

Algorytmy geometryczne

SPRAWOZDANIE Z ĆWICZEŃ NR. 1

WOJCIECH ŁOBODA

GRUPA NR. 1

(ŚRODA_13_P_1)

Dane urządzenia na którym wykonano ćwiczenia:

Komputer z systemem Windows 10 x64

- Procesor: Intel Core i7-7700HQ 2.80 GHz
- Pamięć RAM: 8 GB
- Środowisko: Jupyter notebook

Ćwiczenie wykonano przy pomocy języka python3 z wykorzystaniem bibliotek:

- *Numpy* - obliczenia na macierzach.
- *Matplotlib* - wizualizacja danych.
- *Random* - generowanie punktów.
- *Pandas* - zapisywanie danych.

Opis realizacji ćwiczenia:

Ćwiczenie polegało na wygenerowaniu odpowiednich zbiorów punktów, a następnie na klasyfikacji punktów względem ich położenia względem prostej wyznaczonej przez dwa podane punkty. Należało dokonać obserwacji zmian wyników przy różnych wartościach dla tolerancji dla zera.

1. Generacja punktów

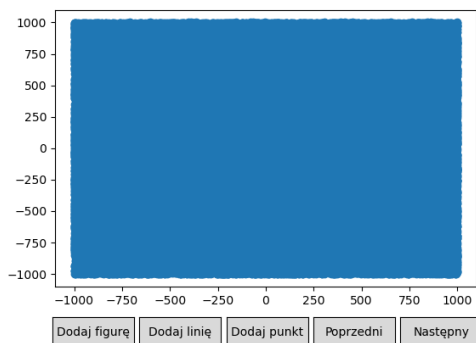
W celu wykonania obliczeń wygenerowane zostały 4 zbioru punktów należących do \mathbb{R}^2 typu double. Punkty zostały umieszczone w osobnych listach:

- Zestaw A (points_a): 100 000 losowych punktów o współrzędnych z przedziału $[1000, 1000]$.
- Zestaw B (points_b): 100 000 losowych punktów o współrzędnych z przedziału $[10^{14}, 10^{14}]$.
- Zestaw C (points_c): 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu $R=100$
- Zestaw D (points_d): 1000 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ leżących na prostej wyznaczonej przez wektor (a, b), $a = [-1.0, 0.0]$, $b = [1.0, 0.1]$

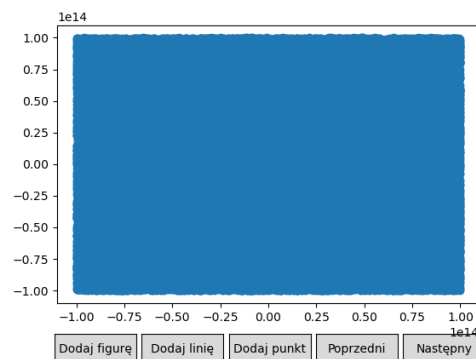
Punkty zostały wygenerowane przy pomocy funkcji random.uniform która zwracała losową liczbę typu double z domkniętego przedziału. Przy generowaniu zestawu C wykorzystane zostało parametryczne równanie okręgu oraz funkcje trygonometryczne z biblioteki numpy. Zestaw D został wygenerowany przy pomocy równania prostej przechodzącej przez punkty a i b.

Wykorzystane równanie okręgu dla zbioru C: $C(t) = \left(\cos\left(\frac{\pi}{2}t\right), \sin\left(\frac{\pi}{2}t\right) \right), t \in [0, 4]$

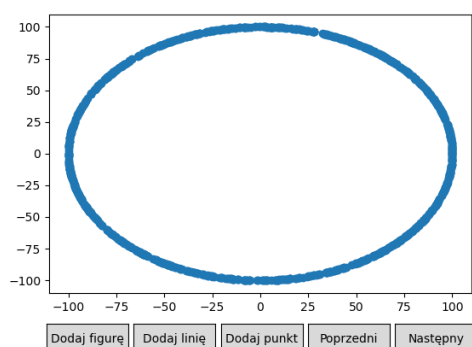
Wykres 1.1 Zestaw A



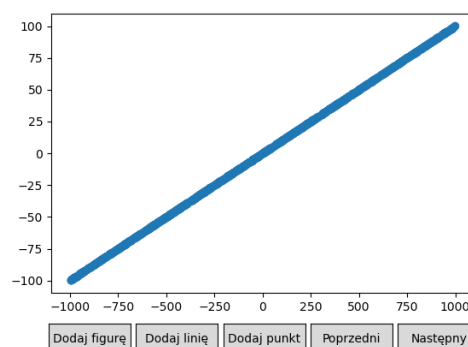
Wykres 1.2 Zestaw B



Wykres 1.3 Zestaw C



Wykres 1.4 Zestaw D



2. Metody obliczania wyznacznika, tolerancje dla zera oraz precyzje obliczeń

Wykorzystane wyznaczniki:

- Wyznacznik 2x2 policzony za pomocą metody z biblioteki numpy
- Wyznacznik 2x2 policzony za pomocą własnej funkcji
- Wyznacznik 3x3 policzony za pomocą metody z biblioteki numpy
- Wyznacznik 3x3 policzony za pomocą własnej funkcji

Przyjmowane tolerancje dla zera (epsilon):

- 10^{-18}
- 10^{-14}
- 10^{-10}
- 10^{-2}

Nazwa	Metoda obliczania
Wyznacznik 2x2 z biblioteki numpy	Metoda <code>numpy.linalg.det()</code>
Wyznacznik 2x2 własna implementacja	$\det(a, b, c) = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$
Wyznacznik 3x3 z biblioteki numpy	Metoda <code>numpy.linalg.det()</code>
Wyznacznik 3x3 własna implementacja	$\det(a, b, c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}$

*a i b to punkty wyznaczające odcinek, c to punkt klasyfikowany

3. Klasyfikacja punktów

Do klasyfikacji punktów wykorzystałem funkcję *qualify()* która przyjmuje jako argumenty zestaw punktów, ustaloną tolerancję dla zera. Funkcja zwraca listę która dla i-tego punktu (i to indeks punktu w wygenerowanej liście) zawiera informacje jak został zakwalifikowany. Zmiana tolerancji odbywała się poprzez zmianę wartości zmiennej *eps*.

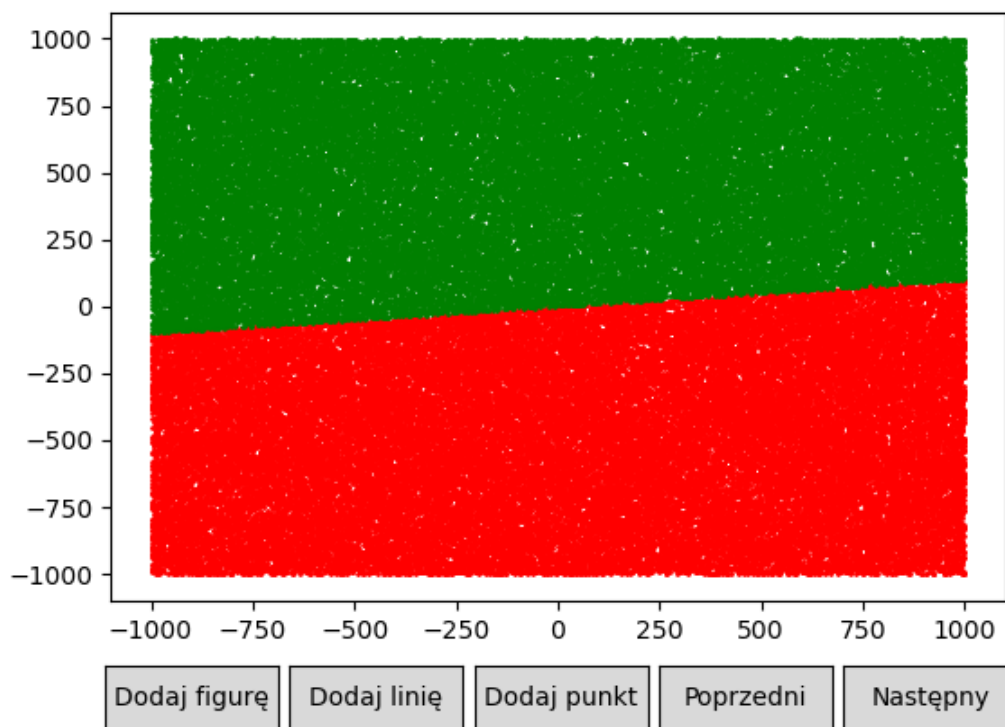
4. Wizualizacja klasyfikacji

Poniżej wyświetlono zwizualizowane wyniki klasyfikacji dla wszystkich wyznaczników w każdym zestawie danych. Kolor czerwony oznacza że punkt znajdował się po prawej stronie prostej, kolor zielony, że punkt znajdował się po lewej stronie prostej a kolor niebieski że punkt był współliniowy z prostą. Na tym etapie tolerancja dla zera była stała i ustalona dla wszystkich przypadków na 10^{-18} . Przy wykresie znajduje się informacja o liczbie punktów w każdej kategorii.

Zestaw A:

Punkty zostały tak samo sklasyfikowane dla każdego sposobu liczenia wyznacznika.

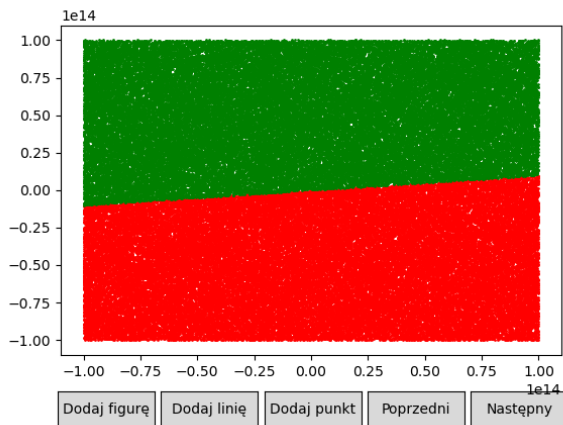
Wykres 2.1 Klasyfikacja punktów z zestawu A dla wszystkich sposobów liczenia wyznacznik



Zestaw B:

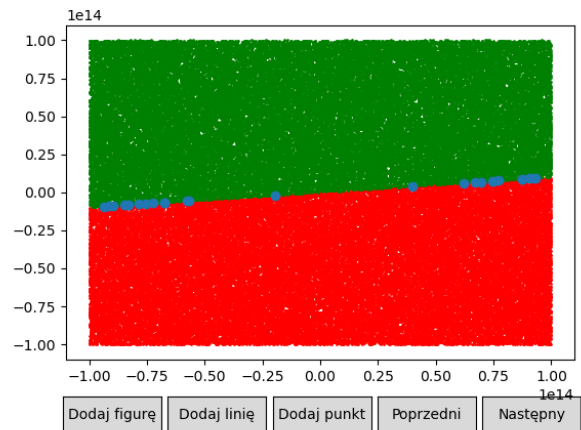
W zestawie B pojawiły się różnice w klasyfikacji. Wyniki dla wyznaczników 3x3 były takie same. W przypadku własnej implementacji wyznacznika 2x2 pojawiło się parę punktów współliniowych. Przy wykorzystaniu bibliotecznego wyznacznika 2x2 jeden punkt został zakwalifikowany inaczej niż dla wyznaczników 3x3.

Wykres 2.2.1 Zestaw B, biblioteczny wyznacznik 2x2



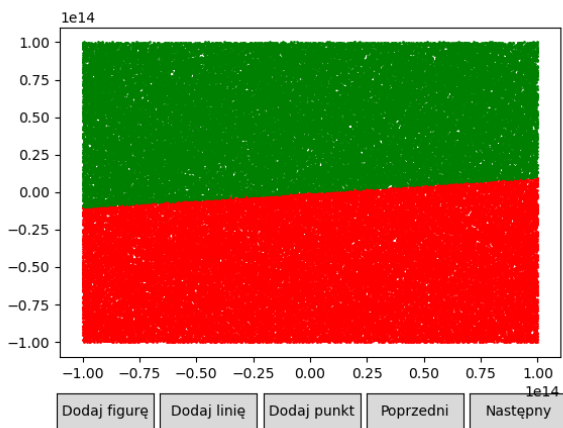
Po prawej: 49979, Po lewej: 50021, Współliniowe: 0

Wykres 2.2.2 Zestaw B, własna implementacja wyznacznika 2x2



Po prawej: 49967, Po lewej: 50006, Współliniowe: 27

Wykres 2.2.3 Klasyfikacja punktów dla wyznaczników 3x3

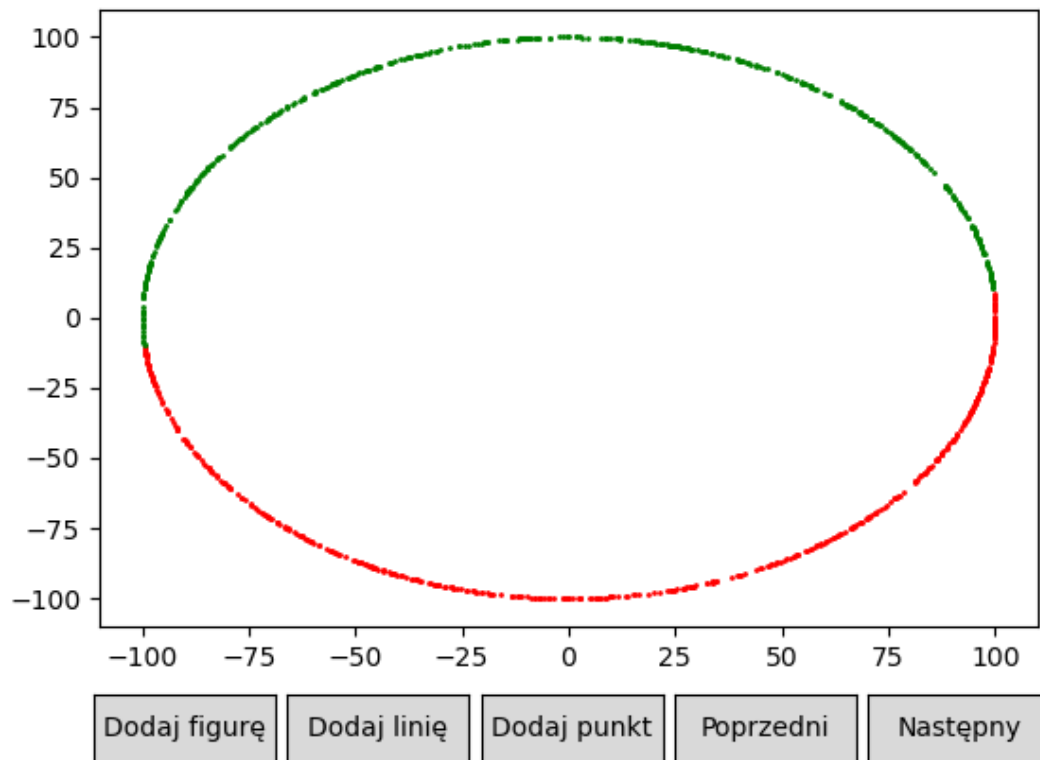


Po prawej: 49978, Po lewej: 50022, Współliniowe: 0

Zestaw C:

W zestawie C wyniki były takie same dla każdego wyznacznika.

Wykres 2.3 Klasyfikacja punktów z zestawu C dla każdego sposobu liczenia wyznacznika

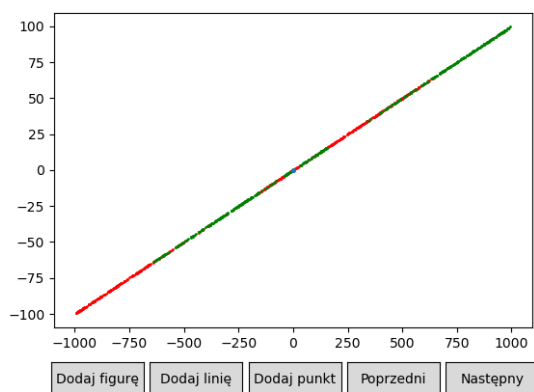


Po prawej: 495, Po lewej: 505, Współliniowe: 0

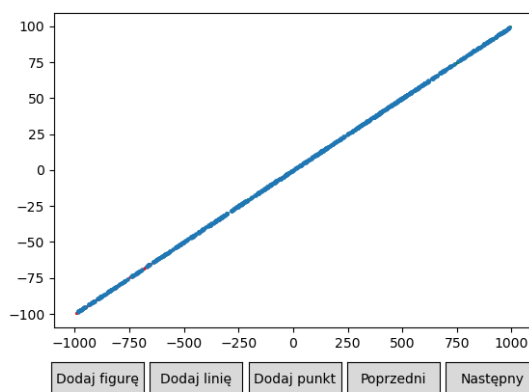
Zestaw D:

W zestawie D wyniki różniły się między sobą w zależności od wykorzystanego sposobu liczenia wyznacznika. Mimo że punkty były generowane na prostej, ograniczona precyzja obliczeń spowodowała, że nie wszystkie zostały sklasyfikowane jako współliniowe.

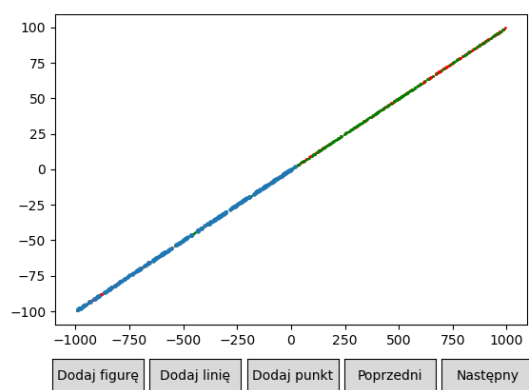
Wykres 2.4.1 Zestaw D, biblioteczny wyznacznik 2x2



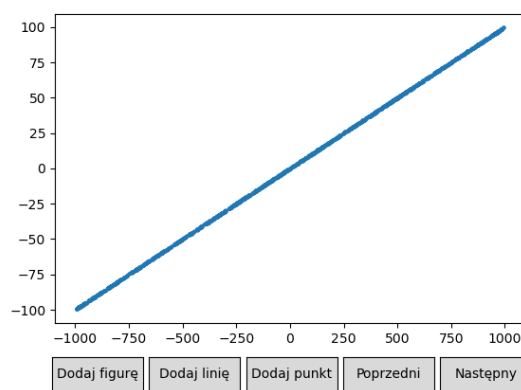
Wykres 2.4.2 Zestaw D, własna implementacja wyznacznik 2x2



Wykres 2.4.3 Zestaw D, biblioteczny wyznacznik 3x3



Wykres 2.4.4 Zestaw D, własna implementacja wyznacznik 3x3



5. Porównanie czasu klasyfikacji

Metody liczenia wyznaczników różniły się liczbą operacji potrzebnych do ich obliczenia a zestawy punktów różniły się licznością, powodowało to różnice w czasie potrzebnym do zakwalifikowania punktów. Poniższa tabela pokazuje różnice w czasie wykonania klasyfikacji ze względu na użytą metodę dla każdego zbioru danych. Wyniki pokazują jednoznacznie że własne implementacje wyznaczników były szybsze niż wyznaczniki biblioteczne. Najszybsza okazała się własna implementacja wyznacznika 3x3.

5.1 Tabela Czas obliczania przy użyciu poszczególnych wyznaczników w sekundach

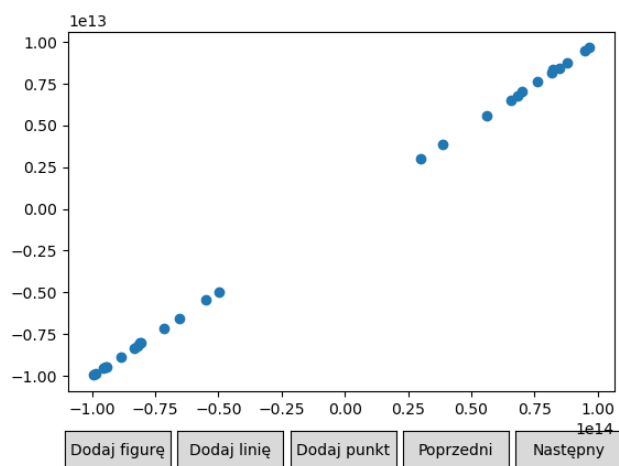
	Biblioteczny 2x2	Własny 2x2	Biblioteczny 3x3	Własny 3x3
Zestaw A	2.74	0.72	4.04	0.92
Zestaw B	2.06	0.43	3.26	0.54
Zestaw C	0.0248	0.0142	0.0301	0.005
Zestaw D	0.0201	0.004	0.0357	0.006

6. Różnice w klasyfikacji

W trakcie wykonywania testów zdarzało się różne metody obliczania wyznacznika inaczej klasyfikowały ten sam punkt. W celu dokładnego zbadania które punkty są inaczej klasyfikowane skorzystałem z funkcji `find_diff()`. Funkcja ta tworzy listę 2-wymiarową, w której wiersz oznacza punkt a 4 kolumny klasyfikacje punktu dla kolejnych metod obliczania wyznacznika. Pozwala ona łatwo znaleźć szukane punkty a następnie je zwizualizować.

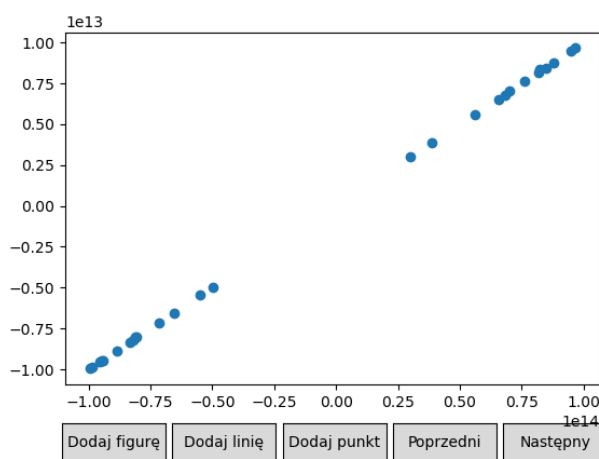
W zestawie A oraz C nie pojawiły się żadne różnice, w zestawie B pojawiły się punkty inaczej zakwalifikowane przy użyciu różnych wyznaczników 2x2, przy użyciu własnej implementacji wyznacznika 2x2 i 3x3 oraz przy użyciu różnych wyznaczników bibliotecznych, punkty zostały tak samo sklasyfikowane jedynie dla przez oba wyznaczniki 3x3. Różnice w klasyfikacji pojawiają się dla punktów przy prostej wyznaczonej przez punkty a i b ale takich których współrzędne to duże liczby. Wynikają one z różnicy precyzji obliczeń dla dużych liczb przy użyciu różnych metod liczenia wyznacznika.

Wykres 6.1.1 Punkty z zestawu B różnie zakwalifikowane przez wyznaczniki 2x2



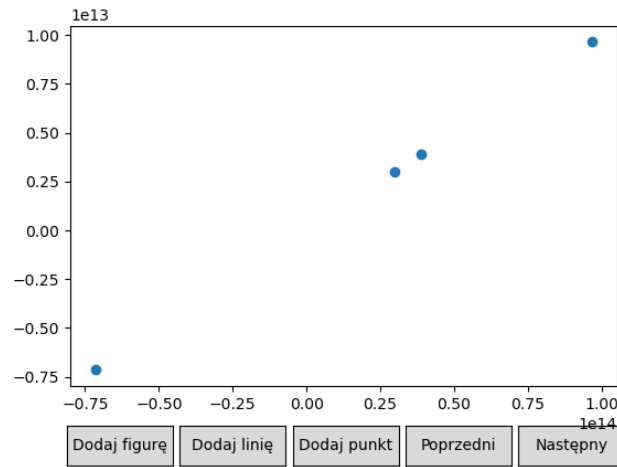
Liczba punktów: 28

Wykres 6.1.2 Punkty z zestawu B różnie zakwalifikowane przez własną implementację wyznaczników



Liczba punktów: 28

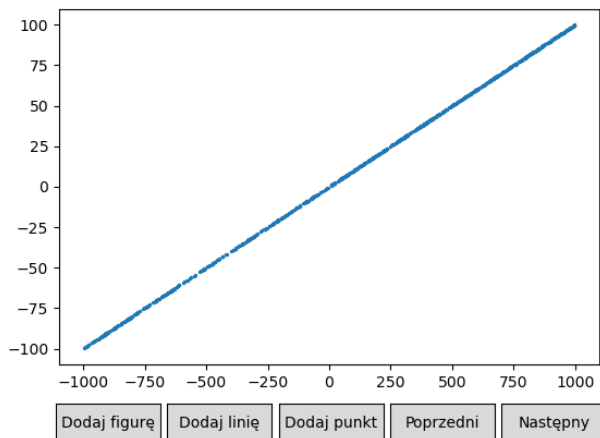
Wykres 6.1.3 Punkty z zestawu B różnie zakwalifikowane przez wyznaczniki biblioteczne



Liczba punktów: 4

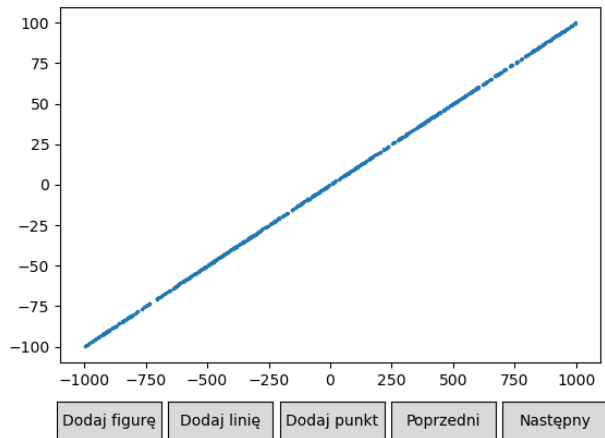
W zestawie D pojawiło się najwięcej różnic w klasyfikacji, największe przy wykorzystaniu różnych wyznaczników 3x3: około 81% punktów inaczej zakwalifikowanych, dla wyznaczników 2x2: około 68% punktów, dla wyznaczników własnej implementacji: około 35% punktów a dla wyznaczników bibliotecznych: około 57% punktów. Różnice podobnie jak w poprzednim zestawie wynikają z różnicy w precyzji obliczeń tu są jednak bardziej widoczne, ponieważ punkty generowaliśmy na prostej według której prowadziliśmy klasyfikację.

Wykres 6.2.1 Punkty z zestawu D różnie zakwalifikowane przez wyznaczniki 3x3



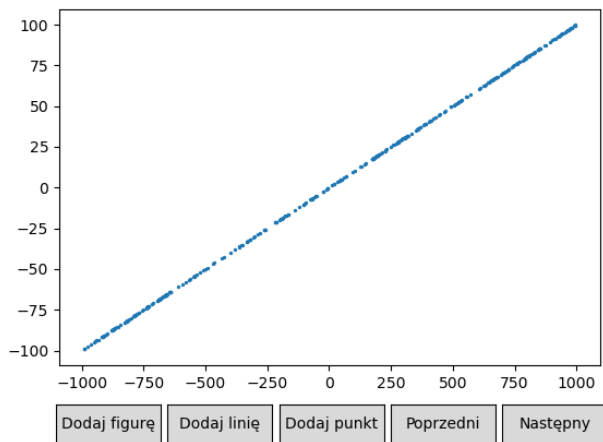
Liczba punktów: 692

Wykres 6.2.2 Punkty z zestawu D różnie zakwalifikowane przez wyznaczniki 2x2



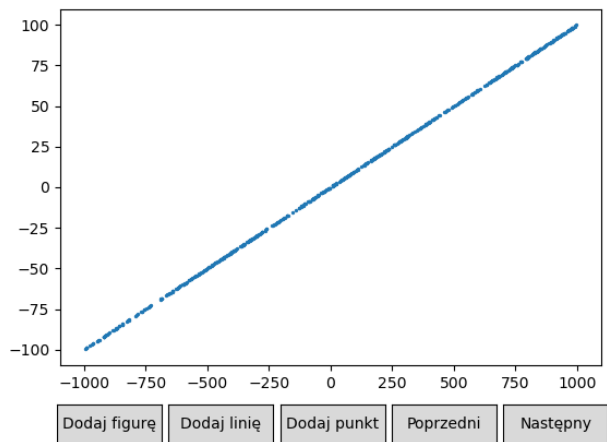
Liczba punktów: 664

Wykres 6.2.3 Punkty z zestawu D różnie zakwalifikowane przez własne implementacje wyznacznika



Liczba punktów: 438

Wykres 6.2.4 Punkty z zestawu D różnie zakwalifikowane przez wyznaczniki biblioteczne



Liczba punktów: 559

7. Wyniki klasyfikacji dla różnych tolerancji na zero

W tej części przedstawione zostały wyniki klasyfikacji z uwzględnieniem zarówno metody obliczania wyznacznika jak i tolerancji dla zera. Różnice w wynikach pojawiły się jedynie przy analizie zestawu D i były one znaczące dlatego też jedynie one będą tu pokazane. W innych zestawach zakres punktów jakie były generowane, ich ilość oraz ich rozkład spowodowały że zmiana tolerancji nie wpłynęła na wynik. W zestawie D widoczne są znaczne różnice, ponieważ punkty były generowane na prostej. Jak można było oczekiwać wynik wskazuje na to że im większa tolerancja tym więcej punktów klasyfikowanych jest jako współliniowe. Różnica w klasyfikacji jest również znaczna dla tej samej tolerancji ale różnych metod obliczania wyznacznika. Własna implementacja wyznacznika 3x3 najdokładniej klasyfikuje punkty na prostej.

Tabela 7.1 Różnice w klasyfikacji punktów z zestawu D dla różnych tolerancji dla zera oraz przy wykorzystaniu różnych wyznaczników

Wyznacznik	Tolerancja dla zera	Po prawej	Po lewej	Współliniowe
Biblioteczny 2x2	0.01	0	0	1000
	1e-10	0	0	1000
	1e-14	464	479	57
	1e-18	488	512	0
Własna implementacja 2x2	0.01	0	0	1000
	1e-10	0	0	1000
	1e-14	173	177	650
	1e-18	176	184	640
Biblioteczny 3x3	0.01	0	0	1000
	1e-10	0	0	1000
	1e-14	117	37	846
	1e-18	460	365	175
Własna implementacja 3x3	0.01	0	0	1000
	1e-10	0	0	1000
	1e-14	20	27	953
	1e-18	92	96	812

7. Analiza wyników i podsumowanie

Powyższe dane pokazują że różnice w klasyfikacji położenia punktu względem prostej były istotne w zależności od wykorzystanej metody obliczania wyznacznika oraz przyjętej tolerancji dla zera. Różnice te pojawiły się jednak tylko przy określonych zestawach danych. Różnic w klasyfikacji dla zestawu A i C nie zaobserwowano. Ponadto metoda obliczania wyznacznika miała również duży wpływ na czas potrzebny do zakwalifikowania punktów.

Różnice w klasyfikacji pojawiły się w zestawie B oraz zestawie D. Punkty których klasyfikacja różniła się w zależności od przyjętej metody znajdują się zawsze blisko prostej i to właśnie w tych zbiorach wygenerowano najwięcej takich punktów. W przypadku zbioru B wynikało to z gęstości rozmieszczenia, natomiast w przypadku zbioru D spowodowane to było sposobem generacji. Wielkość liczb ma również znacznie. Z wykresów 6.1.1, 6.1.2, 6.1.3, widzimy że punkty inaczej zaklasyfikowane w zbiorze B to punkty o dużych współrzędnych. Zmiany tolerancji dla zera oraz miała wpływ na wyniki jedynie w zbiorze D. Teoretycznie wszystkie punkty powinny być tam zakwalifikowane jako współliniowe, w rzeczywistości jednak dobór sposobu liczenia miał tam istotny wpływ na wyniki. Przy analizie tego zbioru widać że tolerancja dla zera 10^{-2} pozwalała sklasyfikować wszystkie punkty poprawnie, przy tak dużej tolerancji błędy wynikające z precyzji obliczeń były pomijalne. Przy zmniejszaniu tolerancji ma już znaczenie dobór metody obliczania wyznacznika.

Najdokładniejszym sposobem obliczania wyznacznika okazała się własna implementacja wyznacznika 3×3 z dokładnością około 80% dla najmniejszej tolerancji dla zera. Ten sposób okazał się być również dużo szybszy od wyznaczników bibliotecznych. W zależności od zbioru zdarzało się że wypadał wolniej od własnej implementacji wyznacznika 2×2 jednak były to nieznaczne różnice a jego dokładność znacząco przewyższała wyznacznik 2×2 .