

Sprawozdanie 1

Antoni Szczepański

Czwartek 16.15B

1. Cel ćwiczenia

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

2. Specyfikacje urządzenia

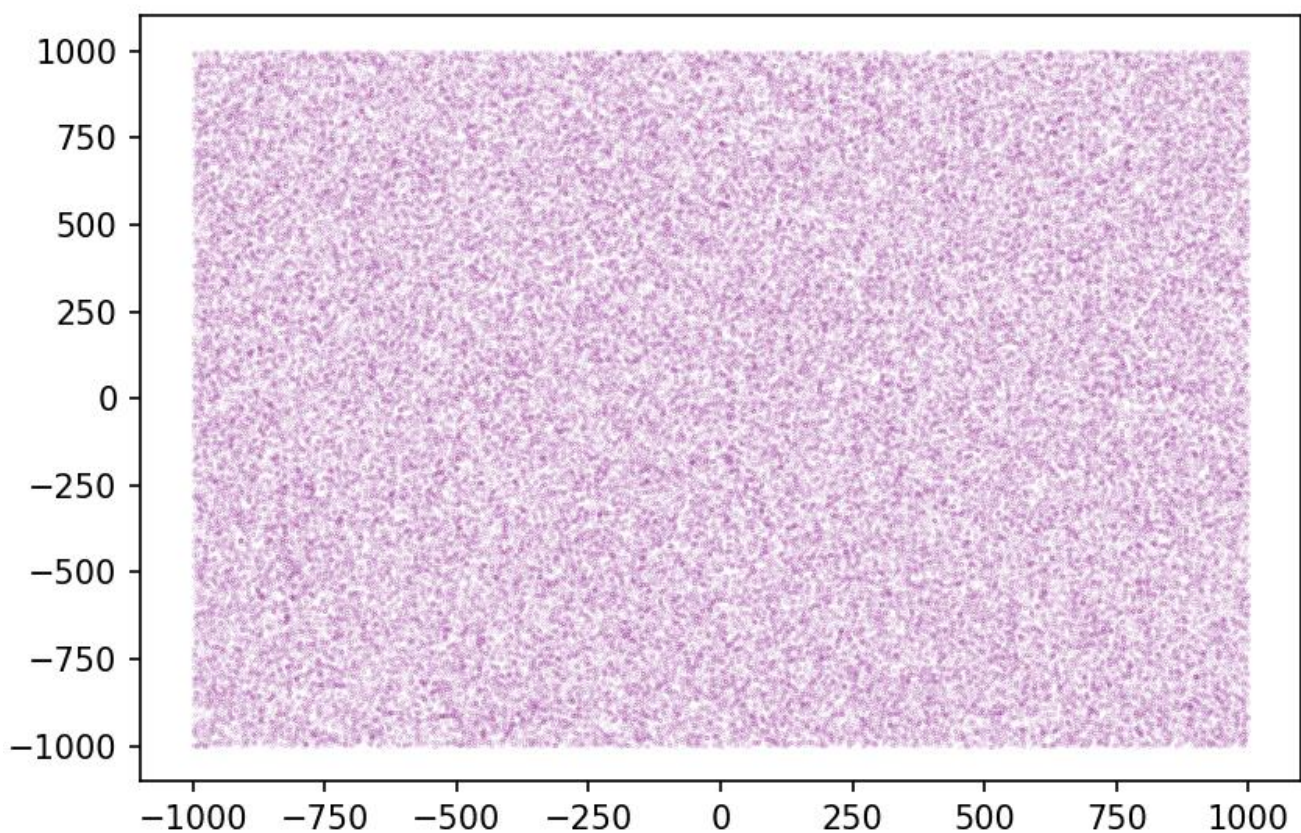
System: Windows 10 Home

Procesor: Intel® Core™ i5-8250U CPU @ 1.60GHz 1.80 GHz

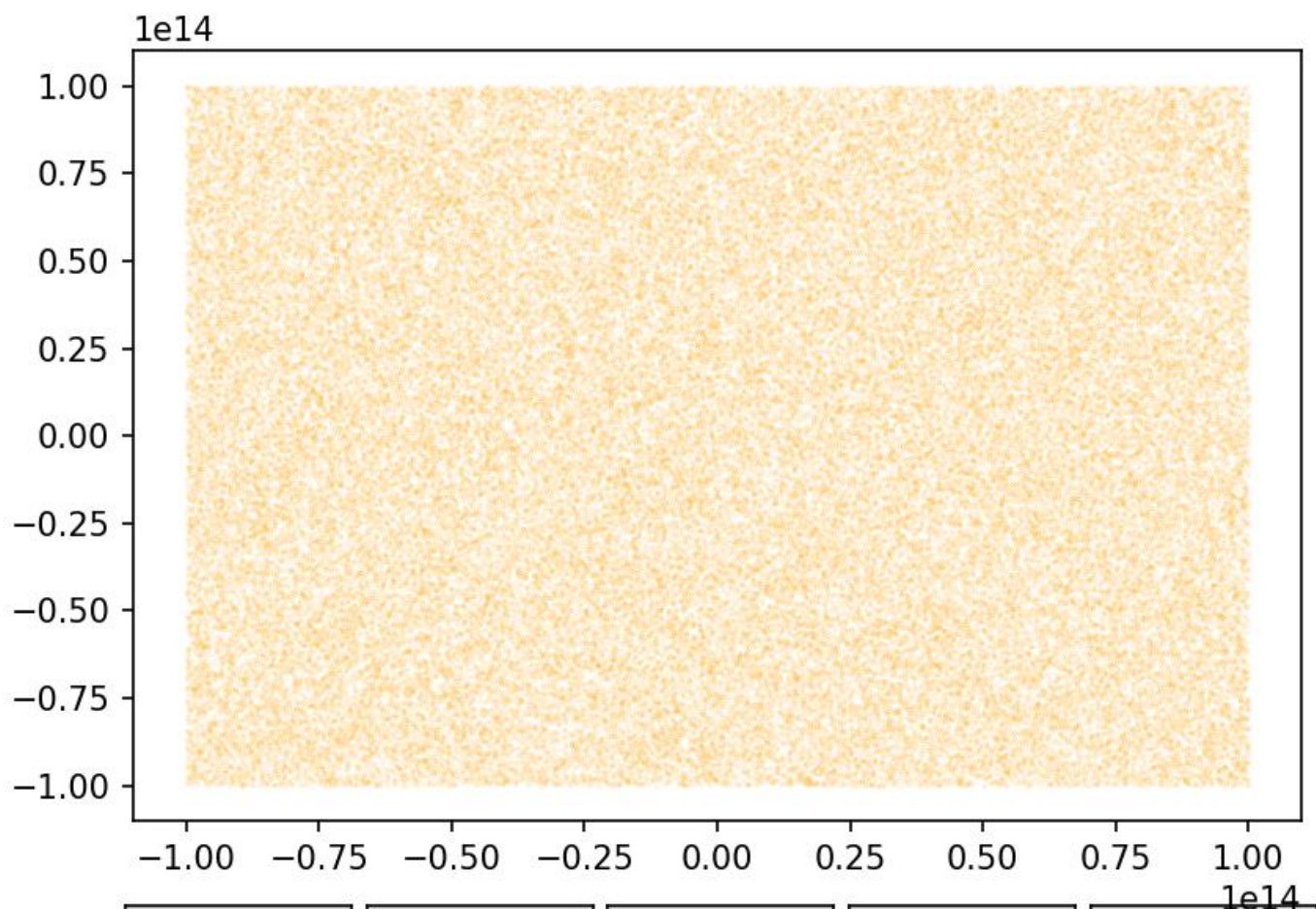
Typ systemu: 64-bitowy system operacyjny, procesor x64

3. Generowanie danych

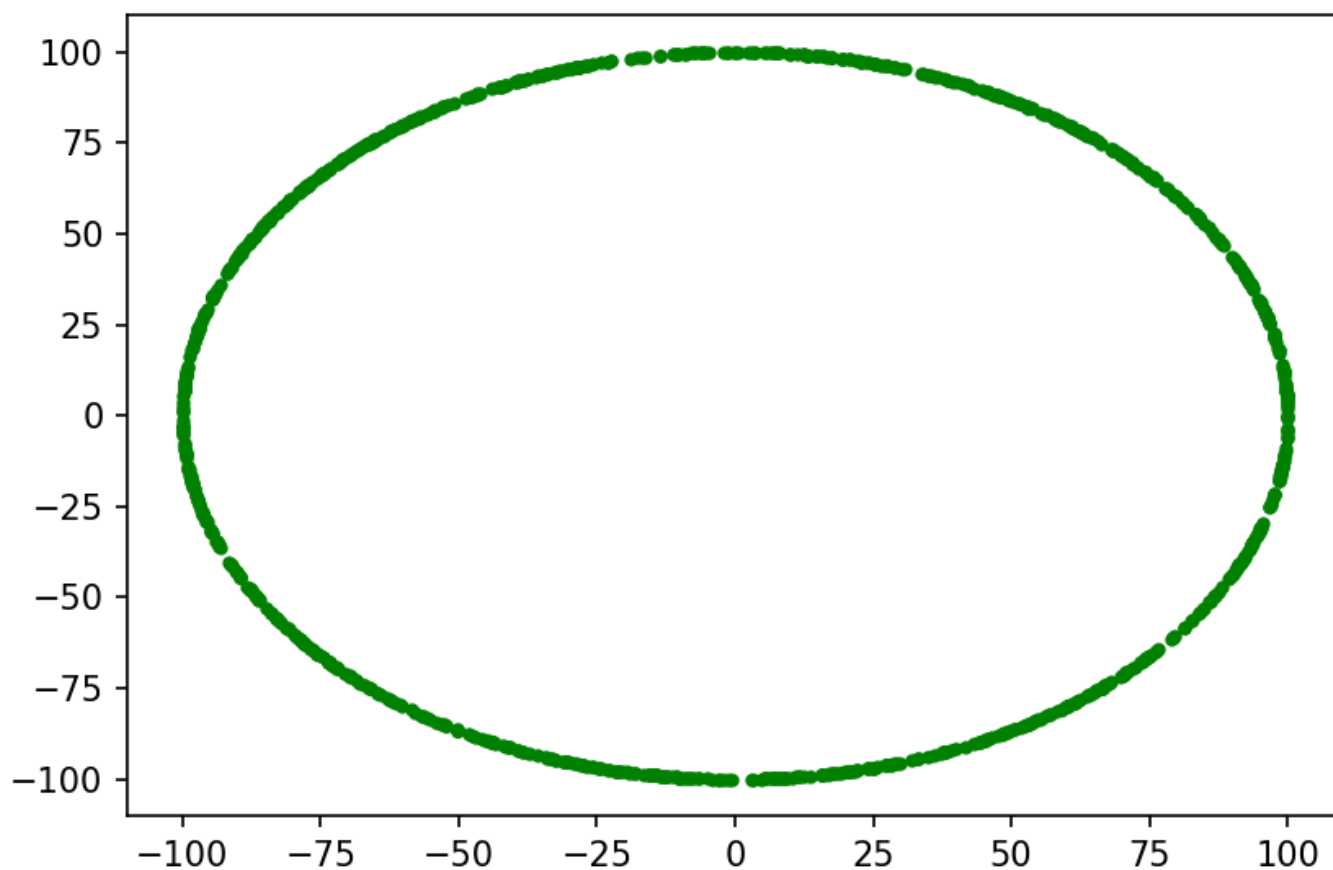
Zbiór A- 10^5 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$.



Zbiór B- 10^5 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$.

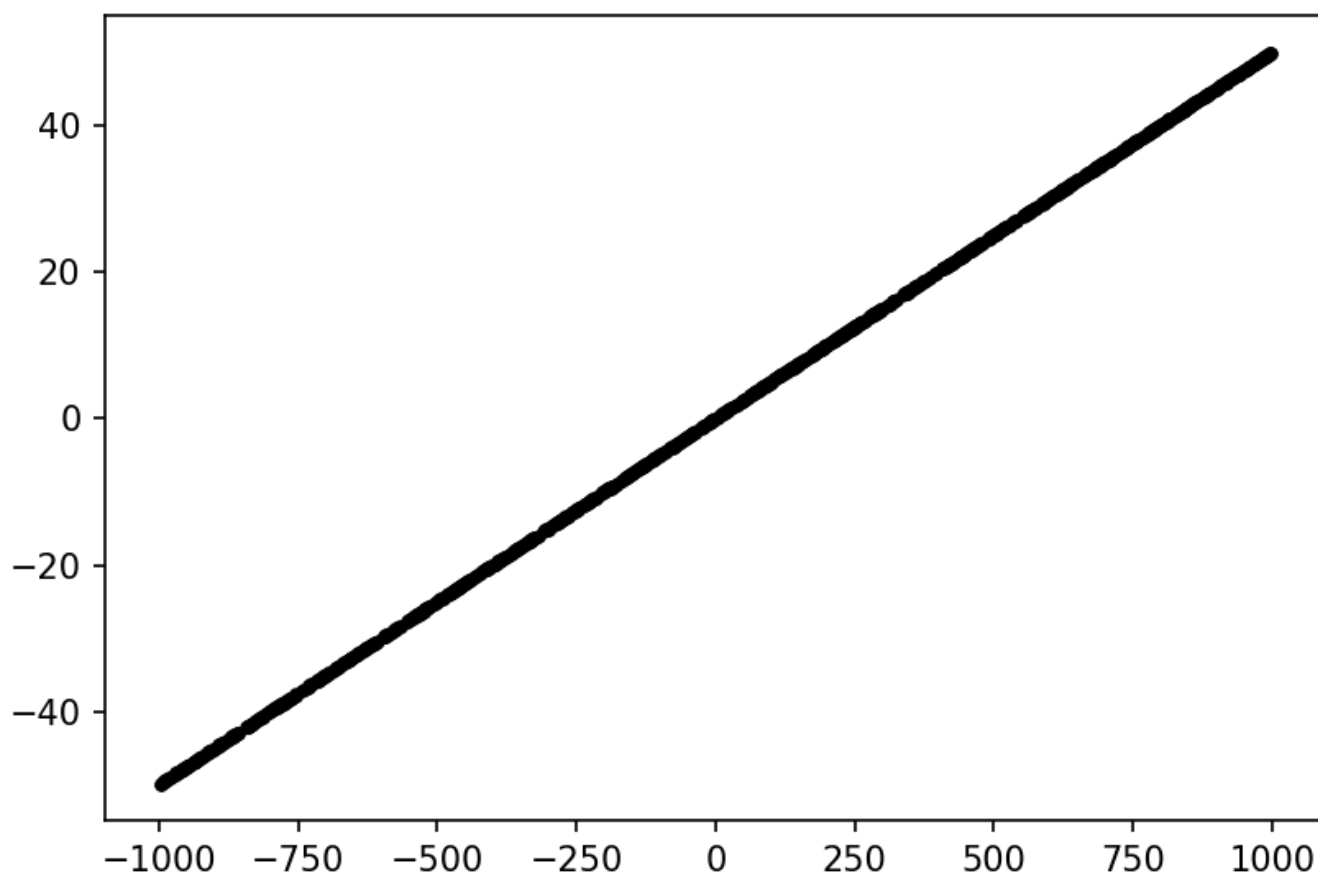


Zbiór C- 1000 losowych punktów leżących na okręgu o środku $(0,0)$ i promieniu $R=100$.



Zbiór D- 1000 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ leżących na prostej wyznaczonej przez wektor (a, b) .

$$a = [-1.0, 0.0] \quad b = [1.0, 0.1]$$



4. Testowanie wyznaczników dla różnych tolerancji dla zera.

W tabelach poniżej zamieszczam wyniki testów różnych wyznaczników dla różnych zbiorów danych, różnych precyzji oraz różnych tolerancji dla zera.

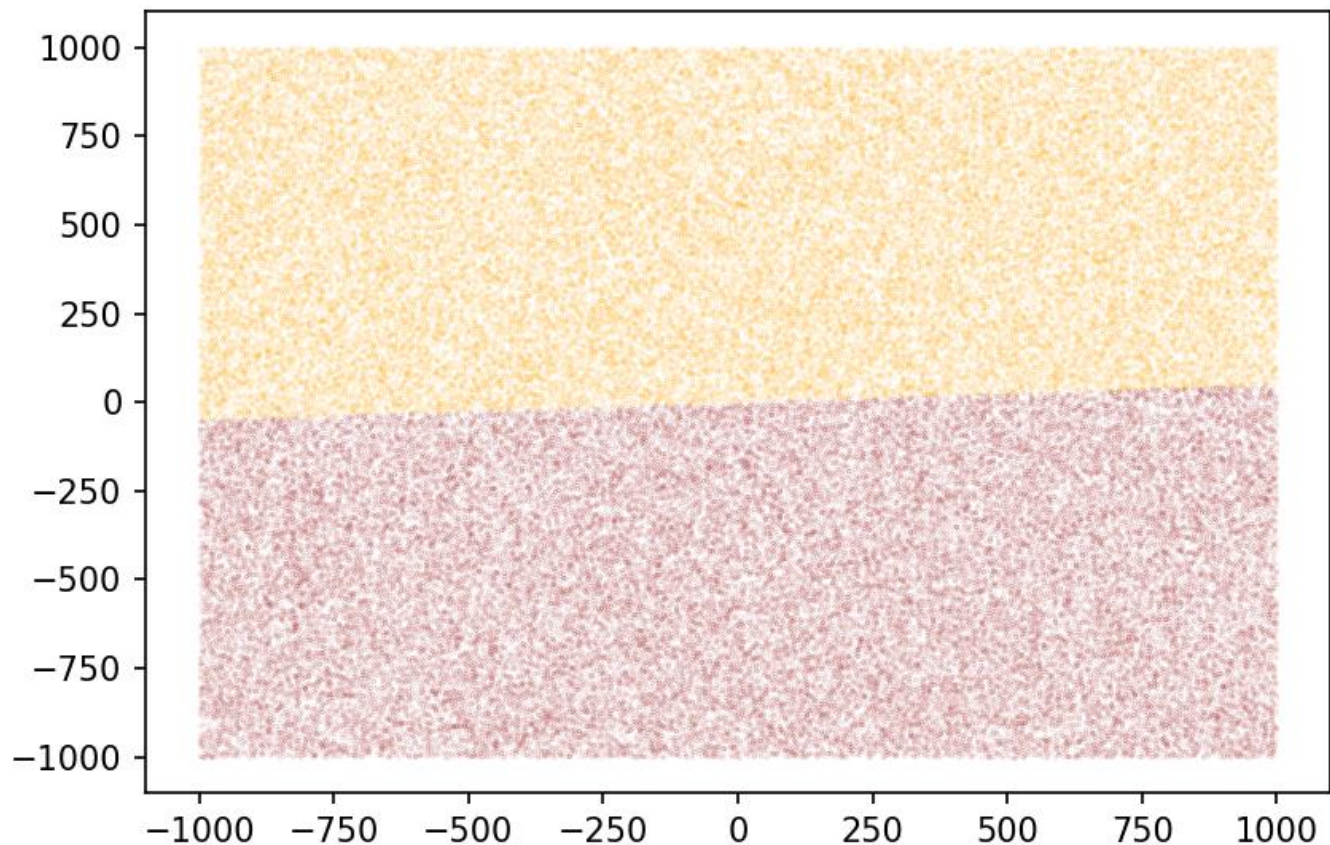
Pierwsza liczba w nawiasie oznacza ilość punktów znajdujących się po lewej stronie prostej, druga liczba oznacza ilość punktów współliniowych a trzecia ilość punktów znajdujących się po prawej stronie prostej.

Zbiór A:

| Pre- cyzja | e | det3x3 | det2x2 | np.det2x2 | np.det3x3 |
|---------------|------------|-------------------|-------------------|-------------------|-------------------|
| float64 | 10^{-8} | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |
| | 10^{-10} | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |
| | 10^{-12} | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |
| | 10^{-16} | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |

| | | | | | |
|---------|--------------|-------------------|-------------------|-------------------|-------------------|
| float32 | $10^{(-8)}$ | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |
| | $10^{(-10)}$ | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |
| | $10^{(-12)}$ | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |
| | $10^{(-16)}$ | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) | (49955, 0, 50045) |

Wizualizacja podziału zbioru danych dla $e=10^{(-16)}$ i precyzji float32.

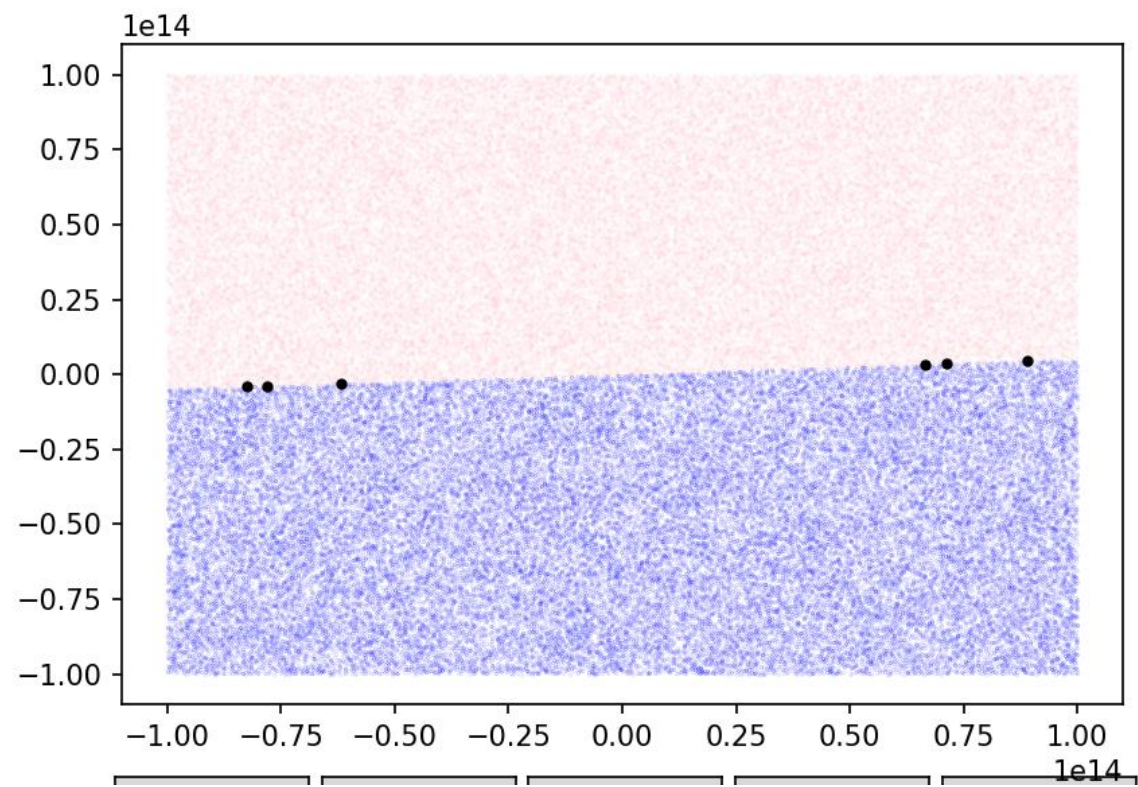


Zbiór A dla każdego przypadku został tak samo podzielony.

Zbiór B:

| Pre- cyzja | e | det3x3 | det2x2 | np.det2x2 | np.det3x3 |
|---------------|--------------|-------------------|-------------------|-------------------|-------------------|
| float64 | $10^{(-8)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (49994, 0, 50006) | (49994, 0, 50006) |
| | $10^{(-10)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (49994, 0, 50006) | (49994, 0, 50006) |
| | $10^{(-12)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (49994, 0, 50006) | (49994, 0, 50006) |
| | $10^{(-16)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (49994, 0, 50006) | (49994, 0, 50006) |
| float32 | $10^{(-8)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (0, 100000, 0) | (49994, 0, 50006) |
| | $10^{(-10)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (0, 100000, 0) | (49994, 0, 50006) |
| | $10^{(-12)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (0, 100000, 0) | (49994, 0, 50006) |
| | $10^{(-16)}$ | (49994, 0, 50006) | (49990, 6, 50004) | (0, 100000, 0) | (49994, 0, 50006) |

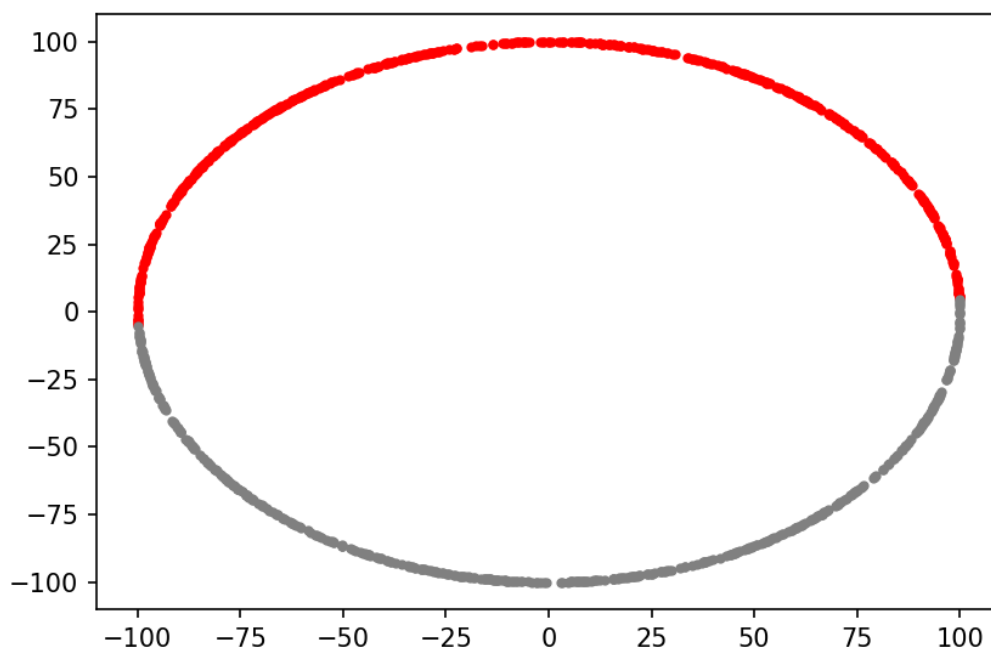
Podział punktów za pomocą własnego wyznacznika 2x2 dla precyzji float32 i $\epsilon=10^{-16}$. Jak widać 6 punktów zostało zakwalifikowane jako współliniowe.



Zauważyć można, że dla własnego wyznacznika 2x2 za każdym razem 6 punktów jest klasyfikowanych jako współliniowe. Spowodowane jest to prawdopodobnie tym, że przy wyliczaniu współczynnika 2x2 odejmujemy od siebie liczby, przez co wyniki są mniej precyzyjne.
Zarówno dla własnej metody wyliczania współczynnika 3x3 jak i wbudowanej, nie występują żadne problemy z podziałem punktów.

Zbiór C:

| Pre- cyzja | e | det3x3 | det2x2 | np.det2x2 | np.det3x3 |
|---------------|---------------------|---------------|---------------|---------------|---------------|
| float64 | 10 ⁽⁻⁸⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| | 10 ⁽⁻¹⁰⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| | 10 ⁽⁻¹²⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| | 10 ⁽⁻¹⁶⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| float32 | 10 ⁽⁻⁸⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| | 10 ⁽⁻¹⁰⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| | 10 ⁽⁻¹²⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |
| | 10 ⁽⁻¹⁶⁾ | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) | (515, 0, 485) |

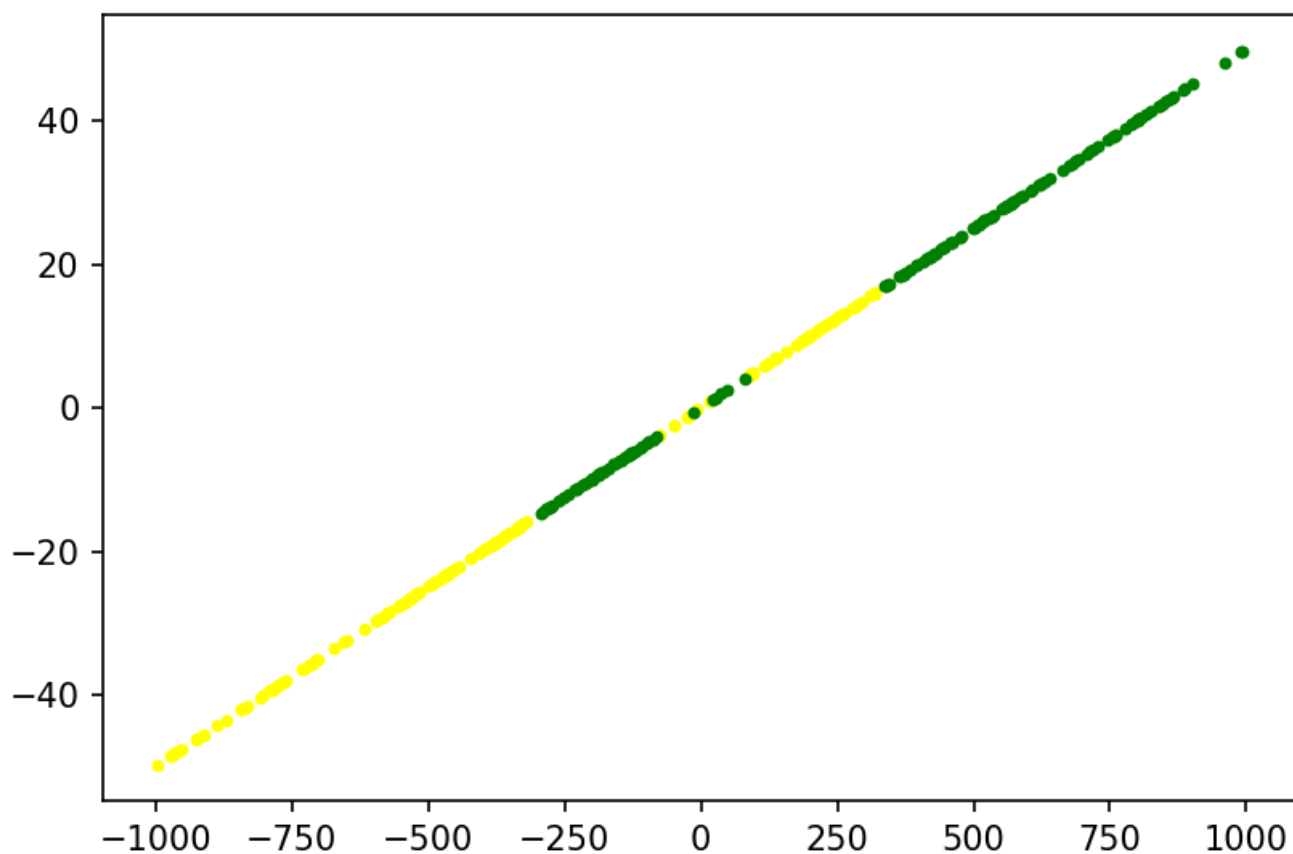


Dla każdej metody wyliczania wyznacznika, tolerancji oraz precyzji dane zostały identycznie podzielone.

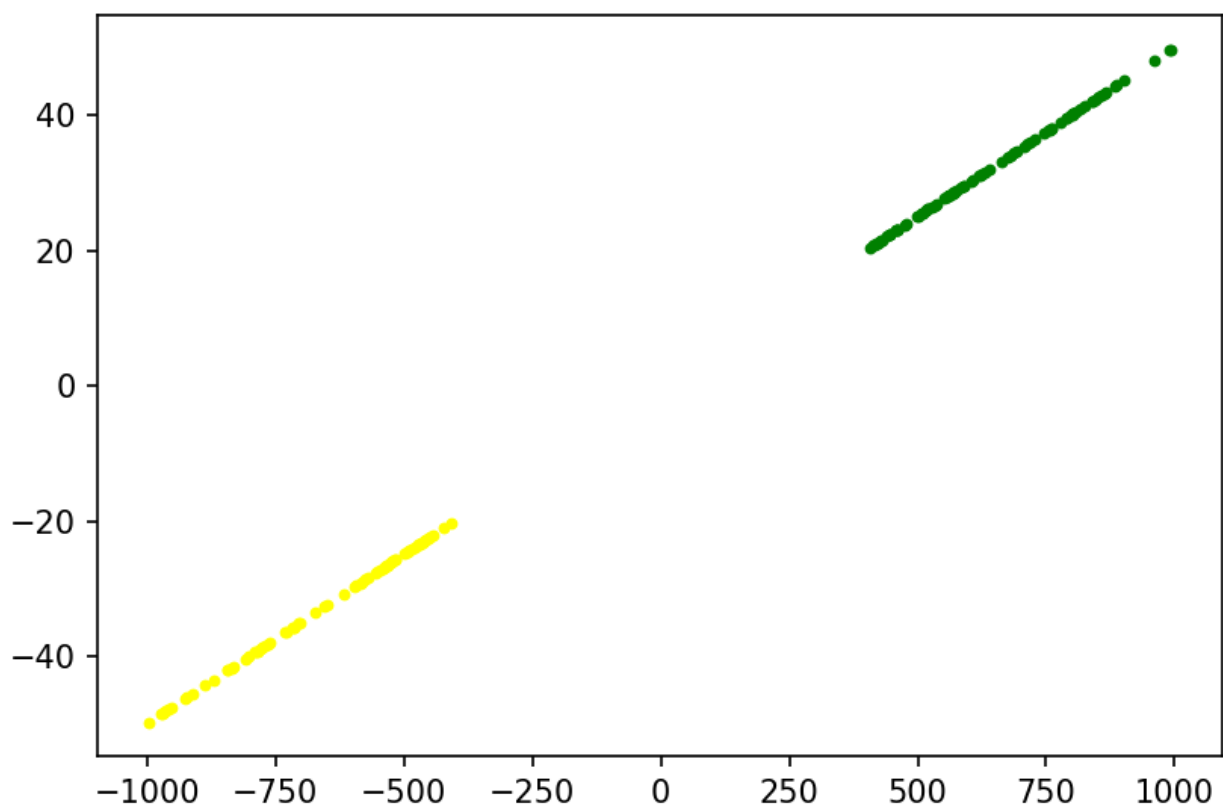
Zbiór D:

| Pre- cyzja | e | det3x3 | det2x2 | np.det2x2 | np.det3x3 |
|---------------|------------|-----------------|-----------------|---------------|----------------|
| float64 | 10^{-8} | (0,1000, 0) | (0,1000, 0) | (0,1000, 0) | (0,1000, 0) |
| | 10^{-10} | (0,1000, 0) | (0,1000, 0) | (0,1000, 0) | (0,1000, 0) |
| | 10^{-12} | (0,1000, 0) | (77,836,87) | (133,716,151) | (0,1000, 0) |
| | 10^{-16} | (176,410,414) | (149,701,150) | (500,8,492) | (442,48,510) |
| | 0 | (176, 405, 419) | (149, 700, 151) | (504, 0, 496) | (466, 0, 534) |
| float32 | 10^{-8} | (0,1000, 0) | (0,1000, 0) | (496, 1, 503) | (495, 16, 489) |
| | 10^{-10} | (0,1000, 0) | (0,1000, 0) | (496, 0, 504) | (504, 0, 496) |
| | 10^{-12} | (0,1000, 0) | (77, 836, 87) | (496, 0, 504) | (504, 0, 496) |
| | 10^{-16} | (176, 410, 414) | (149, 701, 150) | (496, 0, 504) | (504, 0, 496) |
| | 0 | (176, 405, 419) | (149, 700, 151) | (496, 0, 504) | (504, 0, 496) |

Po przeanalizowaniu tabeli stwierdzam, że w większości przypadków własny wyznacznik 3x3 radzi sobie najlepiej, jednak dla bardzo niskich tolerancji nieco lepiej radzi sobie własny wyznacznik 2x2. Po zmniejszeniu precyzji wbudowane wyznaczniki źle kwalifikują większość punktów, dlatego też są nieprzydatne.



Graficzny podział punktów zakwalifikowanych jako leżące na lewo(żółte) oraz prawo(zielone) od prostej dla własnego wyznacznika 2x2 dla precyzji float64 i tolerancji 10^{-16} .



Graficzny podział punktów zakwalifikowanych jako leżące na lewo(żółte) oraz prawo(zielone) od prostej dla własnego wyznacznika 2x2 dla precyzji float32 i tolerancji 10^{-12} .

Różnica w liczbie punktów współliniowych między własnym wyznacznikiem 3x3 a własnym wyznacznikiem 2x2.

| e | różnica(float64) | różnica(float32) |
|------------|------------------|------------------|
| 10^{-8} | 0 | 0 |
| 10^{-10} | 0 | 0 |
| 10^{-12} | 164 | 164 |
| 10^{-16} | 291 | 291 |
| 0 | 295 | 295 |

5. Wnioski.

Analizując wyniki stwierdzam, iż w większości przypadków własny wyznacznik 3x3 sprawdza się najlepiej. Jedynie dla ostatniego zbioru danych przy bardzo małej tolerancji własny wyznacznik 2x2 działa nieco lepiej od własnego 3x3. Wyznaczniki z biblioteki numpy okazały się w większości przypadków gorsze od zaimplementowanych samodzielnie.

Ćwiczenie pokazuje również jak bardzo ważna jest precyzja obliczeń. Dla lepszej precyzji wyniki są zdecydowanie lepsze. Efekt ten głównie można zaobserwować w zbiorze D przy wyliczaniu wyznacznika metodami z biblioteki numpy. Dla większej precyzji wyniki były zdecydowanie lepsze od tych wyliczanych z mniejszą precyzją.