
Po prawej	49709	49709	49709	49709
Współliniowe	0	0	0	0

Tabela 2 Porównanie dla zestawu danych 1 przy $\epsilon=10^{-12}$

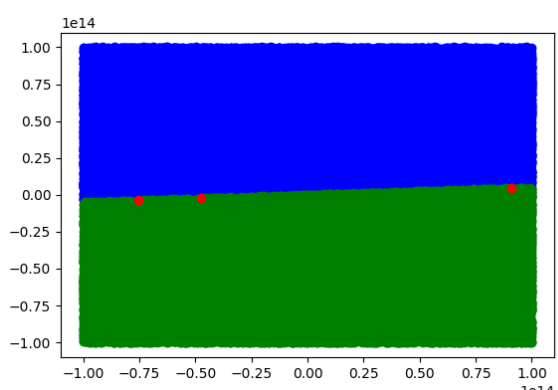


Wykres 5 We wszystkich wariantach identyczny przy $\epsilon=10^{-12}$

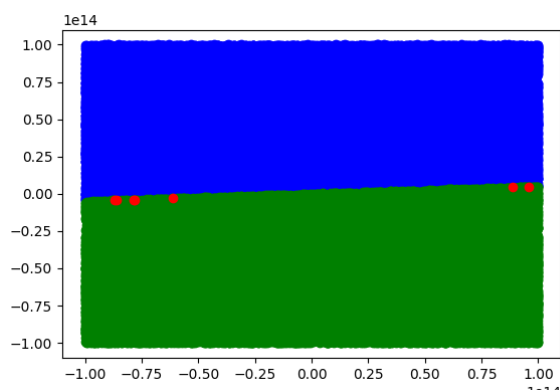
- Zestaw danych 2:

Położenie	<i>detNxN_lib</i> dla 2x2	<i>det2x2</i>	<i>detNxN_lib</i> dla 3x3	<i>det3x3</i>
Left	49738	49736	49739	49739
Right	50259	50259	50261	50261
Collinear	3	5	0	0

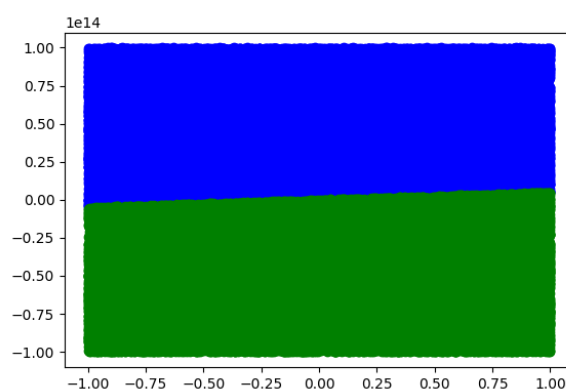
Tabela 3 Porównanie dla zestawu danych 2 przy $\epsilon=10^{-12}$



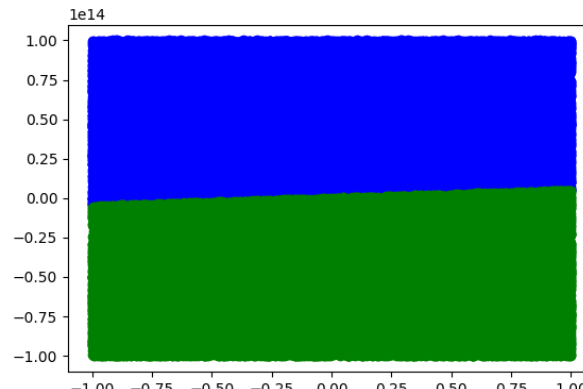
Wykres 6 Wyznacznik 2x2 z Numpy



Wykres 7 Wyznacznik 2x2 własny



Wykres 8 Wyznacznik 3x3 z Numpy

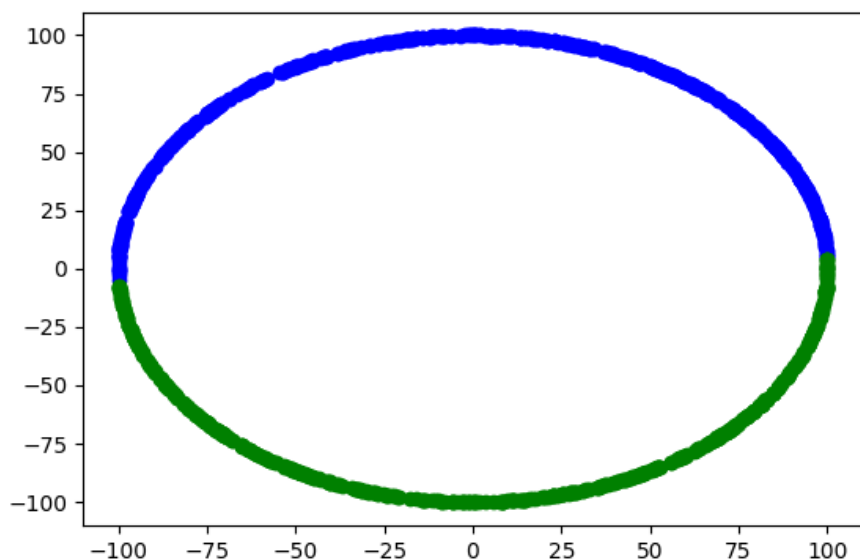


Wykres 9 Wyznacznik 3x3 własny

- Zestaw danych 3:

Położenie	<i>detNxN_lib</i> dla 2x2	<i>det2x2</i>	<i>detNxN_lib</i> dla 3x3	<i>det3x3</i>
Left	494	496	496	494
Right	506	504	504	506
Collinear	0	0	0	0

Tabela 4 Porównanie dla zestawu danych 3 przy $\text{eps} = 10^{-12}$

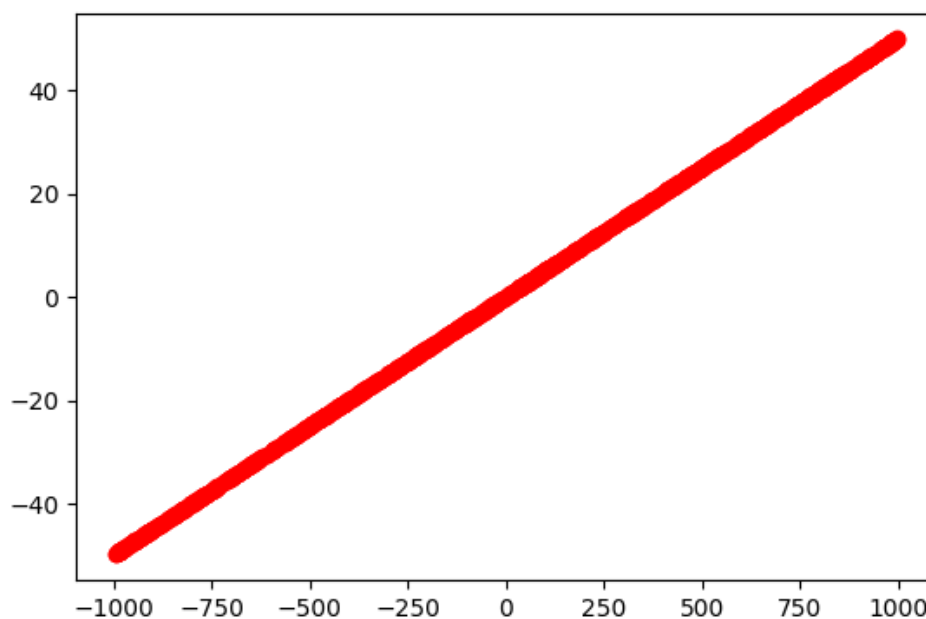


Wykres 10 We wszystkich wariantach identyczny przy $\text{eps} = 10^{-12}$

- Zestaw danych 4:

Położenie	<i>detNxN_lib</i> dla 2x2	<i>det2x2</i>	<i>detNxN_lib</i> dla 3x3	<i>det3x3</i>
Left	88	75	0	0
Right	101	75	0	0
Collinear	811	850	1000	1000

Tabela 5 Porównanie dla zestawu danych 2 przy $\text{eps} = 10^{-12}$



Wykres 10 We wszystkich wariantach praktycznie identyczny przy $\text{eps} = 10^{-12}$

7. Porównywanie wyników dla różnych dokładności oraz wyznaczników:

W tabelach poniżej zamieściłem wyniki obliczeń dla różnych dokładności *epsilon*, zestawów danych oraz funkcji liczących wyznaczniki macierzy.

- Zestaw danych 1

<i>Epsilon</i>	<i>detNxN_lib</i> dla 2x2			<i>det2x2</i>			<i>detNxN_lib</i> dla 3x3			<i>det3x3</i>		
	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.
10^{-6}	50291	49709	0	50291	49709	0	50291	49709	0	50291	49709	0
10^{-12}	50291	49709	0	50291	49709	0	50291	49709	0	50291	49709	0
10^{-18}	50291	49709	0	50291	49709	0	50291	49709	0	50291	49709	0

Tabela 6 Pełne porównanie wyników dla zestawu danych 1

- Zestaw danych 2

<i>Epsilon</i>	<i>detNxN_lib</i> dla 2x2			<i>det2x2</i>			<i>detNxN_lib</i> dla 3x3			<i>det3x3</i>		
	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.
10^{-6}	49738	50259	3	49736	50259	5	49736	50261	0	49736	50261	0
10^{-12}	49738	50259	3	49736	50259	5	49736	50261	0	49736	50261	0
10^{-18}	49738	50259	3	49736	50259	5	49736	50261	0	49736	50261	0

Tabela 7 Pełne porównanie wyników dla zestawu danych 2

- Zestaw danych 3

<i>Epsilon</i>	<i>detNxN_lib</i> dla 2x2			<i>det2x2</i>			<i>detNxN_lib</i> dla 3x3			<i>det3x3</i>		
	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.
10^{-6}	494	506	0	494	506	0	494	506	0	494	506	0
10^{-12}	494	506	0	494	506	0	494	506	0	494	506	0
10^{-18}	494	506	0	494	506	0	494	506	0	494	506	0

Tabela 8 Pełne porównanie wyników dla zestawu danych 3

- Zestaw danych 4

<i>Epsilon</i>	<i>detNxN_lib</i> dla 2x2			<i>det2x2</i>			<i>detNxN_lib</i> dla 3x3			<i>det3x3</i>		
	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.	Lewo	Prawo	Wsp.
10^{-6}	0	0	1000	0	0	1000	0	0	1000	0	0	1000
10^{-12}	88	101	811	75	75	850	0	0	1000	0	0	1000
10^{-18}	144	137	719	150	138	712	398	326	276	195	404	401

Tabela 9 Pełne porównanie wyników dla zestawu danych 4

8. Podsumowanie i wnioski:

Podczas wykonywania ćwiczenia 1. udało mi się dobrze zapoznać z środowiskiem Jupyter Notebook oraz oswoić się z programem graficznym. Zaimplementowałem funkcję tworzącą dane zbiory punktów, określające położenia tych punktów względem podanej prostej oraz wizualizujące je. Z testów czasowych funkcji wyznaczników wynika, że te zaimplementowane przeze mnie są szybsze niż te korzystające z biblioteki Numpy. Z analizy wyników obliczeń można stwierdzić, że arytmetyka komputera jest dosyć ograniczona i nie zawsze istnieje możliwość na uzyskania obliczeń zgodnych z rzeczywistością. Widzimy to szczególnie w tabeli 9, gdzie wszystkie punkty powinny zostać skategoryzowane jako współliniowe. Bardzo niska tolerancja dla zera również (10^{-18}) również powoduje błędy w obliczeniach. Z pełnej analizy danych nasuwa się wniosek, że najwierniejsze wyniki można uzyskać używając funkcji *detNxN_lib* dla macierzy 3x3 lub *det3x3* z tolerancją dla zera równą około 10^{-12} .