# Algorytmy Geometryczne
# Sprawozdanie 1

Aga Patro
czw_13.30_A

# 1. Specyfikacja sprzętu i narzędzia wykorzystane w realizacji

System: Debian Linux Parrot OS x64
Procesor: AMD Ryzen 5 4500U, 6 rdzeni, 6 wątków, 4.00GHz
Pamięć RAM: 16 GB
Środowisko: Jupyter Notebook
Język: Python 3
Narzędzie pomocnicze: plik *geometria.ipynb* dostarczony z poleceniem, biblioteki *numPy*, *random* oraz *matplotlib*

# 2. Temat ćwiczenia

Porównać wyniki klasyfikacji położenia punktów względem odcinka w zależności od wybranej metody obliczania wyznacznika oraz tolerancji dla zera

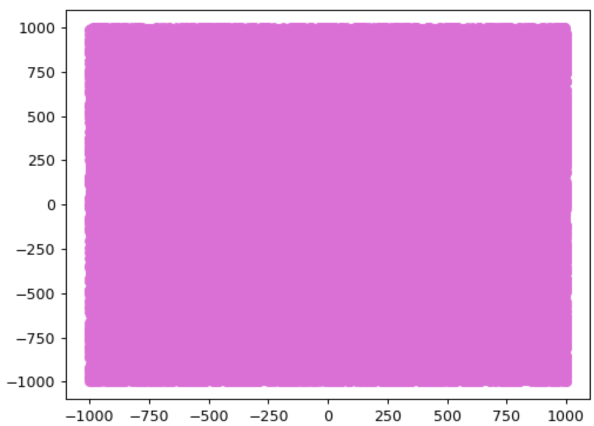
# 3. Realizacja

## 3.1. Generowanie i ilustrowanie punktów

By wygenerować punkty użyłam funkcji *randint()* z biblioteki *random*. Funkcja ta zwraca losową liczbę z przedziału podanego jako argument.
Dodatkowo, w podpunkcie 3 skorzystałam z funkcji *cos()* i *sin()* z biblioteki *math*, natomiast w podpunkcie 4 z podanych punktów wyliczyłam współczynniki prostej.

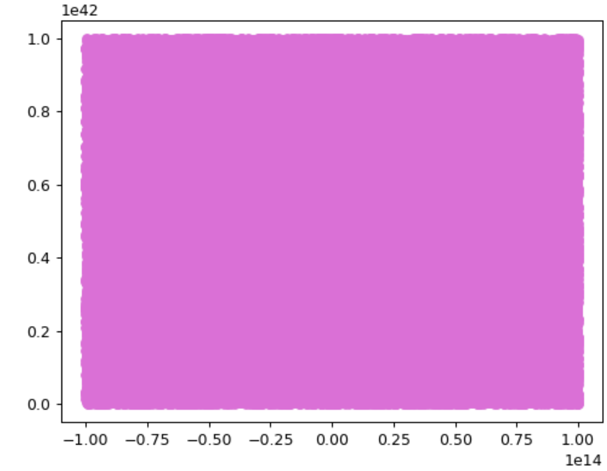
Ilustracje wszystkich zbiory zostały zrealizowane za pomocą funkcji z pliku *geometria.ipynb* opartego o bibliotekę *matplotlib*

1) Zbiór 1 - 10^5 losowych punktów o współrzędnych z przedziału [-1000, 1000]



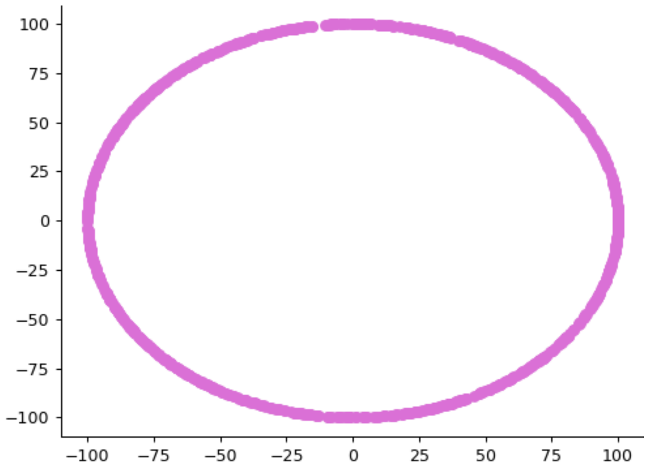

**Wykres 3.1.1 Zbiór punktów 1**

2) Zbiór 2 - 10^5 losowych punktów o współrzędnych z przedziału [-10^14,10^14]



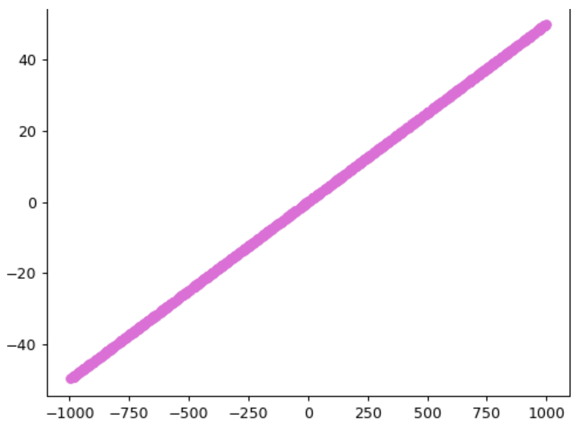

**Wykres 3.1.2 Zbiór punktów 2**

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




**Wykres 3.1.3 Zbiór punktów 3**

4) 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 3.1.4 Zbiór punktów 4**

## 3.2. Obliczanie wyznacznika

By obliczyć wyznaczniki 2x2 i wyznaczniki 3x3 napisałam własne funkcje liczące oraz użyłam biblioteki numPy: *numpy.linalg.det()*

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

**Tabela 1. Zastosowane wyznaczniki i metody ich wyliczenia**

a, b to punkty wyznaczające odcinek na podstawie którego dokonujemy klasyfikacji, c to sprawdzany punkt.

## 3.3. Tolerancje dla zera

Jako rozważane tolerancje dla zera przyjęłam:
- 10^(-14)
- 10^(-18)

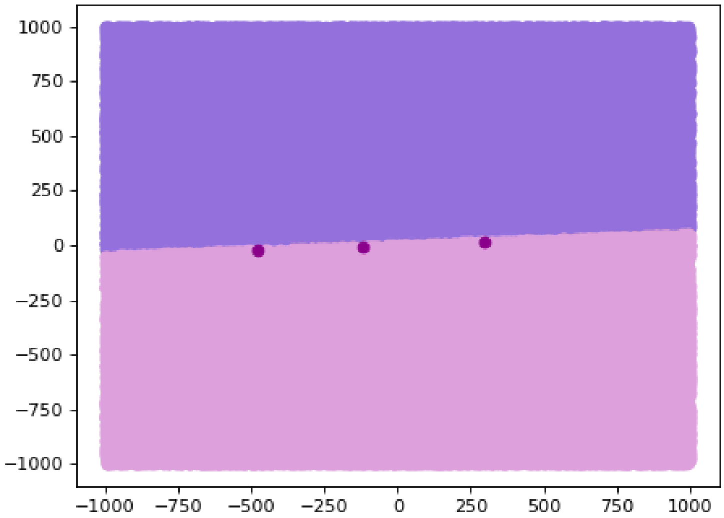
## 3.4. Kategoryzowanie punktów

By kategoryzować punkty napisałam funkcję *points_classifier.* Stworzyłam tablicę list gdzie indeksy 0 oznaczają miejsce po lewej stronie odcinka, 1 miejsce na odcinku a 2 miejsce po prawej stronie odcinka.
Następnie przechodziłam po punktach i dla każdego obliczałam wyznacznik a potem dodawałam go do odpowiedniej listy.

## 3.5. Wizualizacja

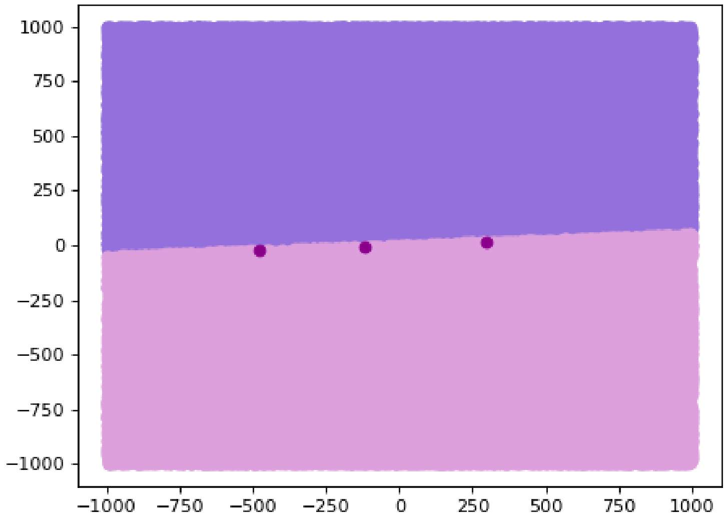
By zwizualizować wykresy napisałam funkcję *show_results*, która jest kombinacją funkcji z pliku *geometria.ipynb*.
Kolor fioletowy oznacza że punkt został sklasyfikowany jako leżący po lewej stronie, kolor różowy jako leżący po prawej stronie, natomiast magenta oznacza punkt współliniowy z danym odcinkiem.

1) Zestaw 1



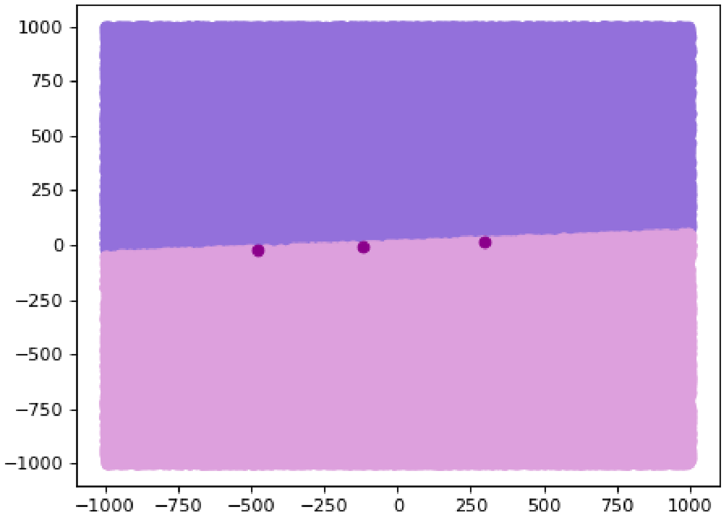

lewa: 50105 prawa: 49892 współliniowo: 3

**Wykres 3.5.1.1 Klasyfikacja punktów z zestawu 1, dla wyznacznika 2x2 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**



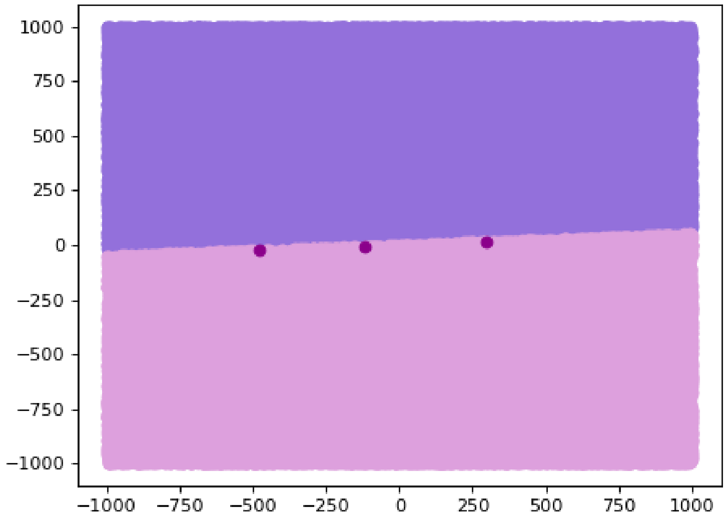

lewa: 50105 prawa: 49892 współliniowo: 3

**Wykres 3.5.1.2 Klasyfikacja punktów z zestawu 1, dla wyznacznika 2x2 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**




lewa: 50105 prawa: 49892 współliniowo: 3

**Wykres 3.5.1.3 Klasyfikacja punktów z zestawu 1, dla wyznacznika 3x3 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**

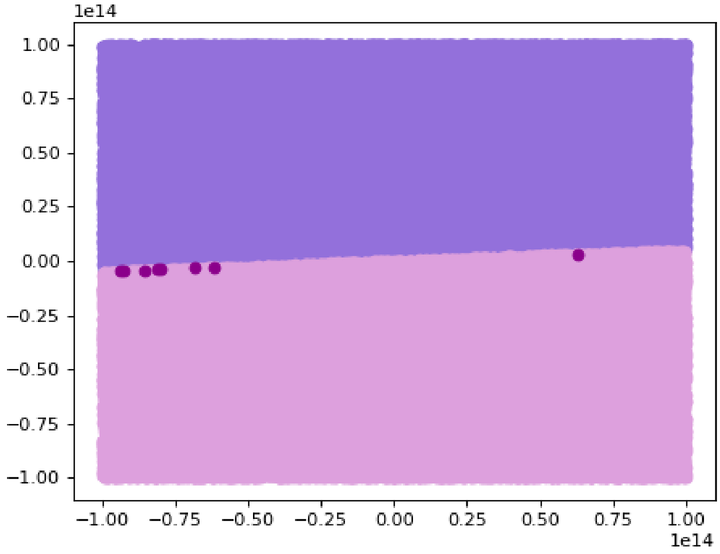



lewa: 50105 prawa: 49892 współliniowo: 3

**Wykres 3.5.1.4 Klasyfikacja punktów z zestawu 1, dla wyznacznika 3x3 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**

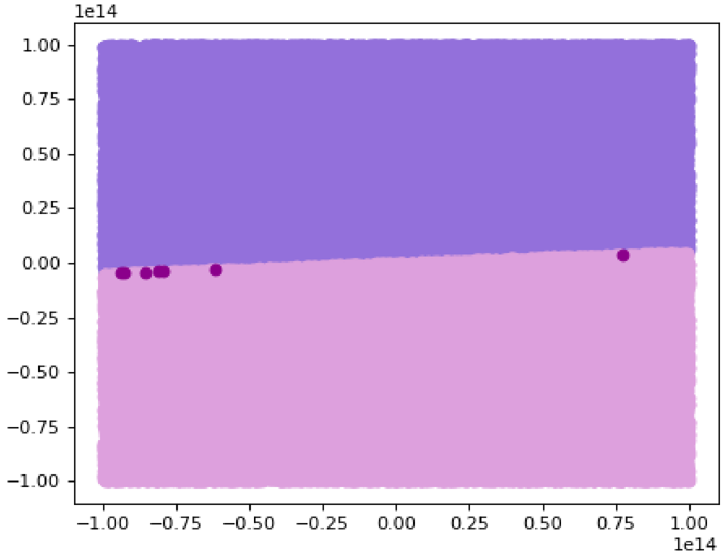
Jak widać, dla pierwszego zestawu wyniki danych dla każdego z wyznaczników były identyczne. Dla tego zestawu wygenerowałam ponownie zbiór punktów by sprawdzić ich współliniową klasyfikację. Za pierwszym ponownym razem otrzymałam 5 punktów współliniowych, za drugim 3 punkty współliniowe a za trzecim razem 1 punkt współliniowy.

2) Zestaw 2



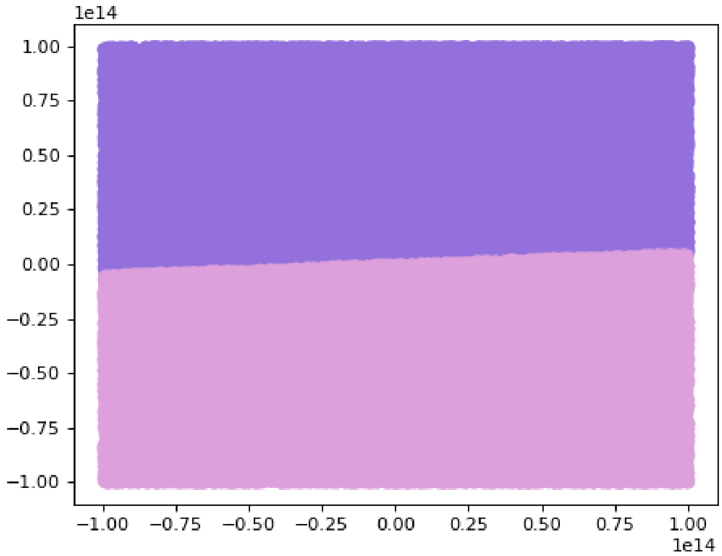
lewa: 49909 prawa: 50083 współliniowo: 8

**Wykres 3.5.2.1 Klasyfikacja punktów z zestawu 2, dla wyznacznika 2x2 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**



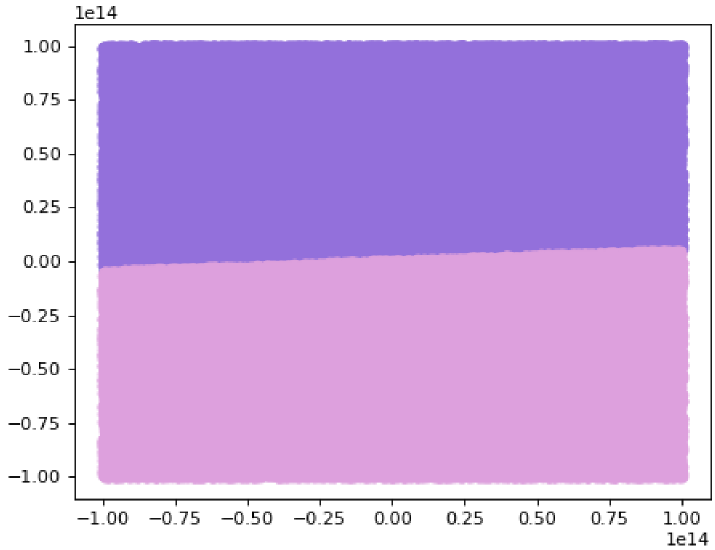
lewa: 49909 prawa: 50084 współliniowo: 7

**Wykres 3.5.2.2 Klasyfikacja punktów z zestawu 2, dla wyznacznika 2x2 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**



lewa: 49912 prawa: 50088 współliniowo: 0

**Wykres 3.5.2.3 Klasyfikacja punktów z zestawu 2, dla wyznacznika 3x3 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**

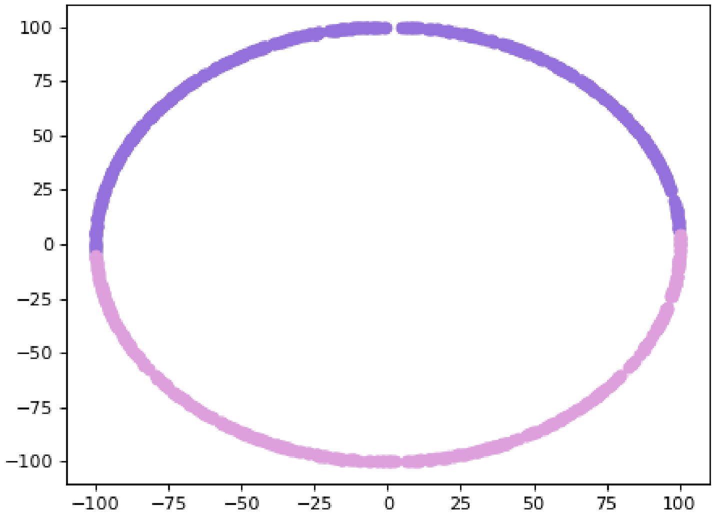


lewa: 49912 prawa: 50088 współliniowo: 0

**Wykres 3.5.2.4 Klasyfikacja punktów z zestawu 2, dla wyznacznika 3x3 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**

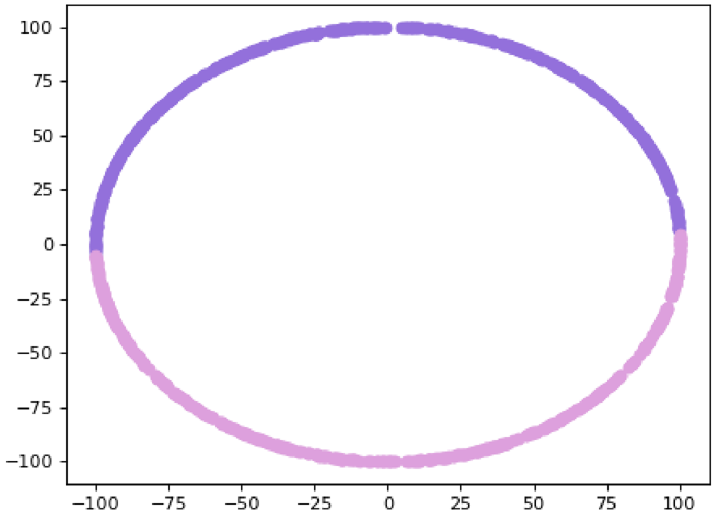
Dla wyznaczników 2x2 wyniki otrzymane różnymi sposobami liczenia są minimalnie różne. Dla wyznaczników 3x3 otrzymane wyniki są takie same.

3) Zestaw 3



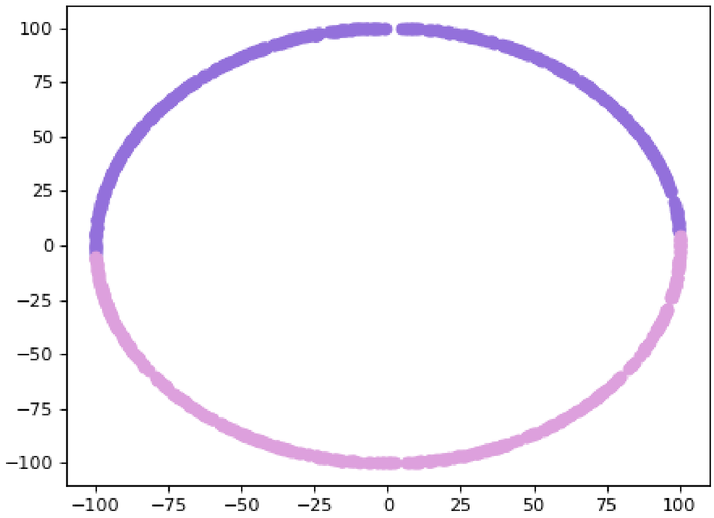

lewa: 513 prawa: 487 współliniowo: 0

**Wykres 3.5.3.1 Klasyfikacja punktów z zestawu 3, dla wyznacznika 2x2 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**



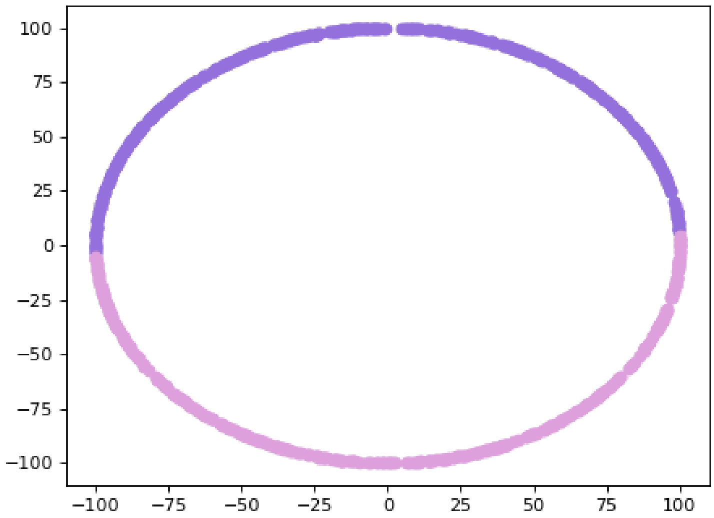

lewa: 513 prawa: 487 współliniowo: 0

**Wykres 3.5.3.2 Klasyfikacja punktów z zestawu 3, dla wyznacznika 2x2 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**




lewa: 513 prawa: 487 współliniowo: 0

**Wykres 3.5.3.3 Klasyfikacja punktów z zestawu 3, dla wyznacznika 3x3 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**

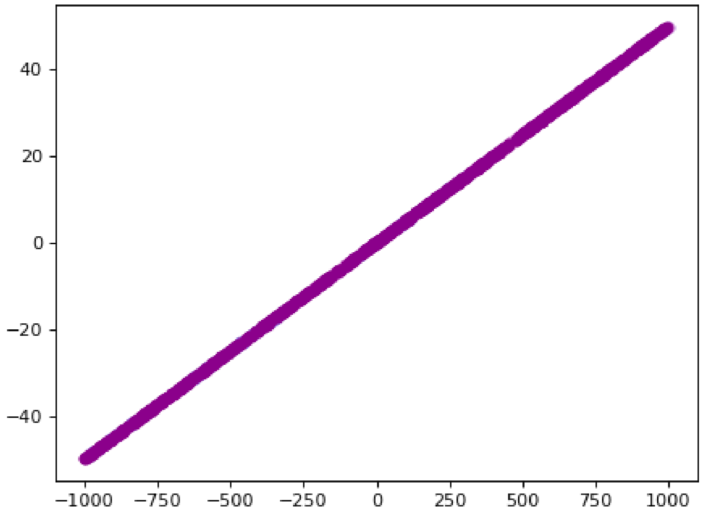



lewa: 513 prawa: 487 współliniowo: 0

**Wykres 3.5.3.4 Klasyfikacja punktów z zestawu 3, dla wyznacznika 3x3 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**

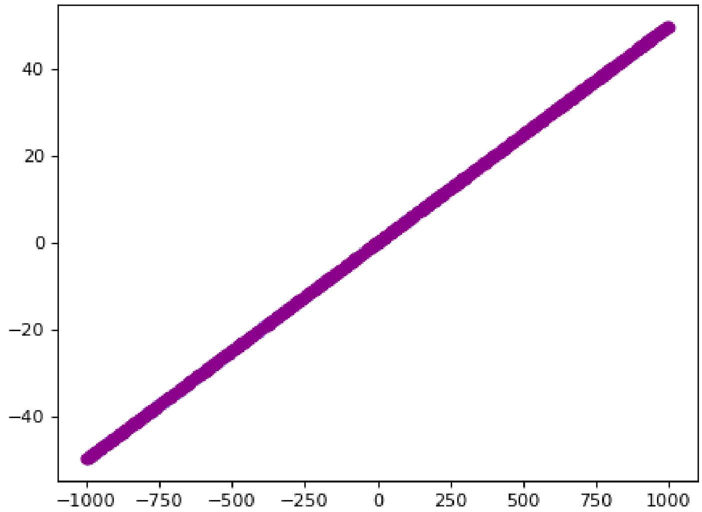
Jak w przypadku zestawu 1, wyniki danych dla każdego z wyznaczników identyczne.

4) Zestaw 4



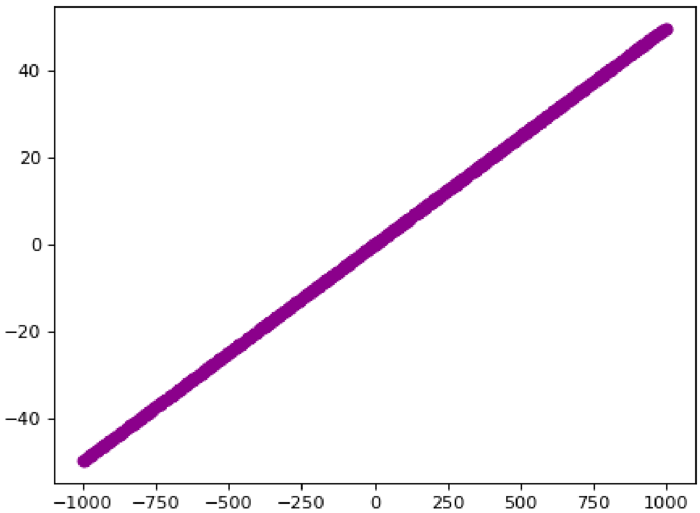

lewa: 115 prawa: 145 współliniowo: 740

**Wykres 3.5.4.1 Klasyfikacja punktów z zestawu 4, dla wyznacznika 2x2 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**



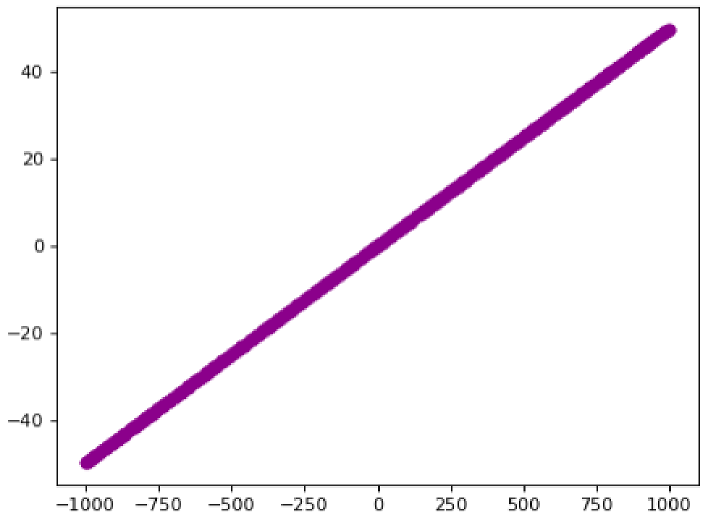

lewa: 165 prawa: 129 współliniowo: 706

**Wykres 3.5.4.2 Klasyfikacja punktów z zestawu 4, dla wyznacznika 2x2 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**




lewa: 7 prawa: 896 współliniowo: 97

**Wykres 3.5.4.3 Klasyfikacja punktów z zestawu 4, dla wyznacznika 3x3 liczonego za pomocą biblioteki NumPy przy tolerancji 10^-14**




lewa: 0 prawa: 1000 współliniowo: 0

**Wykres 3.5.4.4 Klasyfikacja punktów z zestawu 4, dla wyznacznika 3x3 liczonego za pomocą własnej funkcji przy tolerancji 10^-14**

Dla zestawu czwartego teoretycznie wszystkie punkty powinny zostać zaklasyfikowane jako współliniowe, jednak stało się tak tylko dla wyznacznika 3x3 liczonego własną funkcją przy tolerancji 10^-14. Porównując wyniki otrzymane z różnych sposobów liczenia widzimy, że wyniki różnią się od siebie.

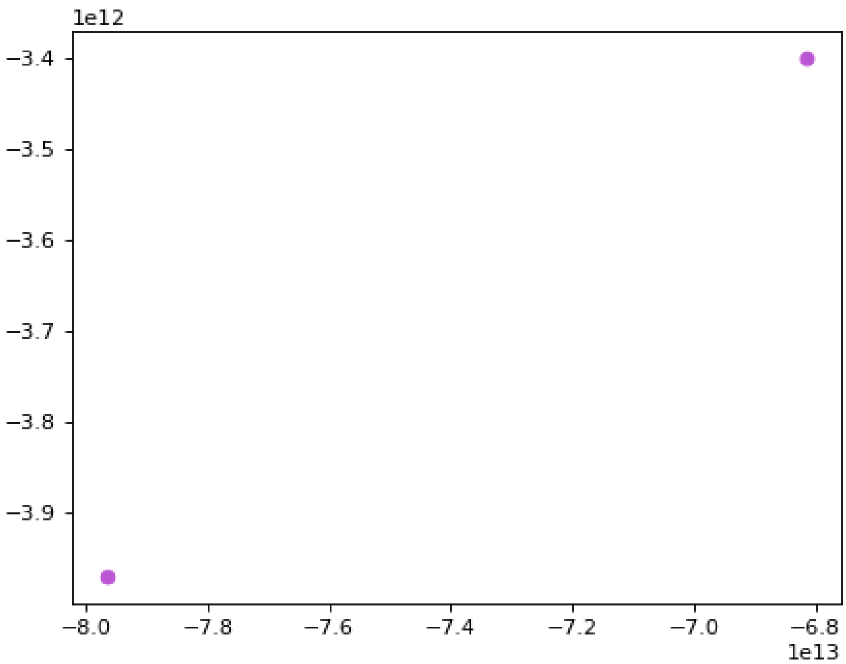
Błędną klasyfikację punktów powodują błędy związane ze skończoną precyzją obliczeń. Dlatego by to zniwelować, w dalszych krokach ćwiczenia będę stosować różne tolerancje dla zera (10^(-14), 10^(-18)).

## 3.6. Porównanie metod liczenia

By znaleźć które punkty różnią się klasyfikacją, napisałam funkcję *find_differece.* Funkcja ta zwraca listę punktów które mają różną klasyfikację w zależności od metody liczenia. Dodatkowo funkcja rysuje wykres ukazujący punkty oraz zwraca ich ilość.

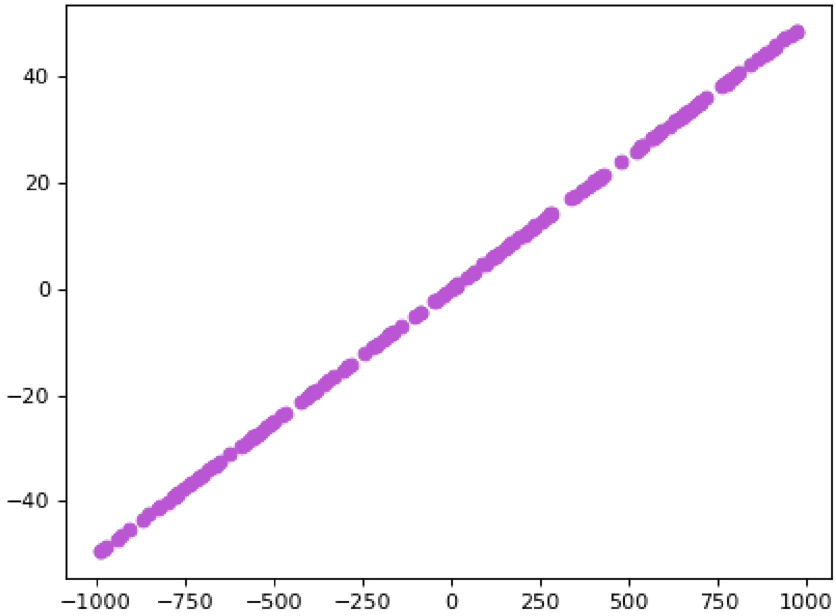
Jako że dla zestawu 1 i zestawu 3 wyniki są takie same, to wykorzystałam tą funkcję by pokazać różnicę w zestawach 2 i 4.

Dla zestawu 2, przy wyznacznikach 2x2 różnica wynosiła 2 punkty.
Dla zestawu 4, dla obu typów wyznaczników różnica wynosiła 170 punktów.



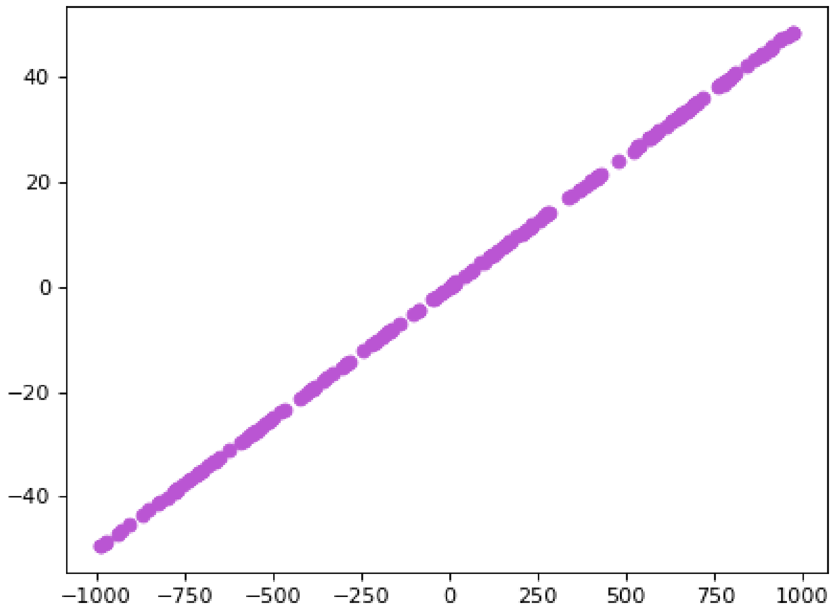

Ilość różnych punktów: 2

**Wykres 3.6.1 Różnica punktów z zestawu 2,**
**dla wyznaczników 2x2 liczonych różnymi metodami za przy**
**tolerancji 10^-14**




Ilość różnych punktów: 170

**Wykres 3.6.2 Różnica punktów z zestawu 4,**
**dla wyznaczników 2x2 liczonych różnymi metodami za przy**
**tolerancji 10^-14**




Ilość różnych punktów: 170

**Wykres 3.6.3 Różnica punktów z zestawu 4,**
**dla wyznaczników 3x3 liczonych różnymi metodami za przy**
**tolerancji 10^-14**

## 3.7. Analiza wyników z uwzględnieniem wszystkich parametrów

| Wyznacznik | Tolerancja dla zera | Po lewej | Współliniowe | Po prawej |
|---|---|---|---|---|
| 2x2 z biblioteki numPy | 10^-14 | 50105 | 3 | 49892 |
| | 10^-18 | 50105 | 3 | 49892 |
| 2x2 funkcja własna | 10^-14 | 50105 | 3 | 49892 |
| | 10^-18 | 50105 | 3 | 49892 |
| 3x3 z biblioteki numPy | 10^-14 | 50105 | 3 | 49892 |
| | 10^-18 | 50106 | 1 | 49893 |
| 3x3 funkcja własna | 10^-14 | 50105 | 3 | 49892 |
| | 10^-18 | 50107 | 0 | 49893 |

**Tabela 3.7.1. Ilość punktów w każdej klasyfikacji dla zestawu 1 w zależności od wyznacznika i tolerancji dla zera**

| Wyznacznik | Tolerancja dla zera | Po lewej | Współliniowe | Po prawej |
|---|---|---|---|---|
| 2x2 z biblioteki numPy | 10^-14 | 49909 | 8 | 50083 |
| | 10^-18 | 49909 | 8 | 50083 |
| 2x2 funkcja własna | 10^-14 | 49909 | 7 | 50084 |
| | 10^-18 | 49909 | 7 | 50084 |
| 3x3 z biblioteki numPy | 10^-14 | 49912 | 0 | 50088 |
| | 10^-18 | 49912 | 0 | 50088 |
| 3x3 funkcja własna | 10^-14 | 49912 | 0 | 50088 |
| | 10^-18 | 49912 | 0 | 50088 |

**Tabela 3.7.2. Ilość punktów w każdej klasyfikacji dla zestawu 2 w zależności od wyznacznika i tolerancji dla zera**

| Wyznacznik | Tolerancja dla zera | Po lewej | Współliniowe | Po prawej |
|---|---|---|---|---|
| 2x2 z biblioteki numPy | 10^-14 | 513 | 0 | 487 |
| | 10^-18 | 513 | 0 | 487 |
| 2x2 funkcja własna | 10^-14 | 513 | 0 | 487 |
| | 10^-18 | 513 | 0 | 487 |
| 3x3 z biblioteki numPy | 10^-14 | 513 | 0 | 487 |
| | 10^-18 | 513 | 0 | 487 |
| 3x3 funkcja własna | 10^-14 | 513 | 0 | 487 |
| | 10^-18 | 513 | 0 | 487 |

**Tabela 3.7.3. Ilość punktów w każdej klasyfikacji dla zestawu 3 w zależności od wyznacznika i tolerancji dla zera**

| Wyznacznik | Tolerancja dla zera | Po lewej | Współliniowe | Po prawej |
|---|---|---|---|---|
| 2x2 z biblioteki numPy | 10^-14 | 115 | 740 | 145 |
| | 10^-18 | 122 | 724 | 154 |
| 2x2 funkcja własna | 10^-14 | 165 | 706 | 129 |
| | 10^-18 | 172 | 693 | 135 |
| 3x3 z biblioteki numPy | 10^-14 | 7 | 97 | 896 |
| | 10^-18 | 372 | 215 | 413 |
| 3x3 funkcja własna | 10^-14 | 0 | 1000 | 0 |
| | 10^-18 | 190 | 414 | 396 |

**Tabela 3.7.4. Ilość punktów w każdej klasyfikacji dla zestawu 4 w zależności od wyznacznika i tolerancji dla zera**

# 4. Wnioski

Jak widać na podstawie powyższych danych, różnice w klasyfikacji położenia punktu względem odcinka w zależności od metody obliczania wyznacznika są różne. Dla zbioru 4 różnice są znaczące, nie tylko dla tolerancji dla zera równej 10^-14 rozważanej przez większą część ćwiczenia ale również dla tolerancji równej 10^-18. Natomiast dla pozostałych zbiorów różnice są niewielkie, również porównując wyniki z tej samej metody liczenia wyznacznika dla różnej tolerancji.

Ciekawe jest to, że w zestawie 4 teoretycznie wszystkie punkty powinny być oznaczone jako współliniowe z odcinkiem, jednak wcale tak nie jest, w zależności od metody liczenia dostajemy zupełnie inne wyniki. Mimo to, można powiedzieć, że w zestawie 4 najlepiej poradził sobie wyznacznik 3x3 liczony przez funkcję własną, przy tolerancji 10^-14.

Podsumowując, nie można wybrać najlepszej metody do takiej klasyfikacji. Wyznacznik należy wybrać w zależności od potrzeb i wymaganych tolerancji.