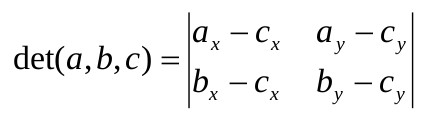
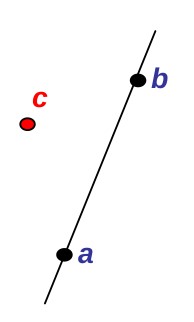
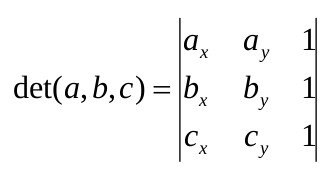
Jan Kaczmarski

Informatyka II rok, grupa 5, 15.10.2024

**Algorytmy geometryczne, laboratorium 1 - sprawozdanie**

# 1.Opis ćwiczenia

Zadaniem które należy wykonać na laboratorium 1 jest określenie po której stronie prostej znajduje się punkt. Można to wykonać przy pomocy algorytmu obliczającego wartość jednego z dwóch wyznaczników:



1. Wyznacznik macierzy 3x3:
2. Wyznacznik macierzy 2x2:

Gdzie ax i ay to współrzędne odpowiednio x i y punktu a (tak samo z punktami b i c). Powyższe wyznaczniki pozwalają określić położenie punktu c względem prostej która jest wyznaczona przez punkty a i b. Jeżeli wyznacznik jest większy od 0 (albo od innej tolerancji dla zera epsilon) to punkt znajduje się z lewej strony prostej, jeżeli jest mniejszy od 0 (albo od innej tolerancji dla zera którą zmieniamy na ujemną) to punkt znajduje się po prawej stronie prostej a jeżeli wartość wyznacznika jest równa 0 (lub równa epsilon) to punkt leży na prostej. Pomimo, że powyższe wyznaczniki są sobie równoważne to na skutek niedoskonałości reprezentacji liczb rzeczywistych w komputerze wyniki mogą się różnić w zależności od użytego wyznacznika.

# Środowisko, biblioteki oraz użyte narzędzia

Ćwiczenie zostało wykonane w Jupyter Notebook i napisane w języku Python z wykorzystaniem narzędzia Visualizer autorstwa koła naukowego BIT do wizualizacji obliczeń i rysowania wykresów. Wszystko było wykonywane na systemie operacyjnym macOS Sonoma14.6.1 I procesorze Apple M2 Pro arm64

# Plan i sposób wykonania ćwiczenia

Wygenerowanie zbiorów punktów typu float:

1. 10^5 losowych punktów o współrzędnych z przedziału [-1000, 1000],
2. 10^5 losowych punktów o współrzędnych z przedziału [-10^14 , 10^14 ],
3. 1000 losowych punktów leżących na okręgu o środku (0,0) i promieniu R=100,
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].

Do wygenerowania poniższych wykresów używam

a) Wykres\_a b) Wykres\_b

A green rectangular object with numbers and lines

Description automatically generatedA green rectangular object with numbers

Description automatically generated

c) Wykres\_c d) Wykres\_d

A green oval with numbers and a graph

Description automatically generatedA green line graph with numbers

Description automatically generated

Następnie użyłem cztery funkcje obliczające wyznaczniki, które pozwolą nam zdeterminować po której stronie prostej wyznaczonej przez dwa punkty leży trzeci punkt:

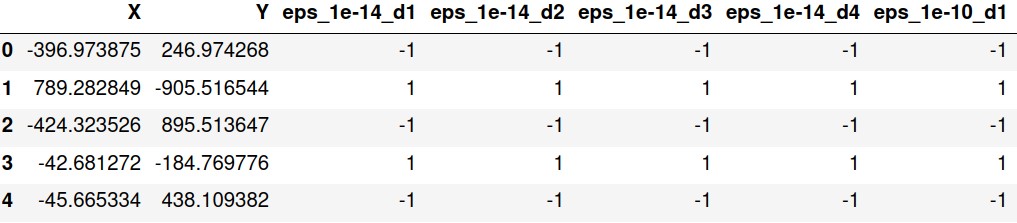
* det\_1 - funkcja obliczająca wyznacznik 3x3, zaimplementowana samodzielnie
* det\_2 - funkcja obliczająca wyznacznik 2x2, zaimplementowana samodzielnie
* det\_3 - funkcja obliczająca wyznacznik 3x3, zaimplementowana przy pomocy biblioteki numpy
* det\_4 - funkcja obliczająca wyznacznik 2x2, zaimplementowana przy pomocy biblioteki numpy

Kolejnym krokiem było sklasyfikowanie punktów z poszczególnych podpunktów ze względu na położenie w stosunku do prostej wyznaczonej przez wektor (a = [-1.0, 0.0], b = [1.0, 0.1]) w celu porównania wyników dla różnych wyznaczników. Do zrealizowania tego została zaimplementowana dodatkowa funkcja classify\_points która klasyfikuje punkty. Zwraca ona wyniki -1 (punkt leży po lewej stronie prostej), 0 (punkt leży na prostej) i 1 (punkt leży po prawej stronie prostej). Klasyfikację punktów powtórzono dla różnych wartości tolerancji błędów wynoszących: ε1 = 10-14, ε2 = 10-12, ε3 = 10-10, ε4 = 10-8. Została również stworzono funkcja assign\_to\_dataframe, która przypisuje do odpowiedniego epsilona oraz odpowiedniego wyznacznika liczbę punktów które zostały sklasyfikowane do -1, 0 lub 1. Na końcu również sprawdzono wyniki dla różnej precyzji obliczeń - float32.

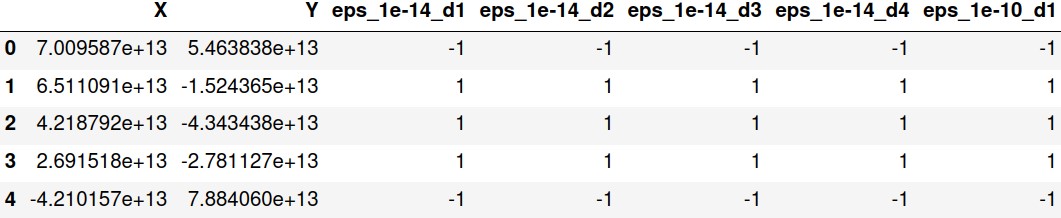
# Klasyfikacja poszczególnych podziałów punktów

Do wszystkich istniejących dataframe które odpowiadają każdemu podpunktowi dołączane są nowe kolumny w których znajdują się wartości (-1, 0, 1) dla różnych epsilonów i różnych wyznaczników.

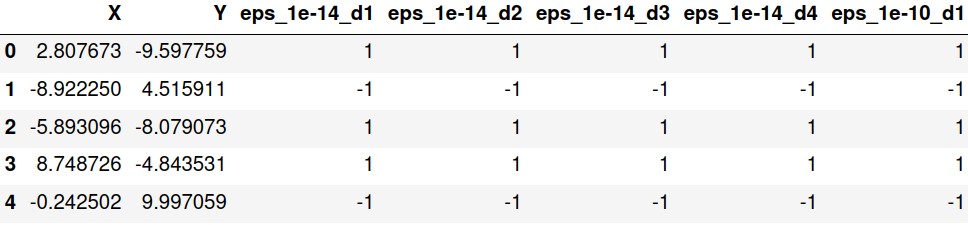
1. Tabela\_5



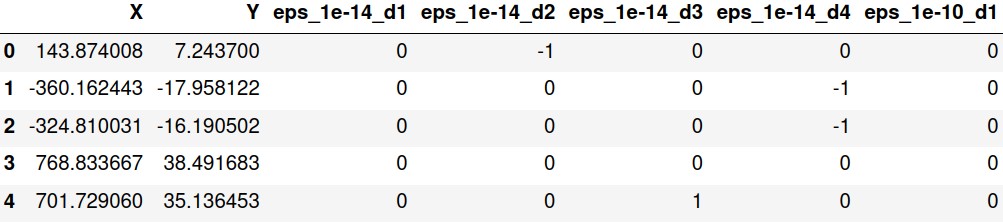
1. Tabela\_6



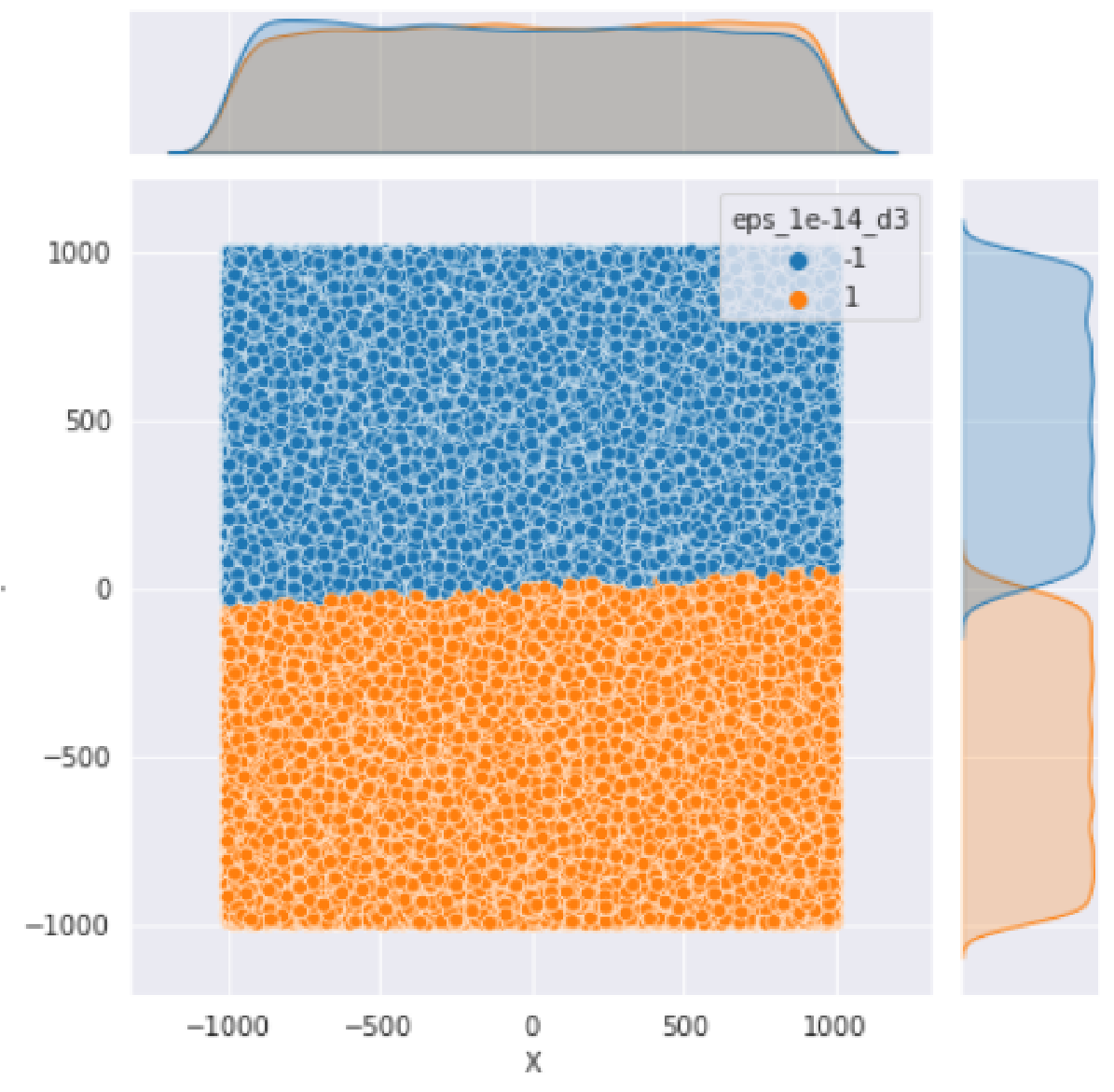
1. Tabela\_7

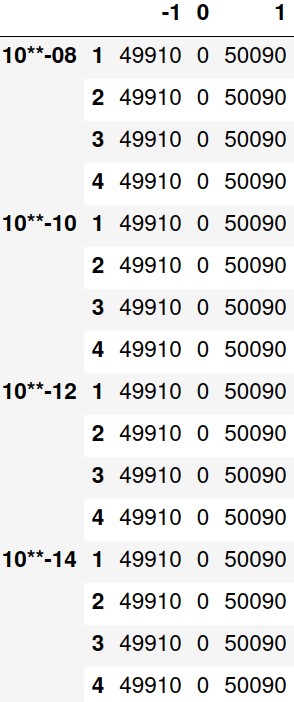
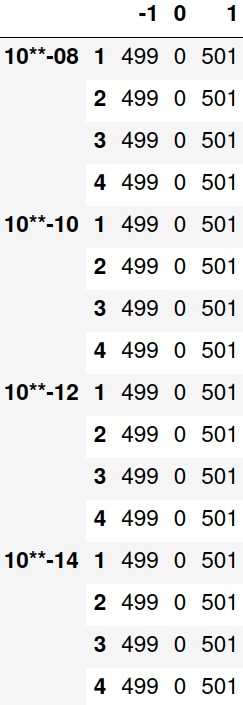
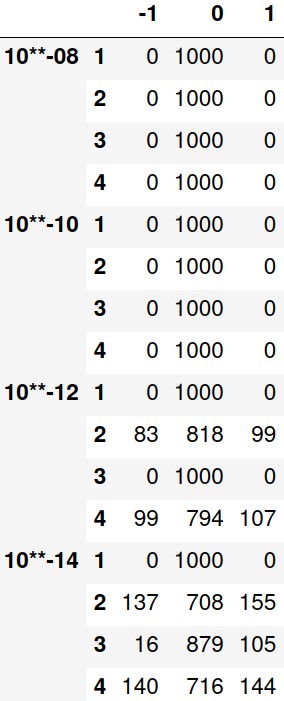
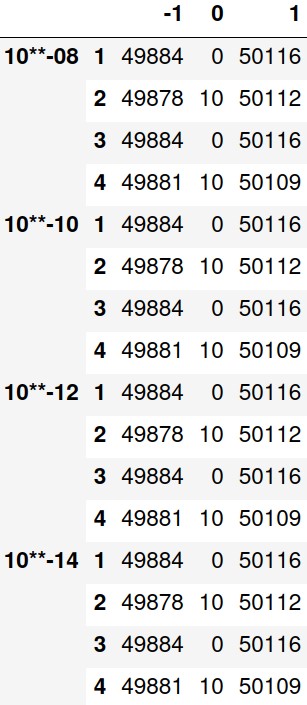


1. Tabela\_8



Powyższe dataframe to tylko fragment całości. Całe dataframe ze wszystkimi obliczonymi wartościami dla różnych epsilonów oraz różnych wyznaczników znajduje się w Jupyter Notebooku. Możemy dzięki tym dataframemom zauważyć, że w przykładzie Tabela\_5 i Tabela\_7 mamy takie same wartości dla różnych epsilonów oraz wyznaczników. W Tabela\_6 i Tabela\_8 możemy zauważyć, że są one różne w zależności od użytego epsilona oraz wyznacznika. Również zostały stworzone dataframe, które zawierają dane ile punktów zostało przydzielonych do odpowiednich wartości (-1, 0, 1) dla różnych wartości epsilon oraz różnych wyznaczników dla odpowiednich podpunktów. W tych dataframe zostało użyte MultiIndexing aby móc zagnieździć dla konkretnych epsilonów wszystkie wyznaczniki.

a) Tabela\_9 b) Tabela\_10 c) Tabela\_11 d) Tabela\_12



1, 2, 3, 4 odpowiada odpowiednio det\_1, det\_2, det\_3, det\_4.

# Wybrane wykresy klasyfikacji punktów dla różnychpodpunktów, epsilonów oraz wyznaczników

Kolory na wykresach są opisane w legendach które znajdują się obok wykresów albo pod nimi.

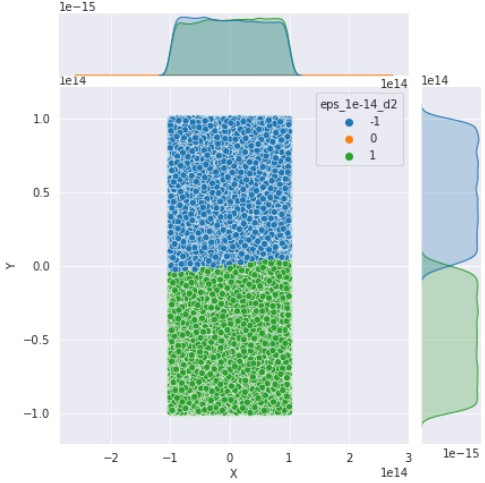
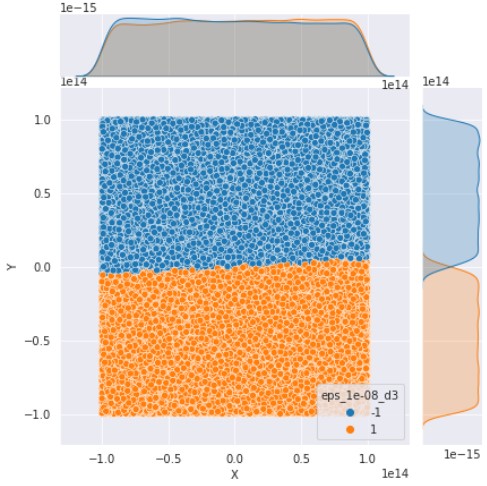
* klasyfikacja punktów z podpunktu a) (wszystkie wykresy są identyczne dla wszystkich ε i wyznaczników)

Wykres\_7

* klasyfikacja punktów z podpunktu b)

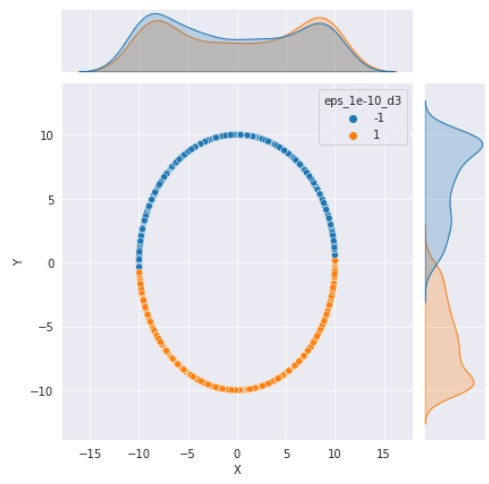
Wykres\_8 Wykres\_9

ε = 10-14 det\_2 ε = 10-8 det\_3



* klasyfikacja punktów z podpunktu c) (wszystkie wykresy są identyczne dla wszystkich ε i wyznaczników)

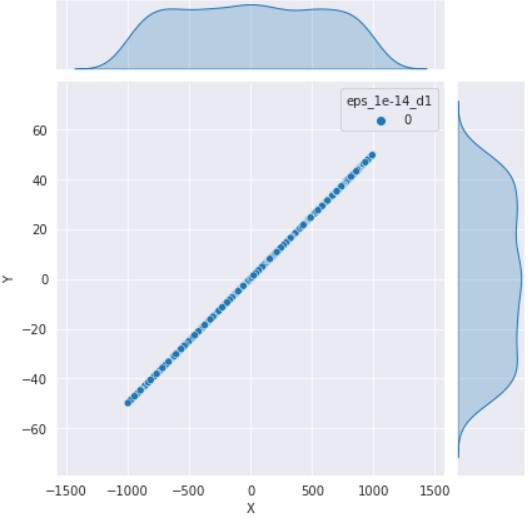
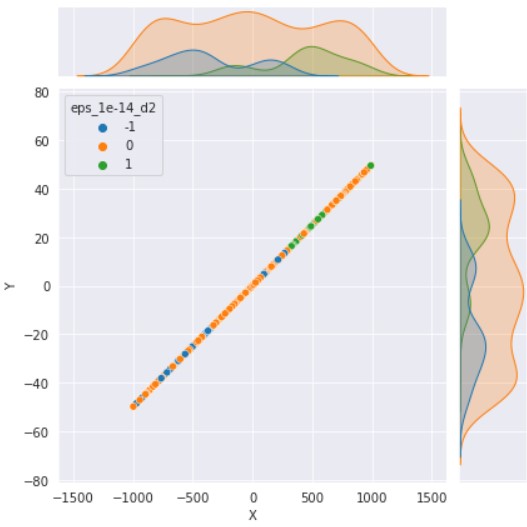
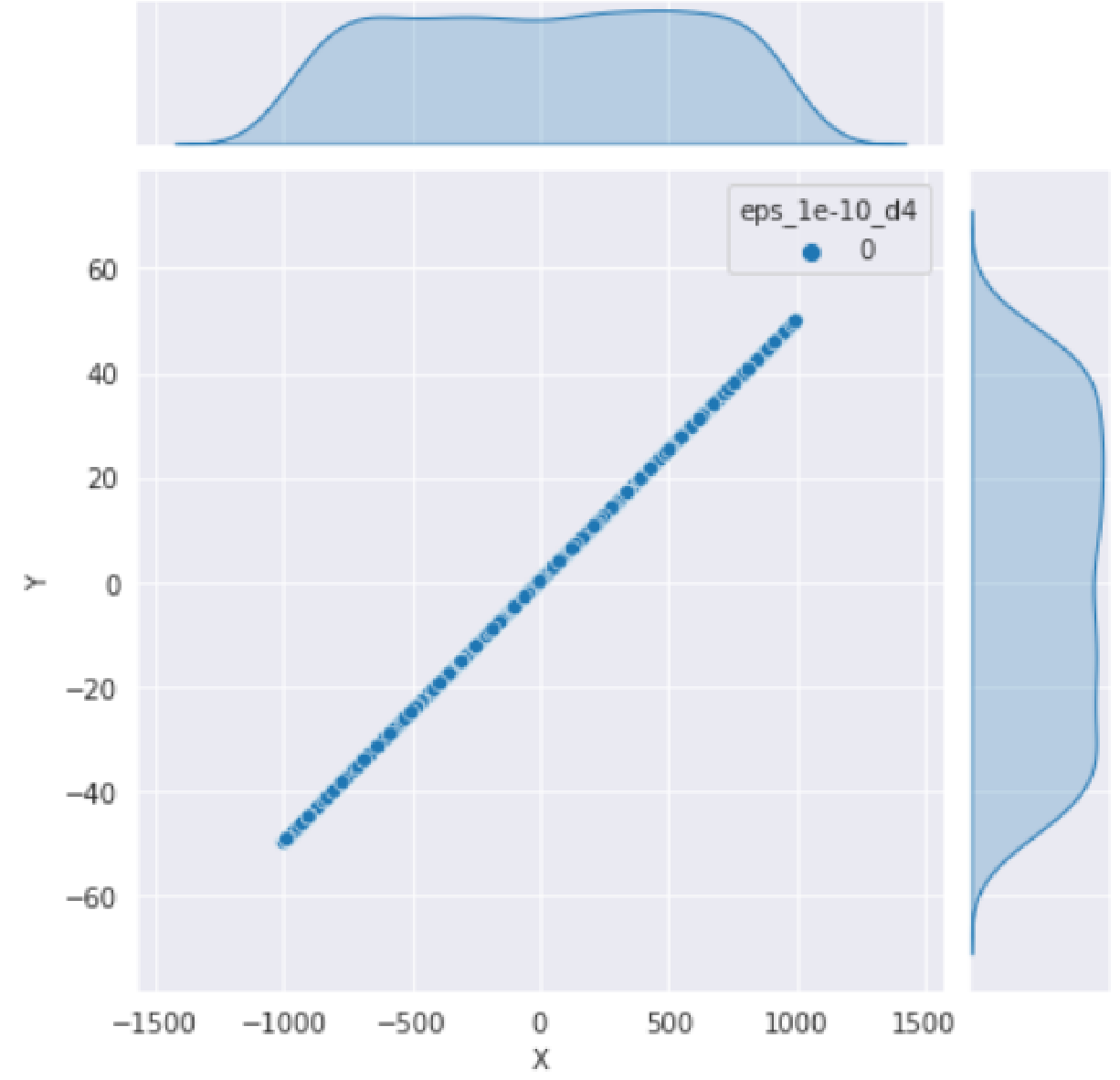
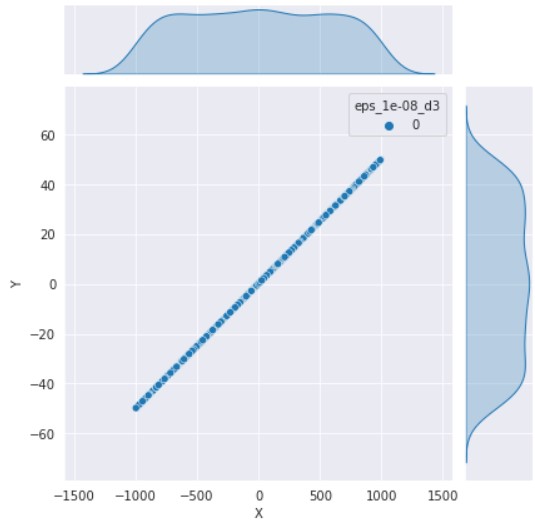
Wykres\_10



* klasyfikacja punktów z podpunktu c)

Wykres\_11 Wykres\_12

ε = 10-14 det\_1 ε = 10-14 det\_2

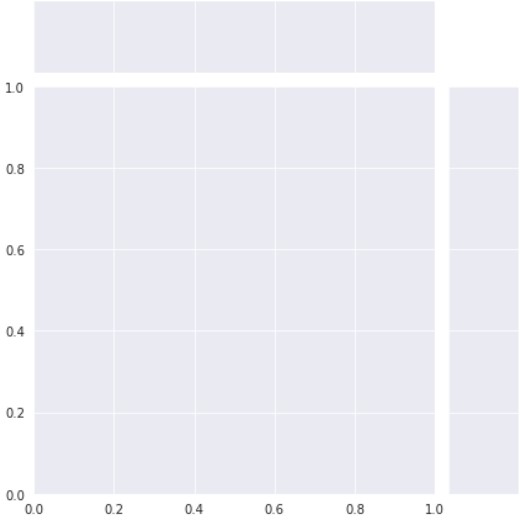


Wykres\_12 Wykres\_13

ε = 10-8 det\_3 ε = 10-10 det\_4 Wykresy dla punktów, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilona

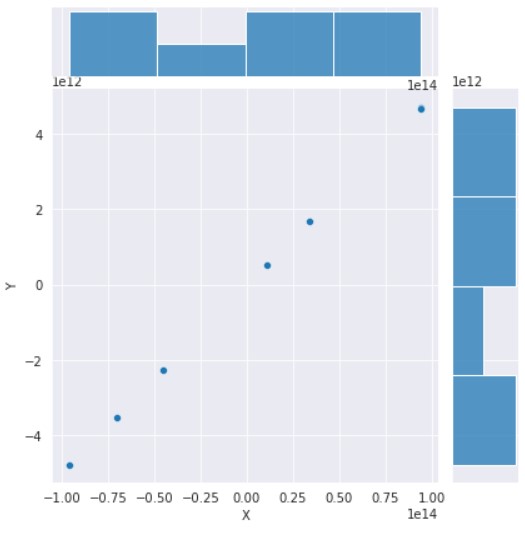
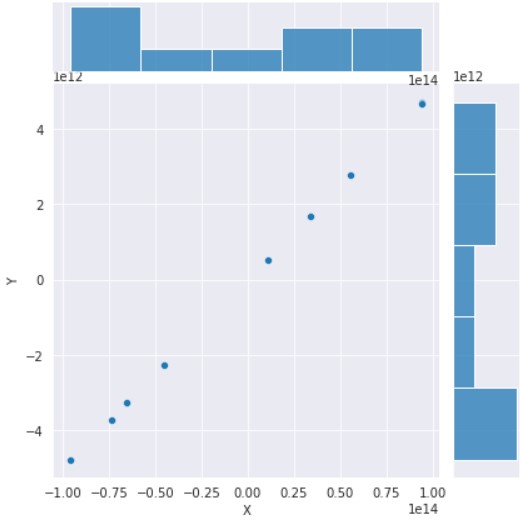
* W podpunkcie a) wyniki dla wszystkich 4 wyznaczników oraz dla różnych epsilonów są takie same, nie różnią się między sobą więc wykresy są puste. Wszystkie punkty zostały sklasyfikowane tak samo.

Wykres\_14



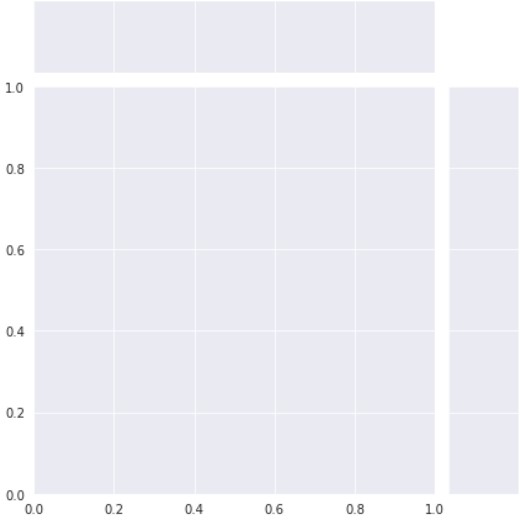
* W podpunkcie b) występują punkty które zostały inaczej zaklasyfikowane przez różne wyznaczniki.

Wykres\_15 Wykres\_16 ε = 10-14 det\_1 z ε = 10-10 det\_3 ε = 10-14 det\_3 z ε = 10-12 det\_1



* W podpunkcie c) wyniki dla wszystkich 4 wyznaczników oraz dla różnych epsilonów są takie same, nie różnią się między sobą więc wykresy są puste. Wszystkie punkty zostały sklasyfikowane tak samo.

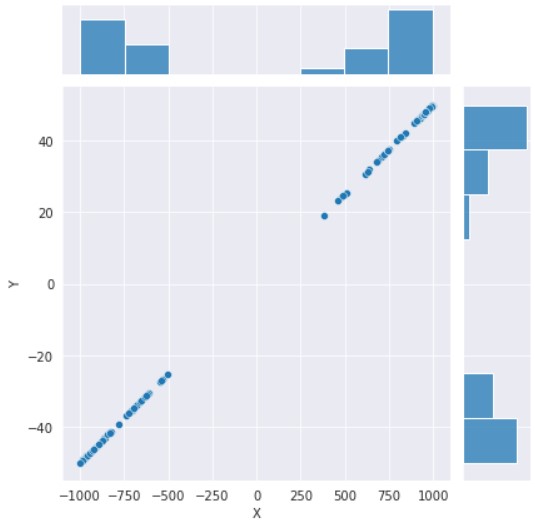
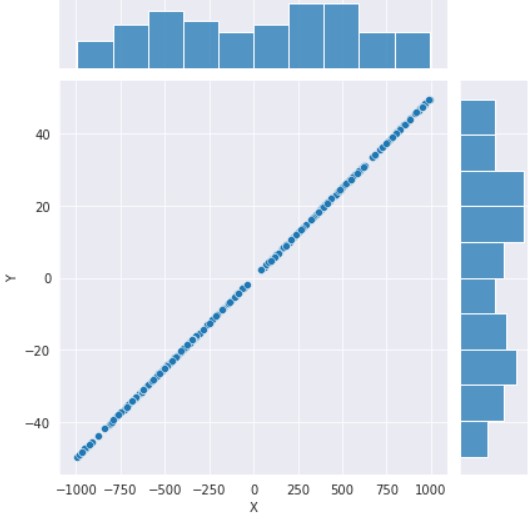
Wykres\_17



* W podpunkcie d) występują punkty które zostały inaczej zaklasyfikowane przez różne wyznaczniki.

Wykres\_18 Wykres\_19

dla ε = 10-14 det\_1 z ε = 10-14 det\_2 ε = 10-14 det\_3 z ε = 10-12 det\_4

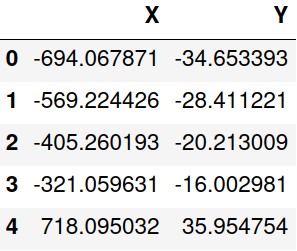
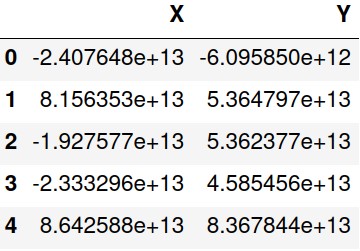


# Użycie innej precyzji (float32)

Wykresy dla podpunktów b) i d) po użyciu innej precyzji. Przedstawione są wykresy tylko dla podpunktów b) i d) ponieważ były to jedyne podpunkty w których występowała rozbieżność w wynikach dla różnych wyznaczników i epsilonów. Zostały stworzone nowe dataframe dla wartości po użyciu innej precyzji.

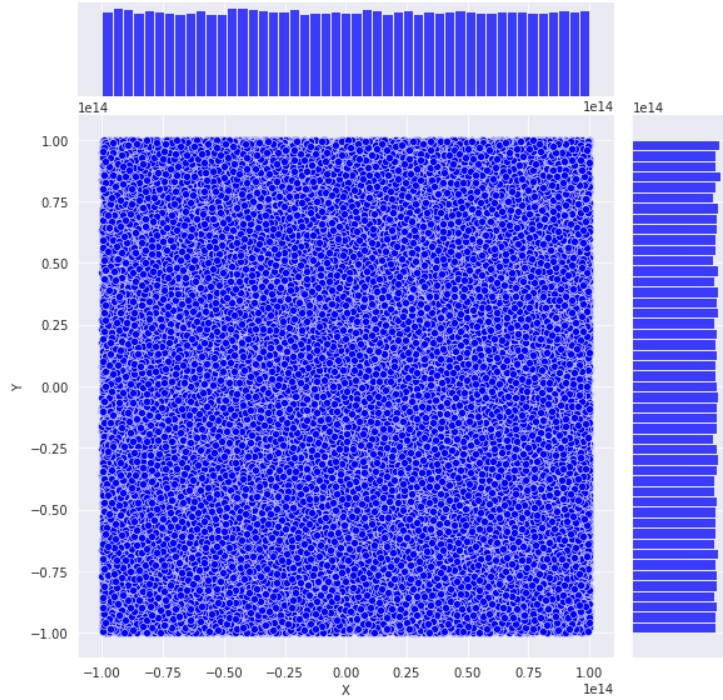
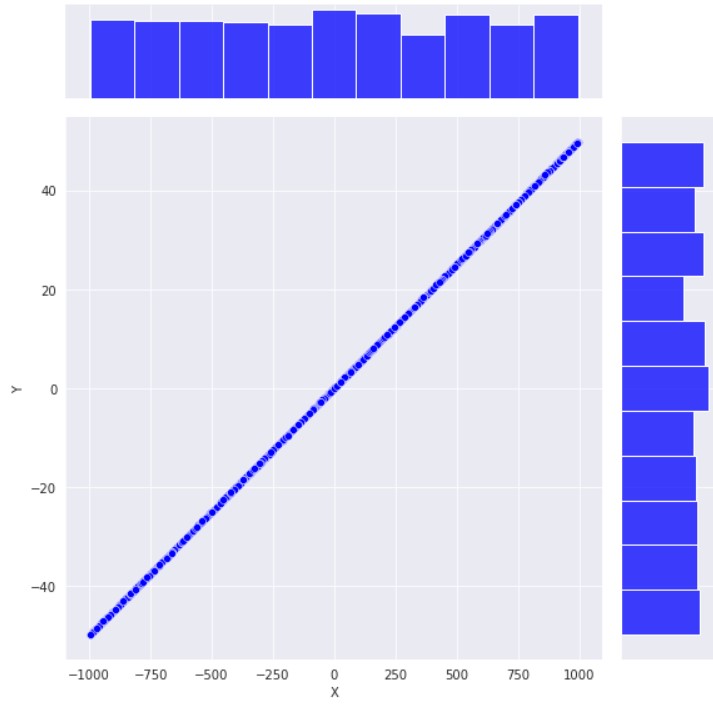
5 pierwszych wierszy z dataframe kolejno z podpunktu b) i d) po zmianie precyzji na float32

b) Tabela\_13 d) Tabela\_14



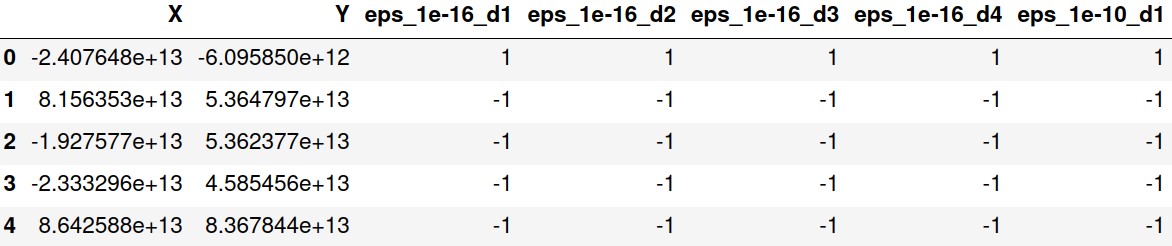
Wygenerowane wykresy rozkładu punktów dla podpunktów b) i d) po zmianie precyzji.

b) Wykres\_20 d) Wykres\_21

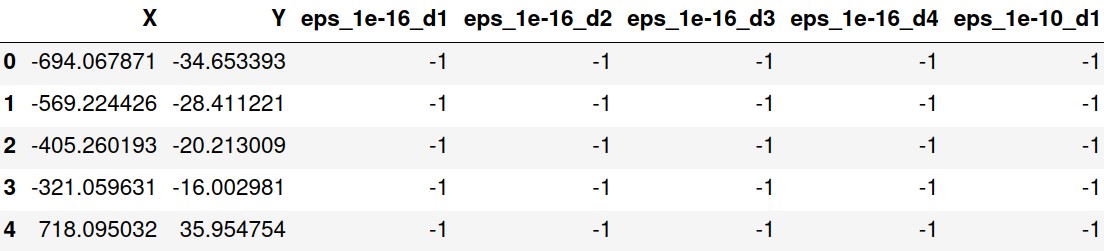


Do nowych dataframe, które odpowiadają podpunktom b) i d) po zmianie ich wartości przez użycie float32 dołączane są nowe kolumny w których znajdują się wartości (-1, 0, 1) dla różnych epsilonów i różnych wyznaczników. Użyte epsilony to ε1 = 10-16 i ε2 = 10-10.

b) Tabela\_15



d) Tabela\_16

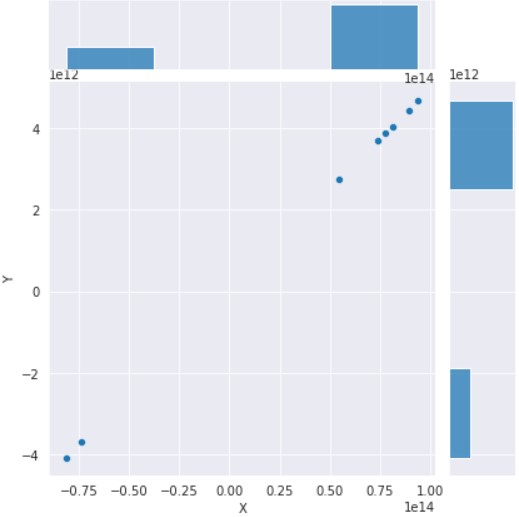
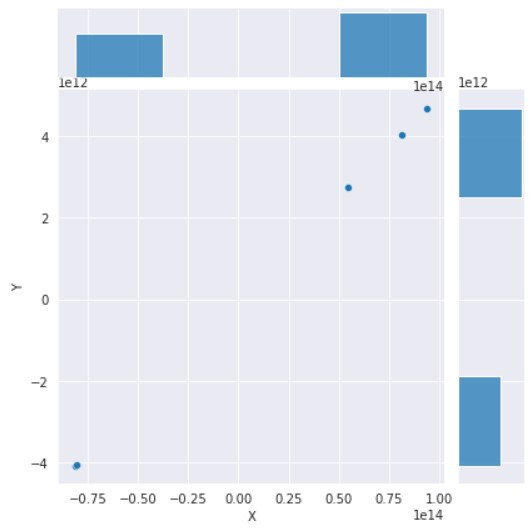


Wybrane wykresy dla punktów o zmienionej precyzji, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilona.

* dla podpunktu b)

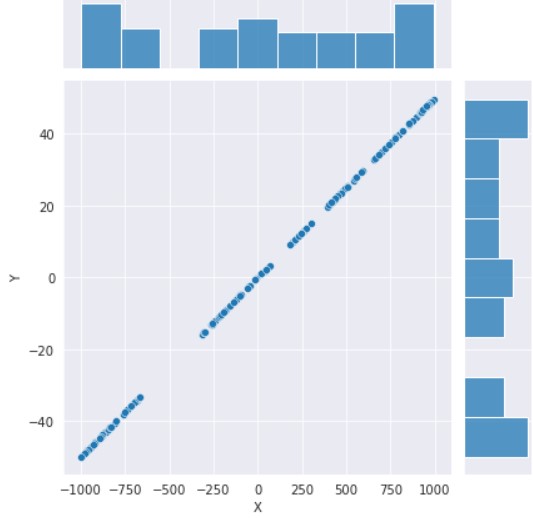
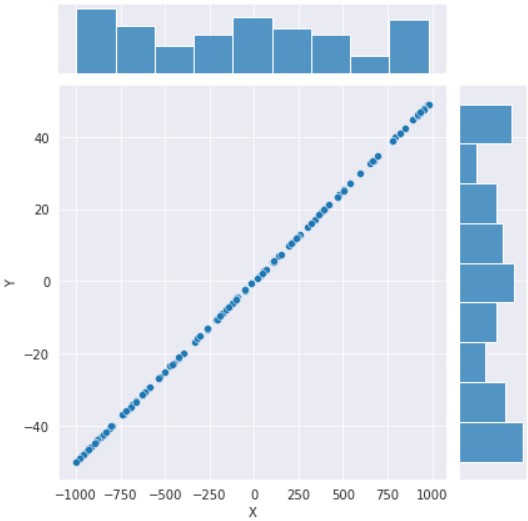
Wykres\_22 Wykres\_23

ε = 10-16 det\_3 z ε = 10-12 det\_1 ε = 10-16 det\_1 z ε = 10-8 det\_4



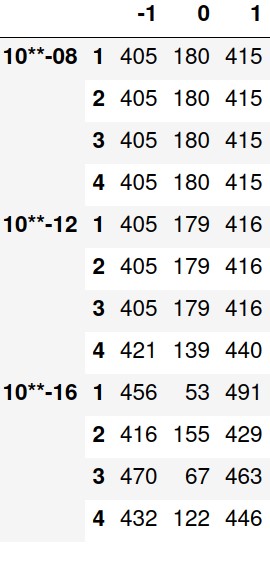
* dla podpunktu d)

Wykres\_24 Wykres\_25 dla ε = 10-16 det\_1 z ε = 10-12 det\_1 ε = 10-16 det\_2 z ε = 10-16 det\_3



Wykresy dla punktów, które zostały sklasyfikowane inaczej dla różnych sposobów liczenia wyznacznika oraz innego epsilona oraz innej precyzji.

b) Tabela\_17 d) Tabela\_18



# Wnioski

Klasyfikacja punktów z podpunktów a) oraz c) dla większości przypadków dawała identyczne wyniki niezależnie od użytego wyznacznika czy epsilona (tolerancji). Spowodowane jest to tym, że istnieje bardzo małe prawdopodobieństwo, iż punkt będzie znajdował się dokładnie na prostej, z dokładnością do użytej tolerancji. W podpunkcie

b) wybór epsilona nie miał znaczenia jednak wybór wyznacznika już tak. Dla wszystkich przypadków dla det\_2 wynikiem ilości punktów leżących na prostej jest 3, natomiast dla det\_4 jest to 7. Jest to spowodowane najprawdopodobniej tym, że współrzędne tych punktów mają duże wartości bezwzględne. Operacja klasyfikacji punktów wymaga jednak dużej precyzji, która jest tracona z powodu dużych wartości które są generowane. Mantysa liczby zmiennoprzecinkowej jest skończona a dla dużej liczby będzie przechowywana informacja o pewnej liczbie cyfr, przez co pomijane są cyfry znajdujące się na odległych miejscach dziesiętnych, które są ważne w tym ćwiczeniu. W dataframeach możemy zauważyć iż najlepsze wyniki otrzymuje się dla wartości ε >= 10-14.

Na wykresach, które ilustrują różnice w klasyfikacji przy użyciu różnych wyznaczników oraz epsilonów można zaobserwować, że ilość różnic zwiększa się wraz ze zwiększeniem wartości bezwzględnej współrzędnych punktów. Możemy dojść do wniosku, że niezależnie od tolerancji wszystkie wyznaczniki działają najlepiej dla punktów o bezwzględnie mniejszych współrzędnych. Można to zauważyć na niektórych wykresach liniowych.

Wyniki otrzymane przy użyciu innej precyzji (zamianie liczb na float32) są zdecydowanie gorsze od tych uzyskanych dla danych o precyzji float64. Mimo tego można jednak zauważyć, iż wyznaczniki biblioteczne sprawdzały się gorzej od tych zaimplementowanych samodzielnie. Najlepsze wyniki otrzymywane były dla wyznacznika det\_1 dla ε >= 10-14 oraz na ogół najgorsze wyniki dawał det\_4 (wyznacznik 2x2 przy pomocy numpy).