# Juliusz Wasieleski Informatyka II rok, grupa 1

Algorytmy geometryczne, laboratorium 1 - sprawozdanie

# 1.Opis ćwiczenia

Na pierwszych zajęciach laboratoryjnych naszym zadaniem było określanie po której stronie prostej leżą punkty. Do wykonania go używaliśmy jednego z wyznaczników:

Wyznacznik macierzy 3x3:



Wyznacznik macierzy 2x2:



W powyższym wzorze ax, ay, bx, by, cx, cy to współrzędne x i y odpowiednio punktów a, b i c. Jeżeli wyznacznik okaże się większy od 0 to punkt znajduje się po prawej stronie od prostej, a jeśli mniejszy od zera to po lewej stronie od prostej. Będę 0 przybliżał wartościami ε, ponieważ porównywanie liczb rzeczywistych w komputerach nie jest idealne i należy brać je z pewnym przybliżeniem. Dodatkowo mimo tego, że powyższe dwa wyznaczniki pod względem matematycznym są identyczne to przez niedokładności obliczeniowe komputerów wyniki uzyskane nimi będą mogły się różnić.

# 2. Środowisko, biblioteki, założenia oraz użyte narzędzia

Ćwiczenie wykonałem w Jupyter Notebook i napisałem w języku Python. Dodatkowo zostały użyłem bibliotekę numpy aby policzyć wyznaczniki funkcjami bibliotecznymi. Do rysowania wykresów użyłem bibliotekę matplotlib, która obrazowała rozkład danych w postaci graficznej. Za wartości ε przyjąłem 10-18, 10-14, 10-10. Do graficznej reprezentacji punktów przyjąłem konwencję:

- na niebiesko punkty powyżej prostej

- na zielono punkty poniżej prostej

- na czerwono punkty na prostej

Wszystkie obliczenia prowadziłem na komputerze Lenovo Y50-70 z systemem Windows 10 Pro w wersji 10.0.19043. Procesor Intel Core i7-4720HQ 2.60GHz, 2601 MHz, Rdzenie: 4, Procesory logiczne: 8

# 3. Plan i sposób wykonania ćwiczenia

Na początku należało wygenerować zbiory punktów o współrzędnych rzeczywistych typu float (w Pythonie mają one 64 bity):

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

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

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

R=100,

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].

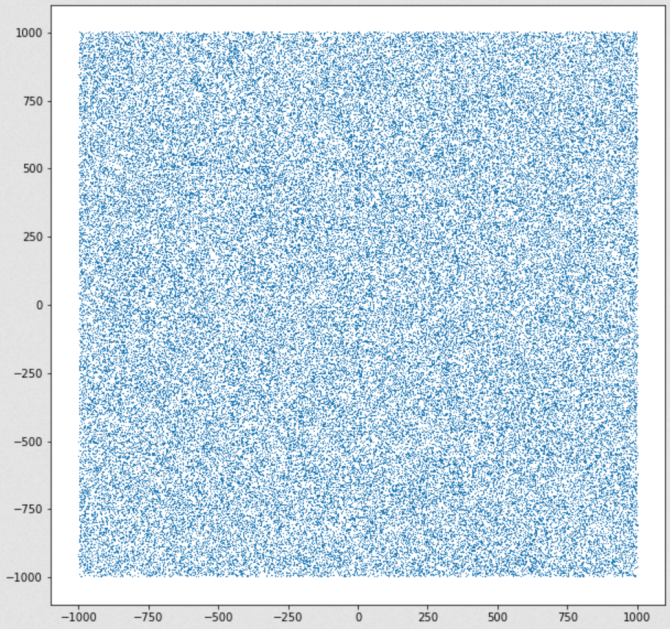
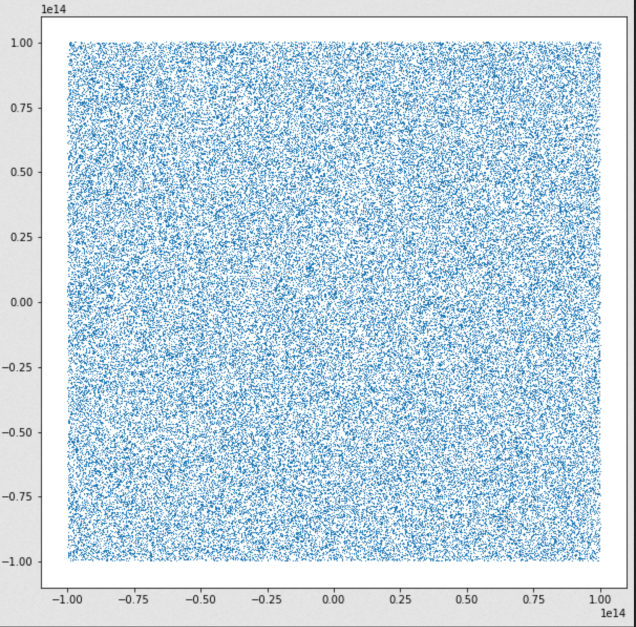
Kolejnym krokiem było zaimplementowanie funkcji obliczających wyznaczniki 2x2, 3x3 oraz skorzystanie z gotowych wersji bibliotecznych.

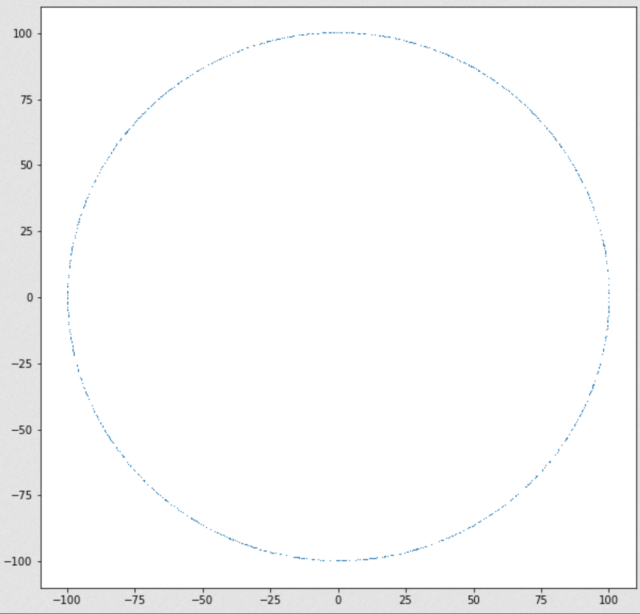
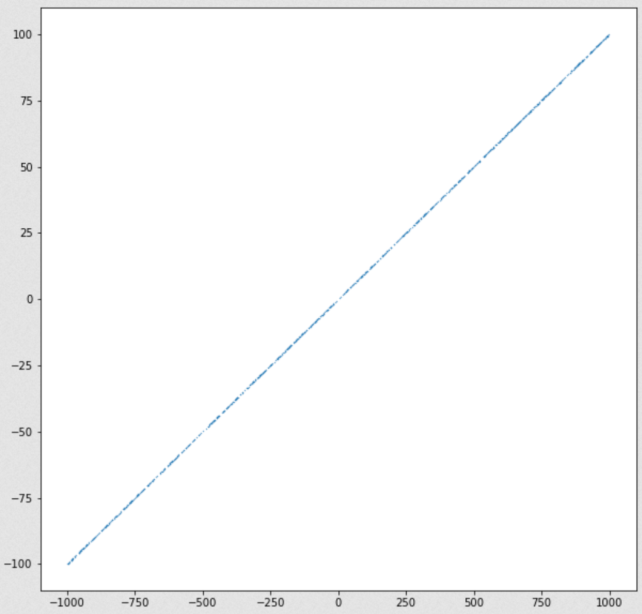
Następnie przedstawiłem uzyskane wyniki na wykresach oraz sprawdziłem jakie są różnice między moim wyznacznikiem 2x2 i bibliotecznym oraz moim wyznacznikiem 3x3 i bilbiotecznym. Na koniec porównałem wyniki które otrzymywałem przy różnych wartościach ε.

# 4. Wykonanie ćwiczenia

## 4.1 Wygenerowanie punktów

Do wygenerowania punktów użyłem funkcji uniform z biblioteki random, następnie przedstawiłem na wykresach otrzymane zbiory punktów:

a) Zbiór A: b) Zbiór B:  
     
(rys. 1) (rys.2)

c) Zbiór C: d) Zbiór D:  
   
(rys.3) (rys.4)

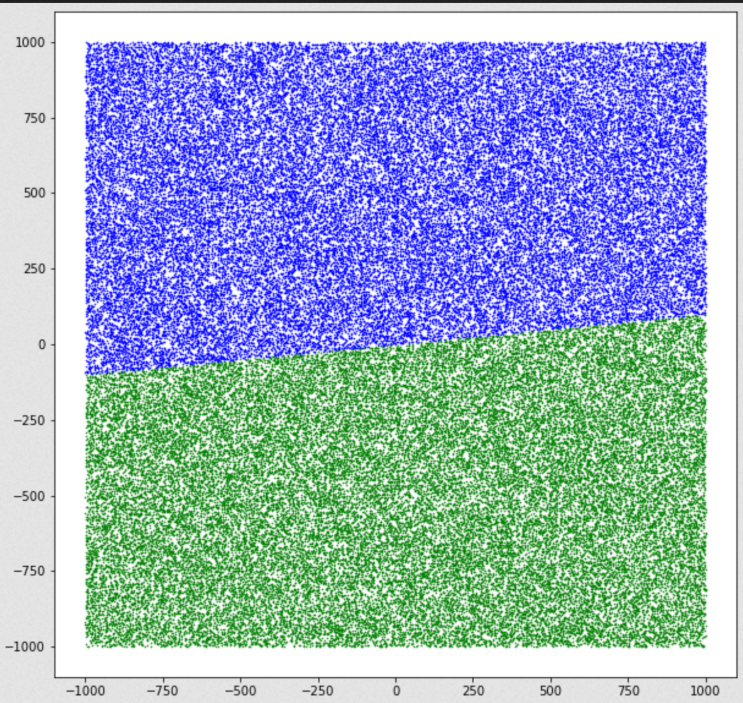
## 4.2 Klasyfikacja punktów zależnie od wyznacznika i tolerancji

Powyższe zadanie wykonałem w funkcji make\_groups która dodatkowo pogrupowała te punkty jako znajdujące się po prawo, po lewo oraz na prostej. To wszystko w trzech wariantach związanych z wartością ε. Na poniższych wykresach pokażę ten podział dla najmniejszej wartości ε = 10-18. Poniżej umieszczam tylko niektóre wykresy:

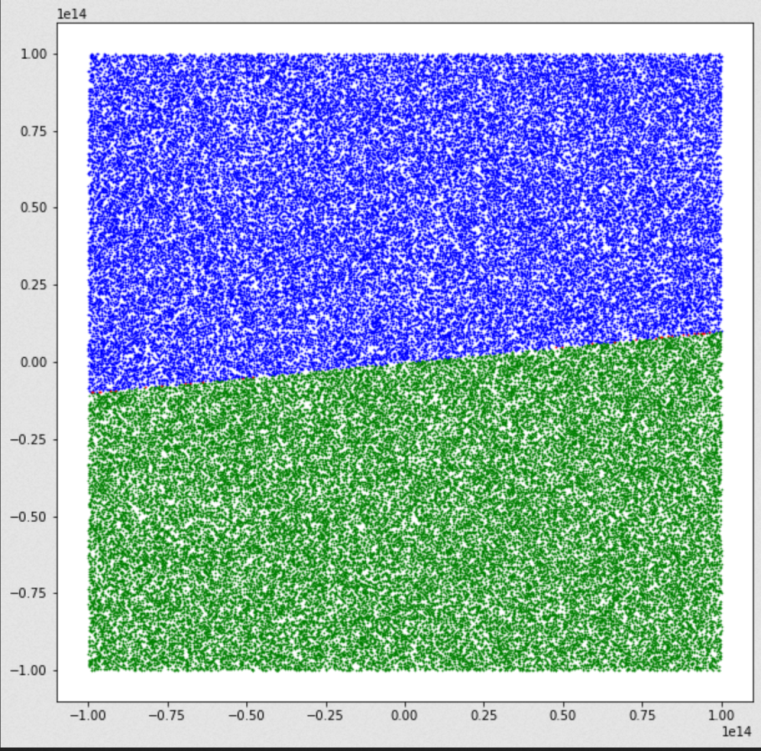
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Wyznacznik: Zbiór: | 2x2 z biblioteki numpy | 2x2 mojej implementacji | 3x3 z biblioteki numpy | 3x3 mojej implementacji |
| A | Nad: 50077  Na: 0  Pod: 49923 | Nad: 50077  Na: 0  Pod: 49923 | Nad: 50077  Na: 0  Pod: 49923 | Nad: 50077  Na: 0  Pod: 49923 |
| B | Nad: 49940  Na:0  Pod: 50060 | Nad: 49926  Na:28  Pod: 50046 | Nad: 49941  Na:0  Pod: 50059 | Nad: 49941  Na:0  Pod: 50059 |
| C | Nad: 509  Na: 0  Pod:491 | Nad: 509  Na: 0  Pod:491 | Nad: 509  Na: 0  Pod:491 | Nad: 509  Na: 0  Pod:491 |
| D | Nad: 531  Na:0  Pod: 496 | Nad: 165  Na:684  Pod:151 | Nad:355  Na: 205  Pod: 440 | Nad: 0  Na: 1000  Pod:0 |

(Tabela 1)

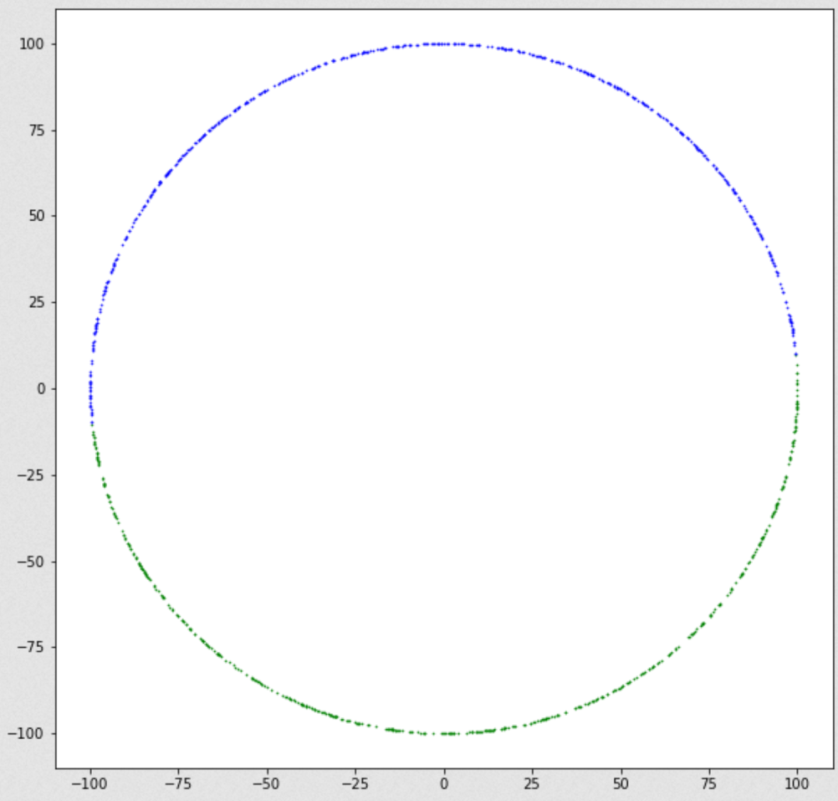
1. Zbiór A podzielony wyznacznikiem 2x2 z bibliolteki numpy:

(rys 5)

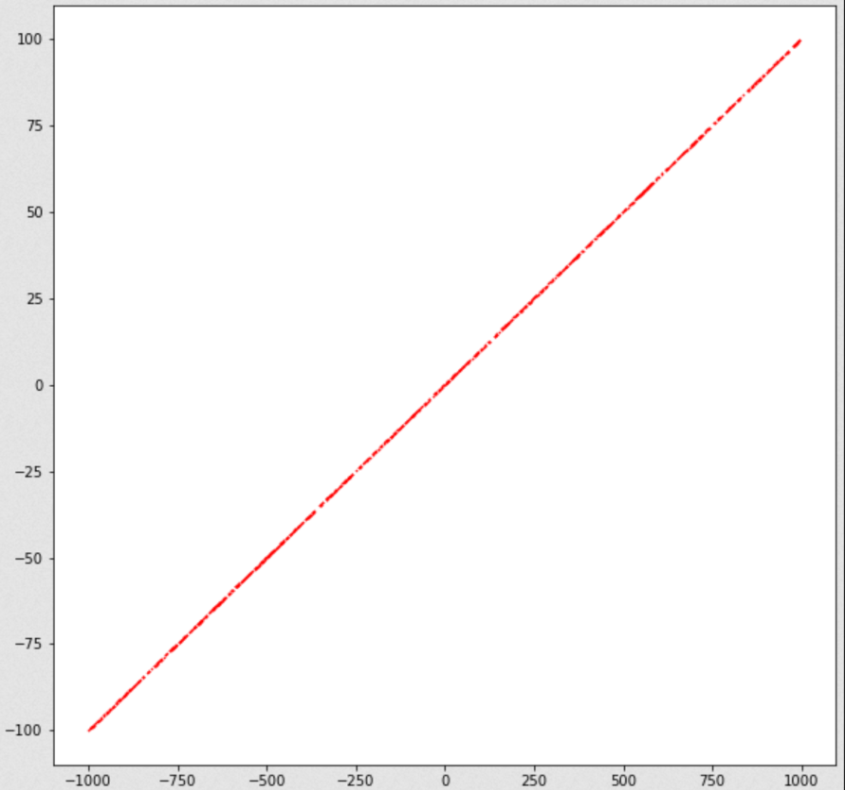
1. Zbiór B podzielony wyznacznikiem 2x2 mojej implementacji:

(rys 6)

1. Zbiór C podzielony wyznacznikiem 3x3 z biblioteki numpy:

(rys 7)

1. Zbiór D podzielony wyznacznikiem 3x3 mojej implementacji:

(rys 8)

## 4.3 Porównanie klasyfikacji punktów ze względu na wyznacznik

Ten podpunkt realizuje funkcja differences, a wizualizację wyników organizuje funkcja show\_differences. Porównywałem wyznaczniki 2x2 ze sobą oraz 3x3 ze sobą następnie wypisywałem które punkty się nie zgadzają między tymi wyznacznikami a następnie na wykresie rysowałem te punkty. Otrzymane wyniki:

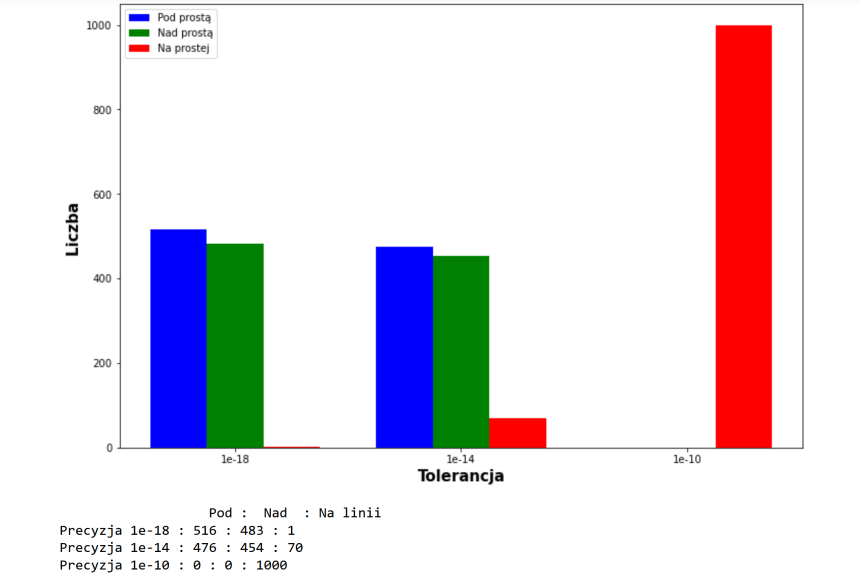
1. W zestawie A nie ma różnic
2. W zestawie B różnice są tylko między wyznacznikami 2x2 (28 punktów jest zaklasyfikowanych inaczej)
3. W zestawie C nie ma różnic
4. W zestawie D wyznaczniki 2x2 różnią się 695 punktami, a 3x3 786 punktami

Muszę zaznaczyć że powyższe wyniki były wyznaczane dla ε = 10-18. Przy niższych tolerancjach nie byłoby aż tak dużych różnic. Dokładniej omówię to w kolejnym podpunkcie. Wykresy punktów są dostępne w pliku Jupyter Notebook, nie umieszczam ich poniżej gdyż wszystkie leżą prawie na prostej wskazywanej przez wektor AB więc na wykresach upodabniają tę prostą

## 4.4 różnice między wynikami ze względu na tolerancję

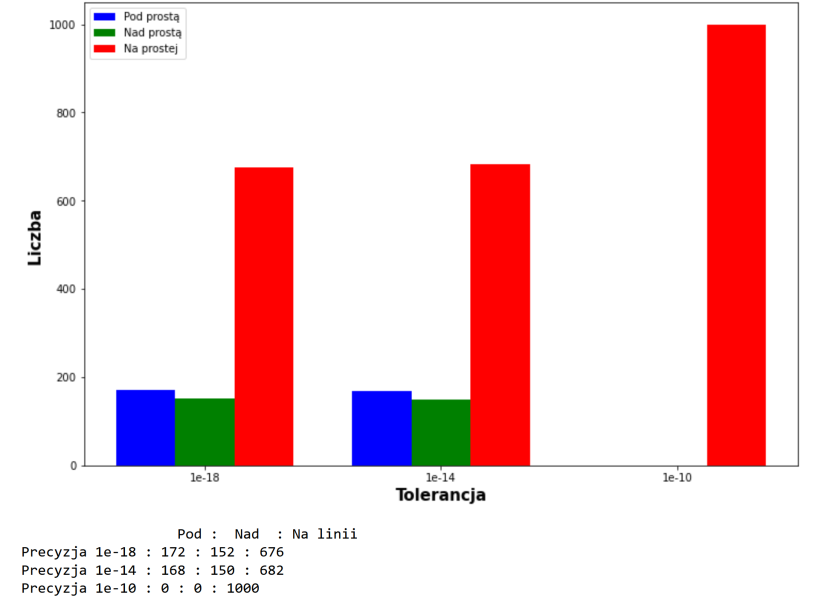
Jako ostatnią zaimplementowałem funkcję generate\_diagram która rysuje wykresy słupkowe przedstawiające dla każdej tolerancji ε ile punktów jest klasyfikowanych pod, ile nad, a ile na prostej. Znaczące różnice pojawiły się jedynie w danych z zestawu D. Poniżej zamieszczam wykresy tych różnic oraz podliczenie ile punktów jak było zaklasyfikowanych. Pozostałe wykresy są w Jupyter Notebook:

a) Obliczenia wyznacznikiem 2x2 z bilbioteki numpy

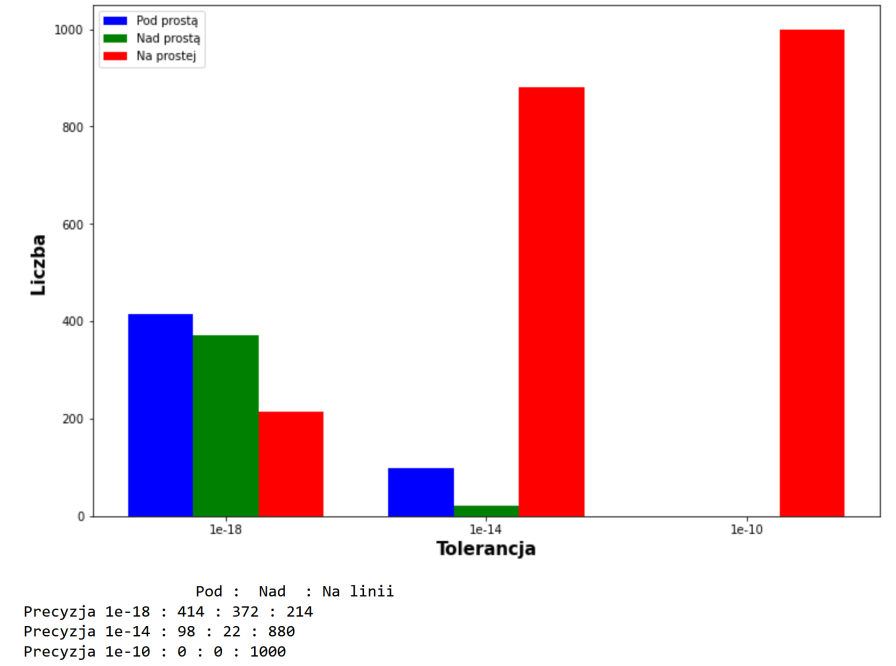


(rys. 9)

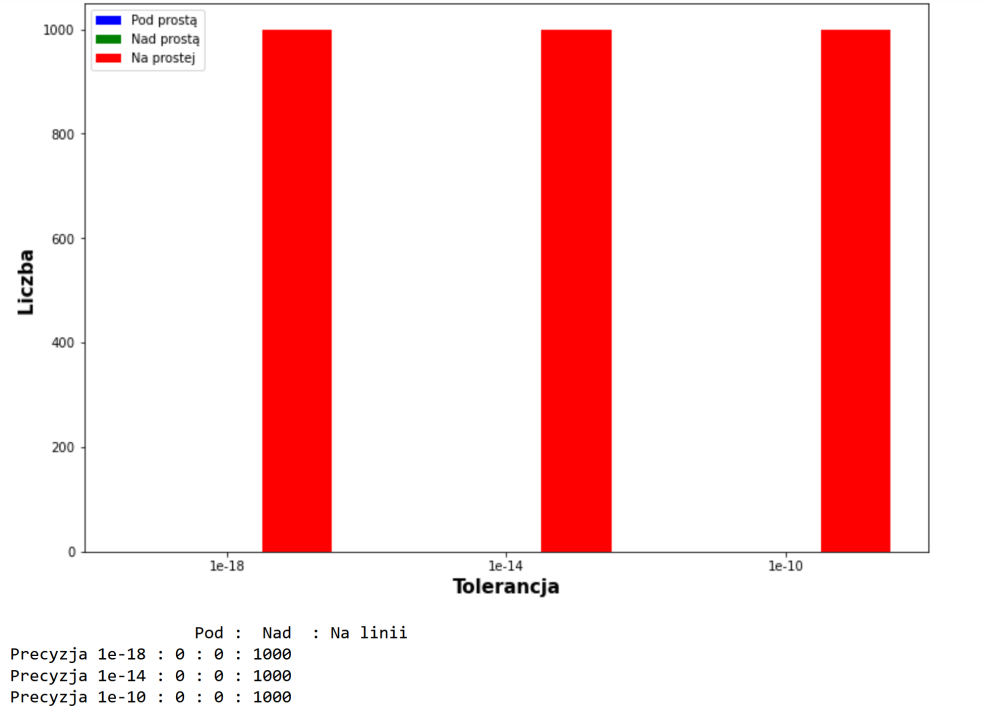
b)Obliczenia wyznacznikiem 2x2 mojej implementacji:

(rys 10)

c)Obliczenia wyznacznikiem 3x3 z biblioteki numpy:

(rys. 11)

d)Obliczenia wyznacznikiem 3x3 mojej implementacji

(rys 12)

# 7. Wnioski

Dla danych z podpunktu:

- A ani wyznacznik ani ε nie mają znaczenia

- B wyznacznik ma znaczenie, ale ε już nie

- C ani wyznacznik ani ε nie mają znaczenia

- D zarówno wyznacznik jak i ε mają znaczenie

Wyniki dla zbiorów A i C są najprawdopodobniej spowodowane tym, że istnieje bardzo małe prawdopodobieństwo, iż punkt będzie znajdował się dokładnie na prostej, z dokładnością do użytej tolerancji oraz punkty są z relatywnie nie aż tak dużego zakresu.  
W B jedynie wyznacznik 2x2 mojej implementacji dawaj inne wyniki od pozostałych. Zapewne wynika to z faktu, że współrzędne z tego zbioru mają duże wartości bezwzględne i podczas klasyfikacji dokładność jest tracona na rzecz dużych liczb.

Dla zbioru D najlepsze wyniki (najwięcej punktów zaklasyfikowanych do prostej uzyskałem przy dokładności 10-10. Wtedy już wszystkie punkty były klasyfikowane do prostej niezależnie od wyznacznika. Co ciekawe lepsze wyniki dawały wyznaczniki mojej implementacji a nie biblioteczne, możliwe, że jest to wynikiem mniejszej dokładności funkcji bibliotecznych która jest uzyskiwana przez chęć optymalizacji obliczeń. Widać też, że wyznaczniki 3x3 „lepiej” radzą sobie z tymi obliczeniami niż 2x2. Najprawdopodobniej jest tak ponieważ w wyznacznikach 3x3 dokonujemy mniej przybliżeń.

Dodatkowo, zaskakujące jest to, że nawet przy tolerancji rzędu 10-18 mój wyznacznik klasyfikuje wszystkie punkty ze zbioru D jako znajdujące się na prostej (czyli poprawnie). Zdarzało się tak za każdym razem podczas generowania nowego zbioru punktów (wykonałem 5 generowań). Nie jestem pewien co może być tego przyczyną, być może przypadkiem generuję punkty zbyt tendencyjnie lub miałem szczęście, że za każdym razem mój wyznacznik „poradził sobie” z taką niedokładnością.