

Algorytmy geometryczne

Labolatorium nr 1

Patryk Klatka

27 października 2022

1 Opis ćwiczenia

Głównym celem labolatorium nr 1 jest określanie położenia punktu względem prostej lub odcinka. Taka klasyfikacja punktów może być dokonana za pomocą obliczania wyznaczników:

$$W_1 = \begin{vmatrix} A_x - C_x & A_y - C_y \\ B_x - C_x & B_y - C_y \end{vmatrix}$$
$$W_2 = \begin{vmatrix} A_x & A_y & 1 \\ B_x & B_y & 1 \\ C_x & C_y & 1 \end{vmatrix}$$

gdzie A oraz B to punkty określające prostą/odcinek, a C to badany punkt. Interpretacji położenia punktu po lewej/prawej stronie dokonujemy w bardzo prosty sposób. Gdy analizujemy odcinek, przykładowo \overline{PK} , to możemy sobie wyobrazić, że idziemy po odcinku z punktu P do punktu K . Gdy wyliczając wyznacznik podstawimy za $A = P$ oraz $B = K$ to badany punkt, np. C , będzie znajdował się odpowiednio:

$$\det(W_n) = \begin{cases} > 0 & \text{Punkt leży po lewej stronie} \\ = 0 & \text{Punkt leży na prostej} \\ < 0 & \text{Punkt leży po prawej stronie} \end{cases}, n \in \{1, 2\}$$

Należy pamiętać, że ze względu na narzędzia z których korzystamy, wyniki wyznaczników w postaci zmiennoprzecinkowej mogą różnić się od danego typu wyznacznika.

2 Użyte narzędzia i środowisko pracy

Cały program został napisany w języku Python za pomocą popularnej, interaktywnej platformy Jupyter Notebook. W narzędziu graficznie zostały wykorzystywane biblioteki takie jak `matplotlib` (do graficznego przedstawiania danych) oraz `numpy` (do zmiany precyzji i innych obliczeń). W otrzymanym narzędziu graficznym zostało dokonane parę zmian, aby narzędzie to mogło obsługiwać nowsze edytory, m.in. `VSCoDe`. Dodatkowo, dodana została możliwość tworzenia tytułów wykresów/podwykresów. W celu powtarzalności wyników (dla zgodności z sprawozdaniem) ustawiono ziarno (seed) dla funkcji z biblioteki `np.random`, osobno dla każdego zbioru danych.

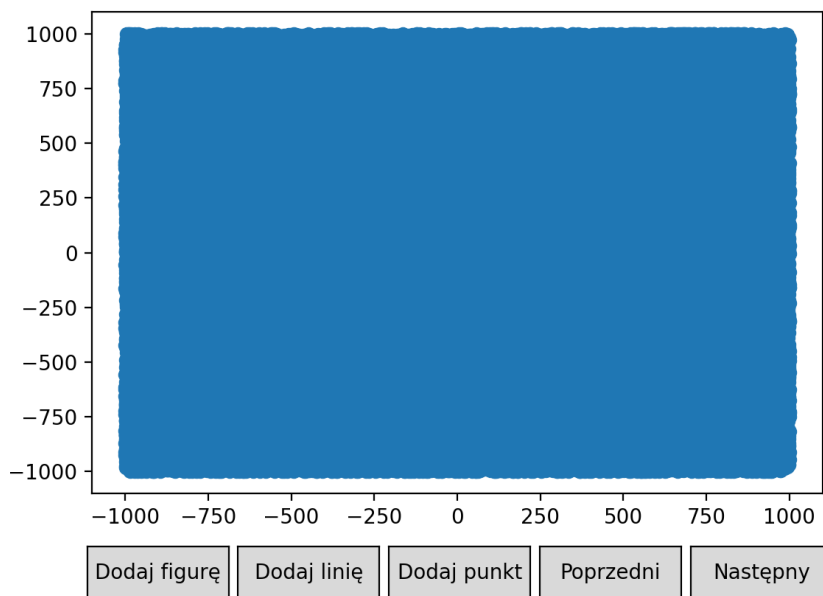
Dane sprzętowe:

- System Windows 10 x64
- Procesor Intel Core i5-8250U
- RAM 8GB
- Wersja Python: 3.10

3 Generacja zestawów punktów

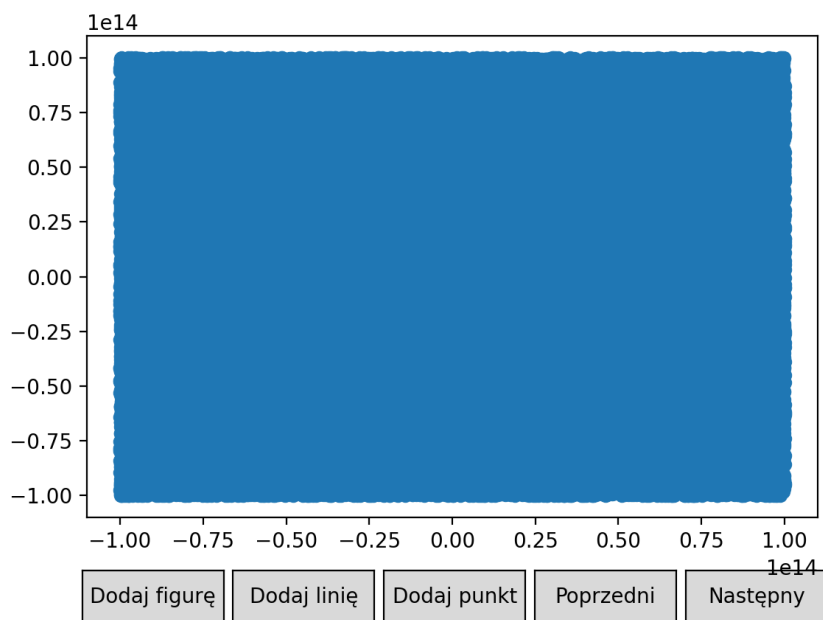
W ramach ćwiczenia należało wygenerować cztery zbiory punktów:

- Zestaw danych A: 10^5 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$



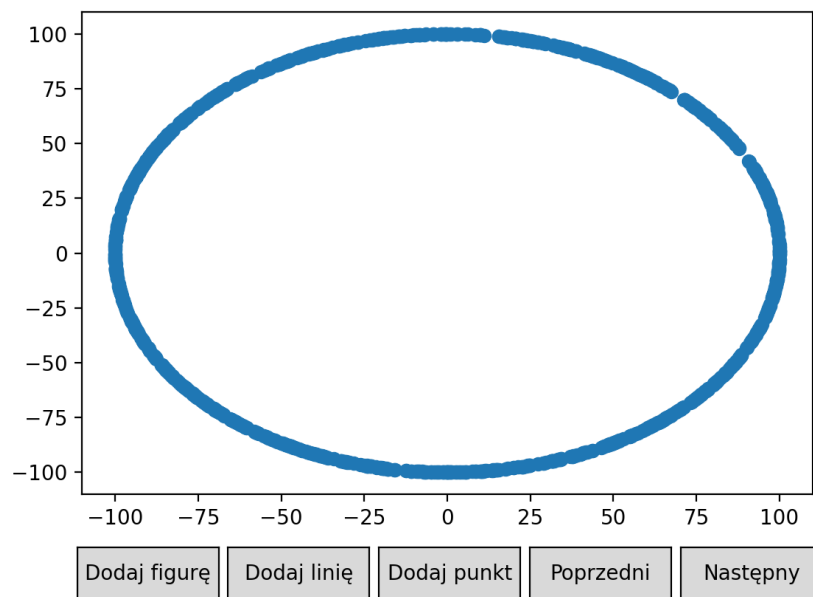
Rysunek 1: Wizualizacja zestawu danych A

- Zestaw danych B: 10^5 losowych punktów o współrzędnych z przedziału $[-10^{14}, 10^{14}]$



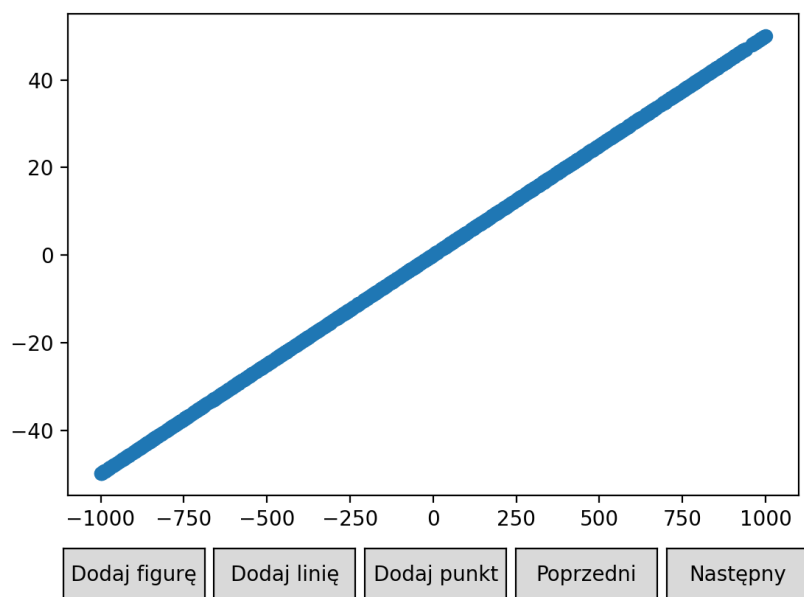
Rysunek 2: Wizualizacja zestawu danych B

- Zestaw danych C: 1000 losowych punktów leżących na okręgu o środku $(0, 0)$ i promieniu $R = 100$



Rysunek 3: Wizualizacja zestawu danych C

- Zestaw danych D: 1000 losowych punktów o współrzędnych z przedziału $[-1000, 1000]$ leżących na prostej wyznaczonej przez wektor \vec{ab} , przyjmując: $a = (-1.0, 0.0)$, $b = (1.0, 0.1)$



Rysunek 4: Wizualizacja zestawu danych D

Powyższe wizualizacje zbiorów danych przedstawiają punkty wygenerowane za pomocą funkcji

`np.uniform` z domyślną precyzją `np.float64`. Wykorzystana została biblioteka `numpy`, w celu poprawienia efektywności programu i prędkości generowania zbiorów. Dla precyzji `np.float32` wizualizacje wyglądają identycznie.

4 Klasyfikacja punktów dla precyzji `np.float64`

Do klasyfikacji punktów zostały wykorzystane własne implementacje wyznaczników, które wykorzystują metodę Sarrusa oraz wyznaczniki dostępne w bibliotece `numpy`. Odpowiadają za to funkcje `det2x2` oraz `det3x3` z odpowiednim argumentem `numpy=True` (do wykonania obliczeń za pomocą biblioteki `numpy`) lub `numpy=False`.

Aby poprawić czytelność kodu został zaimplementowany szereg funkcji pomocniczych. Za klasyfikację punktów odpowiada funkcja `classify_points`, która pozwala klasyfikować punkty z dokładnością do podanej tolerancji dla zera. Funkcja zwraca sklasyfikowane punkty, które są rozszerzone o informacje po której stronie odcinka są oraz listę z punktami sklasyfikowanymi po lewej, prawej stronie odcinka lub na odcinku. Punkty współliniowe są klasyfikowane z tolerancjami: `1e-18`, `1e-14`, `1e-10`, `1e-06`.

Legenda kolorów:

- czerwony - punkt jest współliniowy
- niebieski - punkt znajduje się po lewej stronie
- zielony - punkt znajduje się po prawej stronie

Funkcja `print_classified_points` odpowiada za wygenerowanie scen do wykresu oraz do stworzenia listy z danymi potrzebnej do utworzenia tabeli porównującej dane wyznaczniki.

Aby pokazać tabelę porównującą dane wyznaczniki wraz z odpowiednimi tolerancjami zera należy wywołać funkcję `print_classified_points_table`, która przyjmuje dane z `print_classified_points`.

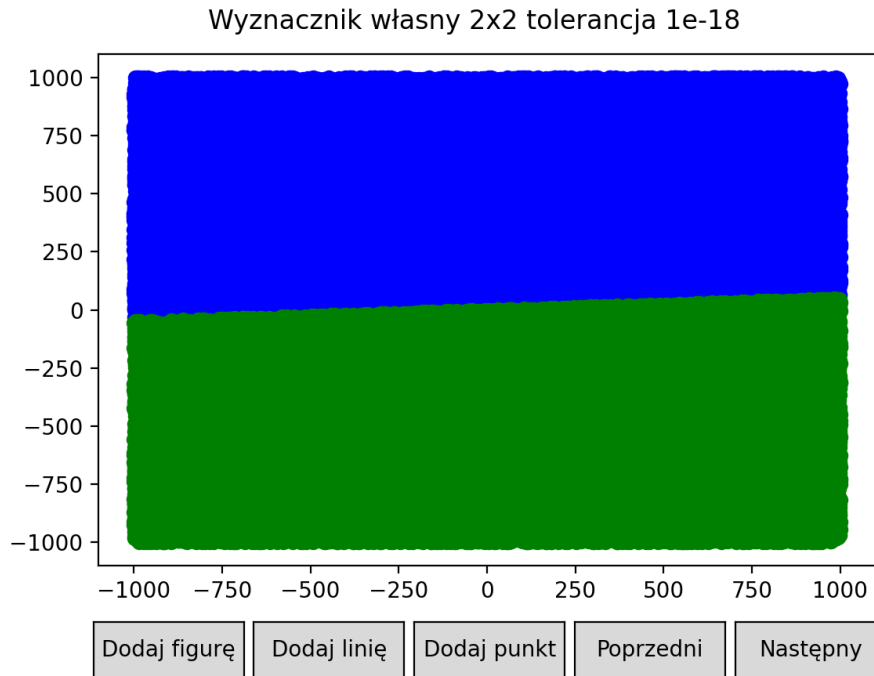
Wykres korzysta ze scen (klasa `Scene`). Aby zmienić wykres danej klasyfikacji punktów należy kliknąć odpowiednio `Następny` lub `Poprzedni`. Po kliknięciu przycisku tytuł wykresu powinien zostać zmieniony.

4.1 Klasyfikacja punktów dla zestawu danych A

Dla zestawu danych A wszystkie punkty zostały sklasyfikowane tak samo, bez względu na tolerancję zera.

Tolerancja	Wyznacznik	Punkty		
		Po lewej	Po prawej	Współliniowe
Wszystkie badane	Wszystkie badane	50192	49808	0

Tabela 1: Tabela klasyfikacji punktów zestawu danych A



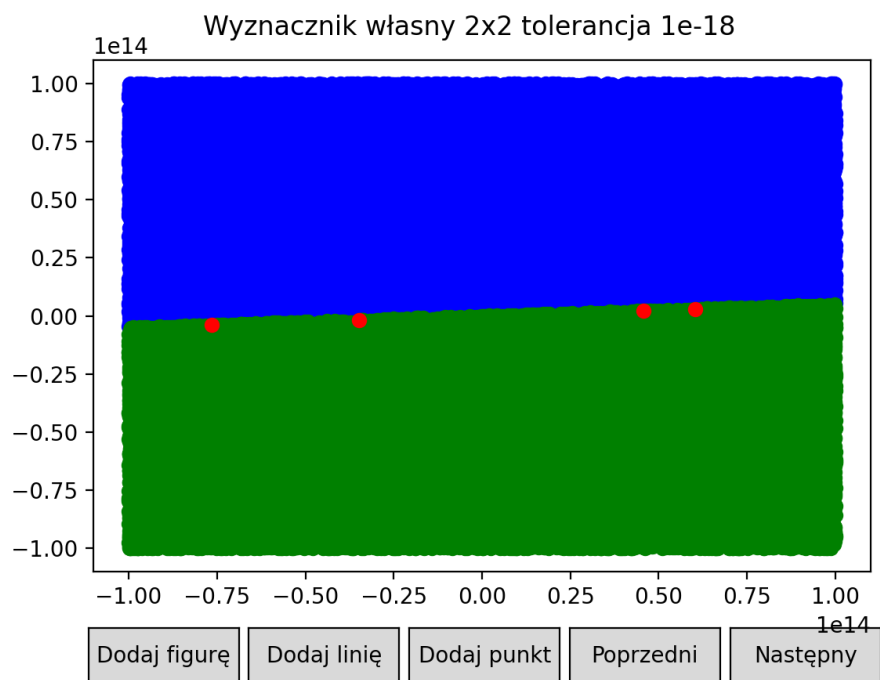
Rysunek 5: Klasyfikacja punktów zestawu danych A

4.2 Klasyfikacja punktów dla zestawu danych B

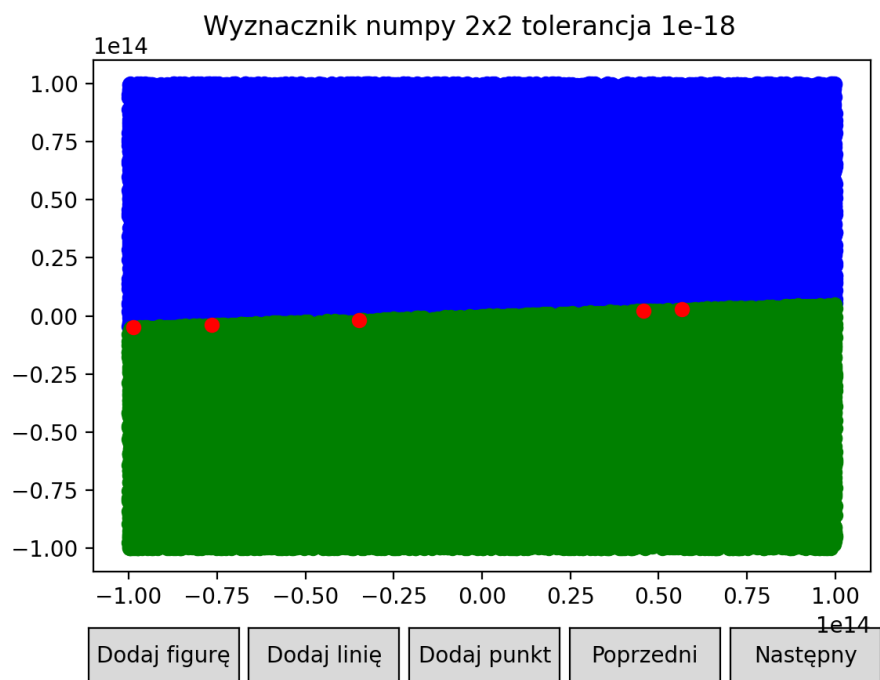
Dla zestawu danych B punkty zostały różnie sklasyfikowane dla wyznaczników, ale tak samo dla różnych tolerancji zera. W przeciwieństwie do klasyfikacji zestawu danych A, wyznaczniki zakwalifikowały niektóre punkty jako współliniowe, co może być spowodowane dużym rozmiarem współrzędnych punktów jak i błędami obliczeniowymi związane z precyzją operacji arytmetycznych.

Tolerancja	Wyznacznik	Punkty		
		Po lewej	Po prawej	Współliniowe
Wszystkie badane	Wyznacznik własny 2x2	49961	50035	4
	Wyznacznik numpy 2x2	49961	50034	5
	Wyznacznik własny 3x3	49964	50036	0
	Wyznacznik numpy 3x3	49964	50036	0

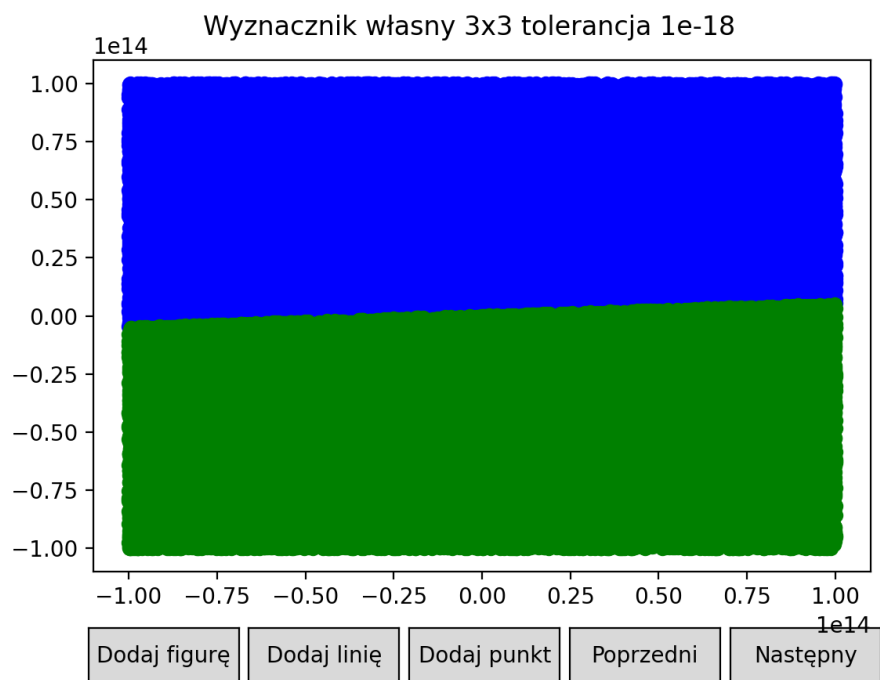
Tabela 2: Tabela klasyfikacji punktów zestawu danych B



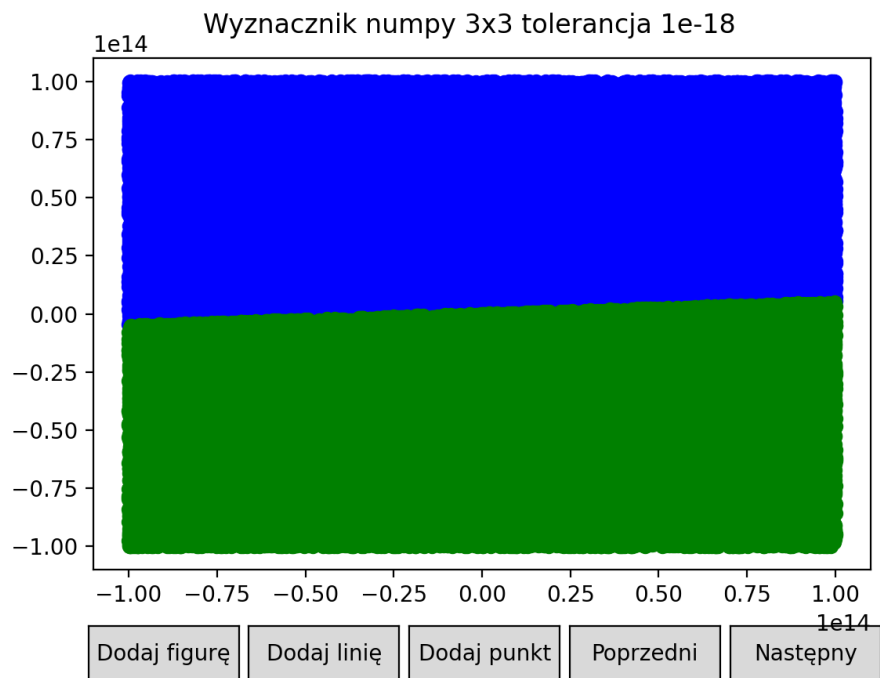
Rysunek 6: Klasyfikacja punktów zestawu danych B



Rysunek 7: Klasyfikacja punktów zestawu danych B



Rysunek 8: Klasyfikacja punktów zestawu danych B



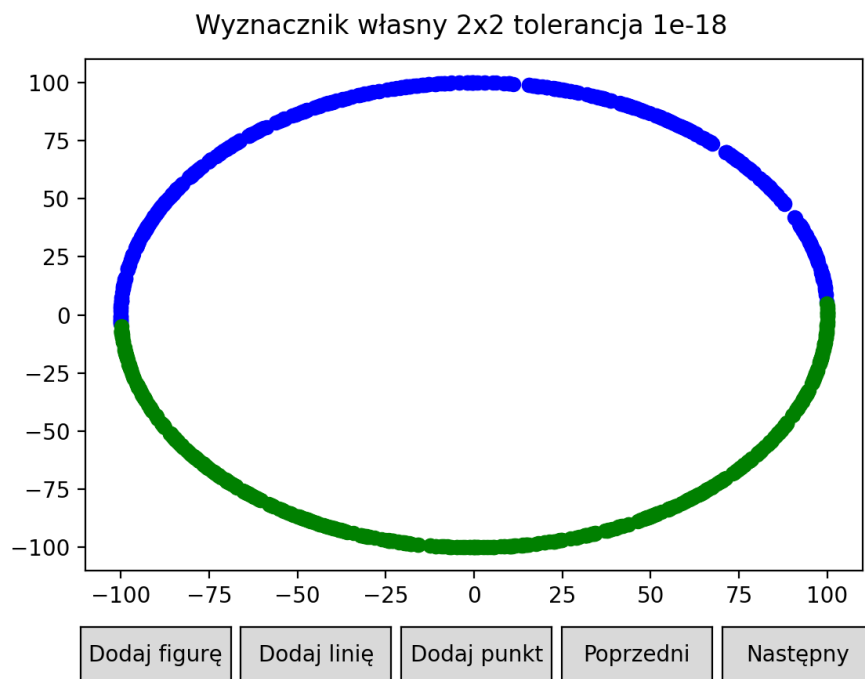
Rysunek 9: Klasyfikacja punktów zestawu danych B

4.3 Klasyfikacja punktów dla zestawu danych C

Dla zestawu danych C sytuacja analogiczna jak dla zestawu danych A.

Tolerancja	Wyznacznik	Punkty		
		Po lewej	Po prawej	Współliniowe
Wszystkie badane	Wszystke badane	452	548	0

Tabela 3: Tabela klasyfikacji punktów zestawu danych C



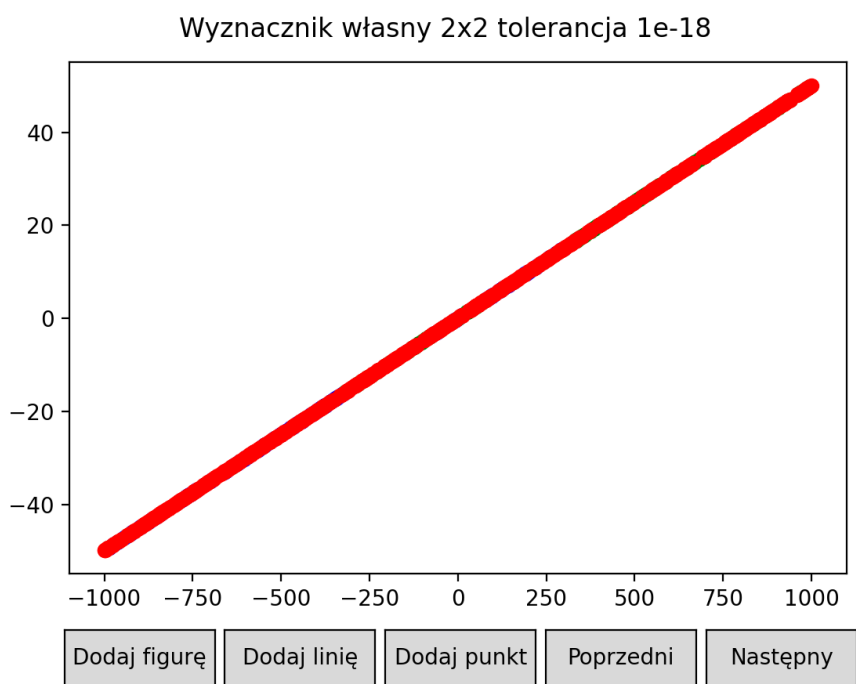
Rysunek 10: Klasyfikacja punktów zestawu danych C

4.4 Klasyfikacja punktów dla zestawu danych D

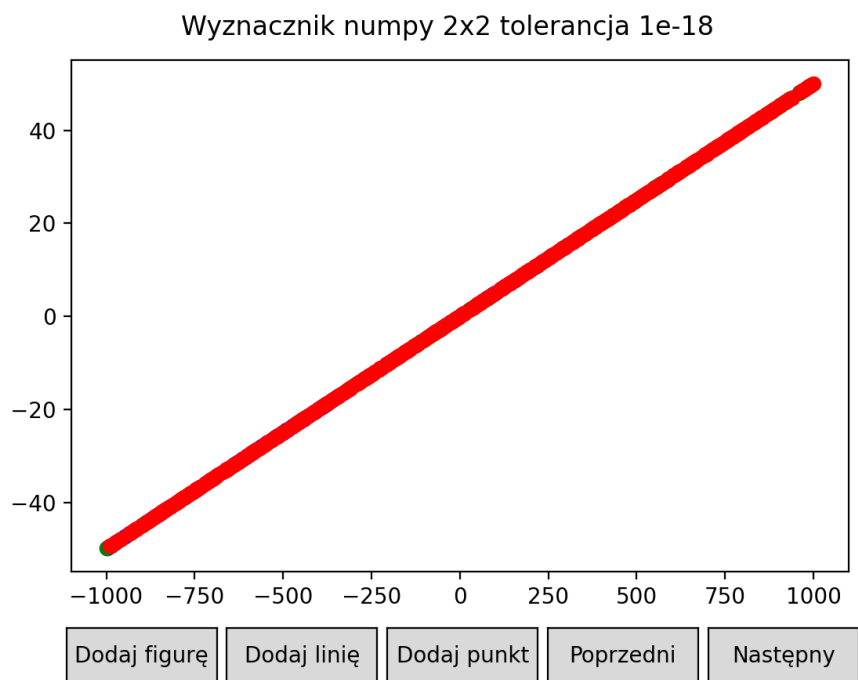
Dla zestawu danych D klasyfikacja punktów jest najbardziej różnorodna ze wszystkich powyższych zestawień. Najlepiej pokazują to tolerancje $1e-18$ oraz $1e-14$, natomiast dla tolerancji $1e-10$ oraz $1e-06$ klasyfikacje są identyczne. Najciekawszym przypadkiem jest klasyfikacja punktów przez wyznacznik własny 3×3 z tolerancją $1e-18$ oraz $1e-14$, który sklasyfikował punkty inaczej, niż inne wyznaczniki. Powodem takiej klasyfikacji mogą być błędy ze skończoną precyzją obliczeń.

Tolerancja	Wyznacznik	Punkty		
		Po lewej	Po prawej	Współliniowe
1e-18	Wyznacznik własny 2x2	145	130	725
	Wyznacznik numpy 2x2	153	149	698
	Wyznacznik własny 3x3	175	401	424
	Wyznacznik numpy 3x3	348	341	311
1e-14	Wyznacznik własny 2x2	136	126	738
	Wyznacznik numpy 2x2	146	141	713
	Wyznacznik własny 3x3	0	0	1000
	Wyznacznik numpy 3x3	10	89	901
1e-10 oraz 1e-06	Wyznacznik własny 2x2	0	0	1000
	Wyznacznik numpy 2x2	0	0	1000
	Wyznacznik własny 3x3	0	0	1000
	Wyznacznik numpy 3x3	0	0	1000

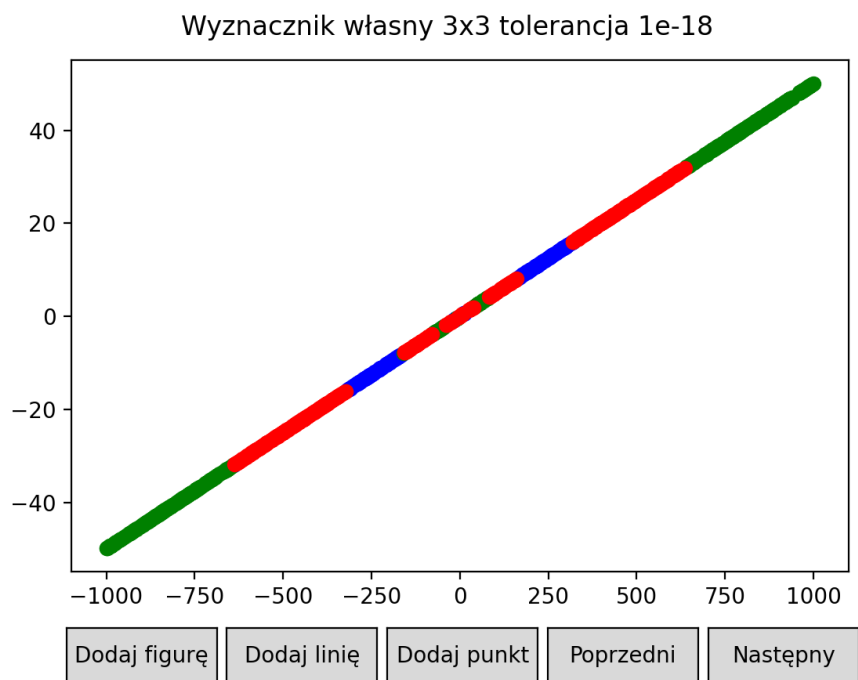
Tabela 4: Tabela klasyfikacji punktów zestawu danych D



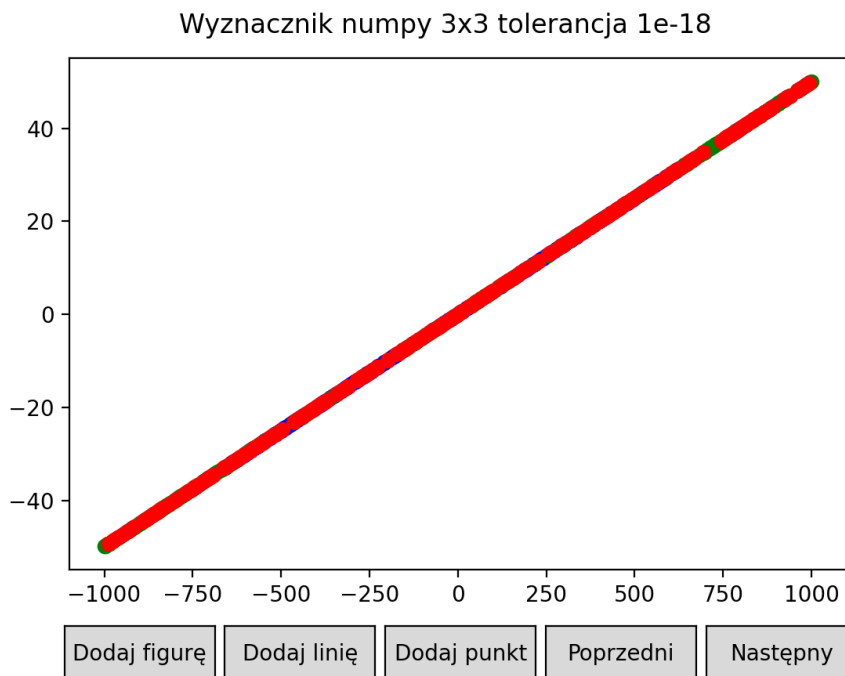
Rysunek 11: Klasyfikacja punktów zestawu danych D



Rysunek 12: Klasyfikacja punktów zestawu danych D



Rysunek 13: Klasyfikacja punktów zestawu danych D



Rysunek 14: Klasyfikacja punktów zestawu danych D

5 Porównanie różnic w klasyfikacji punktów dla precyzji `np.float64`

Również dla porównywania różnic w klasyfikacji zostały zaimplementowane funkcje pomocnicze. Funkcja `compare_points`, pozwala na porównanie punktów, które zwróciła funkcja `classify_points`, dla danego zestawu danych. Funkcja zwraca gotową listę punktów typu `PointsCollection`.

Aby wygenerować sceny oraz dane do tabeli porównującej różnice w klasyfikacji punktów należy użyć funkcji `print_compare_points`.

Tabela wypisująca ile punktów zostało inaczej zakwalifikowanych jest zaimplementowana w funkcji `print_compared_points_table`.

Wykresy korzystają ze scen (klasa `Scene`).

5.1 Różnice klasyfikacji dla zestawu danych A oraz C

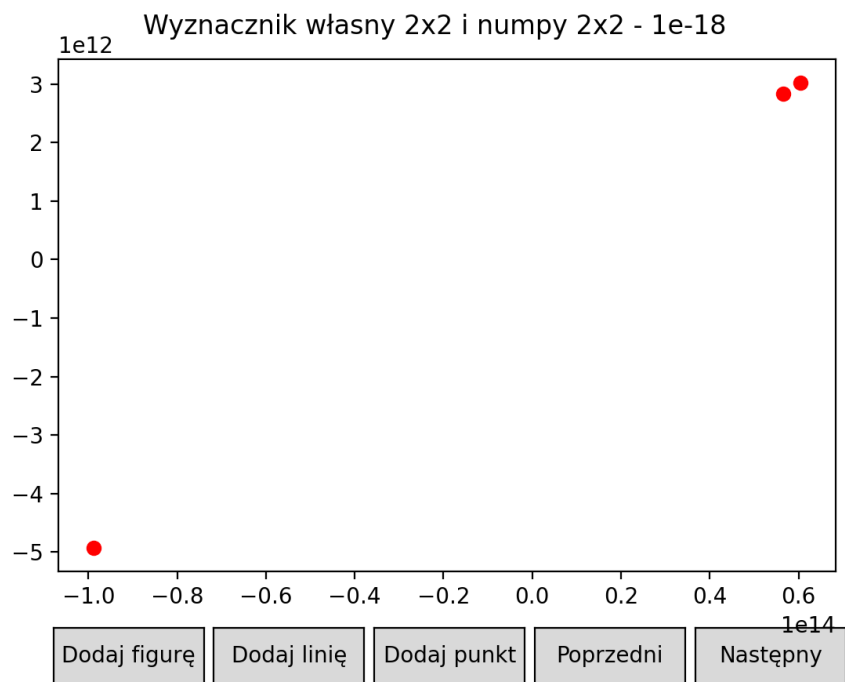
Dla zestawów A oraz C nie zauważono żadnych różnic w klasyfikacji punktów.

5.2 Różnice klasyfikacji dla zestawu danych B

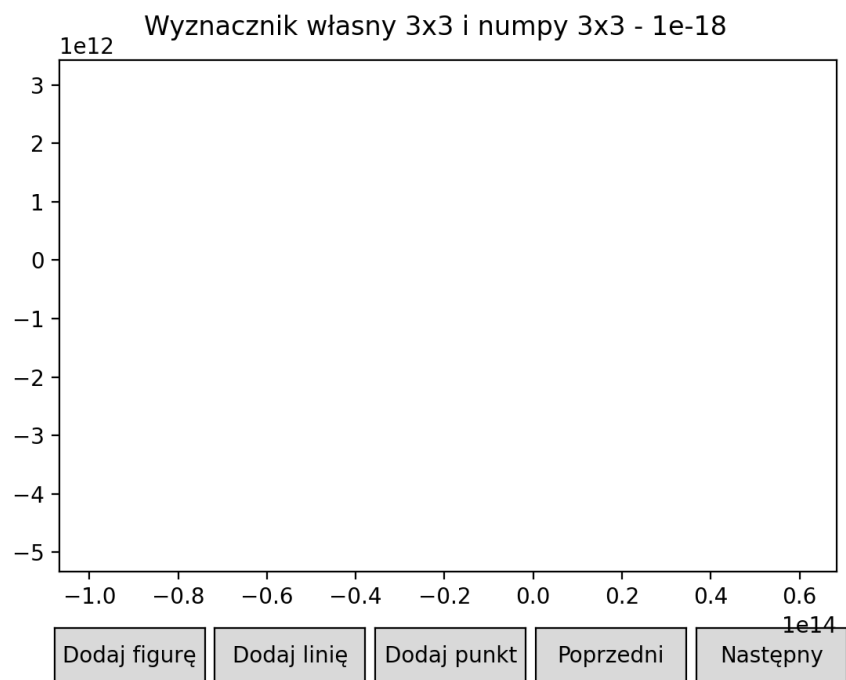
W przypadku zestawu danych B zauważono, w niektórych przypadkach, różnice w klasyfikacji punktów. Na poniższych wizualizacjach inaczej zakwalifikowanych punktów możemy wywnioskować, że są to punkty, które mają dość duże wielkości współrzędnych, co może powodować błędy (związane z precyzją) podczas obliczeń.

Tolerancja	Kombinacja wyznaczników	Różnica
Wszystkie badane	Wyznacznik własny 2x2 i numpy 2x2	3
	Wyznacznik własny 3x3 i numpy 3x3	0
	Wyznacznik własny 2x2 i numpy 3x3	5
	Wyznacznik własny 3x3 i numpy 2x2	5
	Wyznacznik własny 2x2 i własny 3x3	5
	Wyznacznik numpy 2x2 i numpy 3x3	5

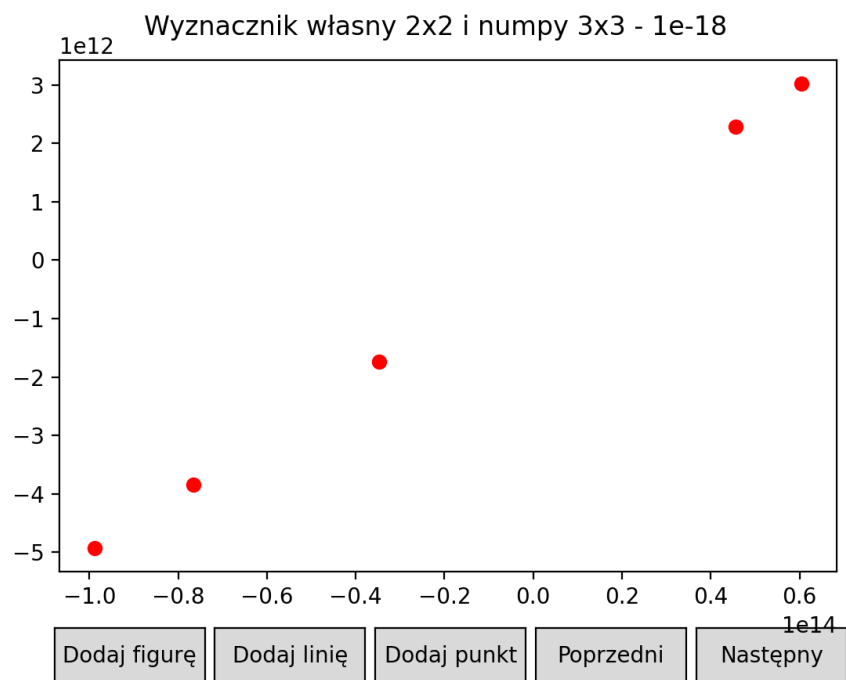
Tabela 5: Tabela różnic klasyfikacji punktów zestawu danych B



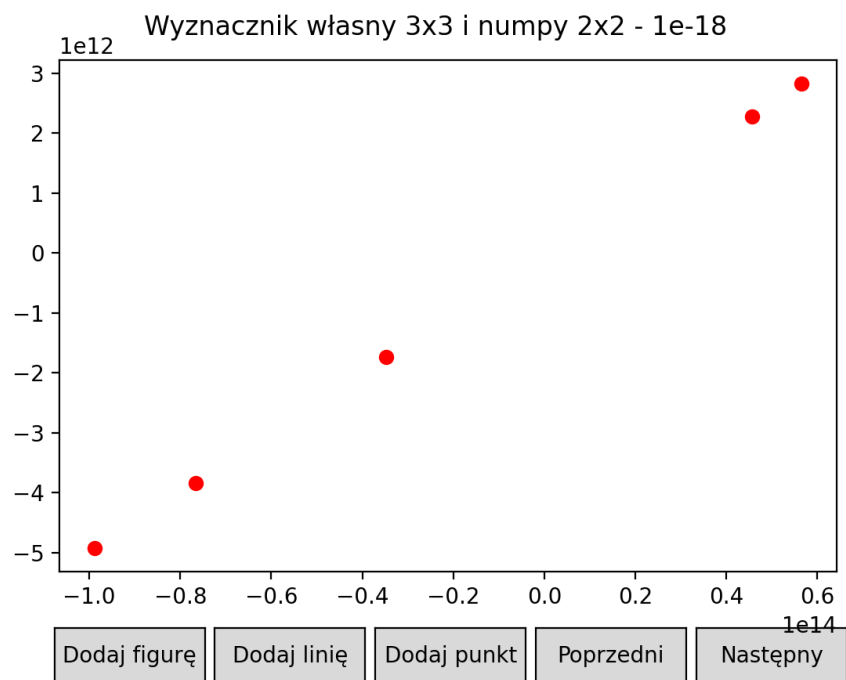
Rysunek 15: Różnice w klasyfikacji punktów zestawu danych B



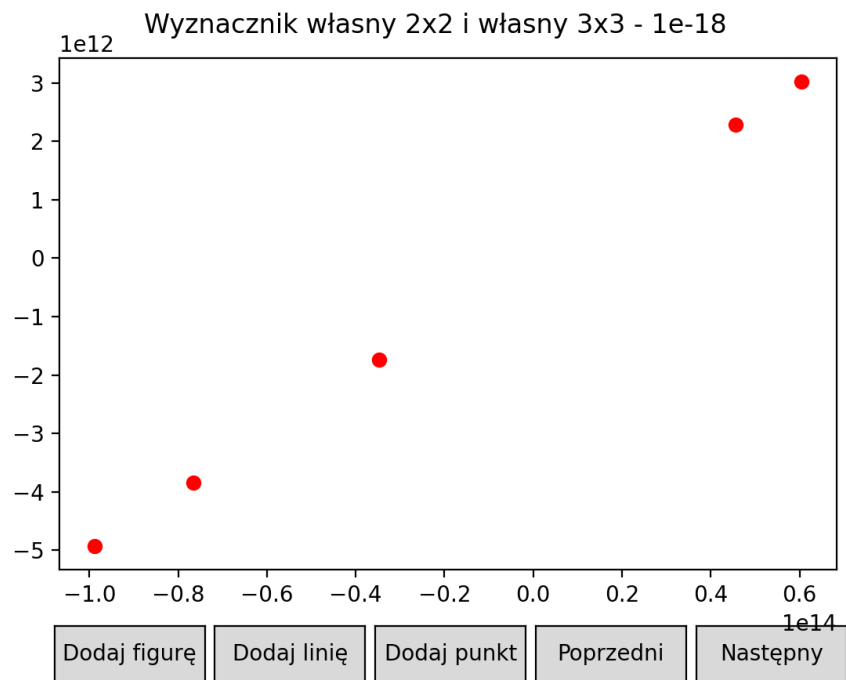
Rysunek 16: Różnice w klasyfikacji punktów zestawu danych B



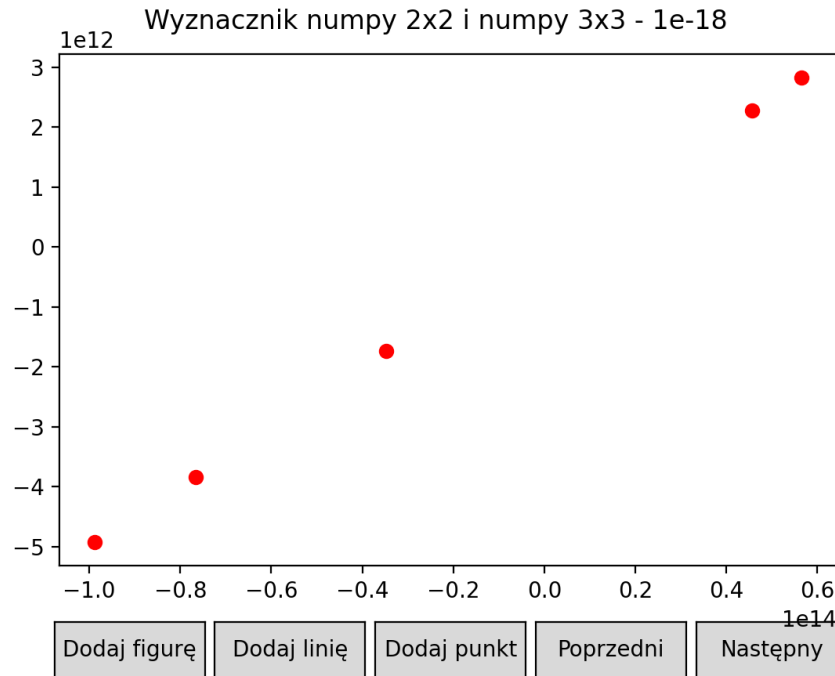
Rysunek 17: Różnice w klasyfikacji punktów zestawu danych B



Rysunek 18: Różnice w klasyfikacji punktów zestawu danych B



Rysunek 19: Różnice w klasyfikacji punktów zestawu danych B



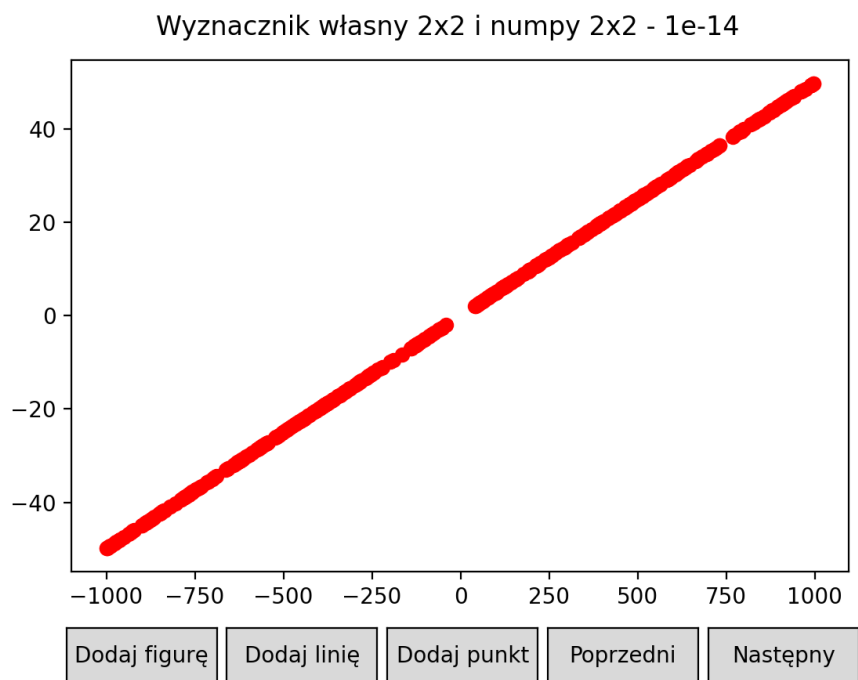
Rysunek 20: Różnice w klasyfikacji punktów zestawu danych B

5.3 Różnice klasyfikacji dla zestawu danych D

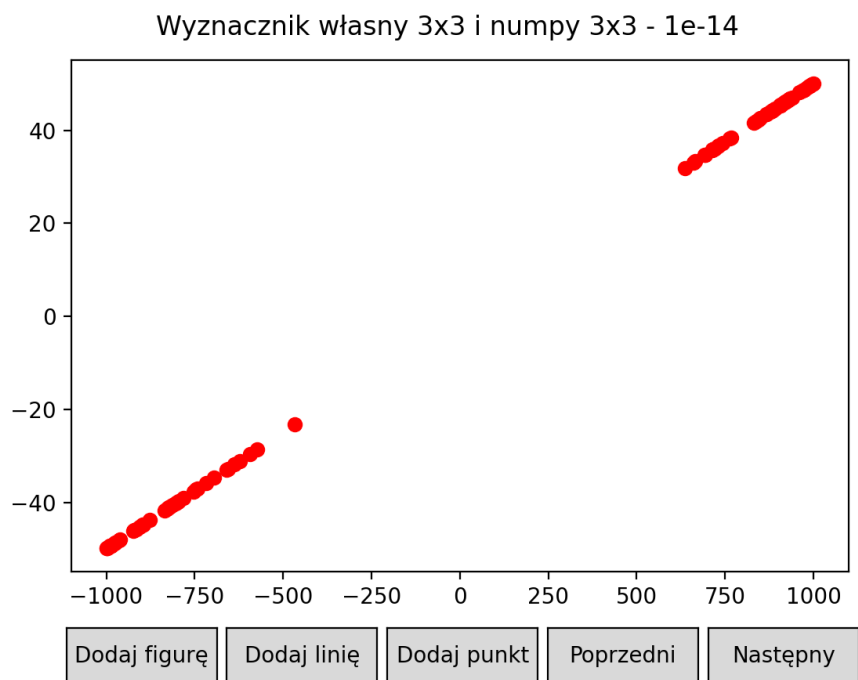
Dla zestawu danych D zauważono największe rozbieżności w klasyfikacji punktów. Ponieważ punkty są bardzo blisko badanej prostej oraz uwzględniając błędy obliczeniowe, to dla mniejszych tolerancji punkty te rzadko klasyfikują się jako współliniowe, przez co te różnice sięgają nawet 69% badanych punktów.

Tolerancja	Kombinacja wyznaczników	Różnica
1e-18	Wyznacznik własny 2x2 i numpy 2x2	407
	Wyznacznik własny 3x3 i numpy 3x3	526
	Wyznacznik własny 2x2 i numpy 3x3	682
	Wyznacznik własny 3x3 i numpy 2x2	674
	Wyznacznik własny 2x2 i własny 3x3	687
	Wyznacznik numpy 2x2 i numpy 3x3	686
1e-14	Wyznacznik własny 2x2 i numpy 2x2	391
	Wyznacznik własny 3x3 i numpy 3x3	99
	Wyznacznik własny 2x2 i numpy 3x3	333
	Wyznacznik własny 3x3 i numpy 2x2	287
	Wyznacznik własny 2x2 i własny 3x3	262
	Wyznacznik numpy 2x2 i numpy 3x3	339
1e-10 oraz 1e-06	Wyznacznik własny 2x2 i numpy 2x2	0
	Wyznacznik własny 3x3 i numpy 3x3	0
	Wyznacznik własny 2x2 i numpy 3x3	0
	Wyznacznik własny 3x3 i numpy 2x2	0
	Wyznacznik własny 2x2 i własny 3x3	0
	Wyznacznik numpy 2x2 i numpy 3x3	0

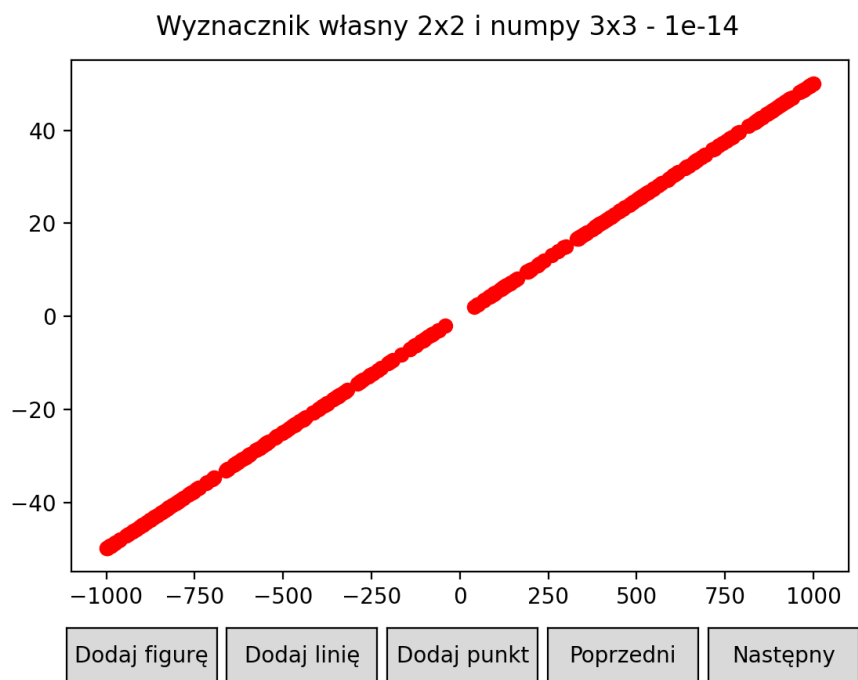
Tabela 6: Tabela różnic klasyfikacji punktów zestawu danych D



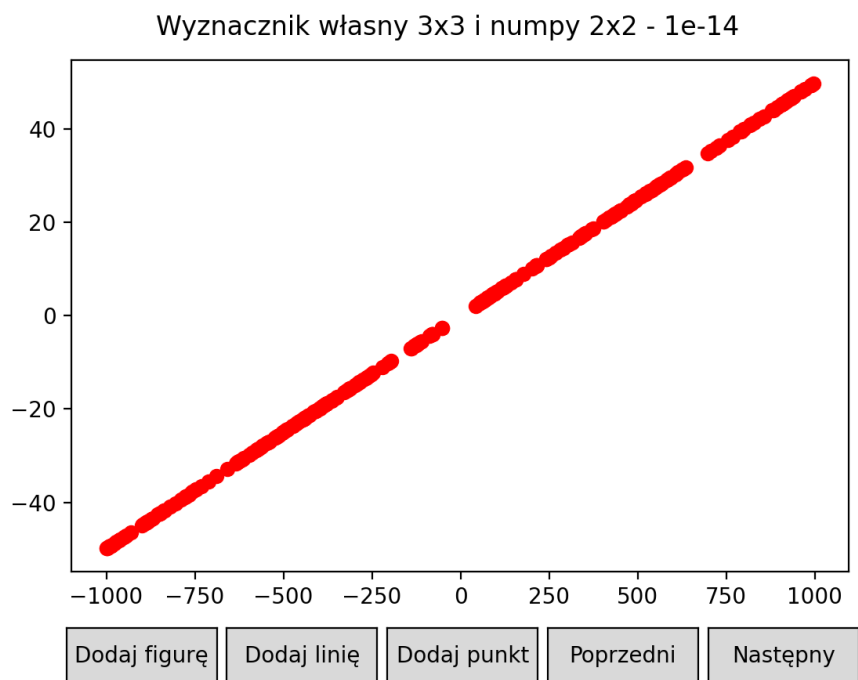
Rysunek 21: Różnice w klasyfikacji punktów zestawu danych D



Rysunek 22: Różnice w klasyfikacji punktów zestawu danych D



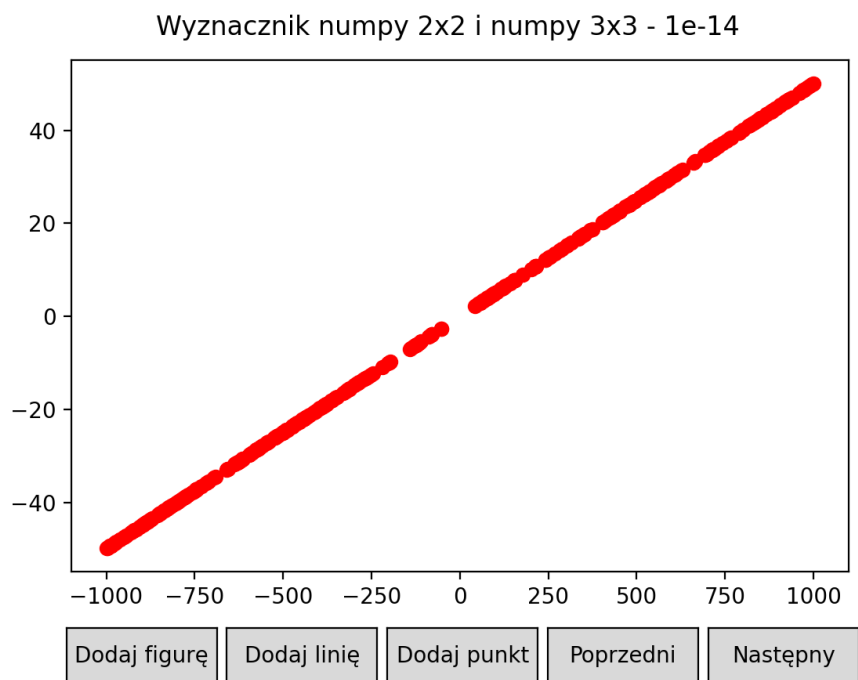
Rysunek 23: Różnice w klasyfikacji punktów zestawu danych D



Rysunek 24: Różnice w klasyfikacji punktów zestawu danych D



Rysunek 25: Różnice w klasyfikacji punktów zestawu danych D



Rysunek 26: Różnice w klasyfikacji punktów zestawu danych D

6 Klasyfikacja punktów dla precyzji np.float32

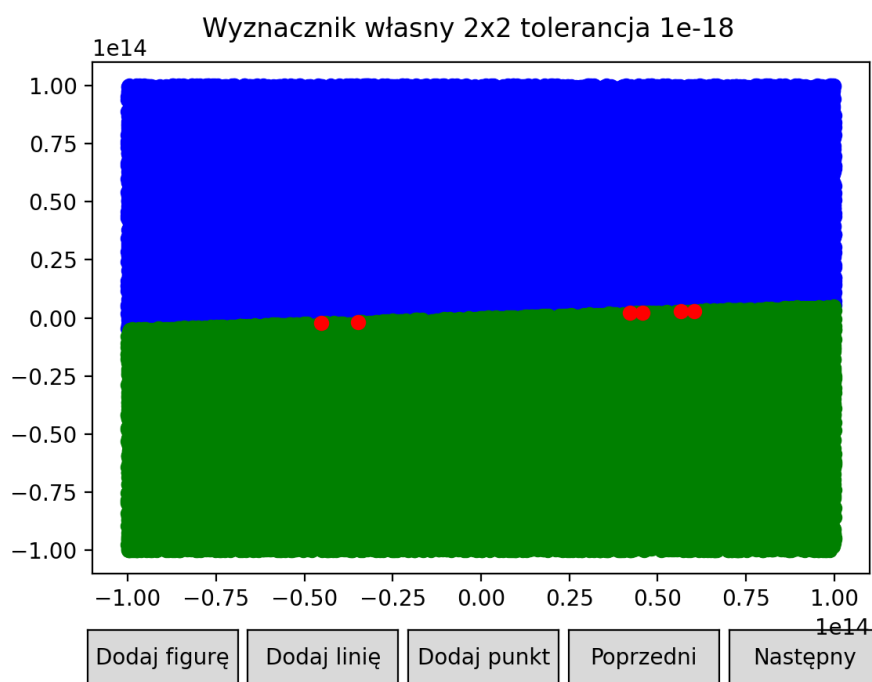
W ramach testów, dla każdego zestawu danych, precyzja została zmniejszona do precyzji `np.float32`. Różnice w klasyfikacjach punktów są najbardziej widoczne w zestawach danych B oraz D. Dla zestawów danych A oraz C nie zauważono różnic.

6.1 Klasyfikacja punktów dla zestawu danych B

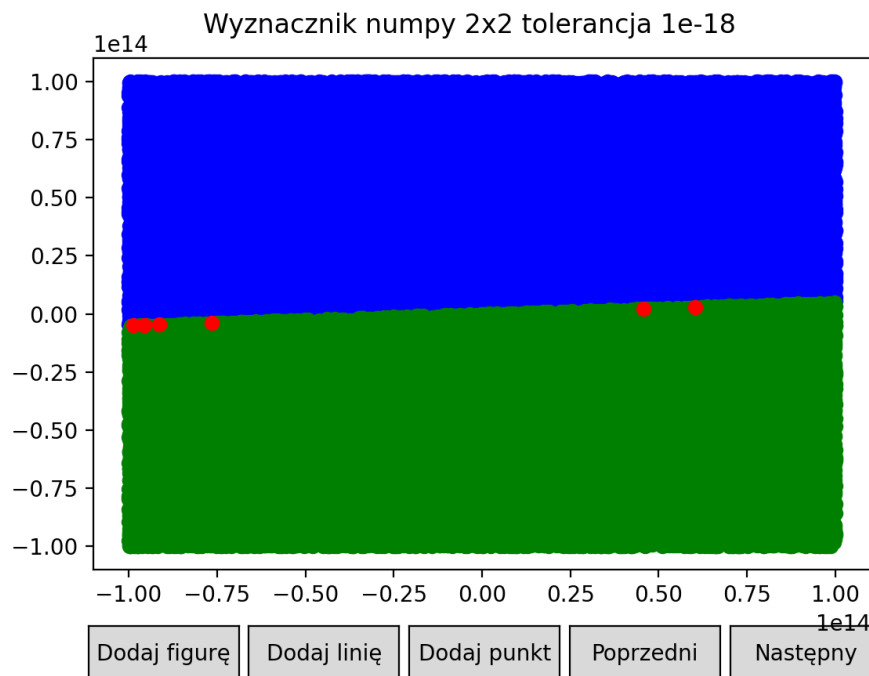
W porównaniu z zestawem danych B z precyzją `np.float64` więcej punktów zostało sklasyfikowanych jako współliniowe.

Tolerancja	Wyznacznik	Punkty		
		Po lewej	Po prawej	Współliniowe
Wszystkie badane	Wyznacznik własny 2x2	49958	50036	6
	Wyznacznik numpy 2x2	49960	50034	6
	Wyznacznik własny 3x3	49964	50036	0
	Wyznacznik numpy 3x3	49964	50036	0

Tabela 7: Tabela klasyfikacji punktów dla zestawu danych B z precyzją float32



Rysunek 27: Klasyfikacja punktów zestawu danych B z precyzją float32



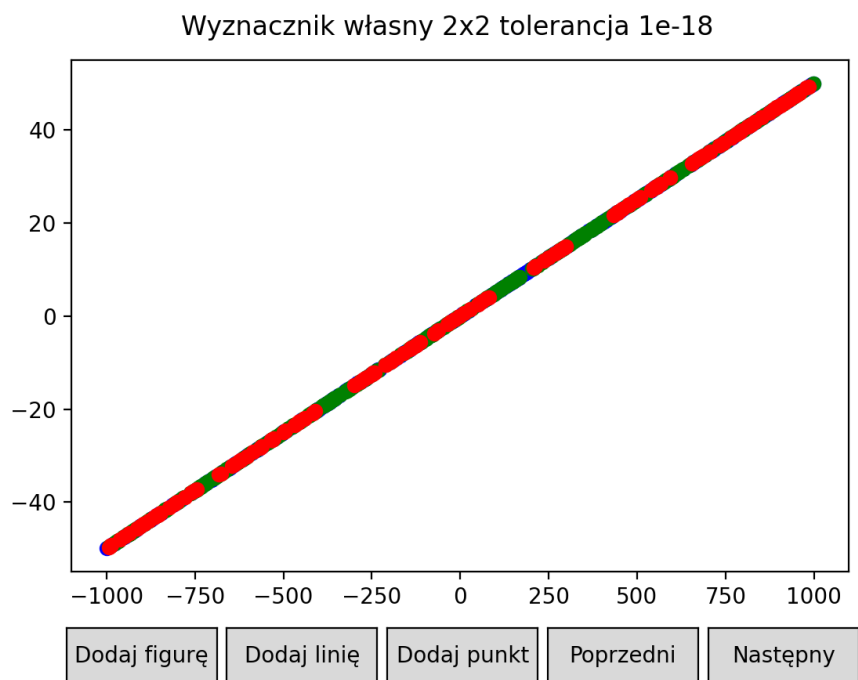
Rysunek 28: Klasyfikacja punktów zestawu danych B z precyzją float32

6.2 Klasyfikacja punktów dla zestawu danych D

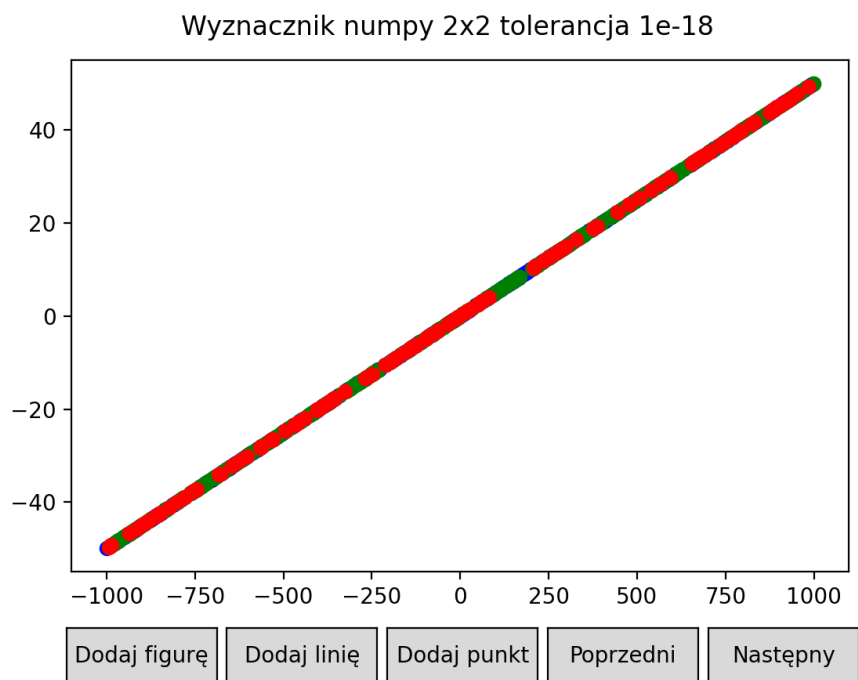
Dla zestawu danych D, liczba punktów zakwalifikowanych jako współliniowe została zmniejszona. Nawet zastosowanie tolerancji dla zera 1e-06 nie pozwoliło na sklasyfikowanie więcej niż 50% punktów jako współliniowych.

Tolerancja	Wyznacznik	Punkty		
		Po lewej	Po prawej	Współliniowe
1e-18	Wyznacznik własny 2x2	424	426	150
	Wyznacznik numpy 2x2	439	438	123
	Wyznacznik własny 3x3	463	491	46
	Wyznacznik numpy 3x3	473	467	60
1e-14	Wyznacznik własny 2x2	424	425	151
	Wyznacznik numpy 2x2	437	436	127
	Wyznacznik własny 3x3	409	417	174
	Wyznacznik numpy 3x3	411	424	165
1e-10	Wyznacznik własny 2x2	409	417	174
	Wyznacznik numpy 2x2	409	417	174
	Wyznacznik własny 3x3	409	417	174
	Wyznacznik numpy 3x3	409	417	174
1e-06	Wyznacznik własny 2x2	252	246	502
	Wyznacznik numpy 2x2	252	246	502
	Wyznacznik własny 3x3	252	246	502
	Wyznacznik numpy 3x3	252	246	502

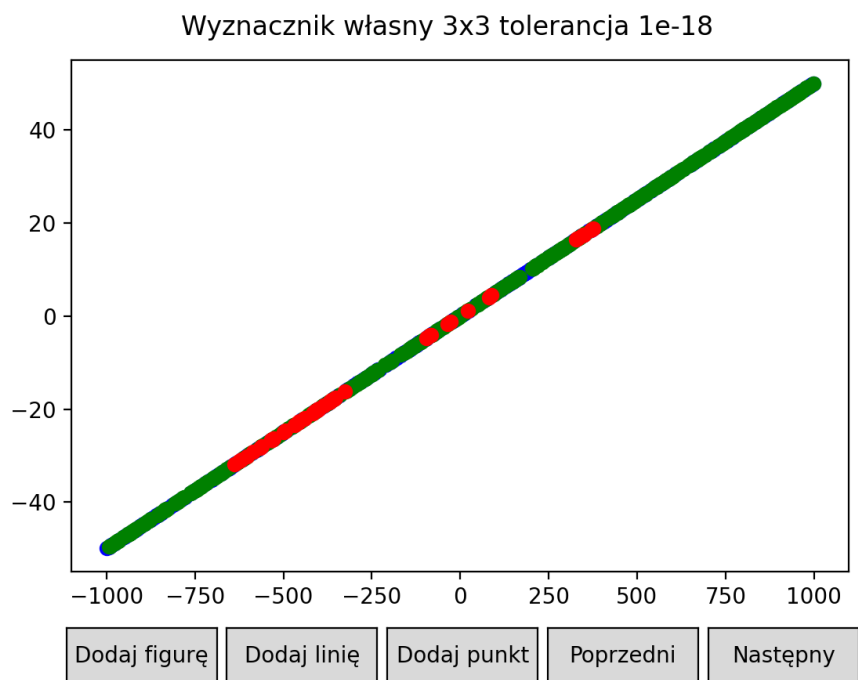
Tabela 8: Tabela klasyfikacji punktów zestawu danych D z precyzją float32



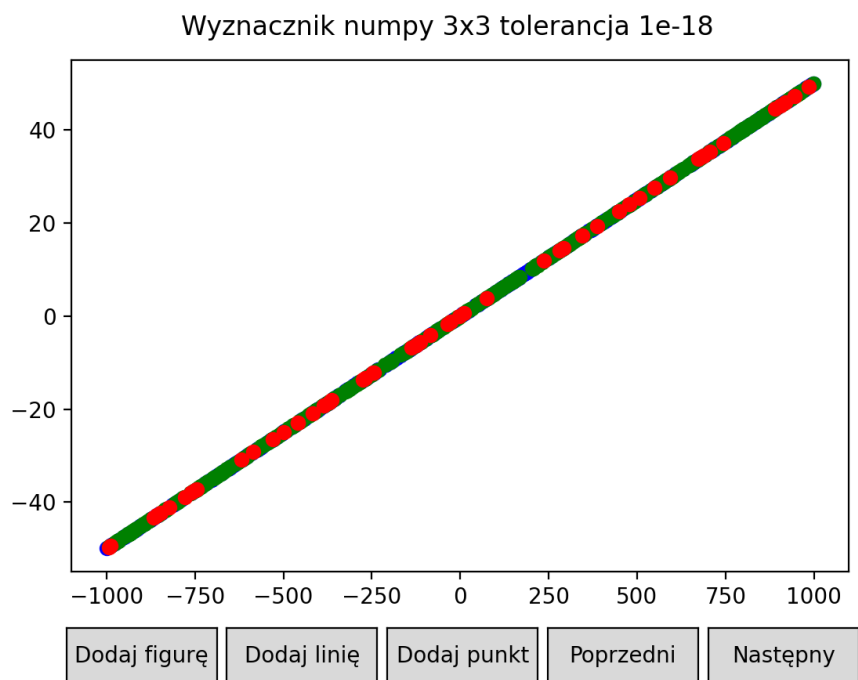
Rysunek 29: Klasyfikacja punktów zestawu danych D z precyzją float32



Rysunek 30: Klasyfikacja punktów zestawu danych D z precyzją float32



Rysunek 31: Klasyfikacja punktów zestawu danych D z precyzją float32



Rysunek 32: Klasyfikacja punktów zestawu danych D z precyzją float32

7 Porównanie różnic w klasyfikacji punktów dla precyzji np.float32

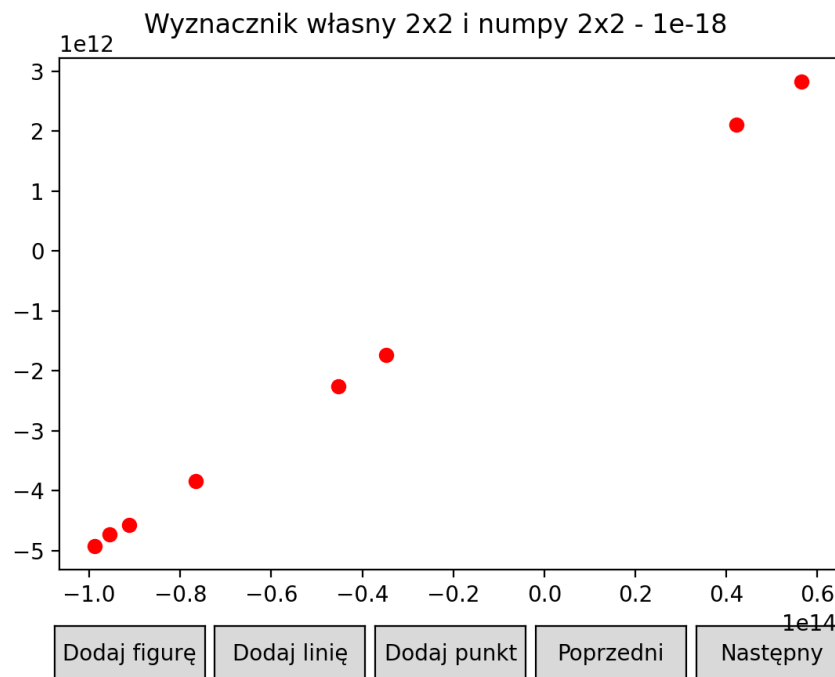
Dla precyzji `np.float32` będą rozważane zestawy danych B oraz D, ponieważ tylko dla tych zestawów zauważono rozbieżności w klasyfikacjach punktów.

7.1 Różnice klasyfikacji dla zestawu B precyzja float32

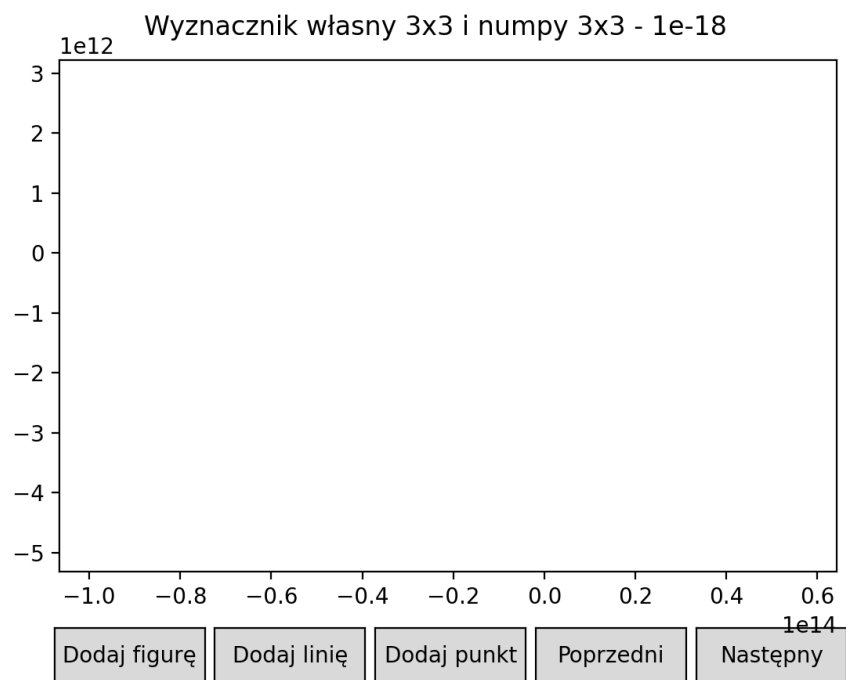
Analogicznie jak dla zestawu danych B z precyzją `np.float64` zauważono różnice w klasyfikacji punktów, które są spowodowane dużymi wielkościami współrzędnych punktów oraz błędami obliczeniowymi związane z precyzją.

Tolerancja	Kombinacja wyznaczników	Różnica
Wszystkie badane	Wyznacznik własny 2x2 i numpy 2x2	8
	Wyznacznik własny 3x3 i numpy 3x3	0
	Wyznacznik własny 2x2 i numpy 3x3	7
	Wyznacznik własny 3x3 i numpy 2x2	6
	Wyznacznik własny 2x2 i własny 3x3	7
	Wyznacznik numpy 2x2 i numpy 3x3	6

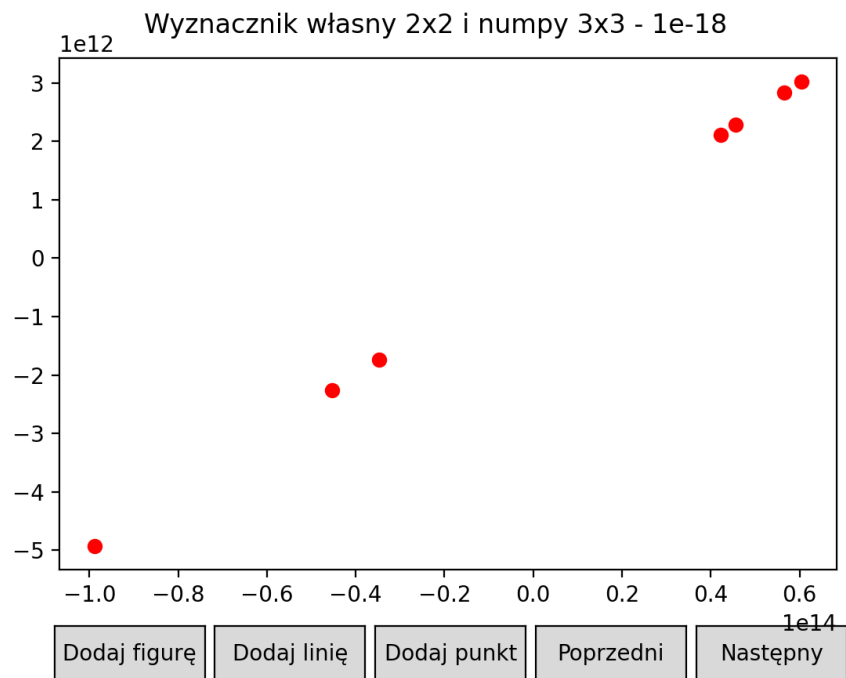
Tabela 9: Tabela różnic klasyfikacji punktów zestawu danych B z precyzją float32



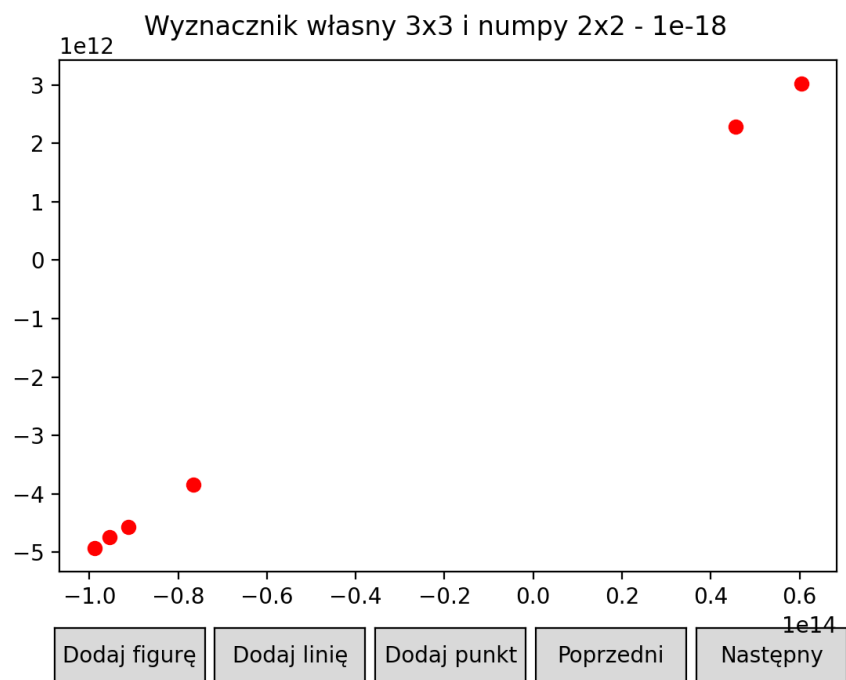
Rysunek 33: Różnice w klasyfikacji punktów zestawu danych B z precyzją float32



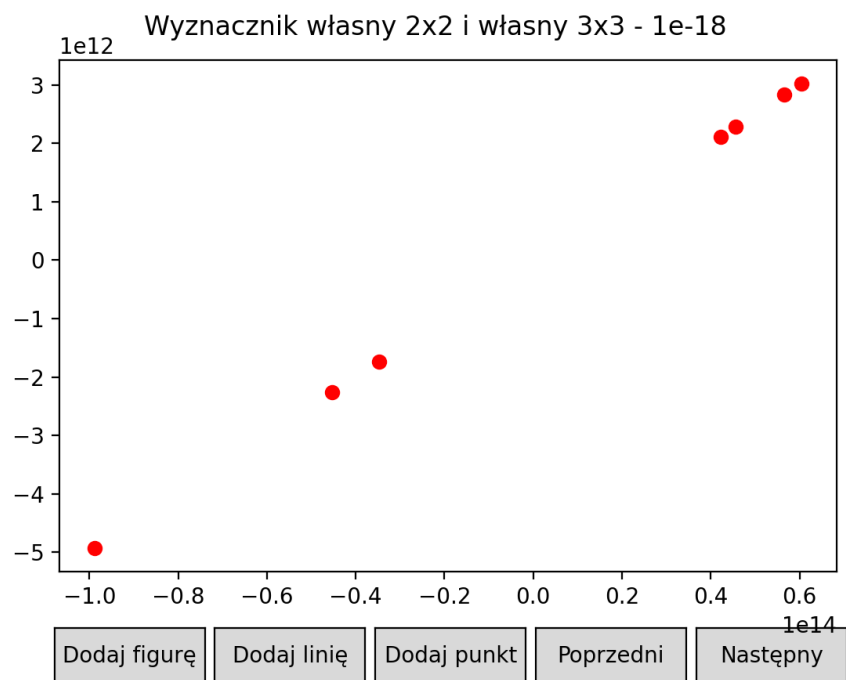
Rysunek 34: Różnice w klasyfikacji punktów zestawu danych B z precyzją float32



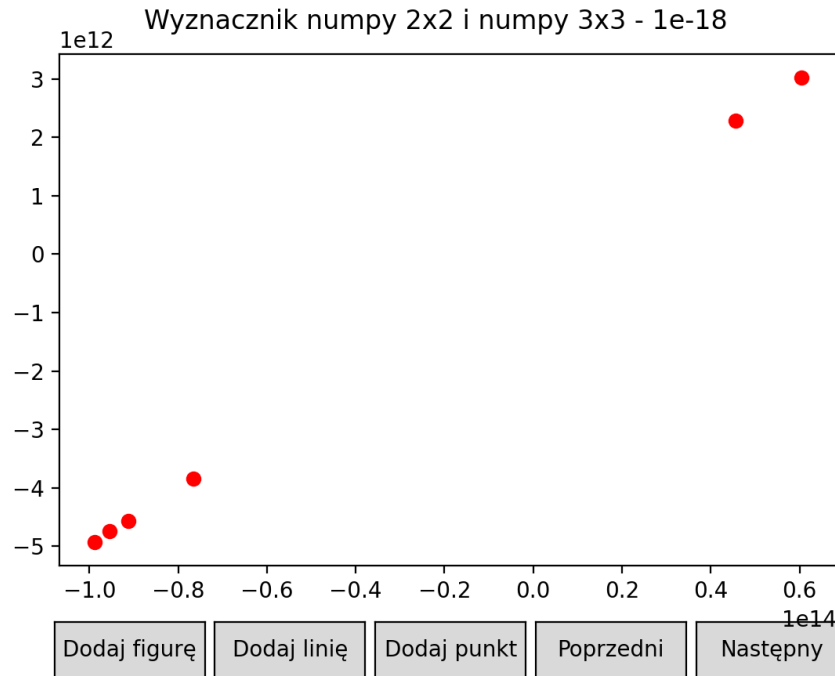
Rysunek 35: Różnice w klasyfikacji punktów zestawu danych B z precyzją float32



Rysunek 36: Różnice w klasyfikacji punktów zestawu danych B z precyzją float32



Rysunek 37: Różnice w klasyfikacji punktów zestawu danych B z precyzją float32



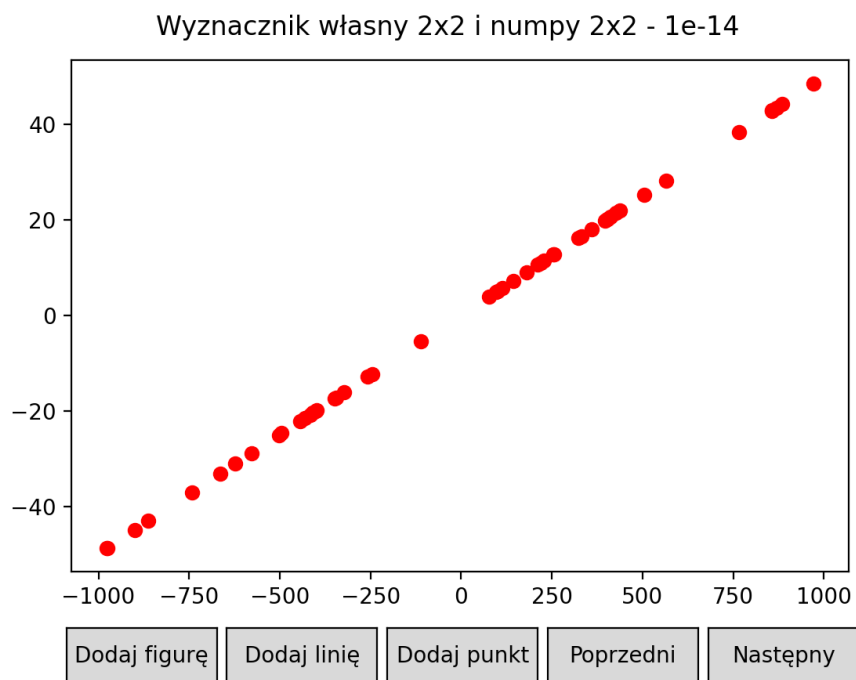
Rysunek 38: Różnice w klasyfikacji punktów zestawu danych B z precyzją float32

7.2 Różnice klasyfikacji dla zestawu D precyzja float32

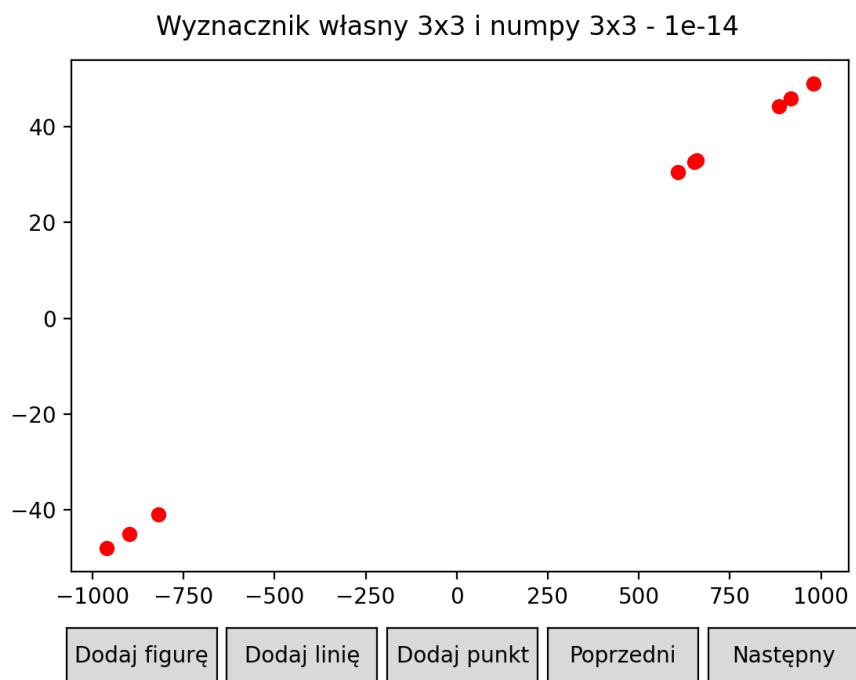
W porównaniu z różnicami klasyfikacji dla precyzji `np.float64`, różnice te są mniejsze, jednakże jest to spowodowane mniejszą precyzją jak i mniejszą liczbą punktów zakwalifikowanych jako współliniowe.

Tolerancja	Kombinacja wyznaczników	Różnica
1e-18	Wyznacznik własny 2x2 i numpy 2x2	53
	Wyznacznik własny 3x3 i numpy 3x3	98
	Wyznacznik własny 2x2 i numpy 3x3	114
	Wyznacznik własny 3x3 i numpy 2x2	126
	Wyznacznik własny 2x2 i własny 3x3	142
	Wyznacznik numpy 2x2 i numpy 3x3	109
1e-14	Wyznacznik własny 2x2 i numpy 2x2	50
	Wyznacznik własny 3x3 i numpy 3x3	9
	Wyznacznik własny 2x2 i numpy 3x3	32
	Wyznacznik własny 3x3 i numpy 2x2	47
	Wyznacznik własny 2x2 i własny 3x3	23
	Wyznacznik numpy 2x2 i numpy 3x3	52
1e-10 oraz 1e-06	Wyznacznik własny 2x2 i numpy 2x2	0
	Wyznacznik własny 3x3 i numpy 3x3	0
	Wyznacznik własny 2x2 i numpy 3x3	0
	Wyznacznik własny 3x3 i numpy 2x2	0
	Wyznacznik własny 2x2 i własny 3x3	0
	Wyznacznik numpy 2x2 i numpy 3x3	0

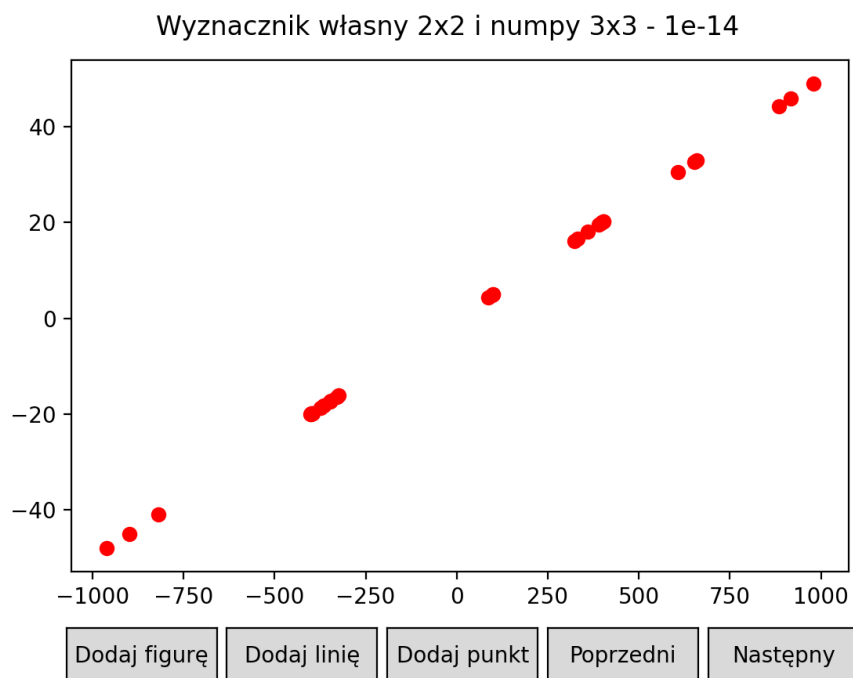
Tabela 10: Tabela różnic klasyfikacji punktów zestawu danych D z precyzją float32



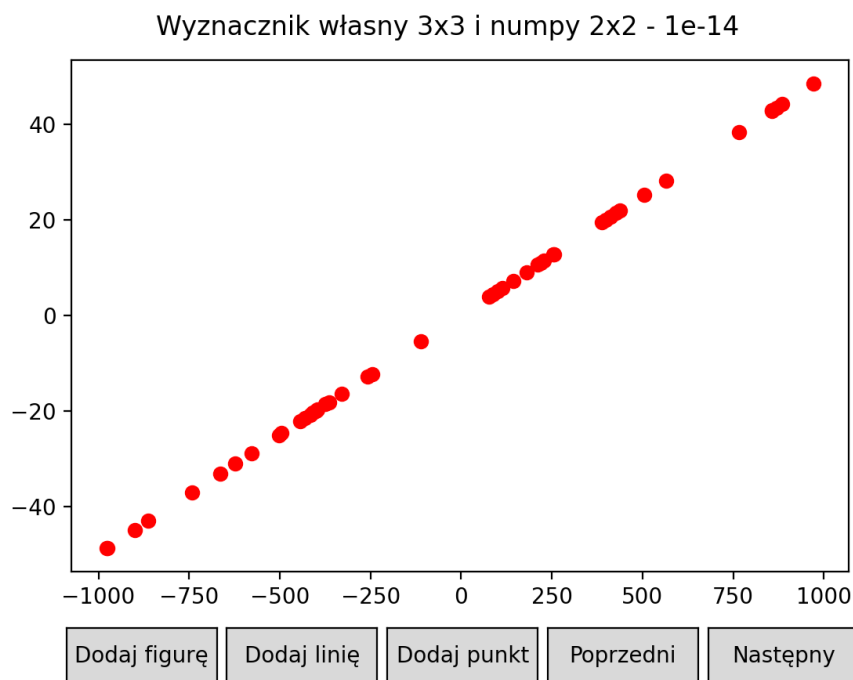
Rysunek 39: Różnice w klasyfikacji punktów zestawu danych D z precyzją float32



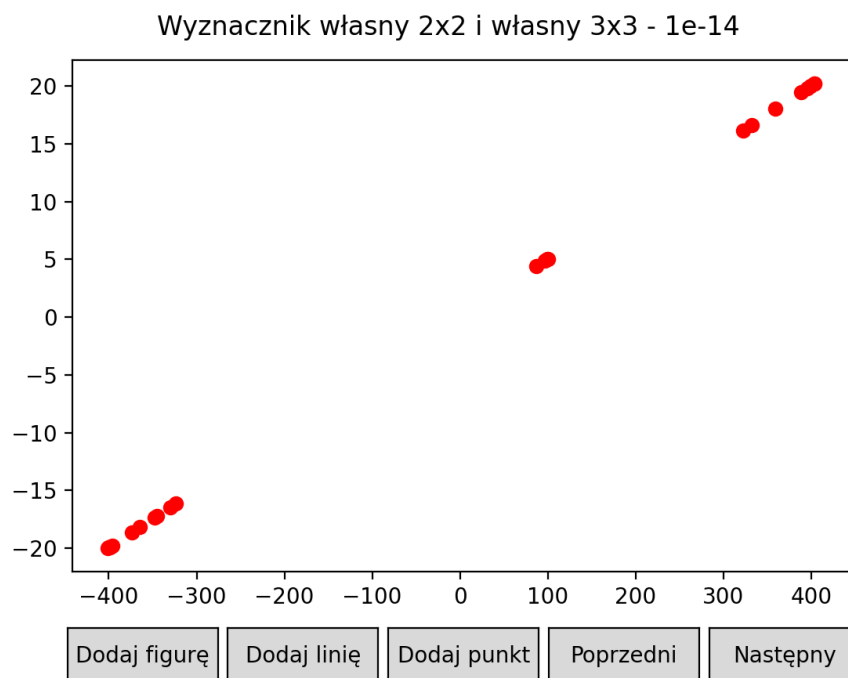
Rysunek 40: Różnice w klasyfikacji punktów zestawu danych D z precyzją float32



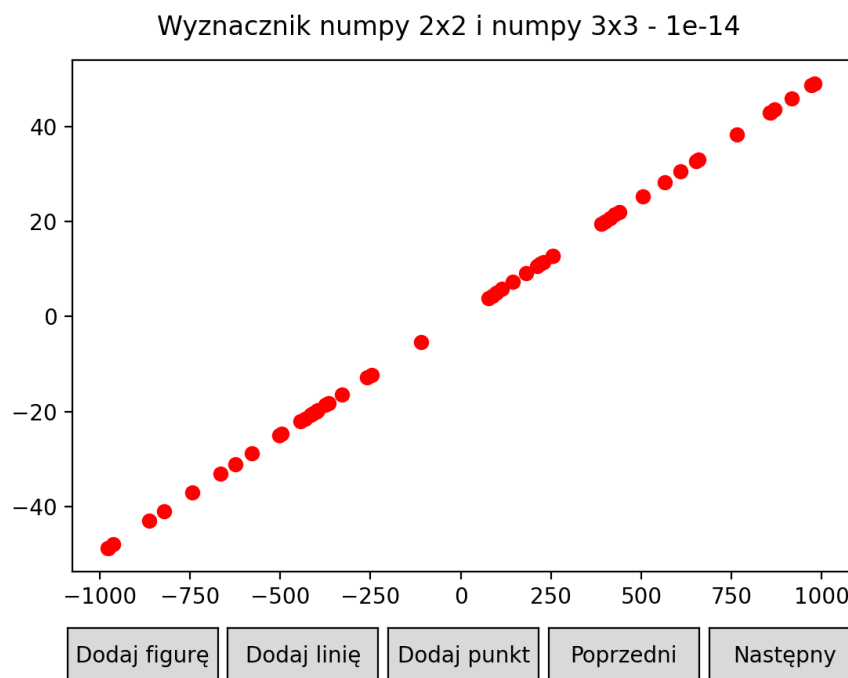
Rysunek 41: Różnice w klasyfikacji punktów zestawu danych D z precyzją float32



Rysunek 42: Różnice w klasyfikacji punktów zestawu danych D z precyzją float32



Rysunek 43: Różnice w klasyfikacji punktów zestawu danych D z precyzją float32



Rysunek 44: Różnice w klasyfikacji punktów zestawu danych D z precyzją float32

8 Czasy obliczania wyznaczników

Czasy obliczania wyznaczników zostały wyznaczone podczas dokonywania klasyfikacji punktów dla badanych zestawów danych za pomocą funkcji `process_time` z biblioteki `time`. Funkcja ta, w porównaniu do innych funkcji z tej biblioteki, oblicza czas działania procesora dla danego procesu (system and user CPU time), nie bierze pod uwagę np. czasu, gdy program korzysta z funkcji `sleep`. Mierzenie czasu obliczania wyznaczników zostało wykonane dla obu precyzji punktów oraz różnych tolerancji dla zera.

Tolerancja	Wyznacznik	Średni czas dla zestawu danych [s]			
		A	B	C	D
1e-18	Wyznacznik własny 2x2	4.68750e-06	2.50000e-06	0.00000e+00	1.56250e-05
	Wyznacznik numpy 2x2	1.06250e-05	1.04688e-05	3.12500e-05	3.12500e-05
	Wyznacznik własny 3x3	3.12500e-06	3.43750e-06	0.00000e+00	0.00000e+00
	Wyznacznik numpy 3x3	1.04688e-05	9.21875e-06	1.56250e-05	1.56250e-05
1e-14	Wyznacznik własny 2x2	1.87500e-06	2.96875e-06	1.56250e-05	0.00000e+00
	Wyznacznik numpy 2x2	1.03125e-05	9.68750e-06	0.00000e+00	1.56250e-05
	Wyznacznik własny 3x3	3.75000e-06	4.84375e-06	0.00000e+00	1.56250e-05
	Wyznacznik numpy 3x3	1.07812e-05	1.09375e-05	1.56250e-05	0.00000e+00
1e-10	Wyznacznik własny 2x2	2.65625e-06	1.87500e-06	0.00000e+00	1.56250e-05
	Wyznacznik numpy 2x2	1.04688e-05	1.12500e-05	1.56250e-05	1.56250e-05
	Wyznacznik własny 3x3	4.37500e-06	4.53125e-06	0.00000e+00	0.00000e+00
	Wyznacznik numpy 3x3	1.17188e-05	1.12500e-05	1.56250e-05	0.00000e+00
1e-06	Wyznacznik własny 2x2	3.75000e-06	2.18750e-06	0.00000e+00	0.00000e+00
	Wyznacznik numpy 2x2	1.09375e-05	1.07812e-05	0.00000e+00	1.56250e-05
	Wyznacznik własny 3x3	3.59375e-06	4.06250e-06	0.00000e+00	0.00000e+00
	Wyznacznik numpy 3x3	1.01562e-05	1.15625e-05	1.56250e-05	1.56250e-05

Tabela 11: Średnie czasy działania wyznaczników dla wszystkich zestawów z precyzją `np.float64`

Tolerancja	Wyznacznik	Średni czas dla zestawu danych [s]			
		A	B	C	D
1e-18	Wyznacznik własny 2x2	8.75000e-06	8.75000e-06	1.56250e-05	0.00000e+00
	Wyznacznik numpy 2x2	1.56250e-05	1.59375e-05	3.12500e-05	1.56250e-05
	Wyznacznik własny 3x3	1.03125e-05	9.68750e-06	1.56250e-05	1.56250e-05
	Wyznacznik numpy 3x3	1.09375e-05	1.10938e-05	1.56250e-05	1.56250e-05
1e-14	Wyznacznik własny 2x2	9.21875e-06	8.12500e-06	0.00000e+00	1.56250e-05
	Wyznacznik numpy 2x2	1.62500e-05	1.71875e-05	3.12500e-05	1.56250e-05
	Wyznacznik własny 3x3	8.75000e-06	1.14062e-05	0.00000e+00	1.56250e-05
	Wyznacznik numpy 3x3	1.17188e-05	1.15625e-05	0.00000e+00	1.56250e-05
1e-10	Wyznacznik własny 2x2	8.28125e-06	8.43750e-06	1.56250e-05	0.00000e+00
	Wyznacznik numpy 2x2	1.60938e-05	1.65625e-05	1.56250e-05	3.12500e-05
	Wyznacznik własny 3x3	1.10938e-05	8.90625e-06	0.00000e+00	0.00000e+00
	Wyznacznik numpy 3x3	1.14062e-05	1.04688e-05	1.56250e-05	0.00000e+00
1e-06	Wyznacznik własny 2x2	8.75000e-06	7.50000e-06	1.56250e-05	1.56250e-05
	Wyznacznik numpy 2x2	1.62500e-05	1.54687e-05	1.56250e-05	1.56250e-05
	Wyznacznik własny 3x3	1.00000e-05	1.03125e-05	1.56250e-05	1.56250e-05
	Wyznacznik numpy 3x3	1.09375e-05	1.10938e-05	1.56250e-05	1.56250e-05

Tabela 12: Średnie czasy działania wyznaczników dla wszystkich zestawów z precyzją `np.float32`

9 Wnioski

9.1 Precyzje współrzędnych punktów

Dla wszystkich zestawów danych wykonano klasyfikacje dla dwóch różnych precyzji współrzędnych punktów (`np.float64` oraz `np.float32`). Po przeanalizowaniu tabel z klasyfikacjami punktów można zauważyć, że punkty z mniejszą precyzją rzadziej kwalifikowały się jako współliniowe, zwłaszcza dla zestawu danych D. Natomiast, dla zestawu danych B, zakwalifikowało się więcej punktów dla precyzji `np.float32`, lecz jest to związane prawdopodobnie z błędami precyzji w obliczeniach. W przypadku chęci klasyfikacji innych zestawów danych, lepiej użyć precyzji `np.float64`, która pozwala na otrzymanie dokładniejszych punktów, a w konsekwencji lepszych wyników klasyfikacji.

9.2 Wyznaczniki oraz różne tolerancje dla zera

Analizując dane dla precyzji `np.float64` zestawy punktów A oraz C dawały identyczne wyniki, bez względu na tolerancje zera czy też użyty wyznacznik. Wynika to z doboru sposobu, jak tworzone są dane. Istnieje małe prawdopodobieństwo, że dany punkt znajdzie się na prostej, nawet odpowiednio manipulując tolerancjami. Dobrym punktem odniesienia są zestawy punktów B oraz D, które różnorodnie klasyfikowały punkty pod względem użytego wyznacznika i tolerancji. Dla małych tolerancji najczęściej punktów jako współliniowe klasyfikowały wyznaczniki 2x2, natomiast po niewielkiej zmianie tolerancji z 1e-18 na 1e-14, w przypadku zestawu danych D, to wyznacznik własny 3x3 sklasyfikował najczęściej punktów jako współliniowe. Pomimo różnych klasyfikacji punktów większość wyznaczników zachowywało się zgodnie z przewidywaniami - gdy tolerancja była zwiększana, to liczba punktów współliniowych rosła.

9.3 Czas potrzebny na obliczenie wyznacznika

Po przeanalizowaniu otrzymanych czasów potrzebnych na obliczenie wyznacznika danego typu, dla zestawów danych C oraz D, wyniki bardzo często się powtarzały lub różnica w wykonaniu pomiarów była tak mała, że odejmowanie czasów zwracało 0. Analizując wyniki dla zestawów A oraz B najszybciej działały wyznaczniki własnej implementacji, zaimplementowane z wykorzystaniem metody Sarrusa. Dla precyzji `np.float32` obliczenia wyznaczników własnych były wolniejsze niż dla precyzji `np.float64`, a dla wyznaczników z biblioteki `numpy` były porównywanle.

9.4 Wybór wyznacznika najlepiej klasyfikującego punkty

Ze względu na błędy obliczeniowe związane z precyzjami liczb oraz otrzymane wyniki klasyfikacji, nie jesteśmy w stanie wybrać najlepszego wyznacznika oraz tolerancji do klasyfikowania punktów. Jeżeli jednak chcemy otrzymywać optymalne wyniki, warto rozważyć wyznacznik 3x3 własnej implementacji, który dawał wyniki najbardziej zbliżone do oczekiwanych.