

Dariusz Marecik

gr. 4, Pon. godz. 15:00 A

Data wykonania: 07.10.2024

Data oddania: 27.10.2024

## Algorytmy Geometryczne - laboratorium 1

# Predykaty geometryczne

### 1. Cel ćwiczenia

Celem ćwiczenia jest implementacja podstawowych predykatów geometrycznych - m. in. określenie po której stronie prostej znajduje się punkt, a także przeprowadzenie testów, wizualizacja i opracowanie danych.

### 2. Wstęp teoretyczny

Do sklasyfikowania punktu względem zadanej prostej skorzystam z odpowiedniego wyznacznika macierzy.

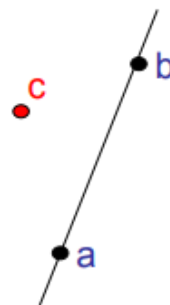
Dane są trzy punkty:

$$a = (a_x, a_y),$$

$$b = (b_x, b_y),$$

$$c = (c_x, c_y).$$

Znak wyznacznika  $\det(a, b, c)$  jednoznacznie klasyfikuje punkt  $c$  względem prostej przechodzącej przez punkty  $a$  i  $b$



Rysunek 1: Przykładowe punkty oraz prosta

$\det(a, b, c)$  można wyrazić dwoma równoważnymi wzorami:

$$\det(a, b, c) = \begin{vmatrix} a_x - c_x & a_y - c_y \\ b_x - c_x & b_y - c_y \end{vmatrix}$$

$$\det(a, b, c) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}$$

Jeżeli:

$\det(a, b, c) > 0 \Rightarrow$  punkt  $c$  leży po lewej stronie prostej  $ab$

$\det(a, b, c) < 0 \Rightarrow$  punkt  $c$  leży po prawej stronie prostej  $ab$

$\det(a, b, c) = 0 \Rightarrow$  punkt leży na prostej  $ab$

### 3. Dane techniczne

#### 3.1. Metodologia

Zbiory danych oparte są na wygenerowanych losowo punktach przy pomocy funkcji `random.uniform()` z biblioteki `numpy`. Każda funkcja obliczająca wyznacznik została napisana w dwóch wersjach: z wykorzystaniem funkcji `linalg.det()` z biblioteki `numpy` (są to funkcje `mat_det_3x3_lib` i `mat_det_2x2_lib`) oraz bez ich używania (`mat_det_3x3`, `mat_det_2x2`). Wyznacznik `mat_det_3x3` był liczony metodą Sarusa. Dla funkcji `mat_det_3x3` i `mat_det_2x2` typ zwracanej zmiennej był identyczny jak typ otrzymany w argumencie danej funkcji. Wyniki zostały wygenerowane dla dwóch typów zmiennych: `float32` i `float64` oraz dla pięciu tolerancji dla zera:  $\epsilon \in \{0, 10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}\}$ .

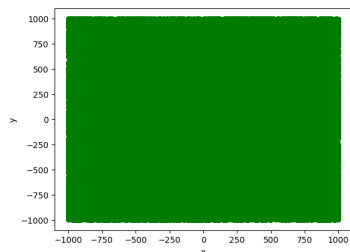
#### 3.2. Specyfikacja narzędzi i sprzętu

Do generowania wykresów i tabel została wykorzystana biblioteka `matplotlib` i `pandas` oraz narzędzie przygotowane przez *koło naukowe Bit*. Program został napisany w języku Python w środowisku *Jupyter Notebook*. Wykorzystany procesor do zebrania danych to Intel Core i5-12500H 4.5 GHz pracujący w systemie Linux Mint 21.4

### 4. Wykonanie ćwiczenia

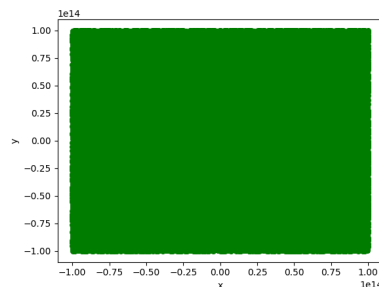
Na początku przygotowałem 4 zbiory danych do analizy:

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



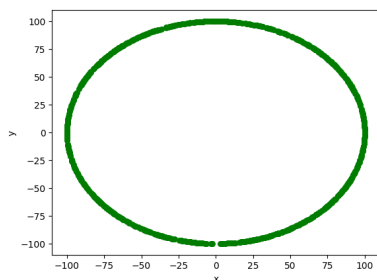
Rysunek 2: Zbiór A

b)  $10^5$  losowych punktów o współrzędnych z przedziału  $[-10^{14}, 10^{14}]$  (zbiór B)



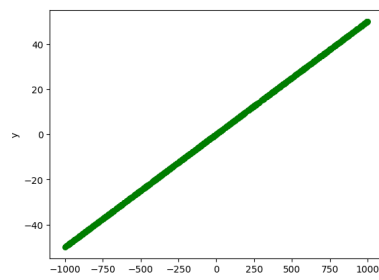
Rysunek 3: Zbiór B

a)  $10^3$  losowych punktów leżących na okręgu o środku w punkcie (0, 0) i promieniu  $R = 100$  (zbiór C)



Rysunek 4: Zbiór C

b)  $10^3$  losowych punktów o współrzędnych z przedziału  $[-1000, 1000]$  leżących na prostej wyznaczonej przez wektor  $[a, b]$ , gdzie  $a = (-1.0, 0.0)$ ,  $b = (1.0, 0.1)$  (zbiór D)



Rysunek 5: Zbiór D

Następnie dla każdego zbioru dokonałem klasyfikacji punktów biorąc pod uwagę ich ułożenie względem prostej przechodzącej przez punkty  $a$  i  $b$ , gdzie  $a = (-1.0, 0.0)$ ,  $b = (1.0, 0.1)$ . Powstały w ten sposób trzy zbiory: zbiór punktów leżących po prawej stronie od prostej, po lewej stronie od prostej i na samej prostej. Do przydzielania punktów do poszczególnych zbiorów wykorzystałem wyznacznik omówiony w rozdziale 2.

Dane pogrupowałem ze względu na cztery metody obliczania wyznacznika podpisane (`mat_det_3x3`, `mat_det_2x2`, `mat_det_3x3_lib` i `mat_det_2x2_lib`), pięć tolerancji dla zera ( $\epsilon$ ) i dwa rodzaje zmiennych użytych do przechowywania współrzędnych punktów (`float32` i `float64`).

Legenda wykresów:

- *kolor zielony* - punkt leży po lewej stronie prostej
- *kolor fioletowy* - punkt leży na prostej
- *kolor żółty* - punkt leży po prawej stronie prostej

## 5. Opracowanie wyników

Jeśli w nagłówku danej podtabeli widnieje więcej niż jedna wartość  $\epsilon$  lub dwa rodzaje zmiennych, sugeruje to, że dane wygenerowane przez program dla tych zestawów parametrów były identyczne. Oznacza to, że przy zastosowanych wartościach tolerancji oraz typach zmiennych wyniki nie uległy zmianie

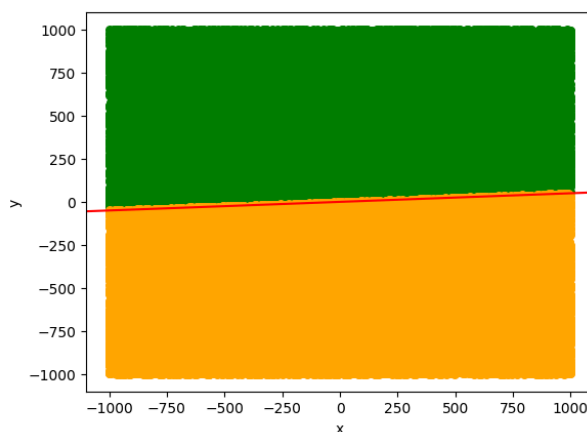
### 5.1. Zbiór A

Funkcja użyta do liczenia wyznacznika	Liczba punktów po lewej stronie	Liczba punktów na prostej	Liczba punktów po prawej stronie
Tolerancja: $\epsilon = 0, 10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}$ , Typ danych: <code>float32</code> i <code>float64</code>			
<code>mat_det_3x3</code>	50080	0	49920
<code>mat_det_2x2</code>	50080	0	49920
<code>mat_det_3x3_lib</code>	50080	0	49920
<code>mat_det_2x2_lib</code>	50080	0	49920

Tabela 1: Rozkład punktów dla zbioru A

Z Tabela 1 wnioskujemy, że dla zbioru A nie ma znaczenia zarówno wykorzystany sposób liczenia wyznacznika, precyzja zera jak i typ użytych zmiennych.

Rysunek 6 przedstawia graficzną wizualizację podziału zbioru A



Rysunek 6: Podział punktów zbioru A względem prostej

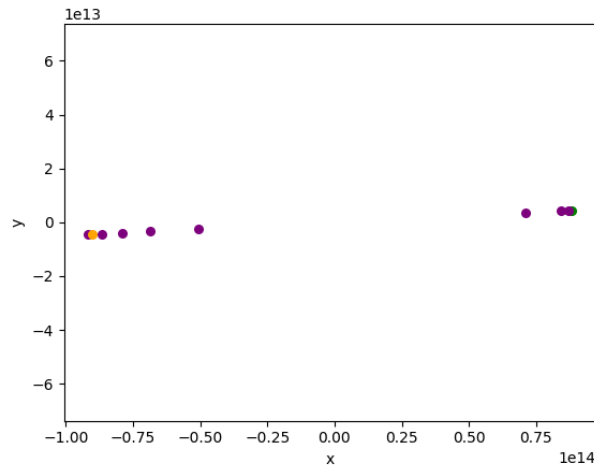
## 5.2. Zbiór B

### 5.2.1. float64

Funkcja użyta do liczenia wyznacznika	Liczba punktów po lewej stronie	Liczba punktów na prostej	Liczba punktów po prawej stronie
Tolerancja: $\varepsilon = 0, 10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}, 0$			
mat_det_3x3	49726	0	50274
mat_det_2x2	49721	8	50271
mat_det_3x3_lib	49726	0	50274
mat_det_2x2_lib	49720	8	50272

Tabela 2: Rozkład punktów dla zbioru B, float64

Na podstawie danych w Tabeli 2 zauważamy drobne różnice pomiędzy metodami liczenia wyznacznika. Zwizualizujemy różnice między nimi:



Rysunek 7: Różnica klasyfikacji punktów między wyznacznikami mat\_det\_2x2 i mat\_det\_3x3

**1 punkt po lewo stronie prostej, 8 punktów na prostej, 1 punkt po prawej stronie prostej**

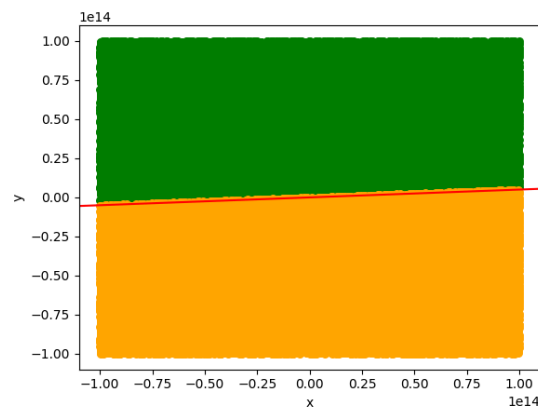
Punkty, które w przypadku funkcji mat\_det\_2x2 zostały zakwalifikowane jako leżące na prostej, dla wyznacznika mat\_det\_3x3 na niej nie leżą. Rysunek 7 przedstawia punkty, które zostały zaklasyfikowane dla mat\_det\_2x2 inaczej niż dla mat\_3x3. Widzimy, że te punkty znajdują się blisko krańcowych wartości swoich współrzędnych. Funkcje mat\_det\_2x2 oraz mat\_det\_2x2\_lib okazały się lepsze dla Zbioru B w porównaniu do wyznaczników mat\_det\_3x3 i mat\_det\_3x3\_lib.

Co ciekawe znalazły się dwa punkty, które dla jednego wyznacznika zostały zaklasyfikowane jako leżące po jednej stronie prostej, zaś dla drugiego po przeciwnej stronie. Oznacza to, że dla tych samych punktów wyznaczniki obliczone metodami mat\_det\_3x3 i mat\_det\_2x2 mają przeciwne znaki.

Warto podkreślić, że przy naszym zakresie współrzędnych prawdopodobieństwo, że punkt znajdzie się dokładnie na prostej, jest bardzo niskie. Dlatego, jeżeli pojedyncze punkty zostały zaklasyfikowane jako leżące na prostej, jest to najprawdopodobniej wynikiem braku precyzji dla float64 przy bardzo dużych wartościach składowych operacji liczenia wyznacznika.

Analizując cały Zbiór B, ponownie nie obserwujemy różnic wynikających z zastosowania różnych wartości  $\varepsilon$ , co wskazuje na niską wrażliwość tolerancji zera dla tego zbioru.

Rysunek 8 przedstawia graficzną wizualizację podziału zbioru B:

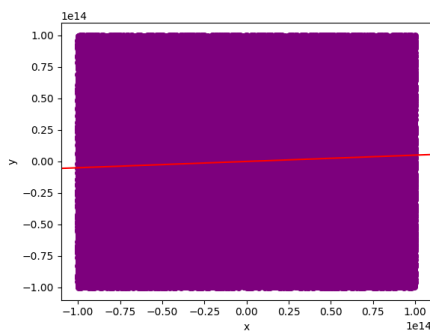


Rysunek 8: Podział punktów zbioru B względem prostej dla float64

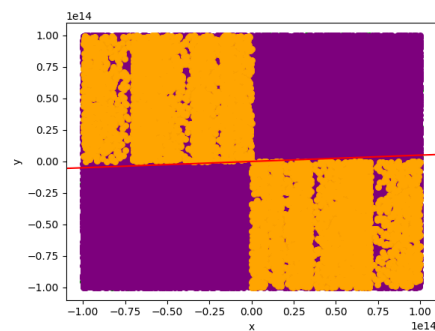
### 5.2.2. float32

Funkcja użyta do liczenia wyznacznika	Liczba punktów po lewej stronie	Liczba punktów na prostej	Liczba punktów po prawej stronie
Tolerancja: $\varepsilon = 0, 10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}$			
mat_det_3x3	49726	0	50274
mat_det_2x2	0	100000	0
mat_det_3x3_lib	49726	0	50274
mat_det_2x2_lib	6602	86769	6629

Tabela 3: Rozkład punktów dla zbioru B, float32



Rysunek 9: Podział punktów zbioru B względem prostej dla float32 i mat\_det\_2x2



Rysunek 10: Podział punktów zbioru B względem prostej dla float32 i mat\_det\_2x2\_lib

Analizując dane z Tabela 3, zauważamy anomalie – wszystkie punkty dla mat\_det\_2x2 (Rysunek 9) i większość dla mat\_det\_2x2\_lib (Rysunek 10) zostały zaklasyfikowane jako leżące na prostej, co jest oczywiście błędne. Różnica ta wynika ze specyfiki reprezentowania liczb zmiennoprzecinkowych. Choć liczba bitów mantysy jest stała, to przy dużych wartościach liczbowych odstęp między kolejnymi możliwymi do reprezentacji liczbami stają się większe. W efekcie, przy obliczaniu wyznacznika macierzy 2x2, odejmowanie podobnych, dużych wartości może skutkować utratą dokładności, ponieważ te same bity mantysy odpowiadają coraz większym różnicom między liczbami.

Dla funkcji mat\_det2x2\_lib większość punktów z 1 i 3 ćwiartki układu współrzędnych zostały sklasyfikowane jako leżące na prostej, zaś dla 2 i 4 ćwiartki jako położone po prawej stronie prostej.

Dla funkcji liczących wyznacznik macierzy 3x3 nie występują różnice między użyciem float32 i float64, co podkreśla wyższość tych metod nad pozostałymi. Ponadto, w tym przypadku, wartość  $\varepsilon$  również nie ma żadnego wpływu na wyniki dla żadnej metody liczenia wyznacznika.

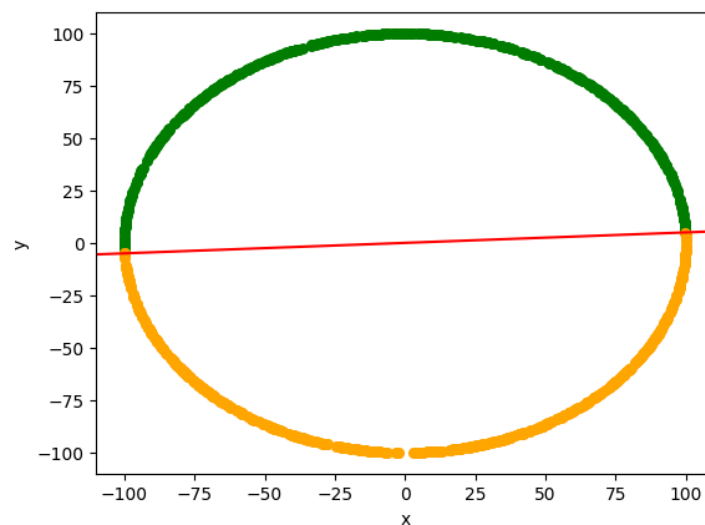
### 5.3. Zbiór C

Funkcja użyta do liczenia wyznacznika	Liczba punktów po lewej stronie	Liczba punktów na prostej	Liczba punktów po prawej stronie
Tolerancja: $\varepsilon = 0, 10^{-14}, 10^{-12}, 10^{-10}, 10^{-8}$			
Typ danych: float32 i float64			
mat_det_3x3	488	0	512
mat_det_2x2	488	0	512
mat_det_3x3_lib	488	0	512
mat_det_2x2_lib	488	0	512

Tabela 4: Rozkład punktów dla zbioru C

Analizując dane z Tabela 4, zauważamy brak różnic między metodami obliczania wyznacznika, wartościami dokładności  $\varepsilon$ , jak i typami użytych zmiennych. Niewrażliwość zbioru C na zastosowane parametry wynika z niewielkiej liczby analizowanych punktów oraz z niskiego prawdopodobieństwa, że którykolwiek z punktów znajdzie się blisko prostej ab. Taka konfiguracja sprawia, że różnice w dokładności i precyzji obliczeń nie mają istotnego wpływu na wyniki klasyfikacji.

Rysunek 11 przedstawia graficzną wizualizację podziału zbioru C.



Rysunek 11: Podział punktów zbioru C względem prostej

## 5.4. Zbiór D

Zbiór ten składa się z punktów idealnie położonych na prostej, co powinno skutkować jednoznaczną ich klasyfikacją. Przeanalizujemy to szczegółowo.

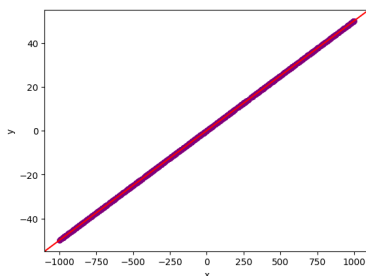
### 5.4.1. float64

Funkcja użyta do liczenia wyznacznika	Liczba punktów po lewej stronie	Liczba punktów na prostej	Liczba punktów po prawej stronie
Tolerancja: $\varepsilon = 10^{-8}, 10^{-10}$			
mat_det_3x3	0	1000	0
mat_det_2x2	0	1000	0
mat_det_3x3_lib	0	1000	0
mat_det_2x2_lib	0	1000	0
Tolerancja: $\varepsilon = 10^{-12}$			
mat_det_3x3	0	1000	0
mat_det_2x2	83	842	75
mat_det_3x3_lib	0	1000	0
mat_det_2x2_lib	107	783	110
Tolerancja: $\varepsilon = 10^{-14}$			
mat_det_3x3	0	1000	0
mat_det_2x2	177	681	142
mat_det_3x3_lib	19	883	98
mat_det_2x2_lib	157	685	158
Tolerancja: $\varepsilon = 0$			
mat_det_3x3	173	455	372
mat_det_2x2	184	668	148
mat_det_3x3_lib	379	280	341
mat_det_2x2_lib	171	660	169

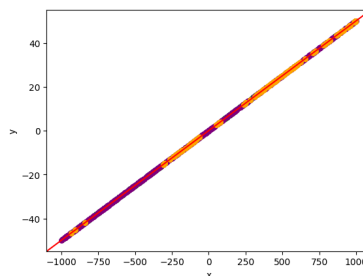
Tabela 5: Rozkład punktów dla zbioru D, float64

Dane z Tabela 5 wyraźnie ukazują znaczenie dokładności parametru  $\varepsilon$  dla zbioru D. Przy dużej tolerancji zera (np.  $10^{-8}$  i  $10^{-10}$ ), program prawidłowo klasyfikuje wszystkie punkty na osi (Rysunek 12), niezależnie od metody obliczania wyznacznika. Jednak wraz ze zmniejszeniem wartości  $\varepsilon$ , wyniki stają się coraz mniej dokładne. Dla  $\varepsilon = 10^{-14}$ , funkcja mat\_det\_2x2\_lib błędnie zaklasyfikowała aż 32% punktów ze zbioru (Rysunek 13). Warto zauważyć istotne różnice między metodami obliczania wyznacznika. Funkcja mat\_det\_3x3 nie zwróciła żadnych błędnych wyników dla żadnej wartości  $\varepsilon$  poza  $\varepsilon = 0$ . Widoczny jest duży przeskok pomiędzy liczbą punktów prawidłowo zaklasyfikowanych pomiędzy  $\varepsilon = 10^{-14}$  i  $\varepsilon = 0$  dla wyznaczników macierzy 3x3. Spadek ten wynosi od 54,5% dla mat\_det\_3x3 do 60,3% dla mat\_det\_3x3\_lib, co czyni z nich najmniej skuteczne metody klasyfikacji dla tego  $\varepsilon$ .

- Wykresy:



Rysunek 12: Podział punktów zbioru D względem prostej ( $\varepsilon = 10^{-8}$ , mat\_det\_2x2, float64)

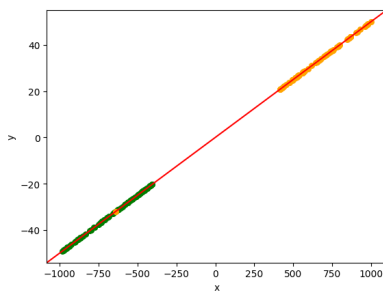


Rysunek 13: Podział punktów zbioru D względem prostej ( $\varepsilon = 10^{-14}$ , mat\_det\_2x2\_lib, float64)

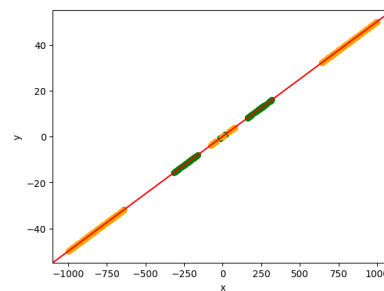
Przeanalizuję teraz, które punkty zostały błędnie sklasyfikowane dla  $\varepsilon = 10^{-12}$  i funkcji `mat_det_2x2` oraz różnicę pomiędzy dokładnościami  $\varepsilon = 10^{-12}$  i  $\varepsilon = 10^{-14}$

Z wykresu Rysunek 14 wynika, że dla funkcji `mat_det_2x2` najbardziej problematyczne ponownie okazały się punkty o dużych wartościach współrzędnych  $x$  i  $y$ . Co ciekawe, punkty z obydwojema współrzędnymi ujemnymi były klasyfikowane po lewej stronie od prostej, natomiast punkty o dodatnich współrzędnych zostały przypisane do zbioru po prawej stronie prostej.

Wykres Rysunek 15 ilustruje punkty błędnie zaklasyfikowane dla  $e = 0$ , które przy tolerancji  $e = 10^{-14}$  zostały prawidłowo przyporządkowane. Widoczne jest zgromadzenie nieprawidłowo sklasyfikowanych punktów w 5 grupach, które są symetrycznie ułożone względem punktu (0.0). Co ciekawe punkty daleko od środka układu współrzędnych zostały w zdecydowanej większości zaklasyfikowane jako leżące po prawej stronie prostej, co oznacza, że wyznacznik dla tych punktów został określony jako ujemny.



Rysunek 14: Błędnie zaklasyfikowane punkty  
( $\varepsilon = 10^{-12}$ , `mat_det_2x2`, `float64`)



Rysunek 15: Różnica w klasyfikacji punktów  
pomiędzy  $\varepsilon = 0$  i  $\varepsilon = 10^{-14}$   
(`mat_det_3x3`, `float64`)

## 5.5. float32

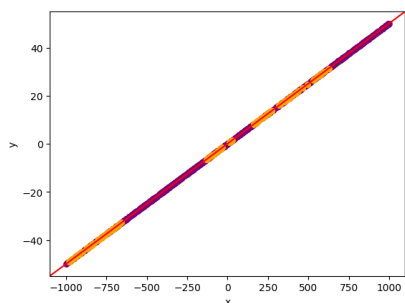
Funkcja użyta do liczenia wyznacznika	Liczba punktów po lewej stronie	Liczba punktów na prostej	Liczba punktów po prawej stronie
Tolerancja: $\varepsilon = 10^{-8}$			
<code>mat_det_3x3</code>	316	376	308
<code>mat_det_2x2</code>	164	685	151
<code>mat_det_3x3_lib</code>	405	200	395
<code>mat_det_2x2_lib</code>	519	1	480
Tolerancja: $\varepsilon = 10^{-10}, 10^{-12}$			
<code>mat_det_3x3</code>	317	375	308
<code>mat_det_2x2</code>	165	684	151
<code>mat_det_3x3_lib</code>	407	197	396
<code>mat_det_2x2_lib</code>	520	0	480
Tolerancja: $\varepsilon = 10^{-14}$			
<code>mat_det_3x3</code>	317	375	308
<code>mat_det_2x2</code>	165	684	151
<code>mat_det_3x3_lib</code>	419	175	406
<code>mat_det_2x2_lib</code>	520	0	480
Tolerancja: $\varepsilon = 0$			
<code>mat_det_3x3</code>	317	375	308
<code>mat_det_2x2</code>	165	684	151
<code>mat_det_3x3_lib</code>	479	60	461
<code>mat_det_2x2_lib</code>	520	0	480

Tabela 6: Rozkład punktów dla zbioru D, `float32`

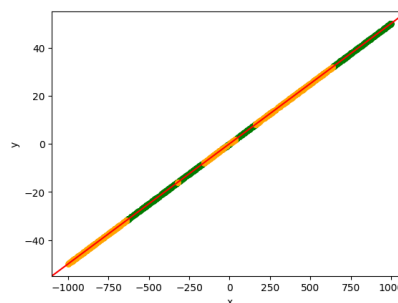


Z danych zawartych w Tabeli 6 wynika, że zmienna typu float32 wykazuje niską wrażliwość na zmienne tolerancje zera. W przypadku float32 zauważalna jest znacznie mniejsza precyzja klasyfikacji punktów dla każdego  $\varepsilon$ . Dodatkowo, widoczne są duże rozbieżności między wynikami uzyskanymi za pomocą różnych metod obliczania wyznacznika. Co zaskakujące, funkcja `mat_det_2x2` (Rysunek 16) uzyskała najwyższą liczbę prawidłowo zaklasyfikowanych punktów dla wszystkich wartości  $\varepsilon$ . Kolejno, najlepsze wyniki z tym zbiorem danych uzyskały funkcje: `mat_det_3x3`, `mat_det_3x3_lib` oraz `mat_det_2x2_lib` (Rysunek 17). Zaimplementowane własnoręcznie funkcje liczące wyznacznik okazały się dużo lepsze. Wynika to z faktu uniwersalności funkcji bibliotecznych, a co za tym idzie, trudniejszej struktury obliczeń zawartych w ich kodzie.

Wykresy:



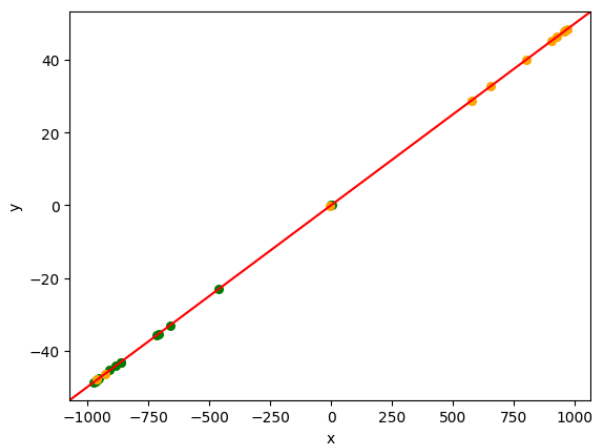
Rysunek 16: Podział punktów zbioru D względem prostej ( $\varepsilon = 10^{-14}$ , `mat_det_2x2`, float32)



Rysunek 17: Podział punktów zbioru D względem prostej ( $\varepsilon = 10^{-14}$ , `mat_det_2x2_lib`, float32)

Przeanalizuję teraz różnicę między  $\varepsilon = 10^{-8}$  i  $\varepsilon = 10^{-14}$

Z danych przedstawionych na Rysunek 18 wynika, że znaczne odchylenia od zera wartości  $\varepsilon$  nie prowadzą do istotnego wzrostu liczby błędnie zaklasyfikowanych punktów. Łączna liczba takich punktów wynosi jedynie 25, a ich rozkład jest zgodny z wcześniejszymi obserwacjami. Zauważalna tendencja wskazuje, że im większa odległość od punktu (0,0), tym większe są trudności w klasyfikacji punktów.



Rysunek 18: Różnica w klasyfikacji punktów pomiędzy  $\varepsilon = 10^{-8}$  i  $\varepsilon = 10^{-14}$  (`mat_det_3x3_lib`, float64)

## 6. Podsumowanie

- Brak różnic w precyzji dla prostych przypadków: Dla zbiorów danych, takich jak Zbiór A i Zbiór C, zarówno metoda liczenia wyznacznika, jak i typ danych (float32, float64) oraz tolerancja zera ( $\epsilon$ ) nie miały wpływu na klasyfikację punktów. Wynika to z braku punktów bliskich prostej oraz małej ilości danych.
- Zbiór B ujawnia ograniczenia precyzji: W przypadku Zbioru B różnice między metodami liczenia wyznacznika były zauważalne. Metoda `mat_det_2x2` wykazała mniejszą czułość, co spowodowało klasyfikację niektórych punktów na prostą, co nie było poprawne. To zjawisko było bardziej widoczne przy użyciu zmiennych typu float32, co sugeruje, że precyzja danych ma znaczenie w bardziej złożonych zbiorach.
- Problemy z dokładnością w Zbiorze D: Zbiór D, który zawiera punkty idealnie leżące na prostej, pokazał wpływ tolerancji na wyniki klasyfikacji. Przy większych tolerancjach ( $\epsilon = 10^{-8}$  i  $10^{-10}$ ), wyniki były zgodne z oczekiwaniami, natomiast przy mniejszych wartościach tolerancji ( $\epsilon = 10^{-12}$  i  $10^{-14}$ ) zaczęły pojawiać się błędy klasyfikacji. Metody `mat_det_3x3` okazały się najbardziej precyzyjne, zwłaszcza dla niższych wartości  $\epsilon$ .
- Float32 znacznie mniej precyzyjny: Zmienna float32 okazała się znacznie mniej dokładna od float64, szczególnie w Zbiorze D. Wartości wyznaczników przyjmowały błędne wyniki, co spowodowało zakwalifikowanie zbyt dużej liczby punktów jako leżących na prostej.
- Wpływ współrzędnych punktów na błędy: W Zbiorze D, błędy klasyfikacji były bardziej widoczne dla punktów o większych współrzędnych, zwłaszcza gdy ich wartość była znacznie oddalona od środka układu współrzędnych (0,0). To sugeruje, że precyzja liczb zmiennoprzecinkowych maleje wraz z rosnącymi wartościami współrzędnych.
- Macierze 3x3 efektywniejsze niż 2x2. Metody `mat_det_3x3` i `mat_det_3x3_lib` okazały się bardziej efektywne w porównaniu z `mat_det_2x2` oraz `mat_det_2x2_lib`. Wynika to z faktu, że w metodzie Sarusa pojawia się wiele operacji mnożenia, których wartości są stosunkowo niewielkie w porównaniu do dwóch mnożeń wykonywanych dla macierzy 2x2, gdzie wyniki mogły powodować utratę wystarczającej precyzji z racji ograniczonego rozmiaru mantysy obu badanych zmiennych.
- Funkcje biblioteczne wygodniejsze, ale mniej precyzyjne. Widoczna jest niewielka przewaga funkcji zaimplementowanych własnoręcznie od funkcji bibliotecznych liczących wyznacznik macierzy. Wynika to z bardziej skomplikowanej struktury funkcji bibliotecznych.