

Algorytmy geometryczne, sprawozdanie z ćwiczenia 1

1. Środowisko, biblioteki oraz dane techniczne urządzenia

Ćwiczenie zostało zrealizowane w Jupyter Notebooku i napisane w języku Python, z wykorzystaniem bibliotek numpy (umożliwiająca zaawansowane operacje numeryczne), pandas (do przedstawiania wyników poszczególnych obliczeń w DataFrame). Dodatkowo do rysowania wykresów (Visualizer) i sprawdzenia poprawności niektórych funkcji wykorzystano projekt opracowany przez studentów Koła Naukowego BIT. Wszystko to było wykonane na laptopie z systemem operacyjnym Windows 11, procesorem AMD Ryzen 5 5600H 3.30 GHz oraz pamięcią RAM 32 GB.

2. Opis realizacji ćwiczenia:

Celem ćwiczenia było określenie po której stronie znajdują się punkty względem odcinka w zależności od wybranej metody obliczania wyznacznika oraz tolerancji dla zera.

2.1. Generacja zbiorów punktów

Żeby wykonać ćwiczenie na początku wygenerowane zostały 4 zbiory punktów (2D, współrzędne typu double):

- Zestaw 1

10000 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$

- Zestaw 2

10000 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$

- Zestaw 3

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

- Zestaw 4

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

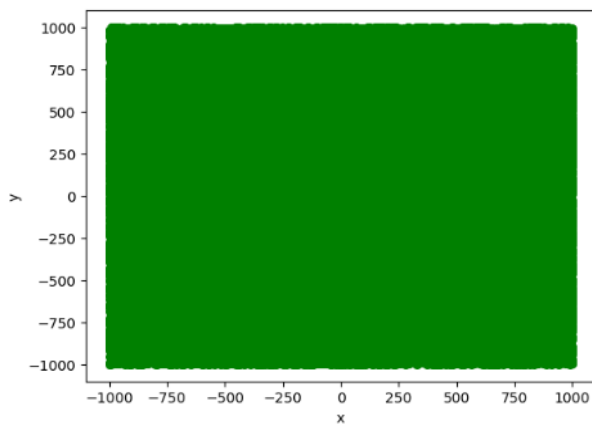
Współrzędne punktów zostały wygenerowane za pomocą metody `random.uniform` z biblioteki `numpy`. Funkcja ta generuje losowe punkty typu `double` w podanym przedziale obustronnie domkniętym.

Punkty z zestawu 3 zostały wygenerowane za pomocą funkcji trygonometrycznych z biblioteki `numpy`.

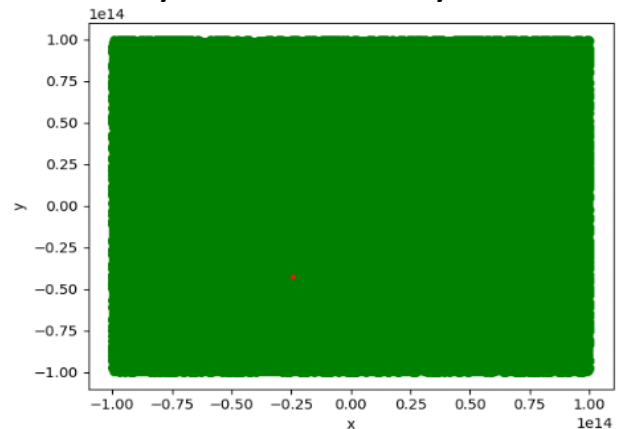
Punkty z zestawu 4 zostały wygenerowane tak by znajdowały się na prostej przechodzącej 2 podane punkty a i b .

Poprawność wykonanych zbiorów została sprawdzona za pomocą testów dostarczonych z projektu Koła Naukowego BIT.

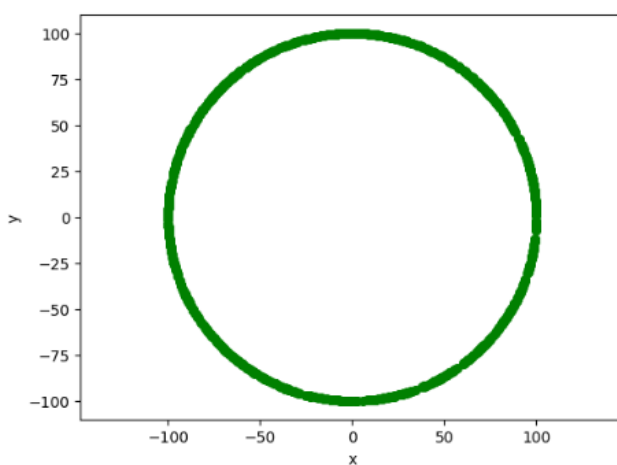
Wykres 1.1 Zestaw danych 1



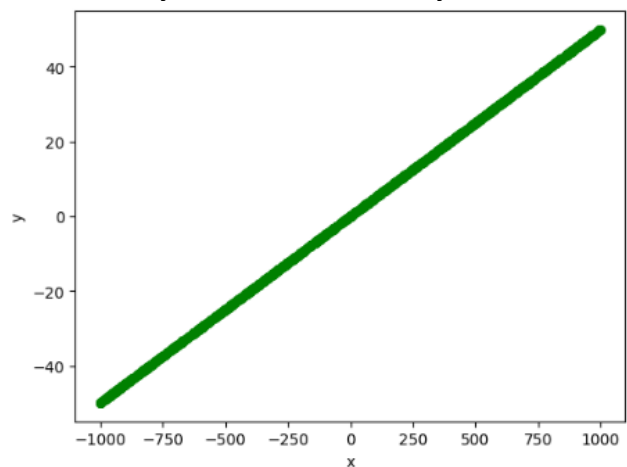
Wykres 1.2 Zestaw danych 2



Wykres 1.3 Zestaw danych 3



Wykres 1.4 Zestaw danych 4



2.2. Metody obliczania wyznacznika i tolerancja dla zera

Następnym krokiem było określenie położenia punktów względem prostej w zależności od wybranej metody obliczania wyznacznika i tolerancji dla zera przy określaniu położenia na podstawie wyznacznika.

Do tego zaimplementowane zostały cztery funkcje obliczające wyznaczniki:

- `mat_det_2x2` – funkcja obliczająca wyznacznik 2x2, zaimplementowany samodzielnie
- `mat_det_2x2_lib` - funkcja obliczająca wyznacznik 2x2, zaimplementowany przy pomocy biblioteki numpy
- `mat_det_3x3` – funkcja obliczająca wyznacznik 3x3, zaimplementowany samodzielnie
- `mat_det_3x3_lib` - funkcja obliczająca wyznacznik 3x3, zaimplementowany przy pomocy biblioteki numpy

Tutaj także wykorzystałem dostarczone testy w celu weryfikacji poprawności zaimplementowanych funkcji `mat_det_2x2`, `mat_det_3x3`.

Natomiast rozpatrywanymi tolerancjami dla zera przyjęto:

- 0
- 10^{-8}
- 10^{-10}
- 10^{-12}
- 10^{-14}

2.3. Klasyfikacja punktów

Do sklasyfikowania punktów wykorzystałem funkcję `categorize_points()`, która klasyfikuje punkty względem prostej przechodzącej przez punkty *a* i *b*, dla podanego wyznacznika i tolerancji dla zera. Ta funkcja zwraca trzy tablice punktów: *L* (punkty leżące po lewej stronie prostej), *M* (punkty leżące na prostej), *R* (punkty leżące po prawej stronie prostej). Za pomocą funkcji `quantity()` tworzę tablice, która dla każdej kombinacji wyznacznika i tolerancji dla zera przypisuje skategoryzowane punkty. Dzięki zastosowaniu tej funkcji nie będzie potrzebna ponowna kategoryzacja punktów.

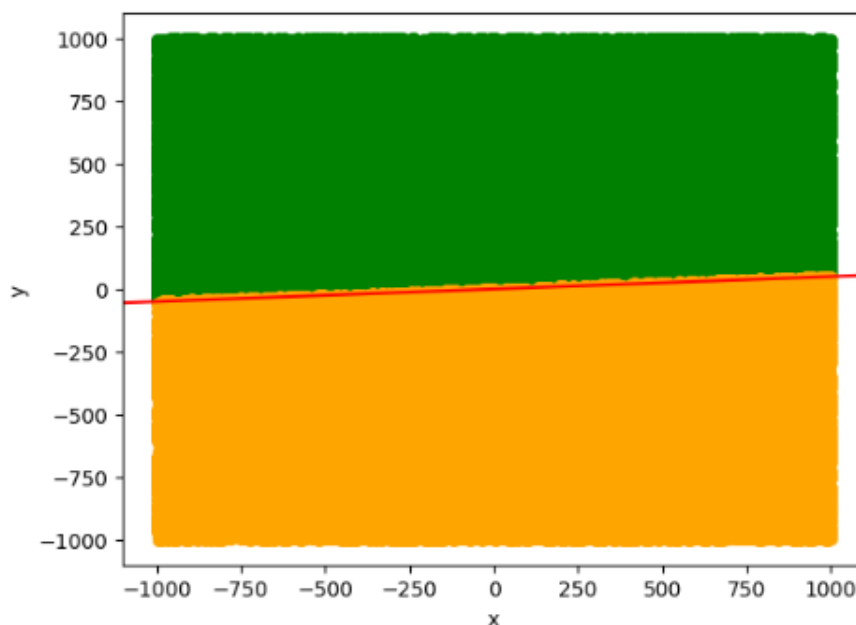
2.4. Wizualizacja wybranych wyników

W tej części ćwiczenia skupiłem się na zwizualizowaniu sklasyfikowanych punktów wszystkich wyznaczników o tolerancji dla zera równej 10^{-14} . Dzięki temu mogłem uzyskać wystarczającą ilość informacji potrzebną do zaobserwowania różnic w klasyfikacji w zależności od wyznacznika.

Kolor zielony oznacza, że punkty zostały sklasyfikowane jako leżące po lewej stronie odcinka, pomarańczowy po prawej stronie odcinka, a fioletowy leżący na odcinku.

Dla Zestawu 1 wyniki były takie same dla wszystkich wyznaczników.

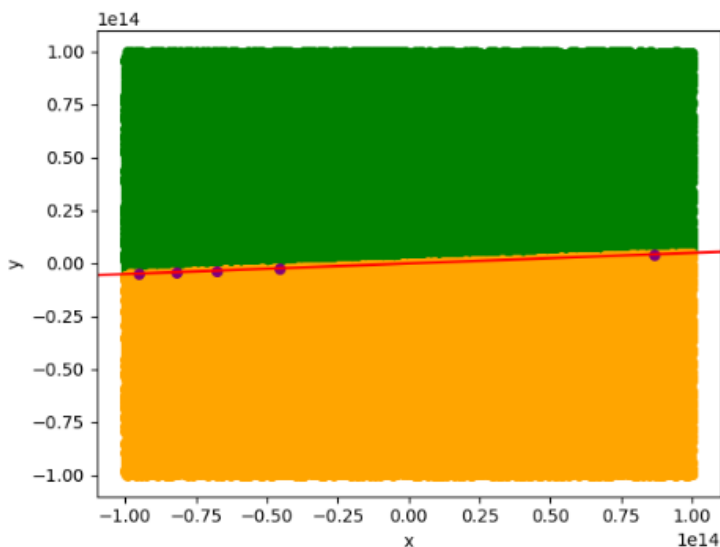
Wykres 2.1 Klasyfikacja punktów dla zestawu 1. dla wszystkich wyznaczników przy tolerancji 10^{-14}



Klasyfikacja: Po lewej: 50014 Na prostej: 0 Po prawej: 49986

Dla Zestawu 2 można zauważyć niewielkie różnice w wynikach. Klasyfikacja punktów między wyznacznikami 3x3 nie różniły się między sobą i żadne punkty nie leżały na prostej. Dla wyznaczników 2x2 pojawia się kilka punktów na prostej. Co ciekawe dla wyznacznika `mat_det_2x2` wystąpiła inna ilość punktów na prostej niż dla wyznacznika `mat_det_2x2_lib`. Cechą wspólną tych punktów, było to, że współrzędne były bardzo dużymi liczbami rzędu 10^{14} . Przykładowy punkt dla obu tych wyznaczników miał współrzędne: $(-67557822541807.47, -3370083784530.625)$. Rozmiar tych liczb i błędy obliczeniowe związane z precyzją operacji na tych liczbach mogły spowodować inną klasyfikację między wyznacznikami

Wykres 2.2.1 Klasyfikacja punktów dla zestawu 2. dla wyznacznika `mat_det_2x2` przy tolerancji 10^{-14}



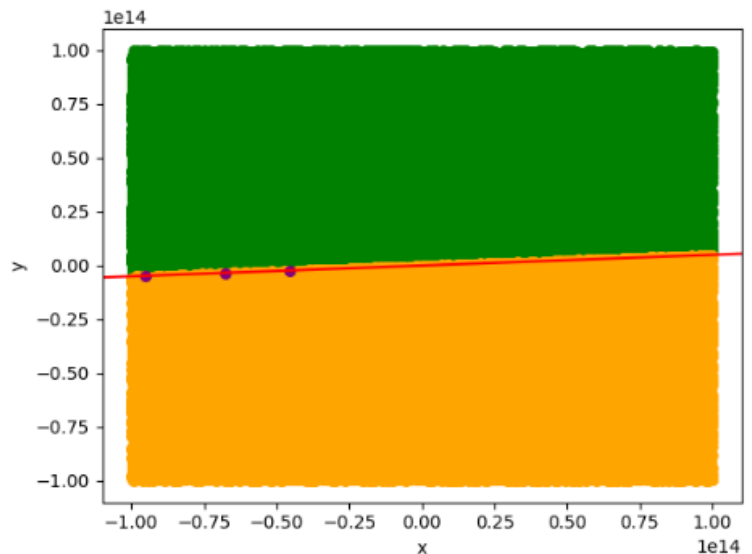
Klasyfikacja:

Po lewej: 49774

Na prostej: 5

Po prawej: 50221

Wykres 2.2.2 Klasyfikacja punktów dla zestawu 1. dla wyznacznika `mat_det_2x2_lib` przy tolerancji 10^{-14}



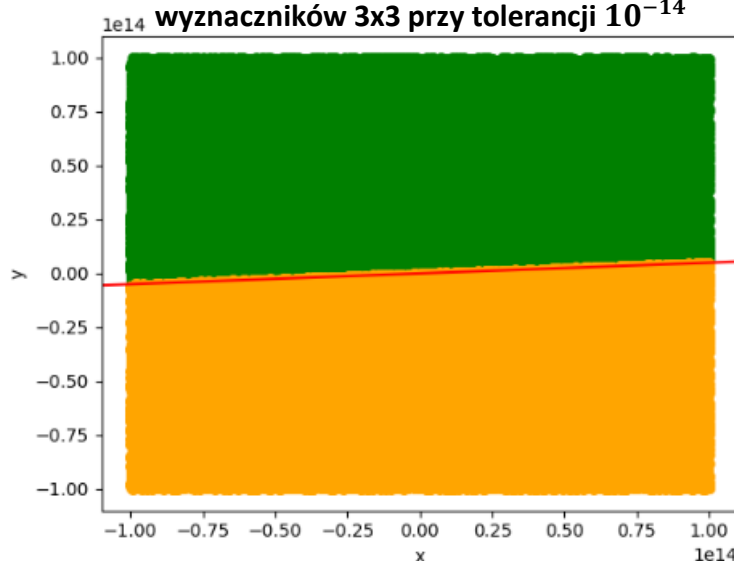
Klasyfikacja:

Po lewej: 49775

Na prostej: 3

Po prawej: 50222

Wykres 2.2.3 Klasyfikacja punktów dla zestawu 2. dla obu wyznaczników 3x3 przy tolerancji 10^{-14}



Klasyfikacja:

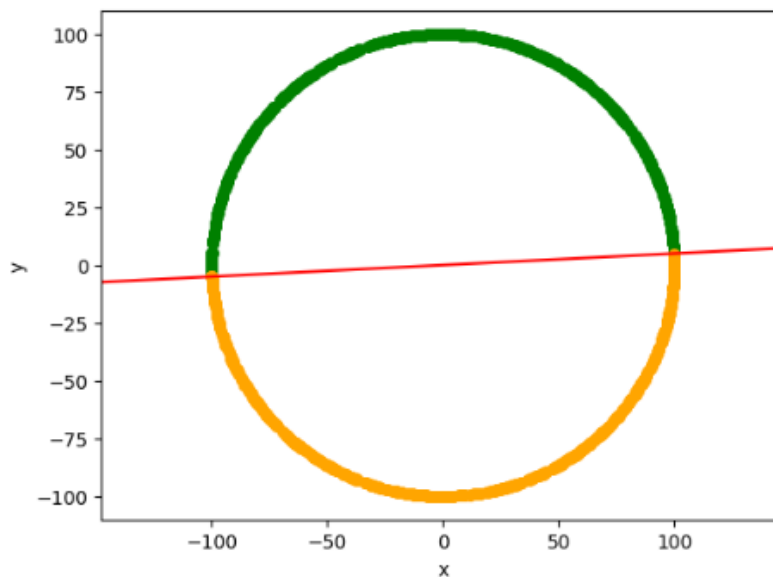
Po lewej: 49777

Na prostej: 0

Po prawej: 50223

Dla Zestawu 3 wyniki również były takie same dla wszystkich wyznaczników.

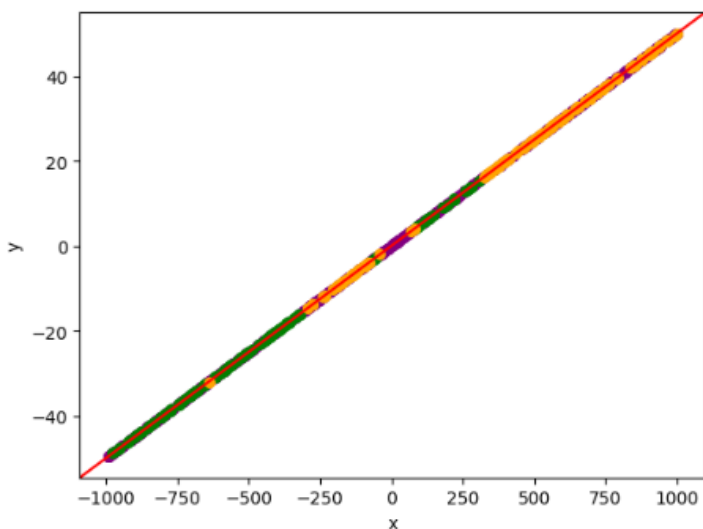
Wykres 2.3 Klasyfikacja punktów dla zestawu 3. dla wszystkich wyznaczników przy tolerancji 10^{-14}



Klasyfikacja: Po lewej: 521 Na prostej: 0 Po prawej: 479

Dla Zestawu 4 można zauważyć dosyć duże różnice podczas wstępnej wizualizacji. W teorii wszystkie punkty powinny leżeć na prostej. Jednak w praktyce tak się nie stało. Powodu błędnej klasyfikacji można się dopatrywać w błędach związanych ze skończoną precyzją obliczeń. W celu usunięcia tego błędu w dalszej części ćwiczenia zostaną przeanalizowane różne tolerancje dla zera. Jednak wstępnie można zauważyć, że wyznacznik 3x3 własnej implementacji był najbardziej dokładny.

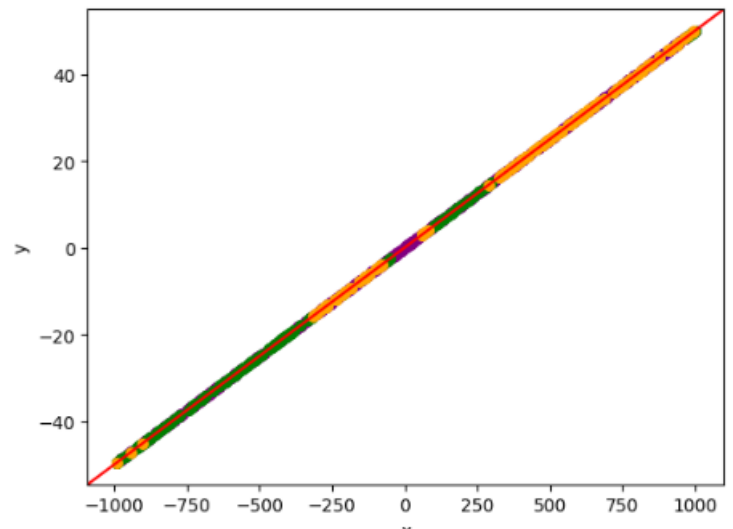
Wykres 2.4.1 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_2x2` przy tolerancji 10^{-14}



Klasyfikacja:

Po lewej: 138 Na prostej: 711 Po prawej: 151

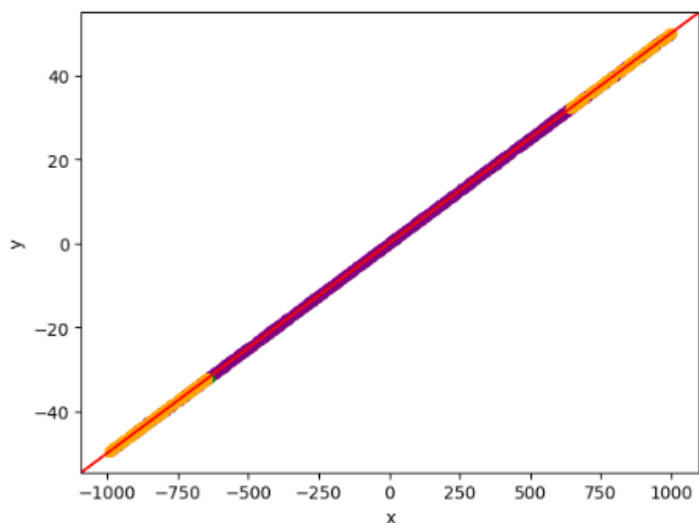
Wykres 2.4.2 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_2x2_lib` przy tolerancji 10^{-14}



Klasyfikacja:

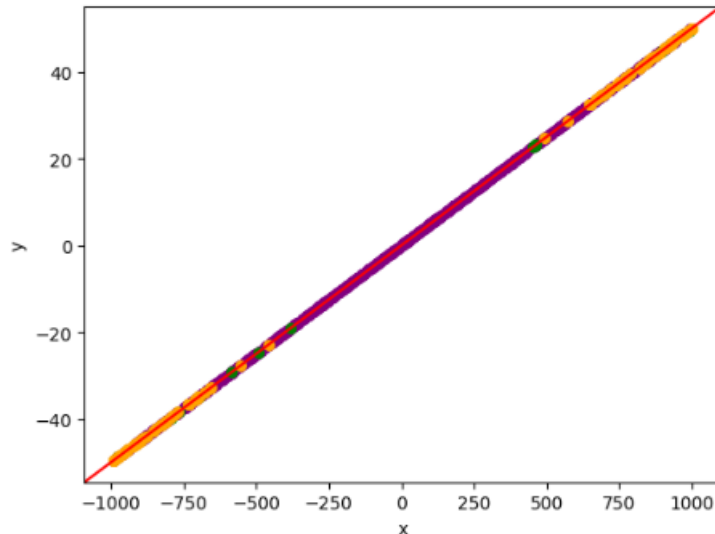
Po lewej: 155 Na prostej: 702 Po prawej: 143

Wykres 2.4.3 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_3x3` przy tolerancji 10^{-14}



Klasyfikacja:
Po lewej: 1 Na prostej: 787 Po prawej: 212

Wykres 2.4.4 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_3x3_lib` przy tolerancji 10^{-14}



Klasyfikacja:
Po lewej: 15 Na prostej: 879 Po prawej: 106

2.5. Wizualizacja wybranych wyników z użyciem innej precyzji (float32)

Następnej części ćwiczenia użyłem innej precyzji w celu sprawdzenia w jakim stopniu zmniejszenie precyzji wpłynie na precyzję obliczeń i klasyfikację wyników. Do tego celu należy zmniejszyć typ danych z float64 (domyślny typ floata w pythonie) na float32. Różnią się one tym, że float32 jest zapisywane na mniejszej ilości bitów, co przekłada się na mniejszą ilość cyfr po przecinku.

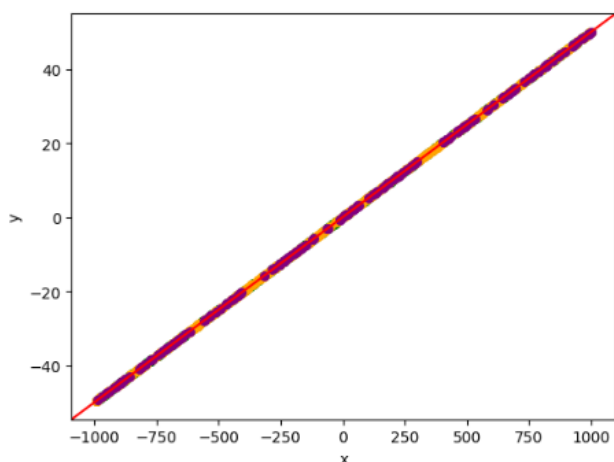
W tym celu do istniejących zbiorów wykorzystałem funkcję `numpy.float32`.

W zestawach nr 1 i 3 w precyzji float32 nie zaobserwowano żadnych zmian. Wykresy są identyczne jak wykresy: Wykres 2.1 i 2.3.

W Zestawie 2 zaobserwowano nieznaczne różnice dla zbiorów wyznaczników 2x2, gdzie ok. 1-2 punkty zostały sklasyfikowane inaczej.

Dopiero w Zestawie 4 można dostrzec znaczące różnice i wpływ mniejszej precyzji float 32. Na podstawie klasyfikacji i wykresów poniżej zaobserwować można, że wyniki są zdecydowanie gorsze. Błąd jest tak duży, że przykładowo dla Wykresu 3.2 tylko 13,2% punktów leży na prostej, gdzie w porównaniu z tym samym Wykresem 2.4.2 dla float64 70,2 % punktów leżało na prostej. W tym celu jedynie te wykresy przedstawiłem poniżej, bo z nich można wyciągnąć jakiekolwiek wnioski.

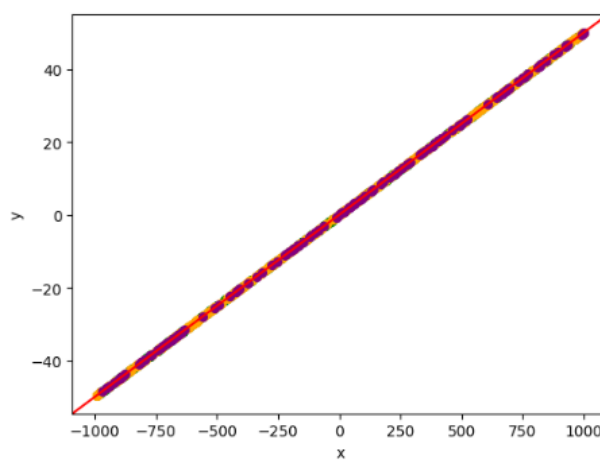
Wykres 3.1 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_2x2` przy tolerancji 10^{-14} o precyzji `float32`



Klasyfikacja:

Po lewej: 382 Na prostej: 163 Po prawej: 455

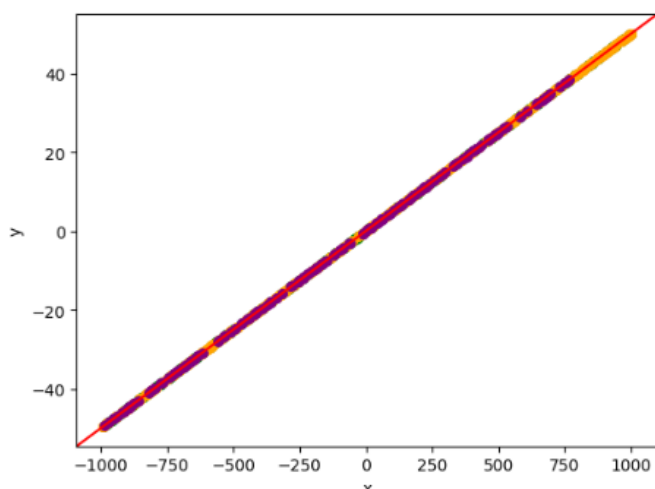
Wykres 3.2 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_2x2_lib` przy tolerancji 10^{-14} o precyzji `float32`



Klasyfikacja:

Po lewej: 401 Na prostej: 132 Po prawej: 467

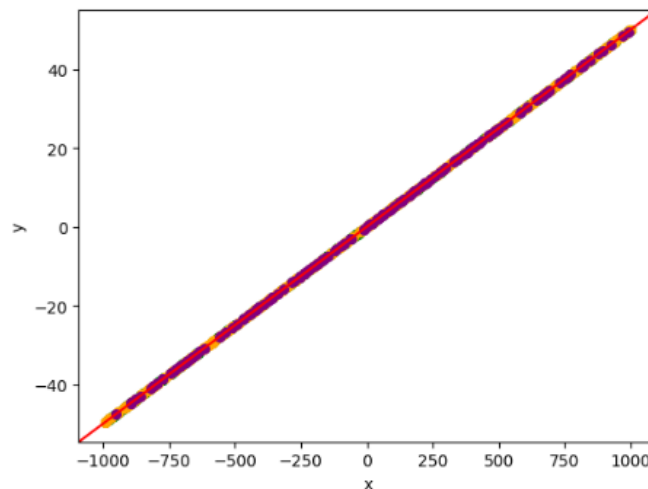
Wykres 3.3 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_3x3` przy tolerancji 10^{-14} o precyzji `float32`



Klasyfikacja:

Po lewej: 375 Na prostej: 163 Po prawej: 462

Wykres 3.4 Klasyfikacja punktów dla zestawu 4. dla wyznacznika `mat_det_3x3_lib` przy tolerancji 10^{-14} o precyzji `float32`



Klasyfikacja:

Po lewej: 387 Na prostej: 158 Po prawej: 455

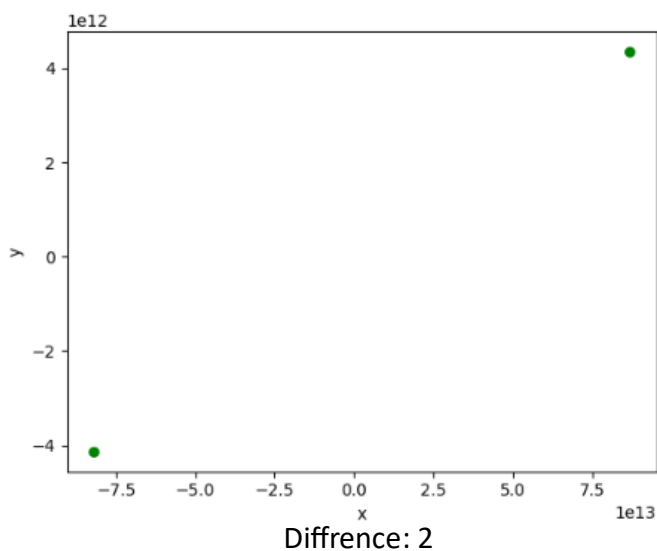
2.6. Porównanie punktów sklasyfikowanych inaczej w zależności od sposobu liczenia wyznacznik

Kolejną część ćwiczenia poświęciłem na różnicach w klasyfikacji między metodami obliczania wyznaczników. Najpierw wykorzystałem zaimplementowaną funkcję `Difference()`, która dla danego zestawu punktów tworzyła listę punktów, które różniły się klasyfikacją w obu zestawach. Na podstawie tej listy punktów funkcja wypisuje liczbę różnic oraz tworzy wykres z tychże punktów.

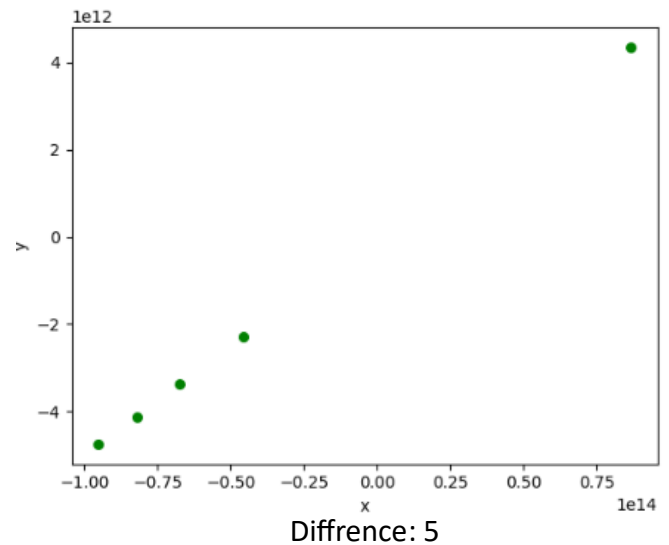
Jak podano w poprzednich podpunktach, dla Zestawu 1 i 3 nie zaobserwowano żadnych różnic.

Dla Zestawu 2 znaleziono 2 różnice porównując wyznacznik 2x2 własnej implementacji z wyznacznikiem biblioteki numpy oraz 5 różnic porównując wyznacznik 2x2 własnej implementacji z wyznacznikiem 3x3 własnej implementacji. Punkty te zgodnie z wcześniej przedstawionymi przypuszczeniami są to punkty dla których współrzędne są dużymi liczbami powodującymi błędy w precyzji obliczeń, co doprowadza do błędnej klasyfikacji.

Wykres 4.1.1 Punkty skategoryzowane inaczej w zestawie 2. przez wyznacznika mat_det_2x2 oraz wyznacznika mat_det_2x2_lib przy tolerancji 10^{-14}

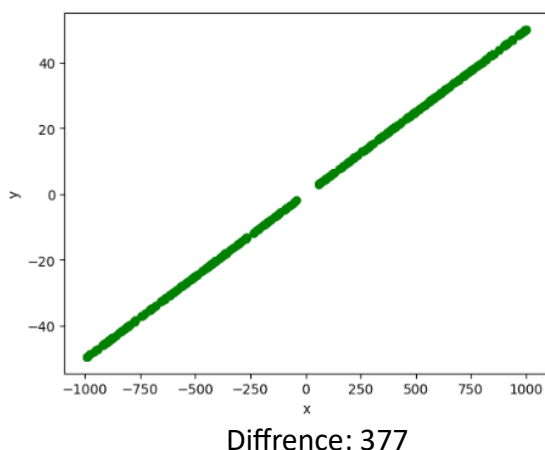


Wykres 4.1.2 Punkty skategoryzowane inaczej w zestawie 2. przez wyznacznika mat_det_2x2 oraz wyznacznika mat_det_3x3 przy tolerancji 10^{-14}

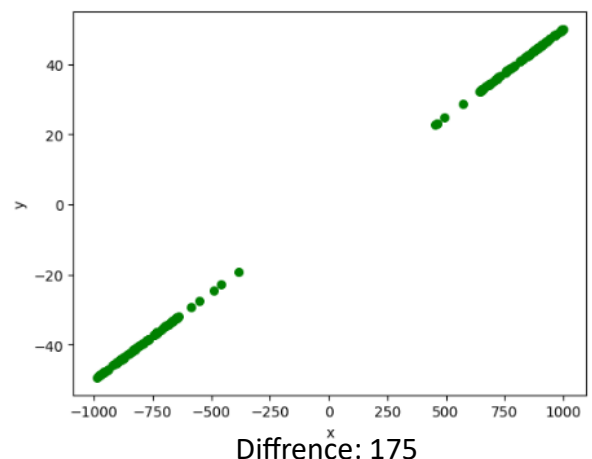


Dla Zestawu 4 różnice w klasyfikacji punktów między wyznacznikami są bardzo znaczące i sięgają nawet do 43,4%. Wynika to z typu dobranych punktów. Ze względu na to, że punkty znajdują się blisko prostej, to istnieje duże prawdopodobieństwo minimalnych błędów obliczeniowych, które mogą wpłynąć na klasyfikacje punktów, ponieważ wartości wyznaczników będą bardzo małymi liczbami bliskimi zera.

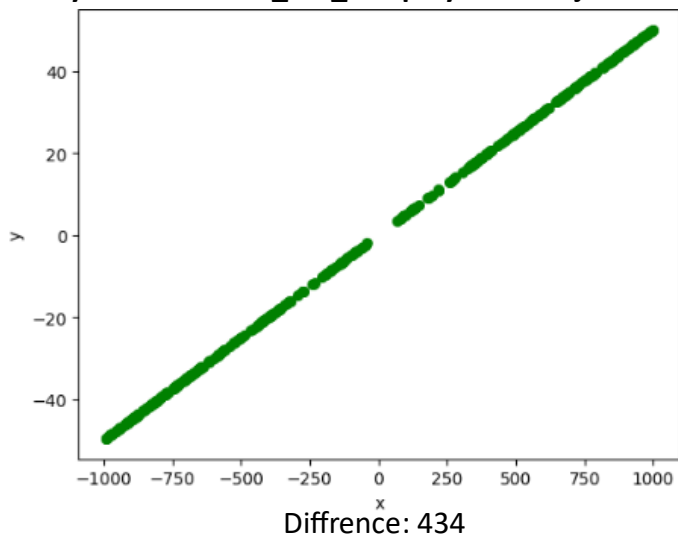
Wykres 4.2.1 Punkty skategoryzowane inaczej w zestawie 2. przez wyznacznika mat_det_2x2 oraz wyznacznika mat_det_2x2_lib przy tolerancji 10^{-14}



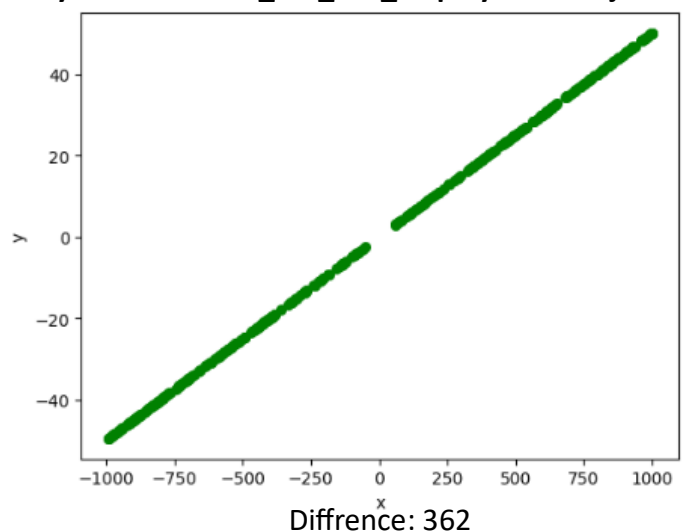
Wykres 4.2.2 Punkty skategoryzowane inaczej w zestawie 2. przez wyznacznika mat_det_3x3 oraz wyznacznika mat_det_3x3_lib przy tolerancji 10^{-14}



Wykres 4.2.3 Punkty skategoryzowane inaczej w zestawie 2. przez wyznacznika `mat_det_2x2` oraz wyznacznika `mat_det_3x3` przy tolerancji 10^{-14}



Wykres 4.2.4 Punkty skategoryzowane inaczej w zestawie 2. przez wyznacznika `mat_det_2x2_lib` oraz wyznacznika `mat_det_3x3_lib` przy tolerancji 10^{-14}



Porównałem także zestawy danych 2 i 4 przy precyzji float32. Znaczące różnice można było zaobserwować dla Zestawu 4, gdzie o wiele mniej było różnic w klasyfikacji punktów między wyznacznikami. Jednak wcale z tego nie wynika, że dokładniejsze były obliczenia, wręcz przeciwnie. Jak zauważono w poprzednim podpunkcie, bardzo duża ilość punktów nawet nie jest klasyfikowana, że leży na prostej, więc dzięki temu mniej różnic wystąpiło w klasyfikacji punktów. W programie są zwizualizowane punkty skategoryzowane inaczej dla tych zbiorów, lecz ze względu na ograniczoną liczbę stron nie zamieściłem ich tu lecz odniosę się do nich we wnioskach.

2.7. Analiza wyników z uwzględnieniem wszystkich parametrów

Ostatnia część ćwiczenia polegała na dokładnej analizie wyników z uwzględnieniem różnych wyznaczników oraz tolerancji dla zera. Za pomocą posiadanych danych wygenerowano tabelę zawierającą informacje dla każdej kombinacji wyznacznika i tolerancji dla zera odnośnie kategoryzacji punktów.

Wnioski odnośnie poniższych danych oraz poprzednich obserwacji przedstawiłem w ostatniej części sprawozdania.

Tabela 1.1 Ilość punktów w każdej klasyfikacji dla zestawu 1. W zależności od użytego wyznacznika, tolerancji dla zera oraz precyzji

eps	mat_det_func	po lewej	leżące na prostej	po prawej
0 dla każdego przypadku	dla każdego przypadku	50014	0	49986

Tabela 1.3 Ilość punktów w każdej klasyfikacji dla zestawu 3. W zależności od użytego wyznacznika, tolerancji dla zera oraz precyzji

eps	mat_det_func	po lewej	leżące na prostej	po prawej
0 dla każdego przypadku	dla każdego przypadku	521	0	479

Tabele 1.2 Ilość punktów w każdej klasyfikacji dla zestawu 2. W zależności od użytego wyznacznika, tolerancji dla zera oraz precyzji

Float64						Float32					
	eps	mat_det_func	po lewej	leżące na prostej	po prawej		eps	mat_det_func	po lewej	leżące na prostej	po prawej
0	1.000000e-08	mat_det_2x2	49774	5	50221	0	1.000000e-08	mat_det_2x2	49775	3	50222
1	1.000000e-10	mat_det_2x2	49774	5	50221	1	1.000000e-10	mat_det_2x2	49775	3	50222
2	1.000000e-12	mat_det_2x2	49774	5	50221	2	1.000000e-12	mat_det_2x2	49775	3	50222
3	1.000000e-14	mat_det_2x2	49774	5	50221	3	1.000000e-14	mat_det_2x2	49775	3	50222
4	0.000000e+00	mat_det_2x2	49774	5	50221	4	0.000000e+00	mat_det_2x2	49775	3	50222
5	1.000000e-08	mat_det_2x2_lib	49775	3	50222	5	1.000000e-08	mat_det_2x2_lib	49775	4	50221
6	1.000000e-10	mat_det_2x2_lib	49775	3	50222	6	1.000000e-10	mat_det_2x2_lib	49775	4	50221
7	1.000000e-12	mat_det_2x2_lib	49775	3	50222	7	1.000000e-12	mat_det_2x2_lib	49775	4	50221
8	1.000000e-14	mat_det_2x2_lib	49775	3	50222	8	1.000000e-14	mat_det_2x2_lib	49775	4	50221
9	0.000000e+00	mat_det_2x2_lib	49775	3	50222	9	0.000000e+00	mat_det_2x2_lib	49775	4	50221
10	1.000000e-08	mat_det_3x3	49777	0	50223	10	1.000000e-08	mat_det_3x3	49777	0	50223
11	1.000000e-10	mat_det_3x3	49777	0	50223	11	1.000000e-10	mat_det_3x3	49777	0	50223
12	1.000000e-12	mat_det_3x3	49777	0	50223	12	1.000000e-12	mat_det_3x3	49777	0	50223
13	1.000000e-14	mat_det_3x3	49777	0	50223	13	1.000000e-14	mat_det_3x3	49777	0	50223
14	0.000000e+00	mat_det_3x3	49777	0	50223	14	0.000000e+00	mat_det_3x3	49777	0	50223
15	1.000000e-08	mat_det_3x3_lib	49777	0	50223	15	1.000000e-08	mat_det_3x3_lib	49777	0	50223
16	1.000000e-10	mat_det_3x3_lib	49777	0	50223	16	1.000000e-10	mat_det_3x3_lib	49777	0	50223
17	1.000000e-12	mat_det_3x3_lib	49777	0	50223	17	1.000000e-12	mat_det_3x3_lib	49777	0	50223
18	1.000000e-14	mat_det_3x3_lib	49777	0	50223	18	1.000000e-14	mat_det_3x3_lib	49777	0	50223
19	0.000000e+00	mat_det_3x3_lib	49777	0	50223	19	0.000000e+00	mat_det_3x3_lib	49777	0	50223

Tabele 1.4 Ilość punktów w każdej klasyfikacji dla zestawu 4. W zależności od użytego wyznacznika, tolerancji dla zera oraz precyzji

Float64						Float32					
	eps	mat_det_func	po lewej	leżące na prostej	po prawej		eps	mat_det_func	po lewej	leżące na prostej	po prawej
0	1.000000e-08	mat_det_2x2	0	1000	0	0	1.000000e-08	mat_det_2x2	375	185	440
1	1.000000e-10	mat_det_2x2	0	1000	0	1	1.000000e-10	mat_det_2x2	375	183	442
2	1.000000e-12	mat_det_2x2	83	836	81	2	1.000000e-12	mat_det_2x2	375	183	442
3	1.000000e-14	mat_det_2x2	138	711	151	3	1.000000e-14	mat_det_2x2	382	163	455
4	0.000000e+00	mat_det_2x2	145	692	163	4	0.000000e+00	mat_det_2x2	382	163	455
5	1.000000e-08	mat_det_2x2_lib	0	1000	0	5	1.000000e-08	mat_det_2x2_lib	375	185	440
6	1.000000e-10	mat_det_2x2_lib	0	1000	0	6	1.000000e-10	mat_det_2x2_lib	375	183	442
7	1.000000e-12	mat_det_2x2_lib	122	776	102	7	1.000000e-12	mat_det_2x2_lib	393	146	461
8	1.000000e-14	mat_det_2x2_lib	155	702	143	8	1.000000e-14	mat_det_2x2_lib	401	132	467
9	0.000000e+00	mat_det_2x2_lib	168	686	146	9	0.000000e+00	mat_det_2x2_lib	403	128	469
10	1.000000e-08	mat_det_3x3	0	1000	0	10	1.000000e-08	mat_det_3x3	375	185	440
11	1.000000e-10	mat_det_3x3	0	1000	0	11	1.000000e-10	mat_det_3x3	375	183	442
12	1.000000e-12	mat_det_3x3	0	1000	0	12	1.000000e-12	mat_det_3x3	375	183	442
13	1.000000e-14	mat_det_3x3	1	787	212	13	1.000000e-14	mat_det_3x3	375	163	462
14	0.000000e+00	mat_det_3x3	113	588	299	14	0.000000e+00	mat_det_3x3	395	107	498
15	1.000000e-08	mat_det_3x3_lib	0	1000	0	15	1.000000e-08	mat_det_3x3_lib	375	185	440
16	1.000000e-10	mat_det_3x3_lib	0	1000	0	16	1.000000e-10	mat_det_3x3_lib	375	183	442
17	1.000000e-12	mat_det_3x3_lib	0	1000	0	17	1.000000e-12	mat_det_3x3_lib	375	183	442
18	1.000000e-14	mat_det_3x3_lib	15	879	106	18	1.000000e-14	mat_det_3x3_lib	387	158	455
19	0.000000e+00	mat_det_3x3_lib	351	316	333	19	0.000000e+00	mat_det_3x3_lib	449	48	503

3. Wnioski

Jak widać na podstawie powyższych danych różnice w klasyfikacji położenia punktu względem odcinka w zależności od metody obliczania wyznacznika są znaczące. Dodatkowe rozpatrywanie różnych tolerancji dla zera oraz precyzji jeszcze bardziej poszerza zakres wyników dla tych samych danych. Różnice te występują jednak tylko w niektórych przypadkach. Dla Zestawu 1 oraz 3 wcale ich nie zauważono.

W Zestawie 2 widzimy że bez względu na tolerancję dla zera oraz precyzji zawsze w tak ogromnym zbiorze kilka punktów było klasyfikowanych jako punkty leżące na prostej dla wyznaczników `mat_det_2x2` i `mat_det_2x2_lib`. Jest to spowodowane tym, że współrzędne są ogromnymi liczbami, a to powodować może błędy obliczeniowe związane z precyzją operacji arytmetycznych przy wyznaczaniu wyznacznika.

Zestaw 4 okazał się najlepszym zbiorem do sprawdzenia poprawności sklasyfikowania punktów zależności od wpływu rodzaju wyznacznika, tolerancji dla zera oraz precyzji.

Najpierw można zauważyć że przez ustawienie na gorszą precyzję (`float32`) w żadnym przypadku punkty nie zostały wystarczająco dobrze sklasyfikowane. Teoretycznie wszystkie punkty powinny znaleźć się na prostej, lecz tak się nie dzieje. Gorsza precyzja, która powoduje mniejszą ilość cyfr po przecinku, zwiększa tylko błędy i dokładność obliczeń.

Także dla `float64` nie zawsze wszystkie punkty zostały sklasyfikowane na prostej. Tutaj można zauważyć, że punkty dopiero są klasyfikowane poprawnie dla tolerancji dla zera równej 10^{-10} . Przy tak dużej tolerancji błędy obliczeniowe spowodowane skończoną precyzją komputera są pomijane dla wszystkich wyznaczników. Dla takiego doboru tolerancji nie ma znaczenia wybór wyznacznika. Jednak dla mniejszych tolerancji już mogą się pojawić problemy.

Dla tolerancji równej 0 najlepiej sobie radzi `mat_det_2x2`, który jest dokładny w ok. 70%. Najgorzej dla tego przypadku poradził sobie `mat_det_3x3_lib`, którego dokładność wynosi jedynie ok. 31 %. Jednak już dla tolerancji dla zera równej 10^{-12} sytuacja się zmienia. Dla tego przypadku oba wyznaczniki 2x2 zaledwie poprawiają swoją dokładność o kilka procent, natomiast wyznaczniki 3x3 sięgają dokładności równej 100%.

Jak można zauważyć nie istnieje jedna najlepsza metoda klasyfikacji. Wyznacznik należy dobrać w zależności od potrzeb i wymaganych tolerancji tak, aby ich kombinacja dawała oczekiwane rezultaty jak najbardziej dokładne.

Z tego względu ja osobiście wybrałbym wyznacznik 3x3 własnej implementacji, ponieważ dla niego można użyć najmniejszego możliwej tolerancji dla zera równej 10^{-12} oraz jest bardziej odporna na wystąpienie błędów obliczeniowych związanych z precyzją operacji arytmetycznych, tak jak to wystąpiło dla wyznaczników 2x2.