Algorytmy Geometryczne
Labolatorium nr 1

Szymon Szarek

28 października 2023

1 Opis ćwiczenia

Celem ćwiczenia było wyznaczenie po której stronie prostej znajduje się dany punkt. Jest to możliwe dzięki obliczeniu wyznaczników:

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

lub

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

gdzie punkty a i b wyznaczają prostą. Wyznaczniki określają położenie punktu c wobec tej prostej. Jeżeli wyznacznik jest większy od 0, oznacza to, że punkty znajduje się po lewej stronie (podczas przechodzenia kolejno po punktach $a \to b \to c$, wykonamy skręt w lewo przechodząc z b do c). Analogicznie, jeżeli wyznacznik jest mniejszy od 0, to punkt znajduje się po prawej stronie. Gdy wyznacznik jest równy 0, oznacza to, że punkt leży na tej prostej.

Wyniki wyznaczników mogą się mninimalnie różnić między sobą przez ograniczoną dokładność obliczeń liczb zmiennoprzecinkowych lub sposób ich kalkulowania.

2 Środowisko pracy

Ćwiczenie zostało napisane w języku Python za pomocą platformy Jupyter Notebook. Jako narzędzie graficzne do wizualizacji figur geometrycznych zostało użyte narzędzie przygotowane przez koło naukowe BIT, które wykorzystuje takie biblioteki jak np. *matplotlib* lub *numpy*.

Specyfikacja sprzętu:

- System Windows 11 x64
- Procesor AMD Ryzen PRO 5650U
- Pamięć RAM 16GB
- Python 3.9

3 Realizacja ćwiczenia

W ramach ćwiczenia na początku zostały wygenerowane 4 zbiory punktów typu *double*:

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

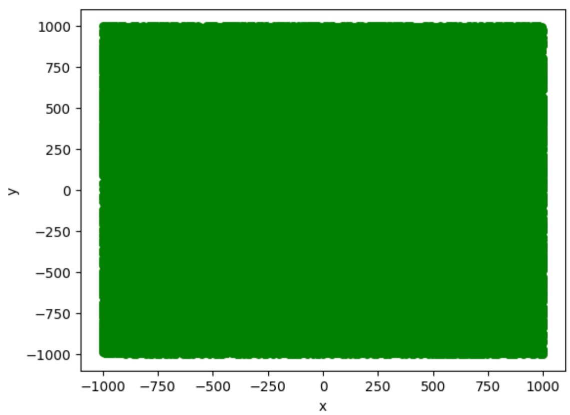
b 10^5 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$,

c 10^5 losowych punktów leżących na okręgu o środku (0,0) i o promieniu R=100,

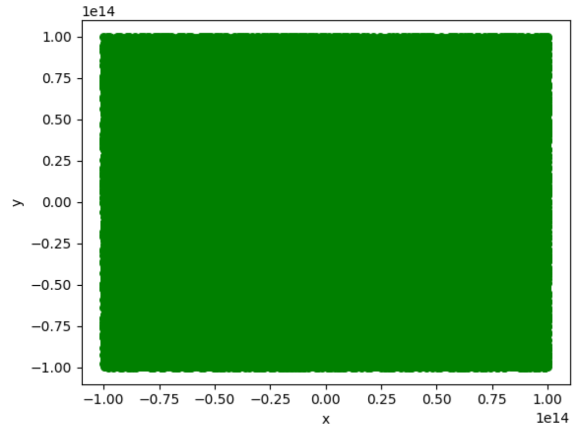
d 10^5 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ leżących na prostej wyznaczonej przez wektor (a,b), gdzie $a = [-1.0, 0.0]$, $b = [1.0, 0.1]$

Następnie zaimplementowano obliczanie wyznaczników 3x3 oraz 2x2. Wyznacznik 2x2 został obliczony z definicji permutacyjnej wyznaczników, a do obliczenia wyznacznika 3x3 wykorzystano regułę Sarrusa. Do porównania wyników skorzystano również z wbudowanej funkcji *det*() obliczających wyznaczniki znajdującej się w bibliotece numpy w module *linalg*.

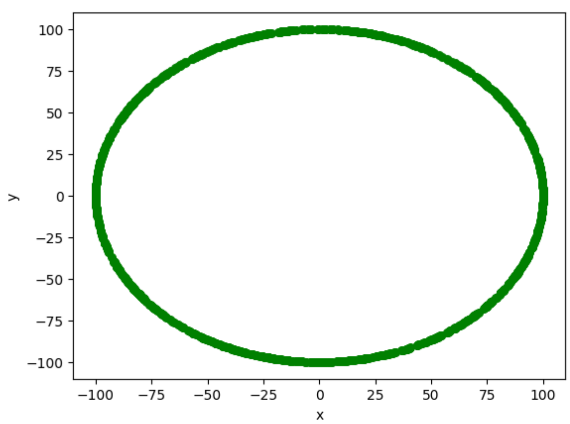
4 Wizualizacja zbiorów



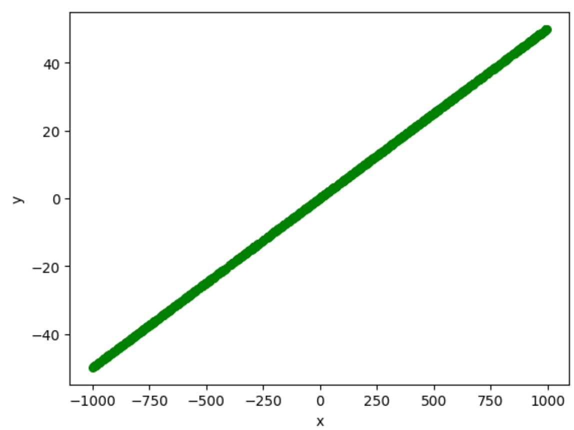
Wykres 1. Zbiór a



Wykres 2. Zbiór b



Wykres 3. Zbiór c



Wykres 4. Zbiór d

5 Klasyfikacja punktów

Zbiory punktów zostały klasyfikowane odpowiednio z tolerancjami: 0, 10^{-8},10^{-10}, 10^{-12}, 10^{-14} oraz z wykorzystaniem wcześniej wspomnianych funkcji obliczających wyznaczniki macierzy.

Legenda kolorów:

- Zielony - punkt po lewej stronie,
- Pomarańczowy - punkty po prawej stronie,
- Fioletowy - punkt współliniowy.

Dla zbiorów a oraz c wyniki klasyfikacji były identyczne niezależnie od użytej tolerancji czy też wyznacznika i prezentują się następująco:

	liczba punktów		
	lewo	środek	prawo
a	50315	0	49685
c	495	0	505

Tabela 1: Wyniki obliczeń dla zbiorów a oraz c

Dla zbioru b wyniki nie różniły się pomiędzy tolerancjami, ale różniły się pomiędzy wyznacznikami.

		liczba punktów		
		lewo	środek	prawo
b	2x2 własny	49832	5	50163
	2x2 biblioteczny	49831	4	50165
	3x3 (oba)	49834	0	50166

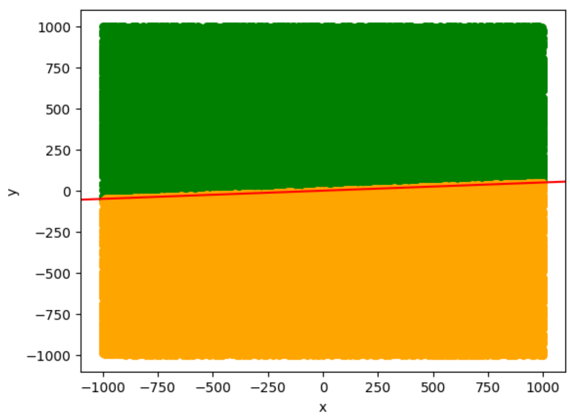
Tabela 2: Wyniki obliczeń dla zbioru b

Największe różnice można zauważyć na wynikach klasyfikacji dla punktów ze zbioru d (punkty leżące na prostej).

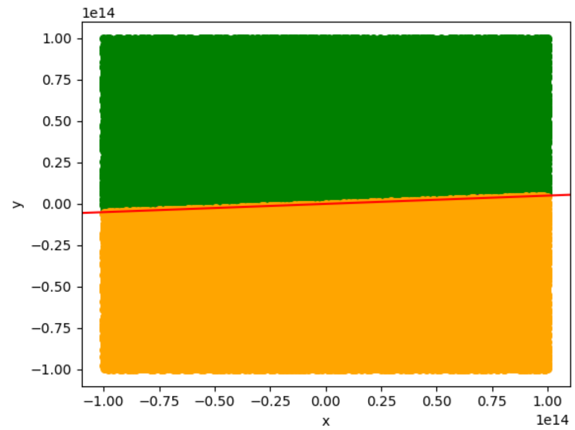
		2x2 własny	2x2 biblioteczny	3x3 własny	3x3 biblioteczny
0	lewo	144	152	373	494
	środek	701	682	346	278
	prawo	155	166	281	228
10^{-14}	lewo	137	146	0	85
	środek	713	700	1000	888
	prawo	150	154	0	27
10^{-12}	lewo	88	96	0	0
	środek	836	791	1000	1000
	prawo	76	113	0	0
10^{-10}	lewo	0	0	0	0
	środek	1000	1000	1000	1000
	prawo	0	0	0	0
10^{-8}	lewo	0	0	0	0
	środek	1000	1000	1000	1000
	prawo	0	0	0	0

Tabela 3: Wyniki obliczeń dla zbioru d

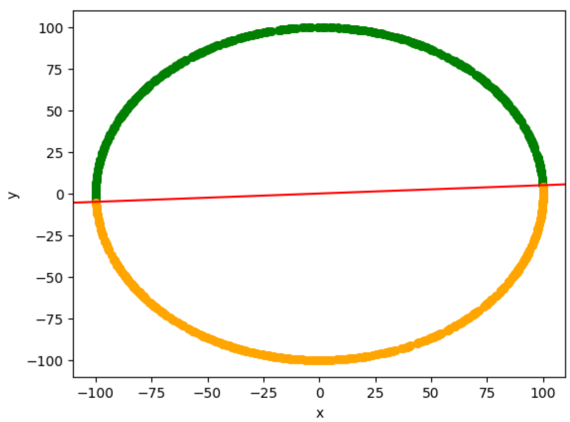
5.1 Wizualizacja klasyfikacji punktów



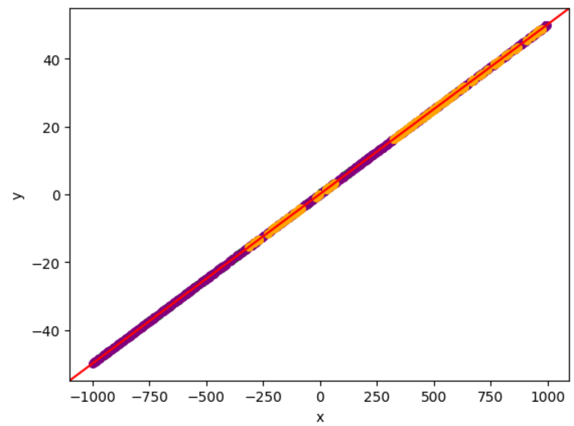
Wykres 5. Zbiór a dla wyznacznika własnego 2x2



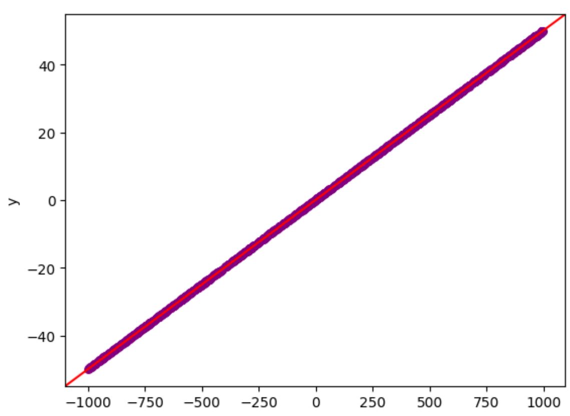
Wykres 6. Zbiór b dla wyznacznika własnego 2x2



Wykres 7. Zbiór c dla wyznacznika własnego 2x2



Wykres 8. Zbiór d dla wyznacznika własnego 2x2 przy tolerancji równej 0



Wykres 9. Zbiór d dla wyznacznika własnego 2x2 przy tolerancji równej 10^{-10}

6 Różnice w klasyfikacji punktów

6.1 Różnice pomiędzy funkcjami własnymi a bibliotecznymi

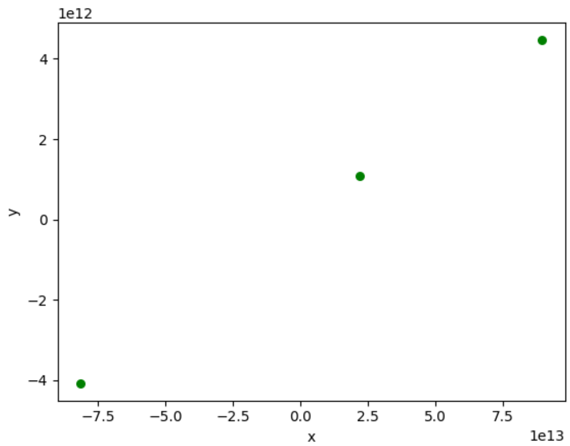
Dla ścisłości analizy porównywano ze sobą tylko funkcje obliczające te same wyznaczniki, tj. funkcje obliczające wyznacznik 2x2 oraz funkcje obliczające wyznacznik 3x3.
Nie zauważono różnic pomiędzy funkcjami własnymi i bibliotecznymi dla zbiorów a oraz c.
Do porównań wykorzystano tolerancję równą 10^{-14}.

W zbiorze b pojawiły się minimalne różnice:

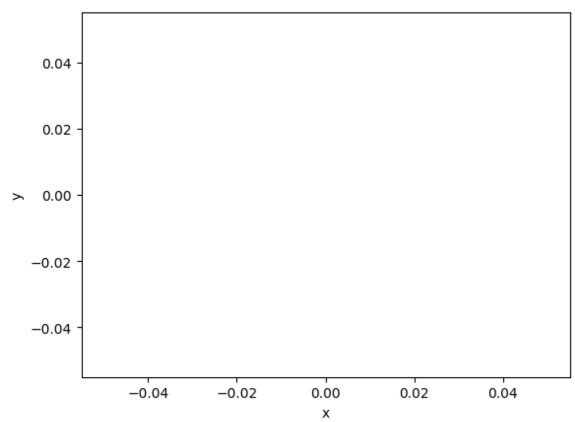
- 3 punkty dla współczynników 2x2
- 0 punktów dla współczynników 3x3

Natomiast w zbiorze d występowały diametralne różnice:

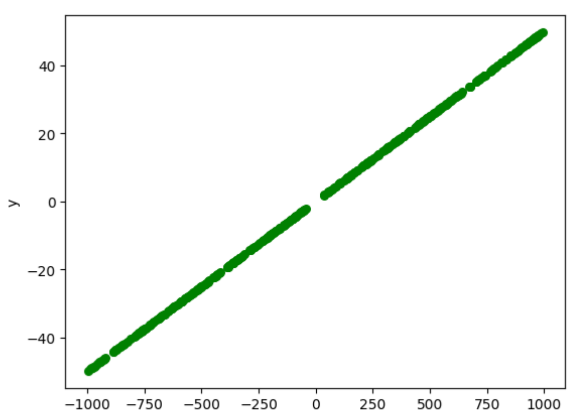
- 338 punktów dla współczynników 2x2
- 112 punktów dla współczynników 3x3



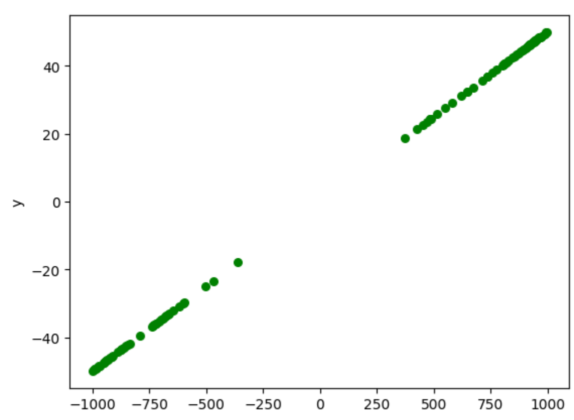
Wykres 10. Zbiór b, różnice w wyznacznikach 2x2



Wykres 11. Zbiór b, różnice w wyznacznikach 3x3



Wykres 12. Zbiór d, różnice w wyznacznikach 2x2



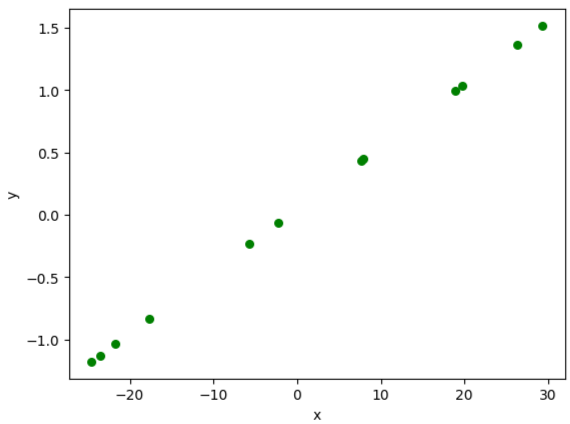
Wykres 13. Zbiór d, różnice w wyznacznikach 3x3

6.2 Różnice pomiędzy tolerancjami

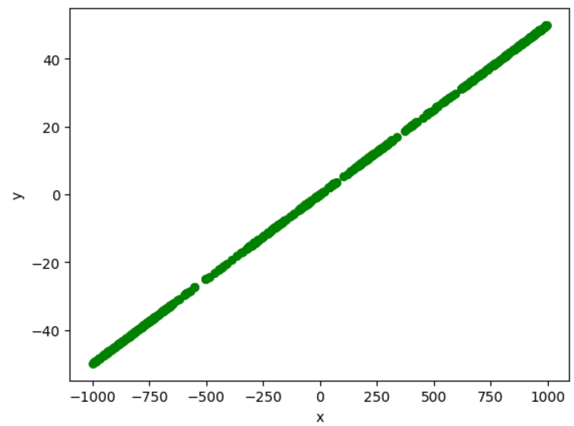
Dla zbiorów a,b,c nie ma żadnych różnic w obliczeniach pomiędzy tolerancjami.
W zbiorze d tolerancje okazały się kluczowe do poprawnych klasyfikacji punktów współliniowych:

- 12 punktów różnie sklasyfikowanych dla współczynników 2x2
- 654 punkty różnie sklasyfikowane dla współczynników 3x3

Porównywane były odpowiednio tolerancje: 0 oraz 10^{-14}.



Wykres 14. Zbiór d, różnice w tolerancjach 2x2



Wykres 15. Zbiór b, różnice w tolerancjach 3x3

6.3 Różnice w czasie wykonywania

Różnice pomiędzy własnoręcznie zaimplementowanymi funkcjami obliczającymi wyznaczniki macierzy, a funkcjami bibliotecznymi pojawiały się również w czasach wykonywania klasyfikacji. Funkcje zaimplementowane okazały się dużo szybsze od tych bibliotecznych. Jest to prawdopodobnie spowodowane poziomem skomplikowania tych funkcji - używają one rozkładu LU. Doprowadzenie do postaci trójkątnej jest naturalnie cięższe obliczeniowo niż metoda Sarrusa lub permutacji.
W tabeli przedstawione zostały czasy klasyfikacji zbiorów a oraz b.

	2x2 własny	2x2 biblioteczny	3x3 własny	3x3 biblioteczny
czas klasyfikacji zbioru [s]	0,0326	0,4482	0,0429	0,4239
	0,0258	0,4021	0,0403	0,4366

Tabela 4: Czasy klasyfikacji

7 Wnioski

W zbiorach a oraz c otrzymane zostały te same wyniki niezależnie od wykorzystanych wyznaczników, funkcji do ich liczenia czy też użytej tolerancji. Jest prawdopodobnie spowodowane tym, że w zbiorach tych nie znalazły się punkty współliniowe. Co innego można powiedzieć o zbiorze b, gdzie konsekwentnie funkcje wyznaczników 2x2 różniły się między sobą o 3 inaczej zklasyfikowane punkty. Punkty te nie leżały bezpośrednio na prostej, ale odległość pomiędzy nimi, a prostą była wystarczająco mała, żeby zostały zakwalifikowane jako współliniowe. Pokazuje to nieidealną naturę algorytmiki komputerowej korzystającej z ograniczeń w reprezentacji liczb zmiennoprzecinkowych. W zbiorze d dla obu wyznaczników najbardziej efektywna okazała się tolerancja do zera równa 10^{-10}, aczkolwiek wyznacznik 3x3 bezbłędnie klasyfikował punkty leżące na tej samej prostej przy dużo mniejszej tolerancji - 10^{-14}. Można więc wywnioskować, że jest on bardziej niezawodny i powinien być stosowany przy takich obliczeniach.

Na wykresie 13. można także zauważyć, że przy małych tolerancjach kluczowe są wielkości współrzędnych punktów. Punkty o (bezwzględnie) większych współrzędnych były częściej różnie klasyfikowane. Wynika z tego, że dla punktów o mniejszych współrzędnych (bliżej środka układu współrzędnych) wyznaczniki lepiej obliczały wynik oraz były bardziej zgodne ze sobą.

Ostatecznie najlepiej korzystać z własnego zaimplementowanego wyznacznika 3x3 z powodu lepszej dokładności obliczeń oraz dużo szybszego działania.