Algorytmy geometryczne

Sprawozdanie z ćwiczenia 1.

Krzysztof Kwiecień
gr. Wt_14.40_B

Dane techniczne urządzenia na którym wykonano ćwiczenie:

Komputer z systemem Windows 10 x64
Procesor: Intel Core I5 2500k @4.5Ghz
Pamięć RAM: 8GB
Środowisko: Jupyter notebook

Ćwiczenie zrealizowano w języku Python 3, z wykorzystaniem bibliotek *numpy* (umożliwiających wiele operacji numerycznych), *random* (do generacji losowych punktów) oraz *matplotlib* (do rysowania wykresów).

Opis realizacji ćwiczenia:

Ćwiczenie polegało na porównaniu wyników klasyfikacji położenia punktów względem odcinka w zależności od wybranej metody obliczania wyznacznika oraz tolerancji dla zera.

1. Generacja punktów

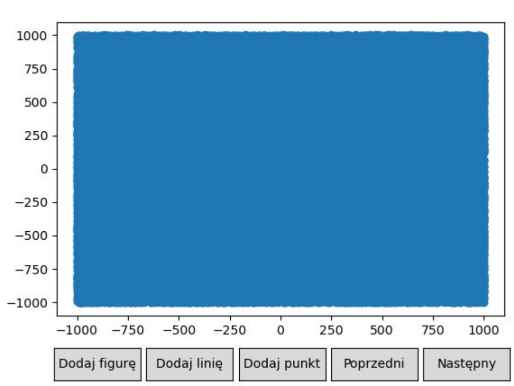
W celu wykonania ćwiczenia wygenerowane zostały 4 zbiory punktów (2D, typu double), które zostały umieszczone w osobnych tablicach:

- Zestaw 1.
  100 000 losowych punktów o współrzędnych z przedziału [-1000, 1000]
- Zestaw 2.
  100 000 losowych punktów o współrzędnych z przedziału [-10^{14}, 10^{14}],
- Zestaw 3.
  1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu R=100,
- Zestaw 4.
  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].

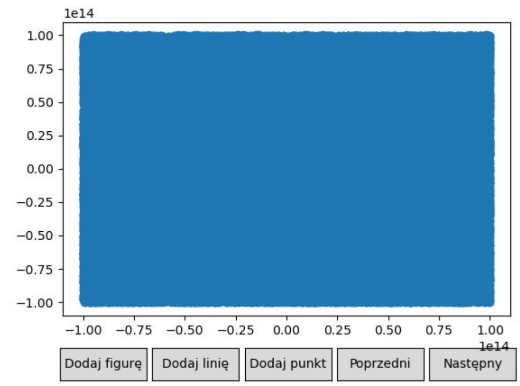
Losowanie położenia punktów odbyło się za pomocą metody random.uniform z biblioteki random. Funkcja ta generuje liczby typu double dla zadanego zakresu obustronnie domkniętego.
Punkty z zestawu 3. zostały wygenerowane z wykorzystaniem funkcji trygonometrycznych z biblioteki numpy.
Punkty z zestawu 4. zostały ułożone na prostej wyznaczonej poprzez równanie prostej przechodzącej przez 2 punkty dla podanych punktów (a, b).

Wszystkie zestawy punktów zostały zwizualizowane za pomocą dostarczonego narzędzia graficznego opartego o bibliotekę *matplotlib*.

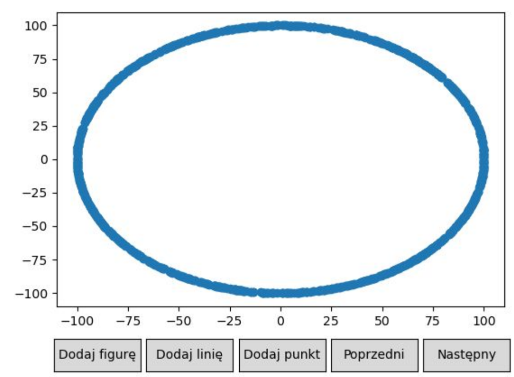
Wykres 1.1 Zestaw danych 1.



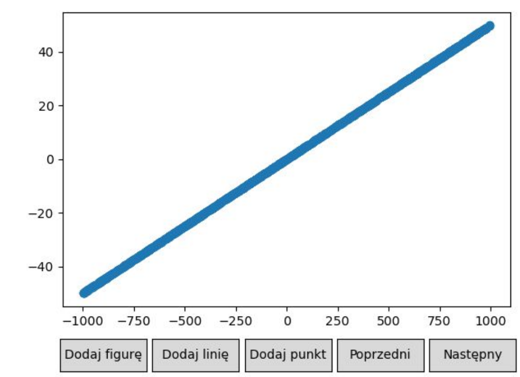
Wykres 1.2 Zestaw danych 2.



Wykres 1.3 Zestaw danych 3.



Wykres 1.4 Zestaw danych 4.



2. Metody obliczania wyznacznika oraz tolerancje dla zera

Kolejnym krokiem było skategoryzowanie położenia punktów względem prostej w zależności od metod obliczania wyznacznika oraz tolerancji dla zera przy określaniu położenia na podstawie wyznacznika.

Wykorzystane zostały następujące wyznaczniki:
- Wyznacznik 2x2 obliczany za pomocą metody z biblioteki numpy
- Wyznacznik 2x2 obliczany za pomocą własnej funkcji
- Wyznacznik 3x3 obliczany za pomocą metody z biblioteki numpy
- Wyznacznik 3x3 obliczany za pomocą własnej funkcji

Natomiast jako rozważanie tolerancje dla zera przyjęto:
- 10^{-18}
- 10^{-14}
- 10^{-10}

Tabela 1. Zastosowane wyznaczniki oraz metody ich obliczania

Nazwa	**Metoda obliczania**
Wyznacznik 2x2 z biblioteki numpy	Metoda z biblioteki numpy: numpy.linalg.det()
Wyznacznik 2x2 własnej implementacji	$\det(a,b,c) = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$
Wyznacznik 3x3 z biblioteki numpy	Metoda z biblioteki numpy: numpy.linalg.det()
Wyznacznik 3x3 własnej implementacji	$\det(a,b,c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}$

Gdzie *a*, *b* to punkty wyznaczające odcinek na podstawie którego dokonujemy klasyfikacji, natomiast *c* jest sprawdzanym punktem.

3. Kategoryzacja punktów

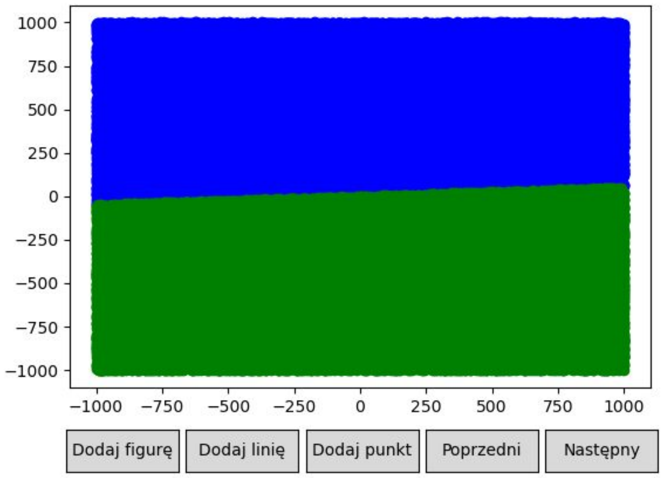
W celu zrealizowania zadania wykorzystana została funkcja tworząca dla każdego zestawu punktów tablicę dwuwymiarową, której kolejne wiersze odpowiadają kolejnym metodom obliczania wyznaczników, natomiast kolejne kolumny odpowiadają kolejnym tolerancjom dla zera. Każde pole tabeli zawiera słownik przechowywujący skategoryzowane punkty dla danej kombinacji metod. Dzięki zastosowaniu tej metody nie trzeba powtarzać ponownej kategoryzacji w różnych częściach programu.

4. Wizualizacja kategoryzacji

Jako pierwszy test wyświetlono zwizualizowane wyniki kategoryzacji dla wszystkich wyznaczników w każdym zestawie danych. Kolor niebieski oznacza, że punkt został sklasyfikowany jako leżący po lewej stronie odcinka, zielony po prawej stronie odcinka, a czerwony jako współliniowy z odcinkiem.
W tej części ćwiczenia tolerancja dla zera była taka sama dla wszystkich testów we wszystkich zestawach i wynosiła 10^{-18}. Dodatkowo wyświetlone zostały tekstowo informacje o liczbie punktów w każdej kategorii.
Już na tym etapie widać, że występują różnice w klasyfikacji w zależności od wyznacznika.

Dla pierwszego zestawu danych wyniki były takie same dla wszystkich wyznaczników.

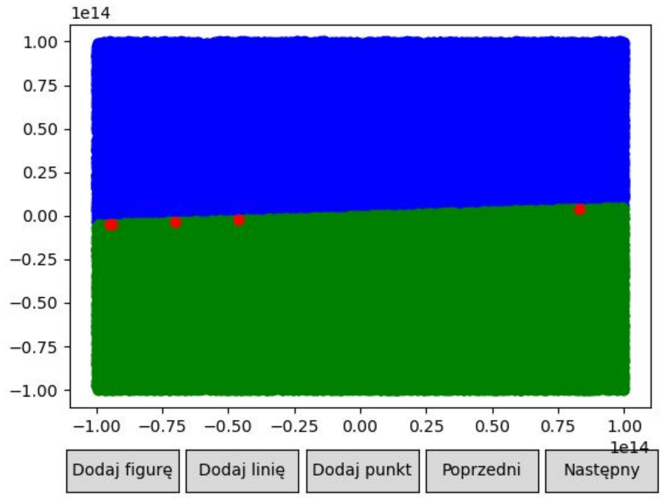
Wykres 2.1 Klasyfikacja punktów dla zestawu 1. dla wszystkich wyznaczników przy tolerancji 10^{-18}



Klasyfikacja: Po lewej: 49688 Po prawej: 50312 Współliniowe: 0

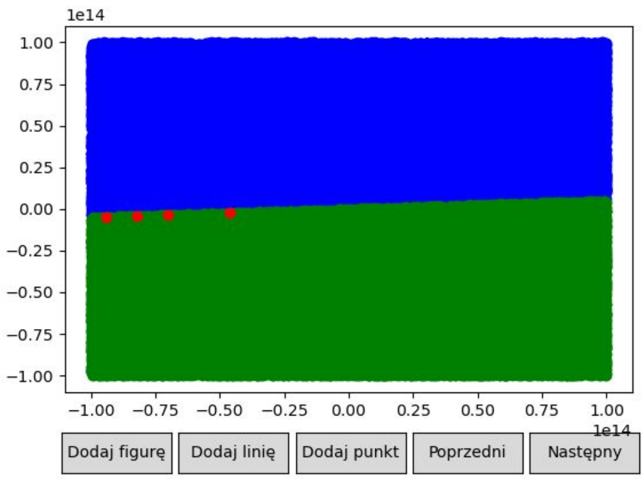
W drugim zestawie jednak wyniki były różne. Klasyfikacje punktów określone przez wyznaczniki 3x3 nie różniły się między sobą i nie zawierały punktów współliniowych. Dla wyznaczników 2x2 zaobserwowano klasyfikację kilku punktów jako współliniowe, co więcej były to różne punkty dla wyznacznika własnej implementacji oraz metody z biblioteki numpy. Cechą wspólną tychże punktów była duża odległość od środka układu współrzędnych, co oznaczało że współrzędne były bardzo dużymi liczbami (rzędu 10^{14}). Przykładowy punkt (dla wyznacznika 2x2 z biblioteki numpy) miał postać (79709734118793.9, 4010237245098.3906). To właśnie rozmiar tych liczb i błędy obliczeniowe związane z precyzją operacji arytmetycznych na tychże liczbach może uzasadniać inną klasyfikację między wyznacznikami.

Wykres 2.2.1 Klasyfikacja punktów dla zestawu 2. Dla wyznacznika 2x2 z biblioteki numpy przy tolerancji 10^{-18}



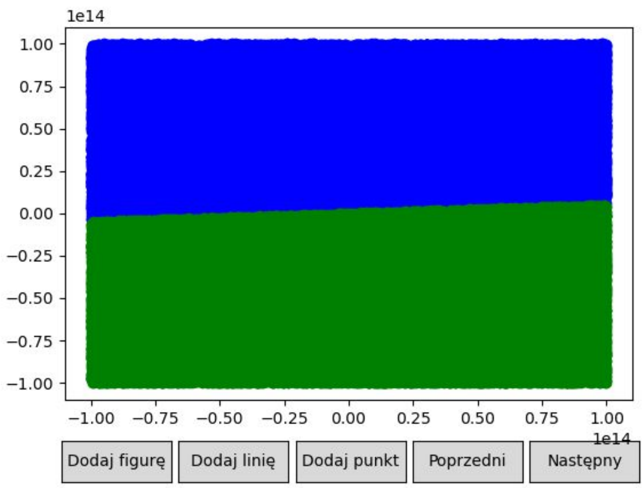
Klasyfikacja: Po lewej: 49953 Po prawej: 50042 Współliniowe: 5

Wykres 2.2.2 Klasyfikacja punktów dla zestawu 2. Dla wyznacznika 2x2 własnej implementacji przy tolerancji 10^{-18}



Klasyfikacja: Po lewej: 49954 Po prawej: 50042 Współliniowe: 4

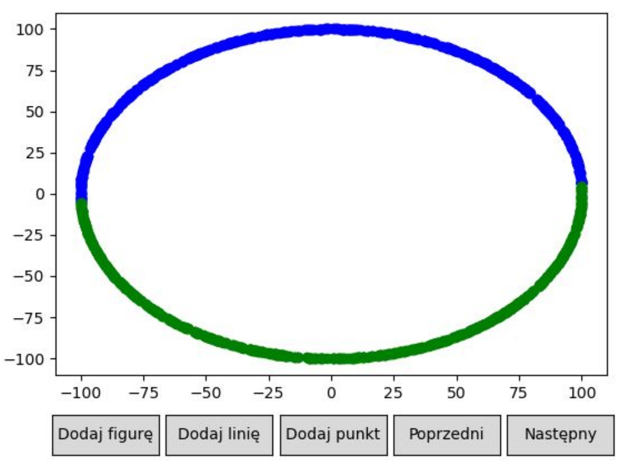
Wykres 2.2.3 Klasyfikacja punktów dla zestawu 2. Dla obu wyznaczników 3x3 przy tolerancji 10^{-18}



Klasyfikacja: Po lewej: 49955 Po prawej: 50045 Współliniowe: 0

W przypadku trzeciego zestawu wyniki ponownie był identycznie dla wszystkich wyznaczników.

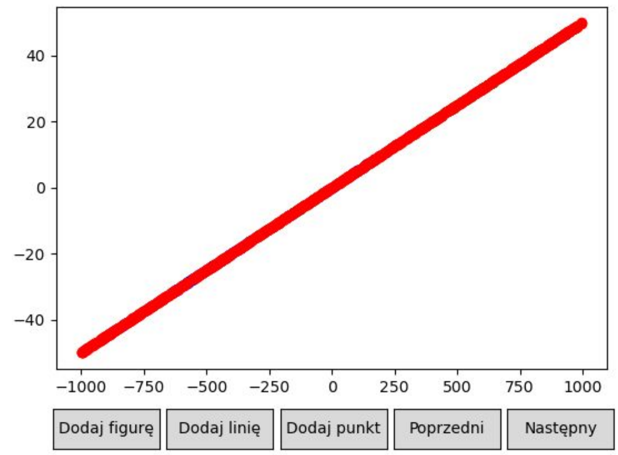
Wykres 2.3 Klasyfikacja punktów dla zestawu 3. Dla wszystkich wyznaczników przy tolerancji 10^{-18}



Klasyfikacja: Po lewej: 496 Po prawej: 504 Współliniowe: 0

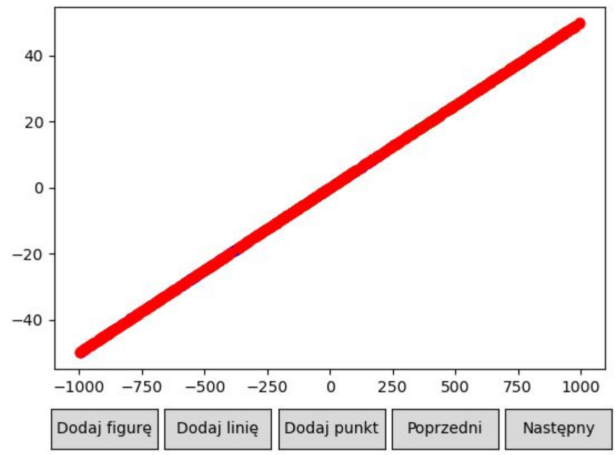
Dla czwartego zestawu zaobserwowano największe różnice podczas wstępnej wizualizacji. Teoretycznie wszystkie punkty powinny być zaklasyfikowane jako współliniowe, jednak w praktyce tak się nie stało. Powodem błędnej klasyfikacji punktów były błędy związane ze skończoną precyzją obliczeń, w celu niwelacji tychże błędów w dalszej części ćwiczenia zastosowano różne tolerancje dla zera. Dla wyznacznika 3x3 własnej implementacji ponownie obserwujemy różne klasyfikacje w zależności od rozmiaru liczb będących współrzędnymi punktów. Są one symetryczne względem środka odcinka.

Wykres 2.4.1 Klasyfikacja punktów dla zestawu 4. Dla wyznacznika 2x2 z biblioteki numpy przy tolerancji 10^{-18}



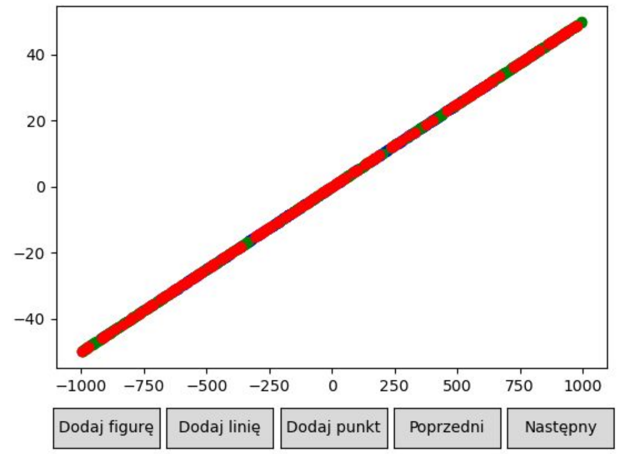
Klasyfikacja:
Po lewej: 95 Po prawej: 101 Współliniowe: 804

Wykres 2.4.2 Klasyfikacja punktów dla zestawu 4. Dla wyznacznika 2x2 własnej implementacji przy tolerancji 10^{-18}



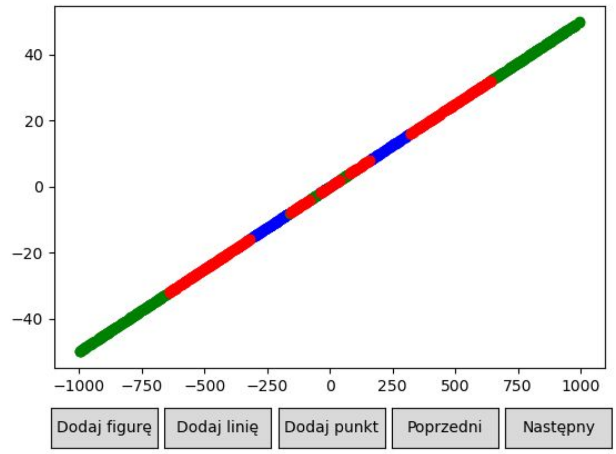
Klasyfikacja:
Po lewej: 127 Po prawej: 164 Współliniowe: 709

Wykres 2.4.3 Klasyfikacja punktów dla zestawu 4. Dla wyznacznika 3x3 z biblioteki numpy przy tolerancji 10^{-18}



Klasyfikacja:
Po lewej: 364 Po prawej: 439 Współliniowe: 197

Wykres 2.4.4 Klasyfikacja punktów dla zestawu 4. Dla wyznacznika 3x3 własnej implementacji przy tolerancji 10^{-18}



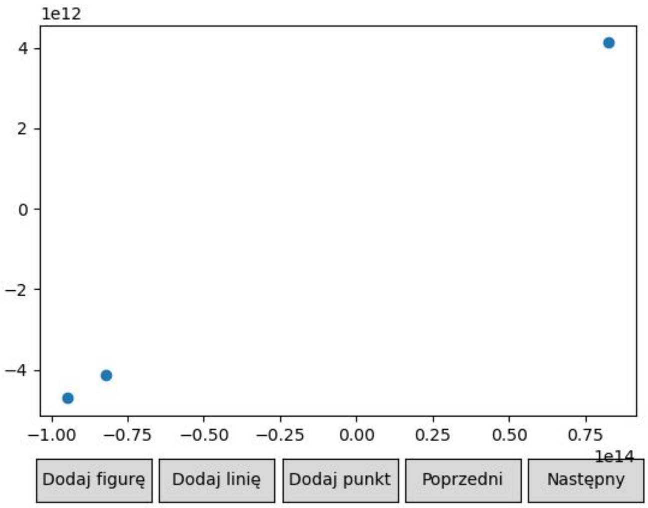
Klasyfikacja:
Po lewej: 158 Po prawej: 413 Współliniowe: 429

5. Porównywanie metod obliczania wyznacznika

Następna część ćwiczenia poświęcona była różnicom w klasyfikacji między metodami obliczania wyznaczników. W tym celu zastosowano funkcję, która dla danego zestawu punktów oraz metod obliczania wyznaczników zwraca listę punktów które różnią się klasyfikacją według tych wyznaczników. Różnice te przedstawiono w formie wykresów oraz dodatkowo tekstowo podano liczbę różnic.

Jak podano wcześniej, dla zestawów 1. oraz 3. nie zaobserwowano żadnych różnic.
W przypadku wyznaczników 2x2 własnej implementacji oraz z biblioteki numpy dla zestawu 2. znaleziono trzy różnice. Zgodnie z wcześniej przedstawioną tezą są to punkty, których współrzędne są ogromnymi liczbami, dla których błędy związane z precyzją obliczeń mogą powodować różne klasyfikacje.

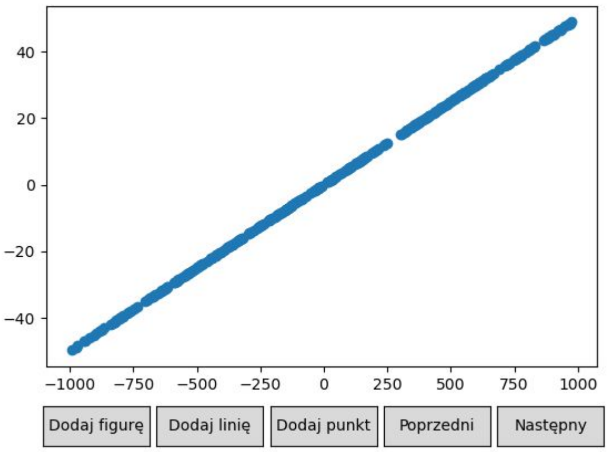
Wykres 3.1 Punkty skategoryzowane inaczej w zestawie 2. przez wyznaczniki 2x2 z biblioteki numpy oraz 2x2 własnej implementacji przy tolerancji 10^{-18}



Ilość różnic: *3*

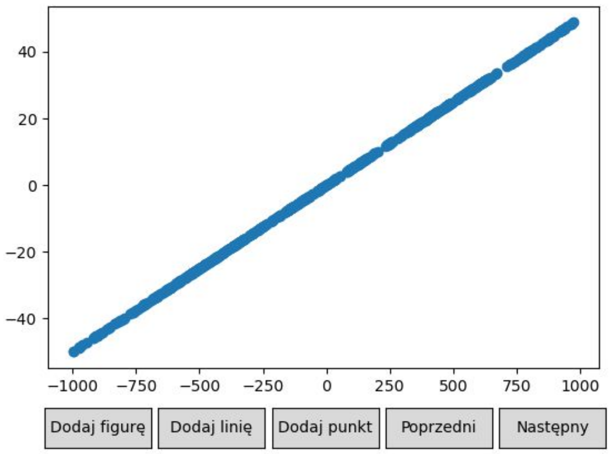
Dla czwartego zestawu danych różnice w klasyfikacji punktów między wyznacznikami są bardzo duże i sięgają nawet 78%. Wynika to z natury wyboru punktów w tym zestawie - nawet drobne różnice w błędach obliczeniowych między wyznacznikami mogą prowadzić do różnych klasyfikacji punktów, ponieważ wartości wyznaczników dla punktów znajdujących się na prostej są bardzo bliskie zera.

Wykres 3.2.1 Punkty skategoryzowane inaczej w zestawie 4. przez wyznaczniki 2x2 z biblioteki numpy oraz 2x2 własnej implementacji przy tolerancji 10^{-18}



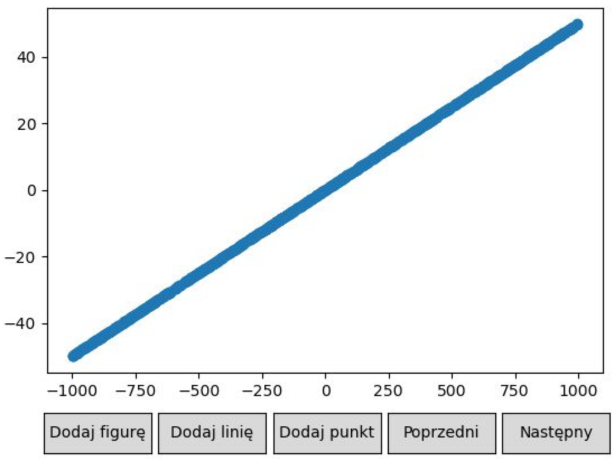
Ilość różnic: *343*

Wykres 3.2.2 Punkty skategoryzowane inaczej w zestawie 4. przez wyznaczniki 3x3 z biblioteki numpy oraz 3x3 własnej implementacji przy tolerancji 10^{-18}



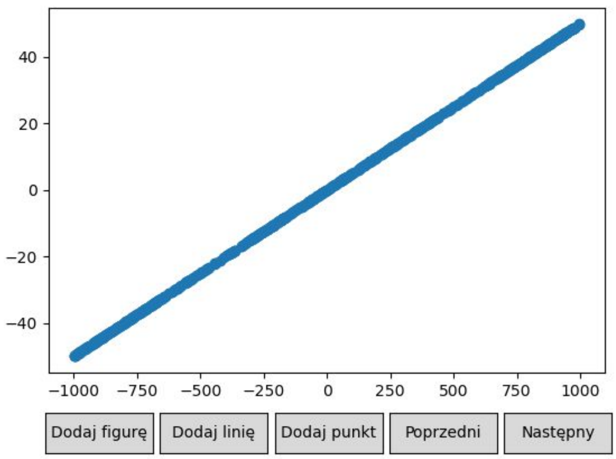
Ilość różnic: *535*

Wykres 3.2.3 Punkty skategoryzowane inaczej w zestawie 4. przez wyznaczniki 2x2 i 3x3 z biblioteki numpy przy tolerancji 10^{-18}



Ilość różnic: *777*

Wykres 3.2.4 Punkty skategoryzowane inaczej w zestawie 4. przez wyznaczniki 2x2 i 3x3 własnej implementacji przy tolerancji 10^{-18}



Ilość różnic: *687*

6. Analiza wyników z uwzględnieniem wszystkich parametrów

Ostatnia część ćwiczenia polegała na kompleksowej analizie wyników z uwzględnieniem zarówno metody obliczania wyznacznika jak i tolerancji dla zera. W tym celu wygenerowano tabele zawierającą informacje o kategoryzacji punktów dla każdej kombinacji wyznacznika i tolerancji dla zera.

Wnioski uzyskane w związku z analizą poniższych danych przedstawiono w ostatniej części sprawozdania.

Tabela 2.1 Ilość punktów w każdej klasyfikacji dla zestawu 1. w zależności od użytego wyznacznika i tolerancji dla zera

Wyznacznik	Tolerancja dla zera	Po lewej	Po prawej	Współliniowe
wszystkie testowane	wszystkie testowane	49688	50312	0

Tabela 2.2 Ilość punktów w każdej klasyfikacji dla zestawu 2. w zależności od użytego wyznacznika i tolerancji dla zera

Wyznacznik	Tolerancja dla zera	Po lewej	Po prawej	Współliniowe
2x2 z biblioteki numpy	10^{-18}	49953	50042	5
	10^{-14}	49953	50042	5
	10^{-10}	49953	50042	5
2x2 własnej implementacji	10^{-18}	49954	50042	4
	10^{-14}	49954	50042	4
	10^{-10}	49954	50042	4
3x3 z biblioteki numpy	10^{-18}	49955	50045	0
	10^{-14}	49955	50045	0
	10^{-10}	49955	50045	0
3x3 własnej implementacji	10^{-18}	49955	50045	0
	10^{-14}	49955	50045	0
	10^{-10}	49955	50045	0

Tabela 2.3 Ilość punktów w każdej klasyfikacji dla zestawu 3. w zależności od użytego wyznacznika i tolerancji dla zera

Wyznacznik	Tolerancja dla zera	Po lewej	Po prawej	Współliniowe
wszystkie testowane	wszystkie testowane	496	504	0

Tabela 2.4 Ilość punktów w każdej klasyfikacji dla zestawu 4. w zależności od użytego wyznacznika i tolerancji dla zera

Wyznacznik	Tolerancja dla zera	Po lewej	Po prawej	Współliniowe
2x2 z biblioteki numpy	10^{-18}	95	101	804
	10^{-14}	85	91	824
	10^{-10}	0	0	1000
2x2 własnej implementacji	10^{-18}	127	164	709
	10^{-14}	111	147	742
	10^{-10}	0	0	1000
3x3 z biblioteki numpy	10^{-18}	364	439	197
	10^{-14}	0	0	1000
	10^{-10}	0	0	1000
3x3 własnej implementacji	10^{-18}	158	413	429
	10^{-14}	0	0	1000
	10^{-10}	0	0	1000

Wnioski i spostrzeżenia

Jak widać na podstawie powyższych danych różnice w klasyfikacji położenia punktu względem odcinka w zależności od metody obliczania wyznacznika są znaczące. Dodatkowe rozpatrywanie różnych tolerancji dla zera podczas klasyfikacji jeszcze bardziej poszerza zakres wyników dla tych samych danych.
Różnice te występują jednak tylko w niektórych przypadkach. Dla zestawów 1. oraz 3. wcale ich nie zaobserwowano.

Różnice występują tylko bardzo blisko linii wyznaczonej przez odcinek na podstawie którego dokonujemy klasyfikacji. Z tego powodu łatwo jest wyjaśnić spójną klasyfikację danych z zestawu 3, ponieważ w pobliżu tejże linii jest bardzo mała liczba punktów. Na poprawność klasyfikacji ma również wielkość liczb.

Najciekawszym zestawem do rozważenia jest zestaw 4. Teoretycznie wszystkie punkty powinny zostać sklasyfikowane jako współliniowe z odcinkiem. Jednakże wcale się tak nie dzieje, w zależności od wybranych metod obliczeń dostajemy zupełnie inne, bardzo różniące się między sobą wyniki. Pierwszym spostrzeżeniem jest poprawna klasyfikacja wszystkich punktów przez wszystkie wyznaczniki dla tolerancji dla zera na poziomie 10^{-10}. Przy tak dużej tolerancji błędy obliczeniowe spowodowane skończoną precyzją komputera są pomijane dla wszystkich wyznaczników. Dla tak dużej tolerancji nie ma znaczenia który wyznacznik weźmiemy. Jeżeli jednak chcemy zaostrzyć tolerancję wybór nie jest już taki prosty.

Dla bardzo niskiej tolerancji na poziomie 10^{-18} w porównaniu najlepiej wypadają wyznaczniki 2x2, w szczególności najlepszy jest ten z biblioteki numpy, z ok. 80% dokładnością. W tym przypadku wyznaczniki 3x3 pozostają dużo w tyle z dokładnością jedynie ok. 30%. Jednakże już dla nieco większej tolerancji, na poziomie 10^{-14} sytuacja zupełnie się zmienia. Dla wyznaczników 2x2 dokładność zwiększa się zaledwie o kilka procent, natomiast dla obu wyznaczników 3x3 sięga ona 100%.

W związku z tymi różnicami widzimy, że nie istnieje jedna najlepsza metoda klasyfikacji, nie da się nawet wyłonić najlepszych składowych tejże klasyfikacji. Wyznacznik należy dobrać w zależności od potrzeb i wymaganych tolerancji tak, aby ich kombinacja dawała rezultaty jak najbardziej zbliżone do teoretycznych.

Jako kombinację wyznacznika i tolerancji dla przyszłych ćwiczeń wybieram wyznacznik 3x3 własnej implementacji przy tolerancji 10^{-14}. Kombinacja ta daje dobre wyniki dla wszystkich zestawów oraz zachowuje stosunkowo małą tolerancję, dzięki czemu znaczna większość klasyfikacji będzie zgodna z teoretyczną.