# Algorytmy geometryczne - Ćwiczenie 1. Paulina Jędrychowska

1. Dane techniczne urządzenia i wykorzystane narzędzia:

System Windows 10 x64

Procesor i5-9300H

Pamięć RAM 16GB

Środowisko Jupyter Notebook

Język programowania Python 3.0

2. Generowanie punktów:

Do wygenerowania były cztery zbiory punktów. Wygenerowane punkty zostały umieszczone w osobnych tablicach:

a) $10^5$ losowych punktów o współrzędnych z przedziału [-1000, 1000] (Wykres1.1.),

b) $10^5$ losowych punktów o współrzędnych z przedziału [-1014, 1014] (Wykres 1.2.),

c) 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu R=100

(Wykres 1.3.),

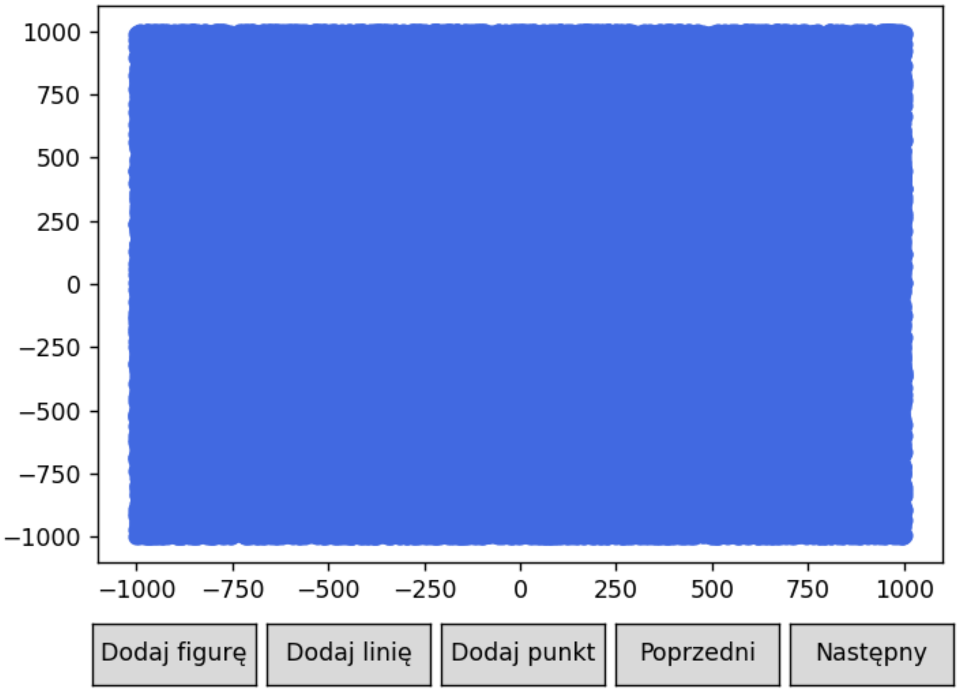
d) 1000 losowych punktów o współrzędnych z przedziału [-1000, 1000] leżących na prostej wyznaczonej przez wektor (a, b), przyjmij a = [-1.0, 0.0], b = [1.0, 0.1] (Wykres 1.4.).

Do wygenerowania zbiorów punktów została użyta funkcja random.uniform(a, b), która generuje liczby typu double z zakresu [a, b].

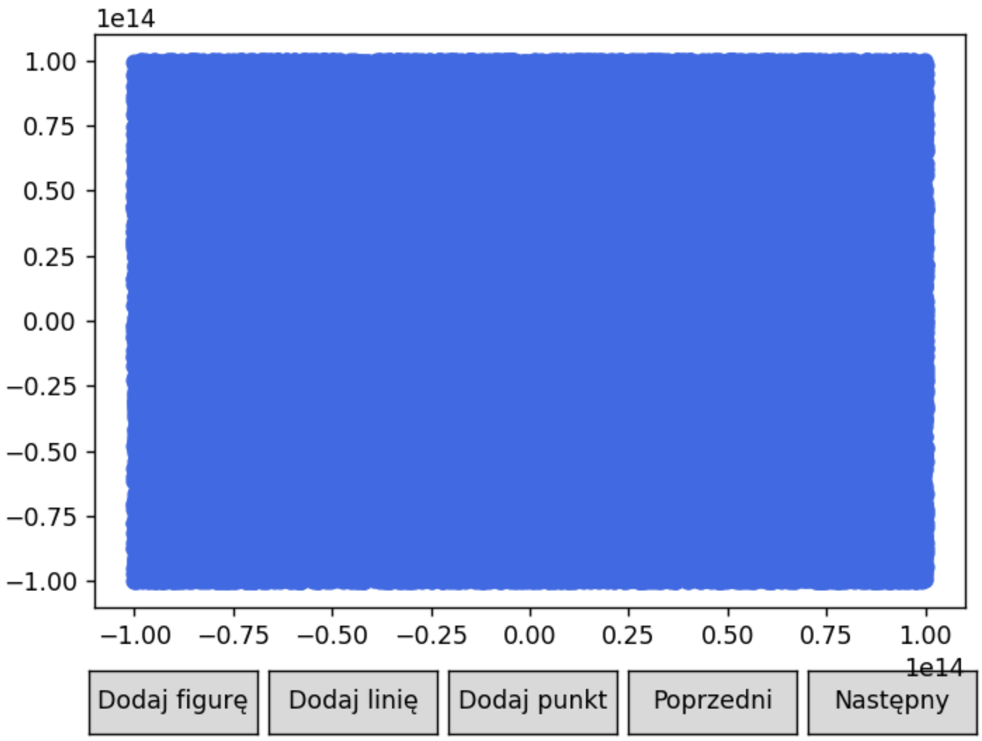
Dla zbiorów a) i b), wygenerowano punkty za pomocą powyższej funkcji. Dla okręgu c) zostały również użyte funkcje pi, sin i cos z biblioteki math. Punkty na okręgu zostały wygenerowane poprzez stworzenie punktów reprezentujących kąt alfa z zakresu $[0, 2\pi]$, i podstawienie odpowiednio: x = 100*cos(alfa) i y = 100*sin(alfa). Do wygenerowania punktów na linii został stworzony punkt x spośród zbioru [-1000, 1000], a punkt y został wyliczony jako 0.05*x + 0.05.

Wszystkie zbiory punktów zostały zwizualizowane przez dostarczone narzędzie graficzne opartego o bibliotekę matplotlib.

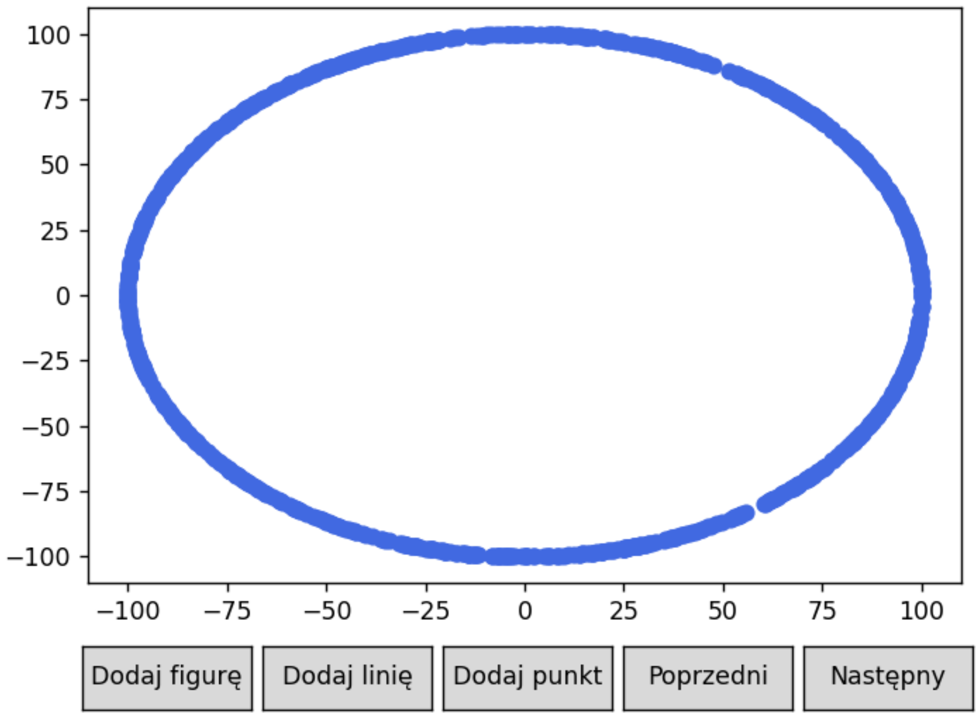
Wykres 1.1.



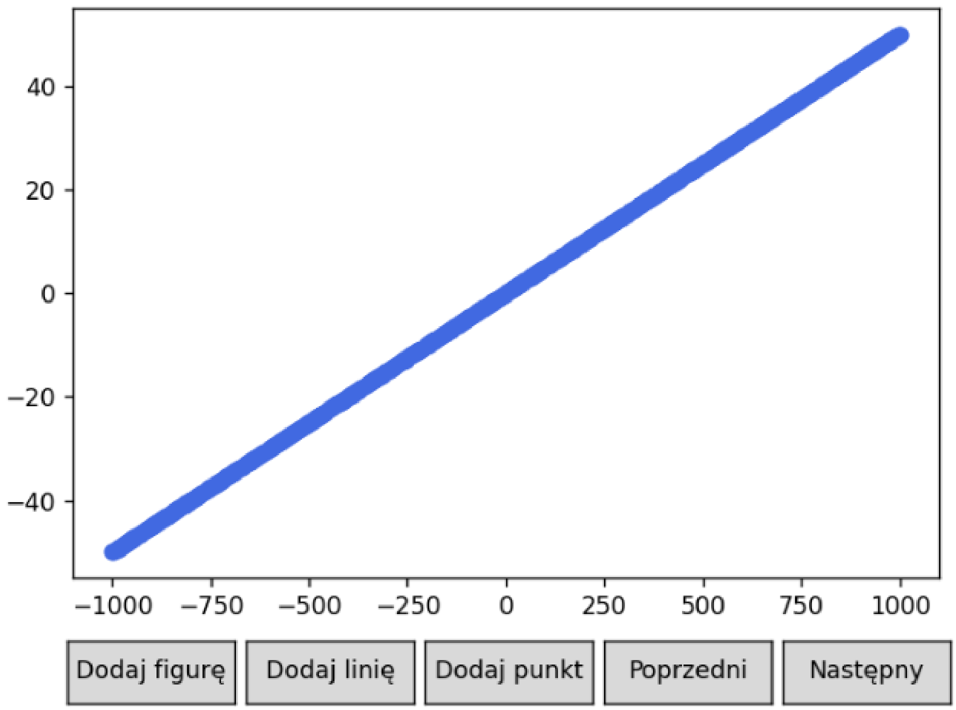

Wykres 1.2



Wykres 1.3




Wykres 1.4



3. Obliczanie wyznacznika:

Wyznacznik został policzony na 4 sposoby:

a) det1 – własna funkcja obliczająca wyznacznik 3x3 liczony ze wzoru 1.:

Wzór 1.

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

b) det2 – własna funkcja obliczająca wyznacznik 2x2 liczony ze wzoru 2.:

Wzór 2.

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

c) det3 – wbudowana funkcja numpy.linalg.det() dla wyznacznika 3x3

d) det4 - wbudowana funkcja z numpy.linalg.det() dla wyznacznika 2x2

4. Wartości tolerancji dla zera:

$10^{-18}$, $10^{-16}$, $10^{-15}$, $10^{-14}$, $10^{-12}$, $10^{-10}$ , $10^{-8}$

Zostały również wykonane obliczenia dla braku tolerancji.

5. Grupowanie punktów przebiega względem odcinka ab ( a = [-1.0, 0.0], b = [1.0, 0.1]). Odbywa się to poprzez wyliczenie wyznacznika dla punktów a, b, oraz wygenerowanego punktu c i przyrównanie wartości wyznacznika do zera. Dla wyznacznika mniejszego niż 0 wartości zaliczane są do prawej strony linii, dla większego do lewej, a dla równego do współliniowej. Wartości są trzymane w trzech osobnych tablicach i wykorzystane przy tworzeniu wykresu.

6. Analiza wyników:

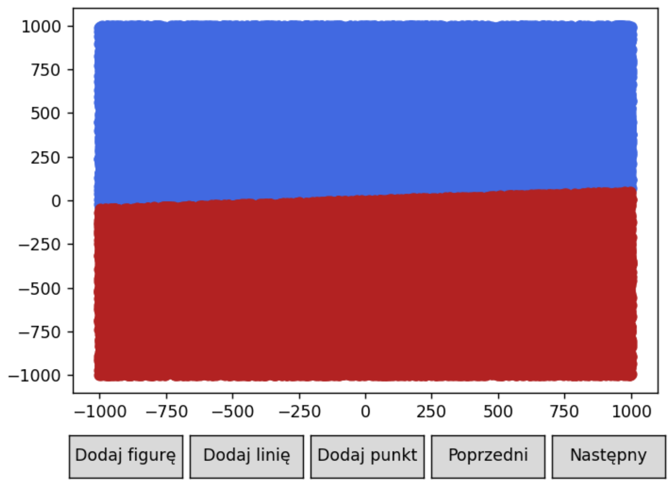
Legenda dla poniższych wykresów:

- Niebieski – punkty na lewo od prostej
- Czerwony – na prawo od prostej
- Zielony – współliniowe

a) Zbiór 1. – T1:

Dla tego zbioru dla każdej przyjętej tolerancji i metodzie wyznaczania wyznacznika nie znaleziono żadnego współliniowego punktu (Wykres 2.):

Wykres 2.

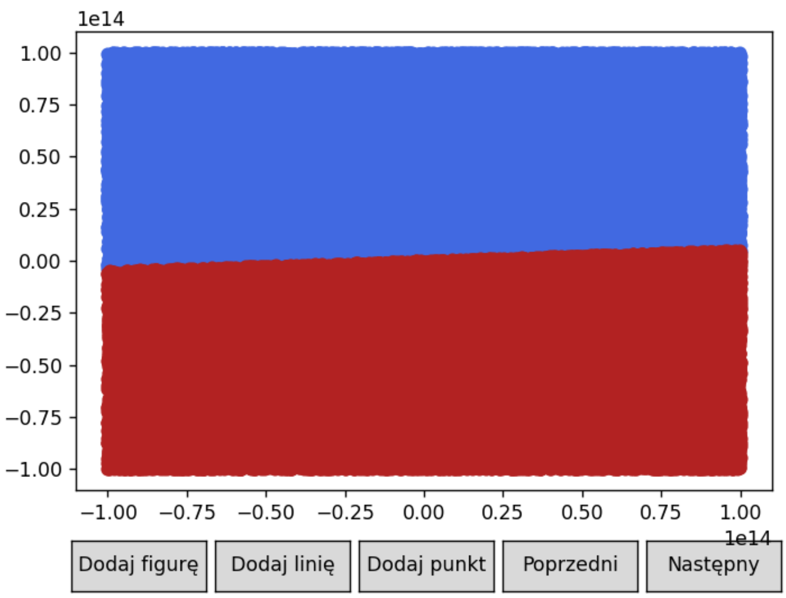



| Po lewej | Na linii | Po prawej |
|---|---|---|
| 49870 | 0 | 50130 |

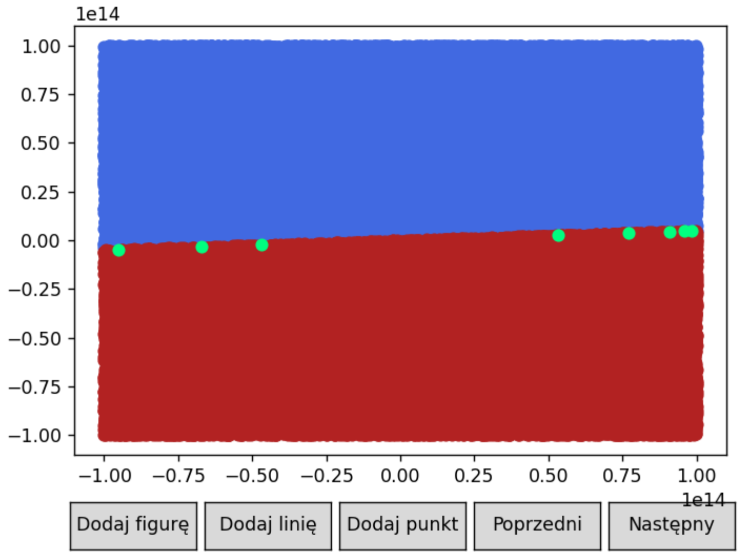
b) Zbiór 2 – T2:

Dla tego zbioru wyniki nie były zależne od wartości tolerancji ( wychodziły również takie same przy jej braku). Dla obu wyznaczników 3x3 nie znajdowało żadnych współliniowych (Wykres 3.1.)punktów, a dla obu wyznaczników 2x2 znajdowało (Wykres 3.2).

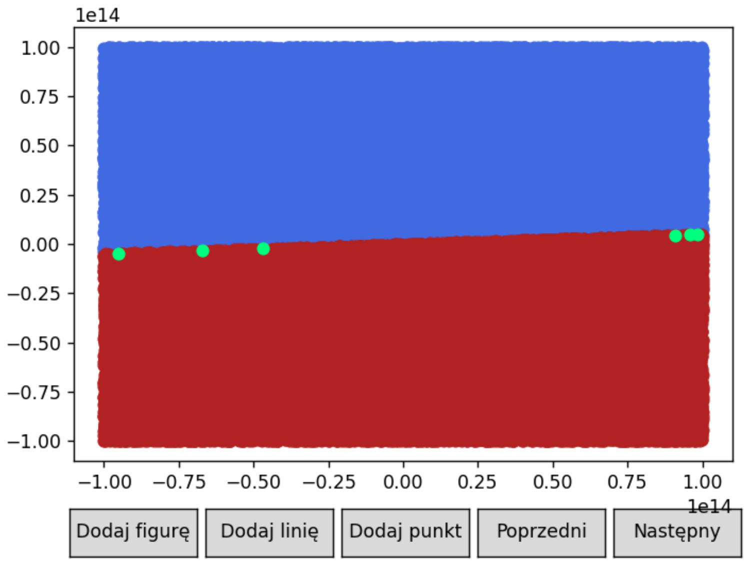
Wykres 3.1.



Wykres 3.2.




Wykres 3.3. Dla wbudowanego wyznacznika 2x2




| Wyznacznik | Po lewej | Na linii | Po prawej |
|---|---|---|---|
| 3x3 | 50021 | 0 | 49979 |
| 2x2 własny | 50020 | 8 | 49972 |
| 2x2 wbudowany | 50020 | 6 | 49974 |

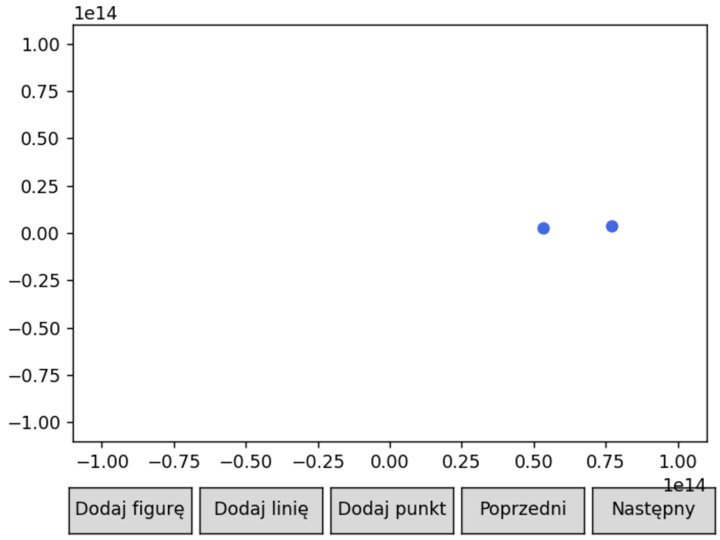
Wyznacznik 2x2 własny znajduje o 2 punkty więcej niż wbudowany, pozostałe punkty pokrywają się ze sobą. Przykładem punktu które się pokrywają dla obu wyznaczników jest:

(-95233044506947.5, -4744978587589.109)

Wartości dwóch punktów, które się nie pokrywają:

(53397466904983.0, 2666693570143.6875), (76910253167041.72, 3841275755044.203)

Wykres 3.4. Różne punkty na wykresie 2x2 własnym i wbudowanym




Dla punktu (-95233044506947.5, -4744978587589.109) stosując wzór 2. składowe

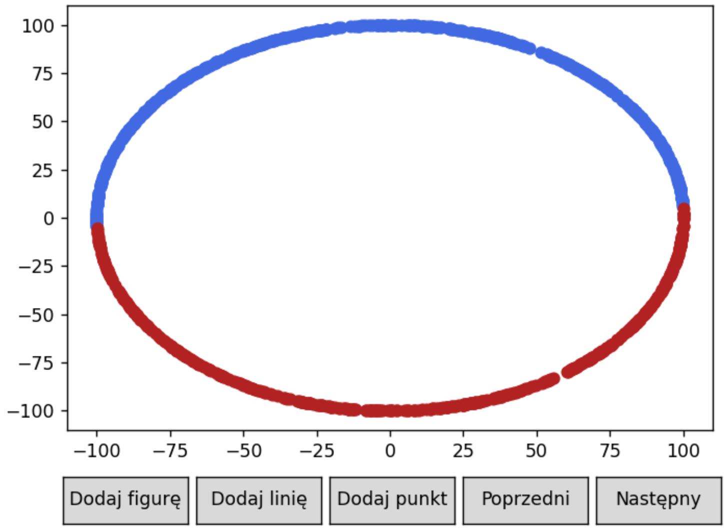
$(a_x - c_x)*(b_y - c_y)$ oraz $(a_y - c_y)*(b_x - c_x)$ wynoszą 4.518787570163913e+26. Dlatego właśnie przy odejmowaniu wyznacznik się zeruje. Jest to spowodowane tym, że po wymnożeniu obie składowe są na tyle duże że liczby jedności nie mają znaczenia i liczby wydają się być sobie równe.

Dla punktów (53397466904983.0, 2666693570143.6875), (76910253167041.72, 3841275755044.203) składowe wyliczane przez funkcję również są sobie równe, a więc wartość własnego wyznacznika 2x2 się zeruje, natomiast wbudowany wyznacznik prawdopodobnie używa jakiejś innej metody na obliczenie wyznacznika.

c) Zbiór 3 – circle:

Dla tego zbioru dla każdej przyjętej tolerancji i metodzie wyznaczania wyznacznika nie znaleziono żadnego współliniowego punktu:

Wykres 4.1 Dla wszystkich wyznaczników i tolerancji




| Po lewej | Na linii | Po prawej |
|---|---|---|
| 504 | 0 | 496 |

d) Zbiór 4 – line:

Dla prostej wpływ miała zarówno tolerancja jak i rodzaj użytego wyznacznika:

Tabela 1.1. Zależność ilości punktów współliniowych od tolerancji oraz wyznacznika:

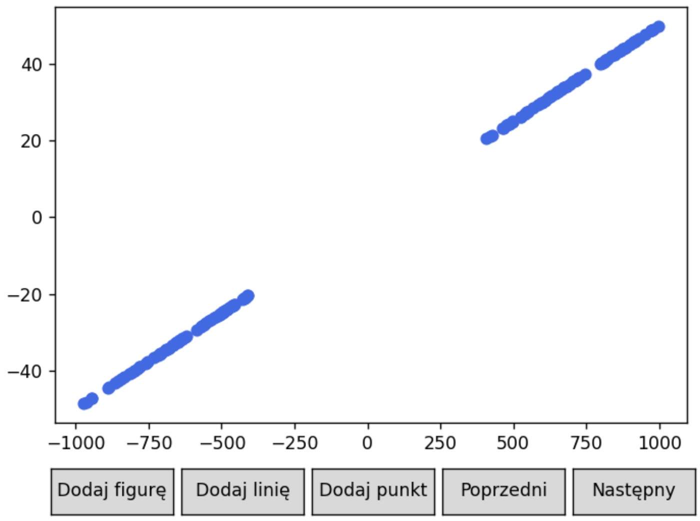
| Tolerancja: | det 3x3 własny | det 2x2 własny | det 3x3 wbudowany | det 2x2 wbudowany |
|---|---|---|---|---|
| $10^{-8}$ | 1000 | 1000 | 1000 | 1000 |
| $10^{-10}$ | 1000 | 1000 | 1000 | 1000 |
| $10^{-12}$ | 1000 | 843 | 1000 | 776 |
| $10^{-14}$ | 1000 | 733 | 883 | 698 |
| $10^{-15}$ | 460 | 725 | 374 | 684 |
| $10^{-16}$ | 414 | 723 | 324 | 681 |
| $10^{-18}$ | 409 | 722 | 317 | 681 |
| brak | 409 | 722 | 317 | 681 |

Na podstawie tabeli 1.1 można zauważyć, że:

- w obu przypadkach własne wyznaczniki działają lepiej
- dla tolerancji > $10^{-12}$ wszystkie wyznaczniki działają tak samo i znajdują wszystkie punkty
- dla tolerancji z zakresu ($10^{-15}$, $10^{-12}$] wyznaczniki 3x3 działają lepiej niż 2x2, a ponadto wyznacznik 3x3 własny znajduje wszystkie punkty
- dla tolerancji z zakresu ($10^{-18}$, $10^{-15}$] efektywność znaczników 3x3 gwałtownie spada, a wyznaczników 2x2 spada nieznacznie, przez co wyznacznik 2x2 dla wartości ≤ $10^{-15}$ znajduje dużo więcej rozwiązań
- dla tolerancji z zakresu [0, $10^{-18}$] wartości wszystkich wyznaczników nie zmieniają się

Wykres 5.1. przedstawia punkty, które są różne dla wyznacznika 3x3 oraz 2x2 (oba własne) dla tolerancji $10^{-12}$.

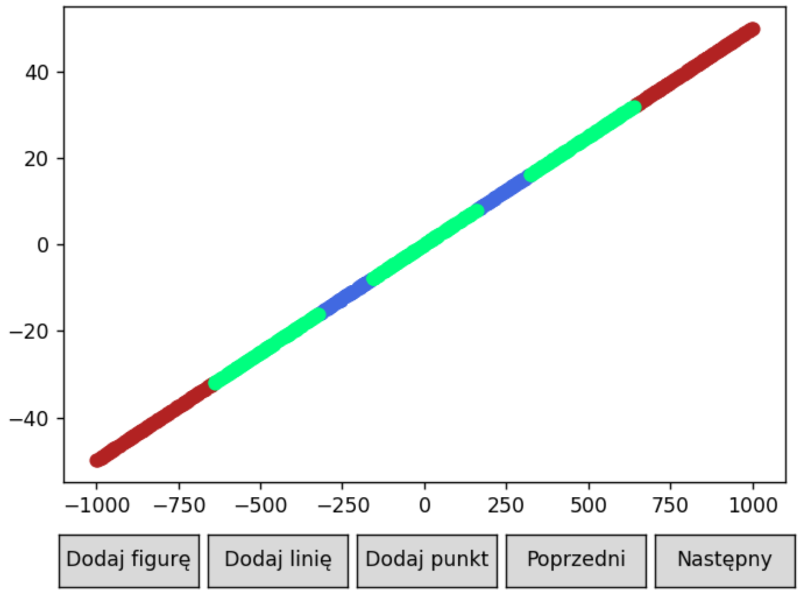
Wykres 5.1.




Z powyższego wykresu możemy wnioskować, że dla det 2x2 większość punktów które kwalifikuje to wartości centralne. Przy zmniejszaniu tolerancji tendencja ta zaczyna stopniowo maleć.

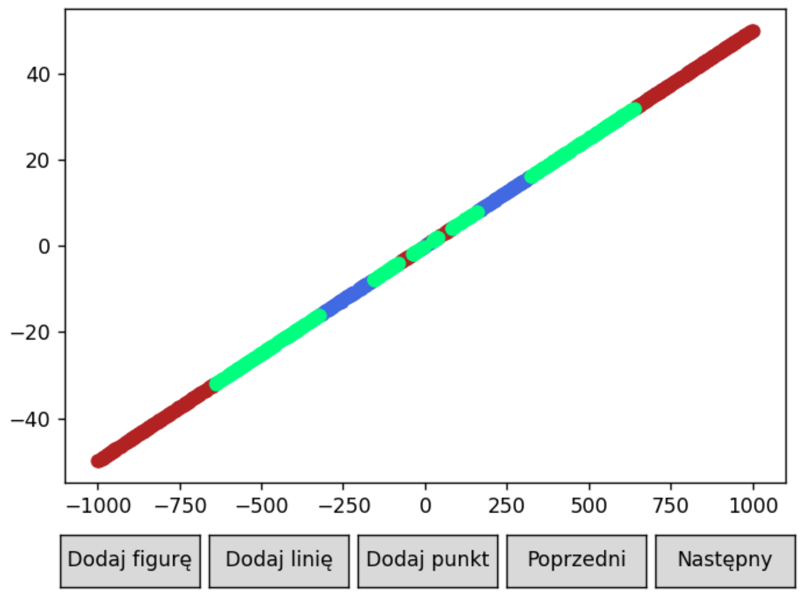
Wykres 5.2. obrazuje rozłożenie punktów, dla współczynnika 3x3 przy tolerancji $10^{-15}$

Wykres 5.2.



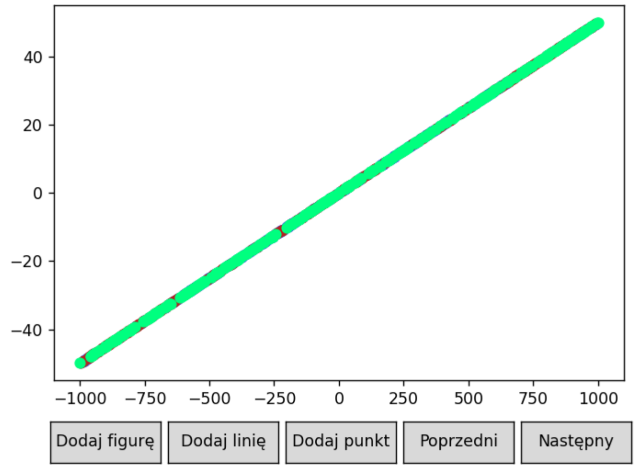

Na powyższym wykresie można wyraźnie zaobserwować centralną i symetryczną tendencję rozkładu znalezionych punktów. Dla zmniejszonej tendencji również można zaobserwować to zjawisko. Wykres 5.3. obrazuje rozłożenie punktów dla współczynnika 3x3 przy tolerancji $10^{-18}$.

Wykres 5.3



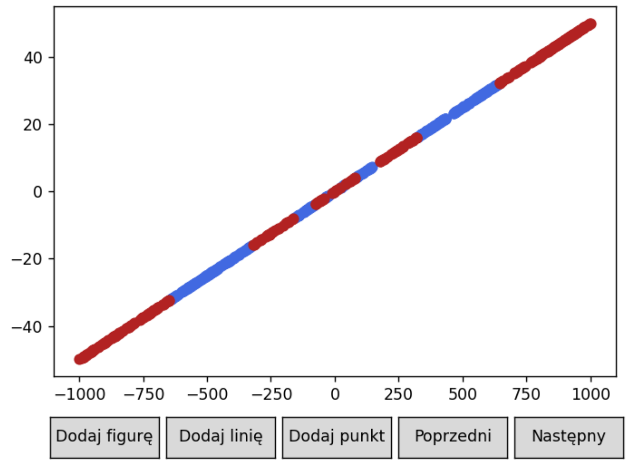
Wykres 6.1 przedstawia rozłożenie punktów dla wyznacznika 3x3 funkcji wbudowanej przy tolerancji $10^{-18}$.

Wykres 6.1.




Można zaobserwować, że pomimo małej ilości punktów, są one równomiernie rozłożone na prostej i są zupełnie inne od punktów. Wykres 6.2. przedstawia punkty które zostały inaczej wyznaczone przez współczynnik 3x3 wbudowany i własny, gdzie na czerwono zostały oznaczone punkty z wbudowanego, a na niebiesko z własnego wyznacznika.

Wykres 6.2.

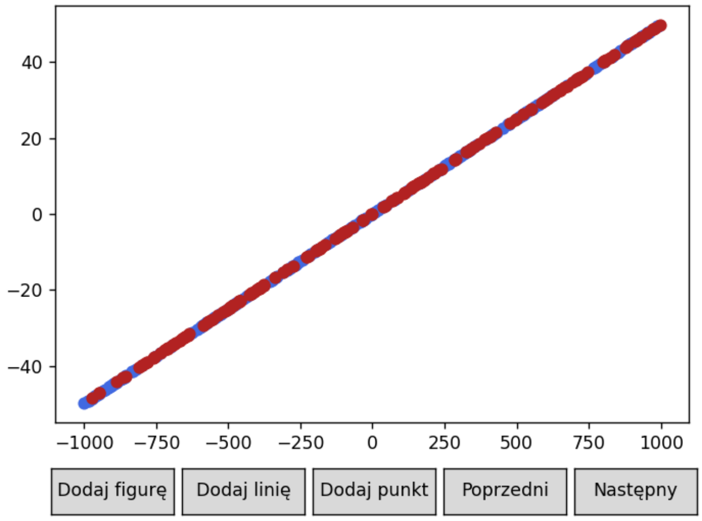



| | |
|---|---|
| det 3x3 własny | 409 |
| det 3x3 wbudowany | 317 |
| Wspólne punkty | 130 |
| Niewspólne punkty | 549 |
| Nie zaklasyfikowane w obu | 321 |

Na podstawie powyższego wykresu i tabeli można zaobserwować, że większość punktów między współczynnikiem własnym i wbudowanym nie pokrywa się.

Wykres 6.3. przedstawia punkty które zostały inaczej wyznaczone przez współczynnik 2x2 wbudowany i własny, gdzie na czerwono zostały oznaczone punkty z wbudowanego, a na niebiesko z własnego wyznacznika.

Wykres 6.3.




| | |
|---|---|
| det 2x2 własny | 722 |
| det 2x2 wbudowany | 681 |
| Wspólne punkty | 501 |
| Niewspólne punkty | 405 |
| Nie zaklasyfikowane w obu | 94 |

Jak w poprzednim przypadku, wiele punktów dla wyznacznika własnego i wbudowanego nie pokrywa się ze sobą.

e) Porównanie czasów dla funkcji własnych i wbudowanych:

Porównuję czasy funkcji grupującej dla dwóch pierwszych zestawów, gdyż mają one najwięcej wartości, a więc najdłuższy czas liczenia.

| Nr. zestawu | Własny | Wbudowany |
|---|---|---|
| Zestaw 1. Wyznacznik 3x3 | 0.06183433532714844 | 1.1479299068450928 |
| Zestaw 1. Wyznacznik 2x2 | 0.04188728332519531 | 1.0013222694396973 |
| Zestaw 2. Wyznacznik 3x3 | 0.0608372688293457 | 1.150923252105713 |
| Zestaw 2. Wyznacznik 2x2 | 0.041887521743774414 | 1.0172793865203857 |

Funkcje własnej implementacji są ok. 20 razy szybsze niż funkcje wbudowane.

7. Wnioski:

Na klasyfikację położenia punktu względem prostej wpływ ma zarówno metoda obliczania wyznacznika jak i tolerancja dla zera. Na podstawie zestawu 1 i 3, możemy wnioskować że przy nie gęstych danych, oddalonych od prostej rzadko kiedy trafią się punkty, które zostaną zakwalifikowane. Natomiast na podstawie zestawu 2, można zaobserwować, że przy wyliczaniu wyznacznika 2x2 dla dużych liczb program popełnia błędy obliczeniowe. W zestawie 4 wiemy, że wszystkie punkty powinny należeć do prostej, natomiast oprócz błędów pomiarowych związanych z wyznacznikiem i tolerancją problem stanowią także drobne błędy programu przy generowaniu punktów na prostej. Z otrzymanych wyników można wywnioskować, że najlepiej dopasowywać sposób wyliczania wyznacznika do tolerancji którą wybieramy, gdyż dla każdej tolerancji inny wyznacznik się lepiej sprawuje. Kolejnym spostrzeżeniem jest, że dla wszystkich wyników funkcje wbudowane są zarówno dużo mniej precyzyjne jak i dużo wolniejsze od funkcji własnych. Jest to spowodowane, tym że są one uniwersalne i nie dostosowane do tych konkretnych danych. Dlatego jeżeli zależy nam na precyzji, a napisanie własnej funkcji nie jest skomplikowane to warto to zrobić.

Wyznacznik, który wybieram do używania w przyszłości to wyznacznik 3x3 własnej implementacji, przy tolerancji $10^{-14}$, ponieważ on przy niskiej tolerancji precyzyjnie klasyfikuje nasze punkty.