**Algorytmy geometryczne**

Sprawozdanie z ćwiczenia 1.

Paweł Lamża

Dane techniczne urządzenia na którym wykonano ćwiczenie:

Laptop z systemem Windows 10 x64

Procesor: AMD Ryzen™ 5 4600H

Pamięć RAM: 16GB

Środowisko: Jupyter notebook

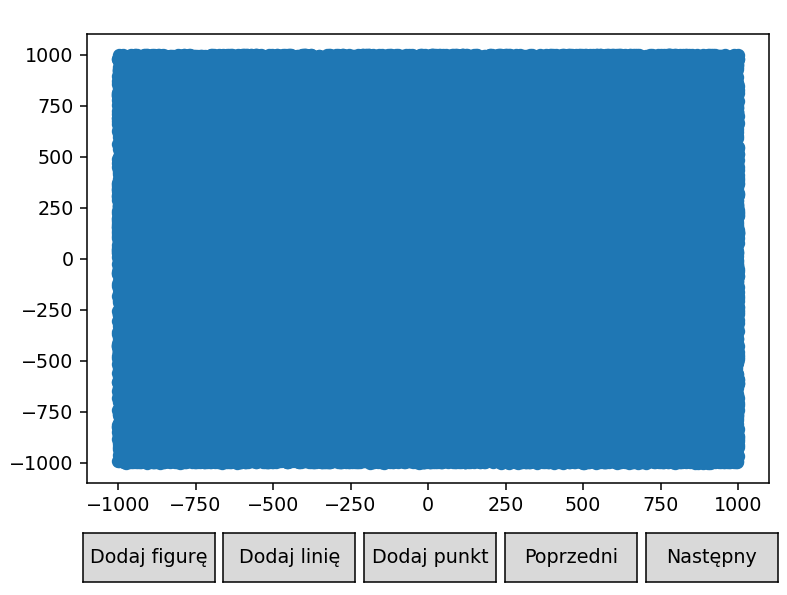
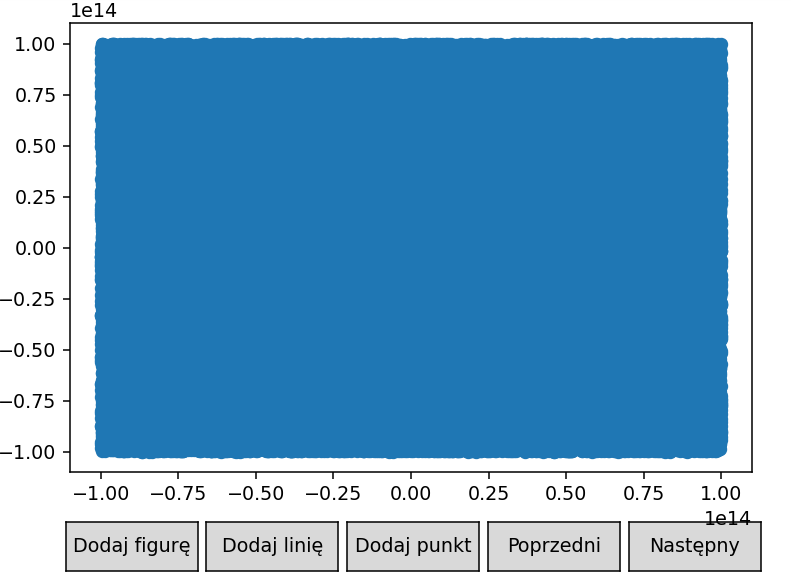
Ćwiczenie zrealizowano w języku Python 3, z wykorzystaniem bibliotek numpy oraz matplotlib

**Opis ćwiczenia**

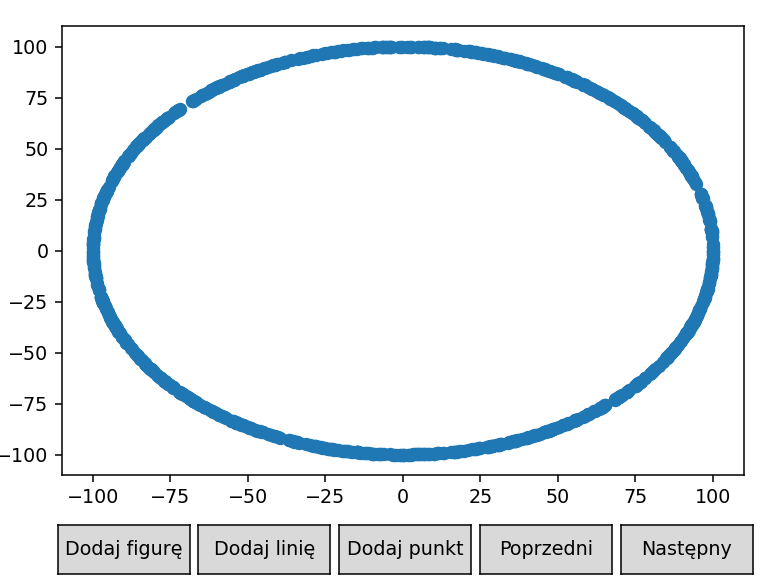
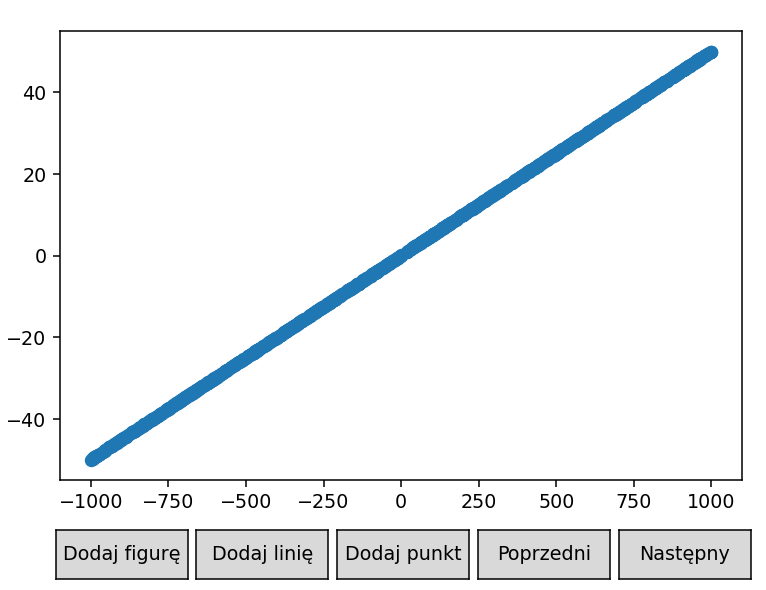
Ćwiczenie polegało na porównaniu wyników klasyfikacji położenia różnych grup punktów względem wektora w zależności od wybranej metody obliczania wyznacznika oraz tolerancji dla zera. (położenie punktów po lewej/prawej stronie wektora lub na nim)

1. Generacja punktów

W celu wykonania ćwiczenia wygenerowałem 4 zestawy punktów A, B, C, D z treści ćwiczenia. Dokonałem losowanie położenia punktów z pomocą metody np.random.rand() z biblioteki numpy. Funkcja ta generuje macierz zadanej wielkości z liczbami typu double z przedziału (0,1). Dodatkowo punkty z zestawu C zostały wygenerowane z pomocą funkcji trygonometrycznych, również z biblioteki numpy. W zestawie D dodatkowo użyłem metodę np.polyfit (w celu wyliczenia ‘a’ i ‘b’ w równaniu y = ax + b) Wszystkie zestawy punktów zostały zwizualizowane za pomocą dostarczonego narzędzia graficznego. Poniżej wykresy dla kolejnych zestawów punktów:

Wykres 1.1, ZESTAW A Wykres 1.2, ZESTAW B

Wykres 1.3, ZESTAW C Wykres 1.4, ZESTAW D

2. Metody obliczania wyznacznika oraz tolerancje dla zera

Kolejnym krokiem ćwiczenia było sklasyfikowanie punktów względem wektora w zależności od metod obliczania wyznacznika oraz tolerancji dla zera przy określaniu położenia na podstawie wyznacznika. Wykorzystałem następujące wyznaczniki:

- Wyznacznik 3x3 obliczany za pomocą metody z biblioteki numpy

- Wyznacznik 2x2 obliczany za pomocą metody z biblioteki numpy

- Wyznacznik 3x3 obliczany za pomocą własnej funkcji

- Wyznacznik 2x2 obliczany za pomocą własnej funkcji

Do obliczania własnych wyznaczników użyłem wzorów zamieszczonych w treści ćwiczenia na obliczanie położenia punktu względem odcinka

Rozważane przeze mnie tolerancje dla zera to: 10^-18, 10^-14, 10^-10 oraz 10^-6

3. Klasyfikowanie punktów

W celu zrealizowania zadania stworzyłem funkcje **classify**, która dla danego zestawu punktów zwraca trzy tablice dwuwymiarowe (kolejno left, collinear, right) w których przechowywane są informacje o sklasyfikowaniu punktów dla różnych metod wyliczania wyznacznika i różnych tolerancji dla zera (odpowiednio: po lewej, na odcinku, po prawej)

4. Wizualizacja sklasyfikowania

W kolejnym kroku wyświetlałem wyniki dla poszczególnych zestawów przy użyciu różnych metod obliczania wyznacznika. W tej części ćwiczenia tolerancja dla zera była taka sama dla wszystkich testów we wszystkich zestawach i wynosiła 10^-18. Dodatkowo na początku wyświetlałem wyniki dla danego zestawu w porównaniu kolejnych wyznaczników. Jeśli chodzi o kolory na wykresach to:

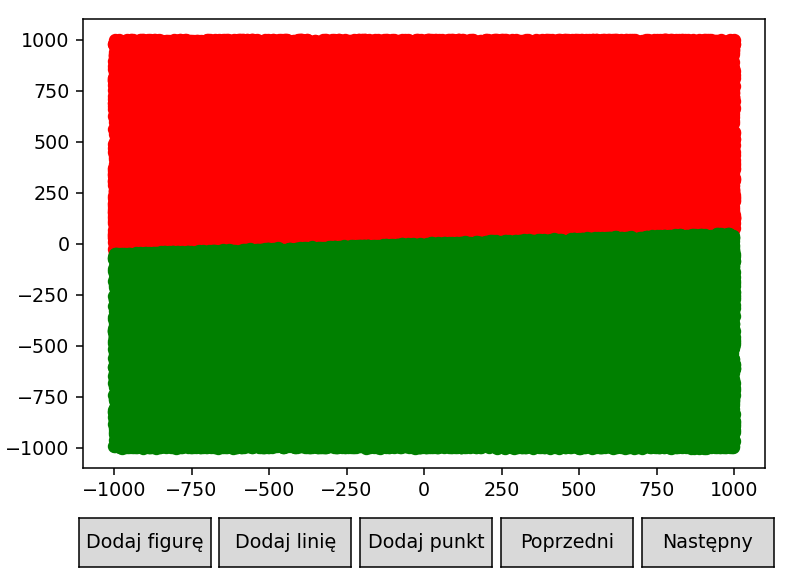
* Czerwony – punkty sklasyfikowane po lewej stronie prostej
* Zielony – punkty sklasyfikowane po prawej stronie prostej
* Niebieski – punkty sklasyfikowane na prostej

**Zestaw A**, wyniki takie same dla wszystkich wyznaczników:

|  |  |  |  |
| --- | --- | --- | --- |
| Wyznacznik | po lewej | na prostej | po prawej |
| 3x3 numpy | 50045 | 0 | 49955 |
| 2x2 numpy | 50045 | 0 | 49955 |
| 3x3 mój | 50045 | 0 | 49955 |
| 2x2 mój | 50045 | 0 | 49955 |

*Tabela 4.A.1*

*KLASYFIKACJA DLA WSZYSTKICH WYZNACZNIKÓW PRZY STAŁEJ TOLERANCJI DLA ZERA (****ZESTAW A****)*



*Wykres 4.A.1*

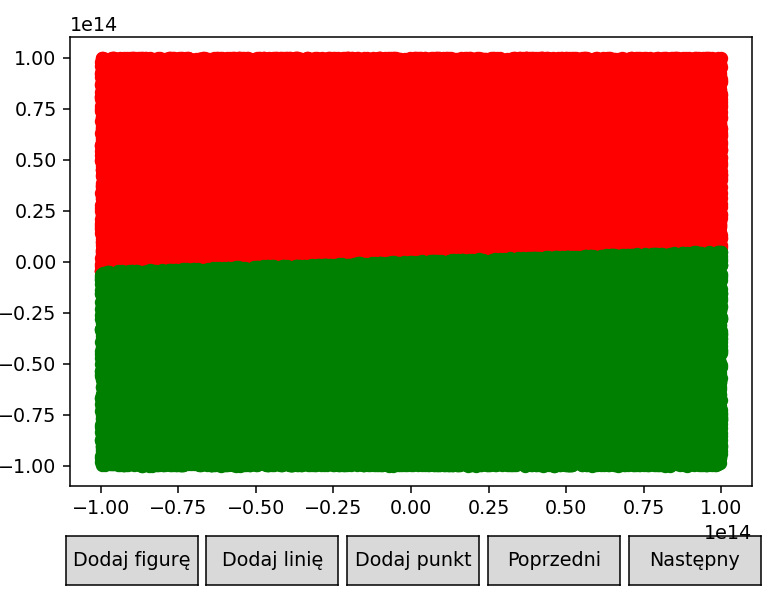
*KLASYFIKACJA DLA WSZYSTKICH WYZNACZNIKÓW PRZY STAŁEJ TOLERANCJI DLA ZERA (****ZESTAW A****)*

**Zestaw B**, wyniki różne dla obliczania wyznaczników 3x3 i 2x2, dla wyznaczników 2x2 12 punktów znalazło się na odcinku, przy czym co ciekawe były to różne punkty dla wyznacznika 2x2 z biblioteki numpy i własnej implementacji wyznacznika 2x2

|  |  |  |  |
| --- | --- | --- | --- |
| Wyznacznik | po lewej | na prostej | po prawej |
| 3x3 numpy | 50152 | 0 | 49848 |
| 2x2 numpy | 50146 | 12 | 49842 |
| 3x3 mój | 50152 | 0 | 49848 |
| 2x2 mój | 50146 | 12 | 49842 |

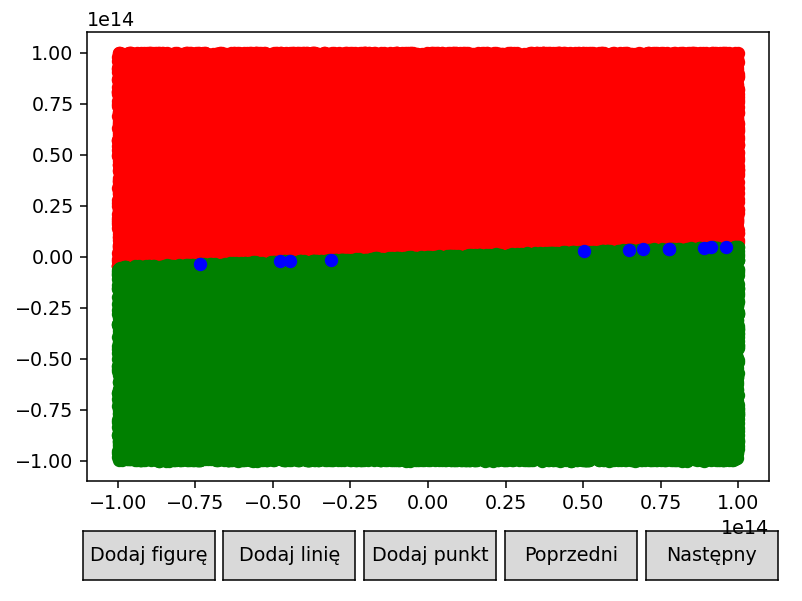
*Tabela 4.B.1*

*KLASYFIKACJA DLA WSZYSTKICH WYZNACZNIKÓW PRZY STAŁEJ TOLERANCJI DLA ZERA (****ZESTAW B****)*



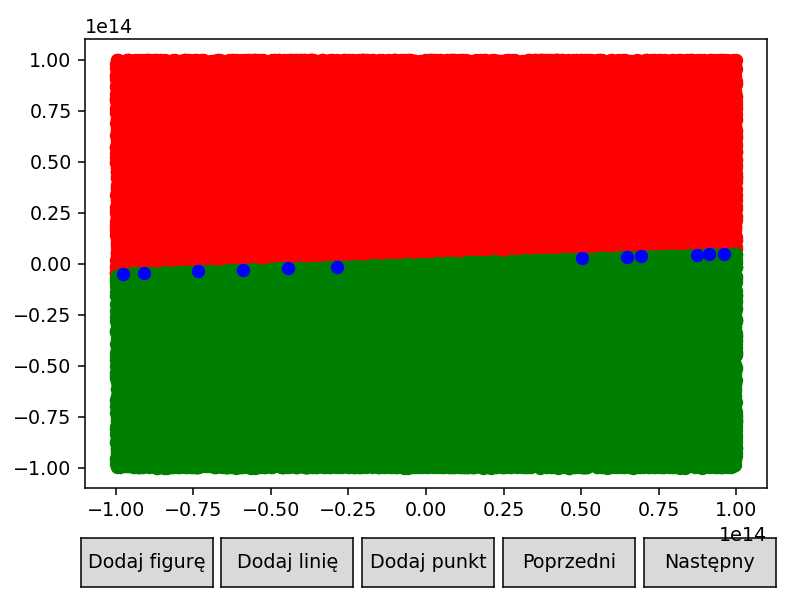
Wykres 4.B.1

DLA WYZNACZNIKA 3x3 Z BIBLIOTEKI NUMPY (**ZESTAW B**)



Wykres 4.B.2

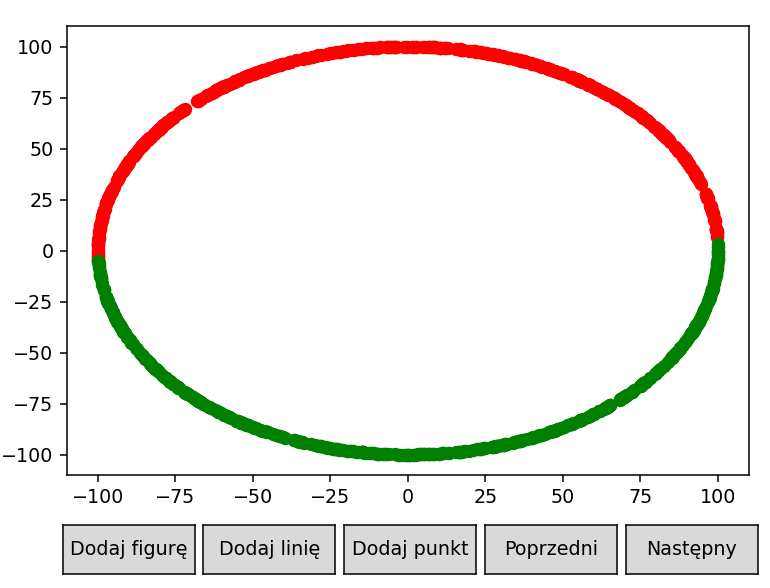
DLA WYZNACZNIKA 2x2 Z BIBLIOTEKI NUMPY (**ZESTAW B**)



Wykres 4.B.3

DLA WYZNACZNIKA 2x2 WŁASNEJ IMPLEMENTACJI (**ZESTAW B**)

**Zestaw C**, wyniki znowu takie same dla wszystkich wyznaczników



Wykres 4.C.1

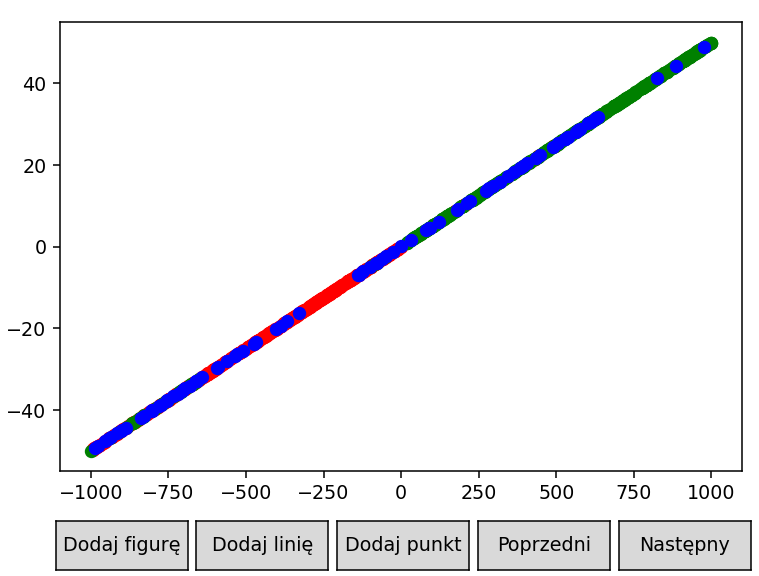
KLASYFIKACJA DLA KAŻDEGO WYZNACZNIKA PRZY STAŁEJ TOLERANCJI DLA ZERA(**ZESTAW C**)

**Zestaw D**, zaobserwowałem największe różnice dla różnych sposobów obliczania wyznacznika. W teorii wszystkie punkty powinny leżeć na prostej ale się tak nie dzieje, wiele z nich leży po prawej bądź lewej stronie co może wynikać z przyjętej tolerancji dla zera wynoszącej 10^-18

|  |  |  |  |
| --- | --- | --- | --- |
| Wyznacznik | po lewej | na prostej | po prawej |
| 3x3 numpy | 444 | 92 | 464 |
| 2x2 numpy | 168 | 675 | 157 |
| 3x3 mój | 354 | 231 | 415 |
| 2x2 mój | 122 | 730 | 148 |

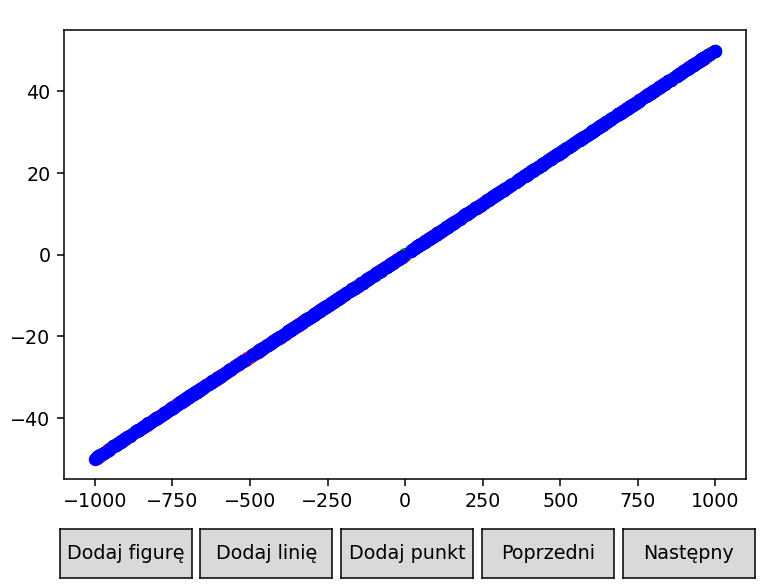
Tabela 4.D.1

KLASYFIKACJA DLA WSZYSTKICH WYZNACZNIKÓW PRZY STAŁEJ TOLERANCJI DLA ZERA(**ZESTAW D**)



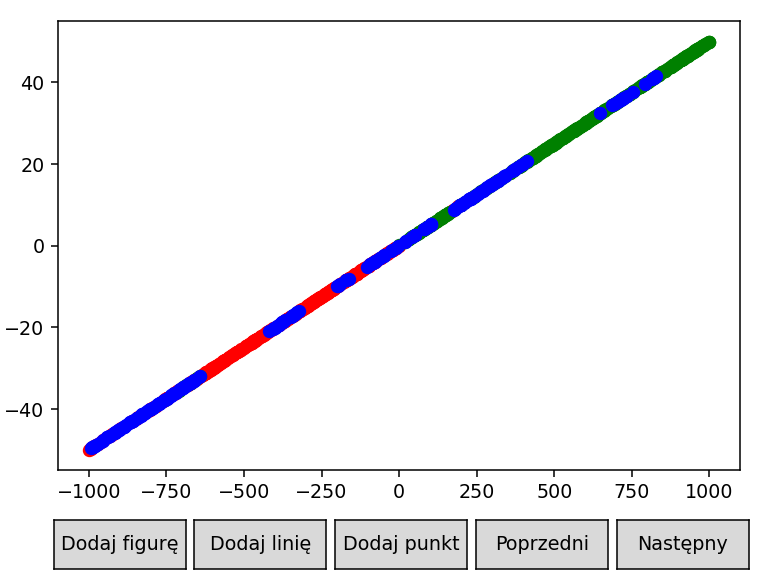
Wykres 4.D.1

KLASYFIKACJA DLA WYZNACZNIKA 3x3 Z BIBLIOTEKI NUMPY (**ZESTAW D**)



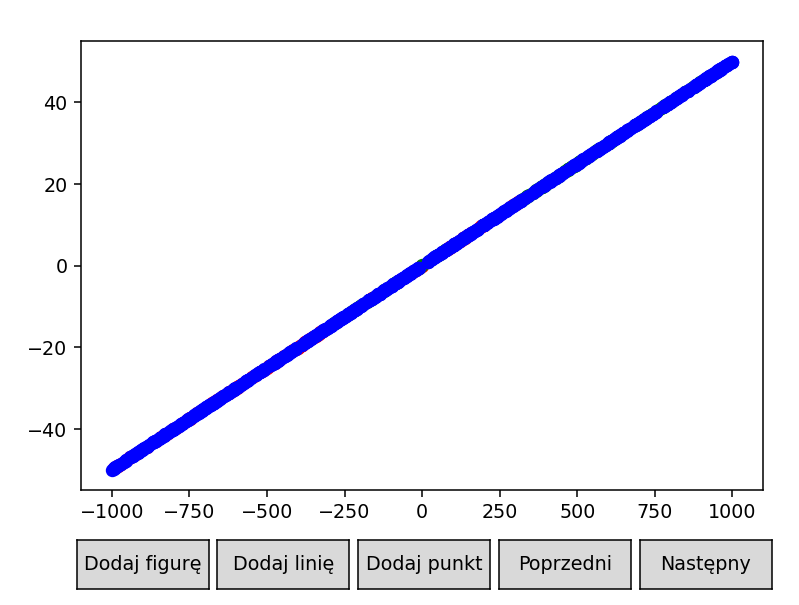
Wykres 4.D.2

KLASYFIKACJA DLA WYZNACZNIKA 2x2 Z BIBLIOTEKI NUMPY (**ZESTAW D**)



Wykres 4.D.3

KLASYFIKACJA DLA WYZNACZNIKA 3x3 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)



Wykres 4.D.4

KLASYFIKACJA DLA WYZNACZNIKA 2x2 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)

5. Różnice dla poszczególnych wyznaczników

W tej części wciąż przyjmuję stałą tolerancję dla zera wynoszącą 10^-18, dla poszczególnych zestawów wyniki prezentują się następująco:

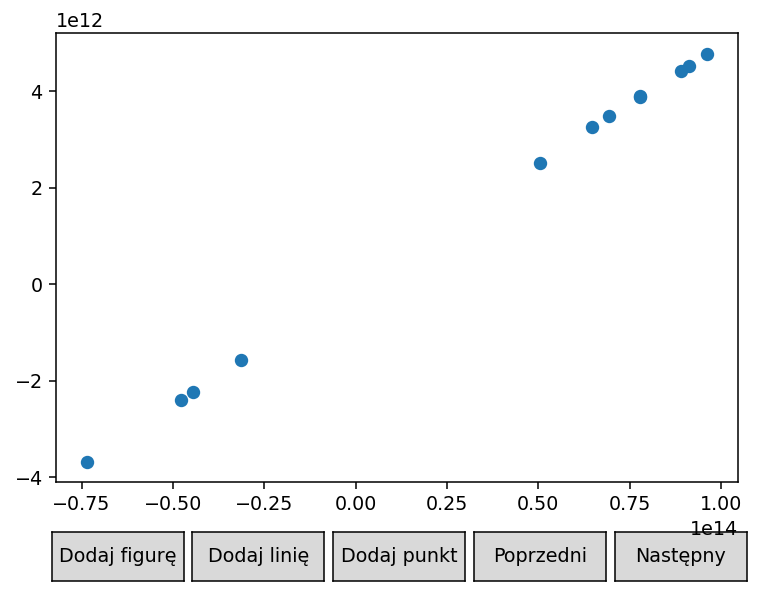
**Zestaw A**, nie zaobserwowałem żadnych różnic

**Zestaw B**, kilkanaście różnic pomiędzy poszczególnymi wyznacznikami

|  |  |  |
| --- | --- | --- |
| wyznacznik 1 | wyznacznik 2 | liczba różnic pomiędzy nimi |
| 3x3 numpy | 2x2 numpy | 12 |
| 3x3 numpy | 3x3 mój | 0 |
| 3x3 numpy | 2x2 mój | 12 |
| 2x2 numpy | 3x3 mój | 12 |
| 2x2 numpy | 2x2 mój | 10 |
| 3x3 mój | 2x2 mój | 12 |

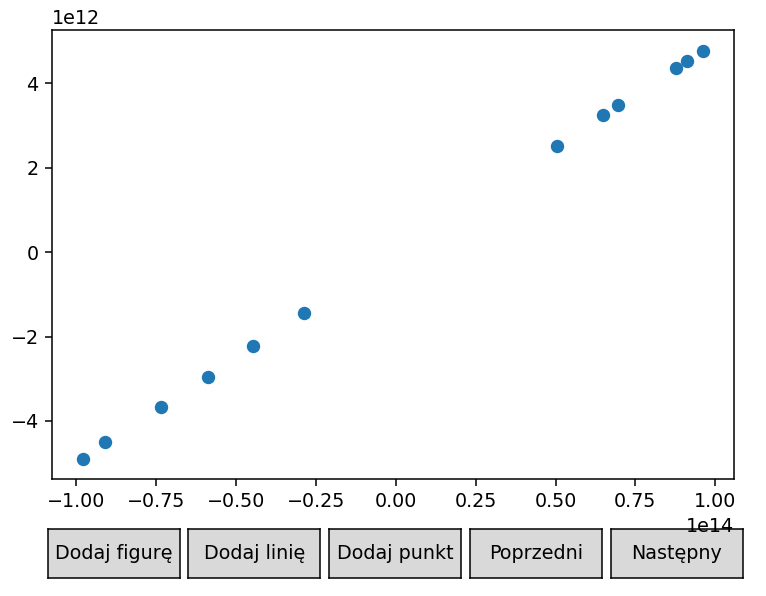
Tabela 5.B.1

LICZBA PUNKTÓW SKLASYFIKOWANYCH INACZEJ DLA RÓŻNYCH WYZNACZNIKÓW(**ZESTAW B**)



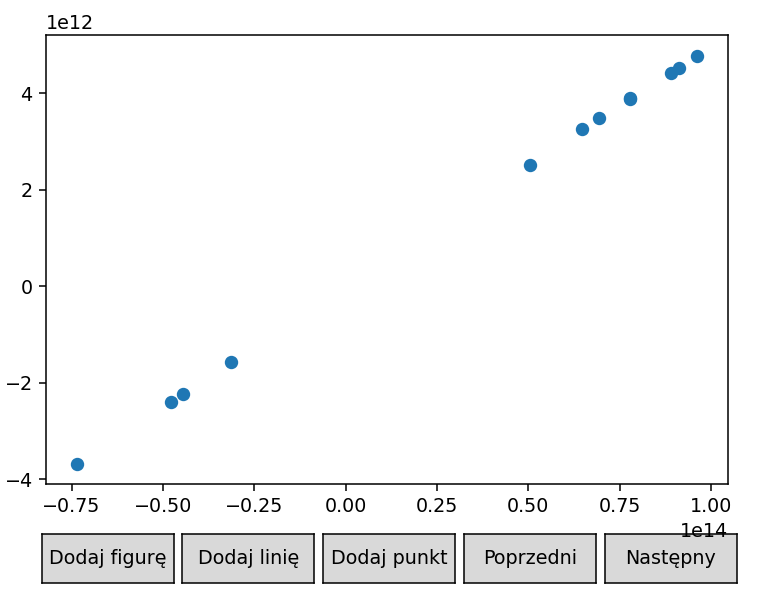
Wykres 5.B.1

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 3X3 NUMPY I 2X2 NUMPY (**ZESTAW B**)



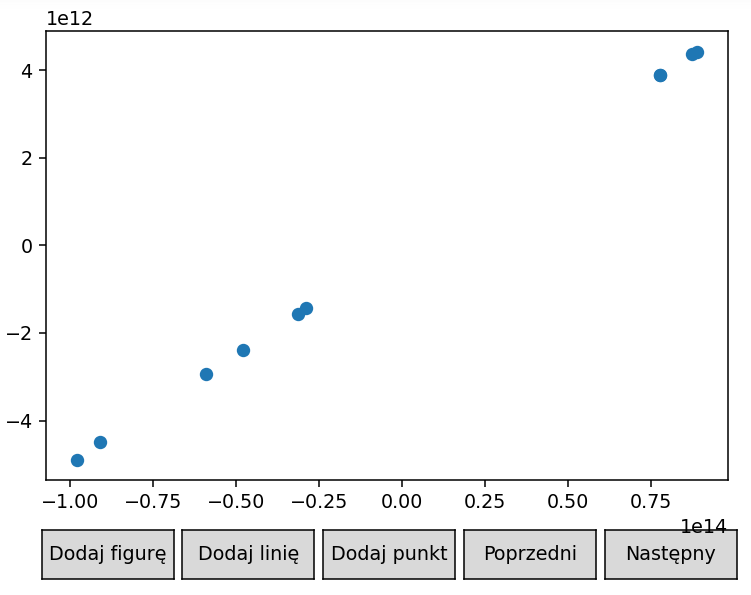
Wykres 5.B.2

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 3X3 NUMPY I 2X2 WŁASNEJ IMPLEMENTACJI (**ZESTAW B**)



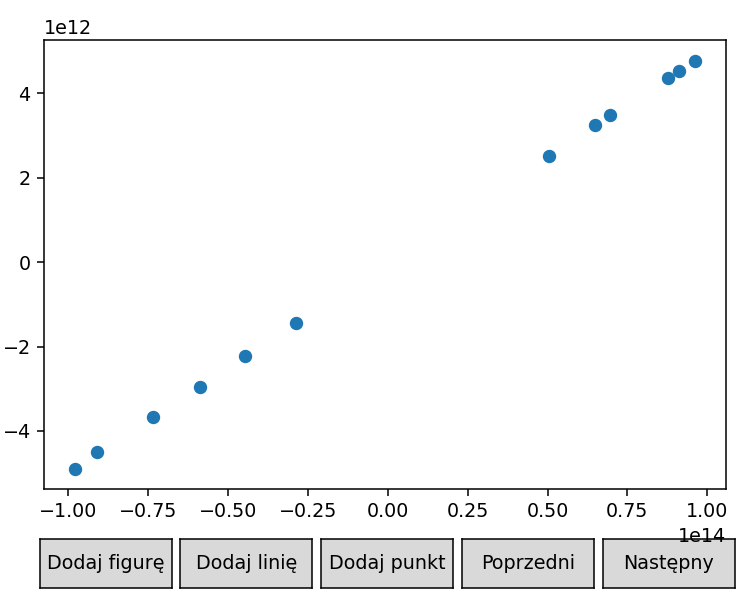
Wykres 5.B.3

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 2X2 NUMPY I 3X3 WŁASNEJ IMPLEMENTACJI (**ZESTAW B**)



Wykres 5.B.4

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 2X2 NUMPY I 2X2 WŁASNEJ IMPLEMENTACJI (**ZESTAW B**)



Wykres 5.B.5

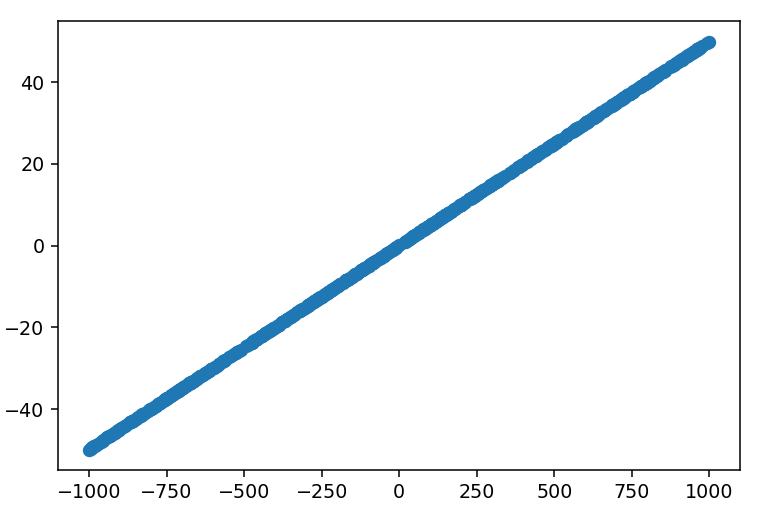
PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 2X2 WŁASNEJ IMPLEMENTACJI I 2X2 WŁASNEJ IMPLEMENTACJI (**ZESTAW B**)

**Zestaw C**, ponownie nie zaobserwowałem żadnych różnic

**Zestaw D**, ogrom różnic (prawie 76%), ponieważ wszystkie punkty są blisko prostej toteż nawet minimalne różnice powodują inną klasyfikację

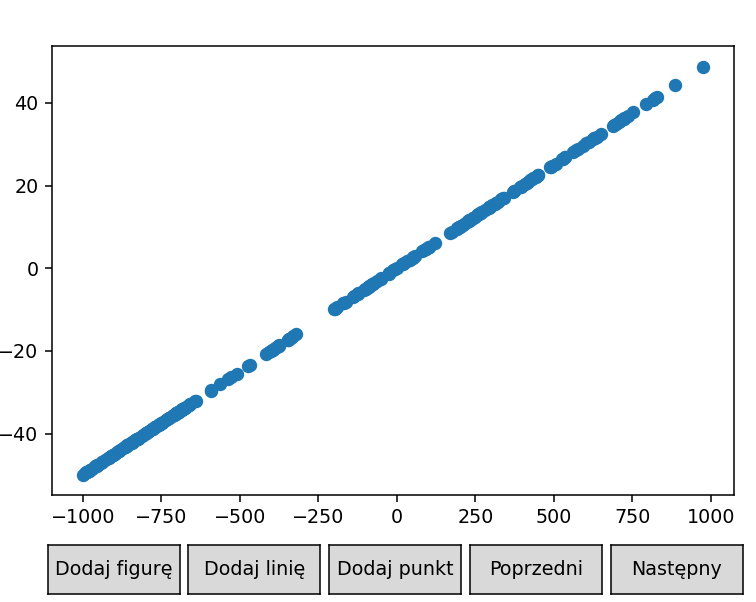
|  |  |  |
| --- | --- | --- |
| wyznacznik 1 | wyznacznik 2 | liczba różnic pomiędzy nimi |
| 3x3 numpy | 2x2 numpy | 749 |
| 3x3 numpy | 3x3 mój | 277 |
| 3x3 numpy | 2x2 mój | 758 |
| 2x2 numpy | 3x3 mój | 667 |
| 2x2 numpy | 2x2 mój | 391 |
| 3x3 mój | 2x2 mój | 678 |

Tabela 5.D.1



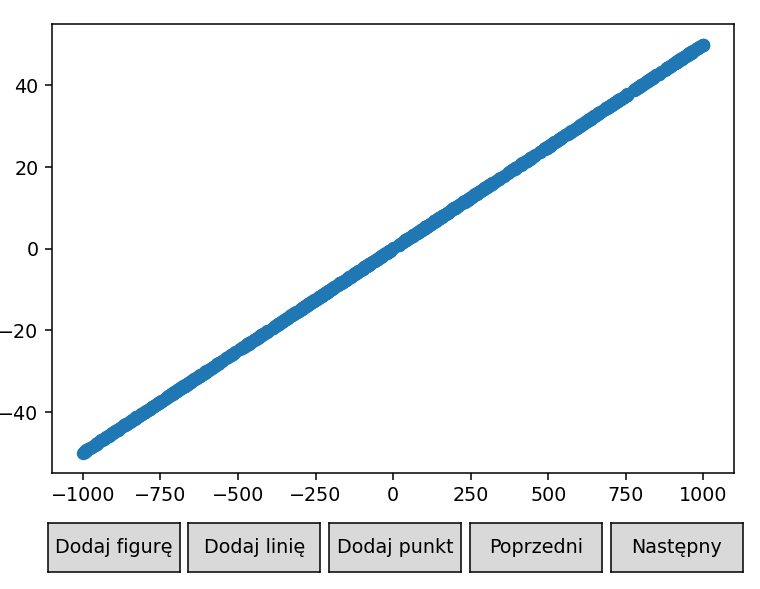
Wykres 5.D.1

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 3X3 NUMPY I 2X2 NUMPY (**ZESTAW D**)



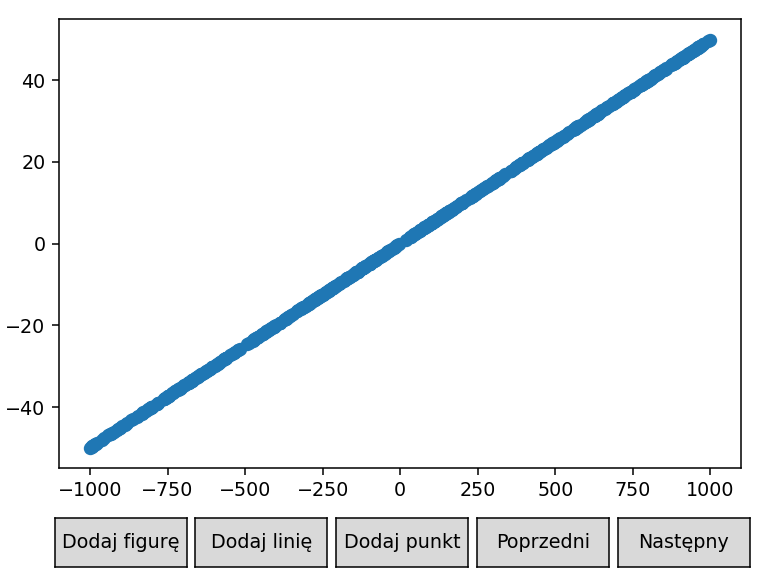
Wykres 5.D.2

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 3X3 NUMPY I 3X3 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)



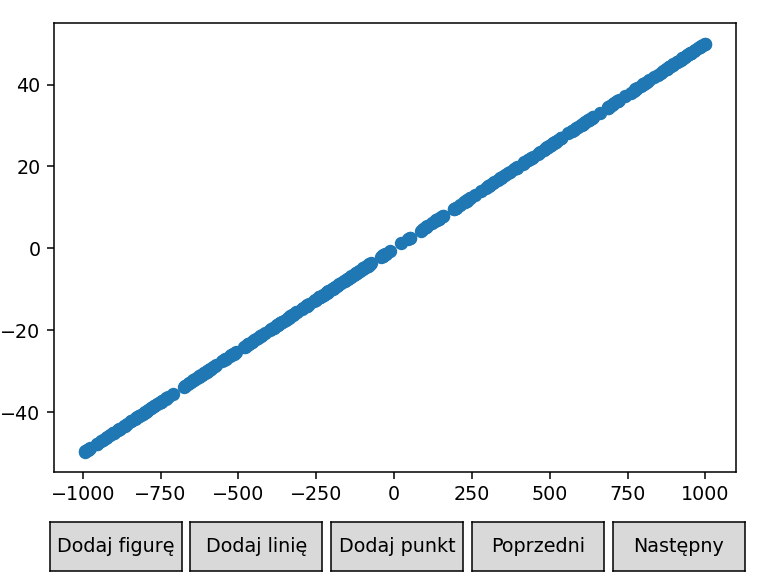
Wykres 5.D.3

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 3X3 NUMPY I 2X2 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)



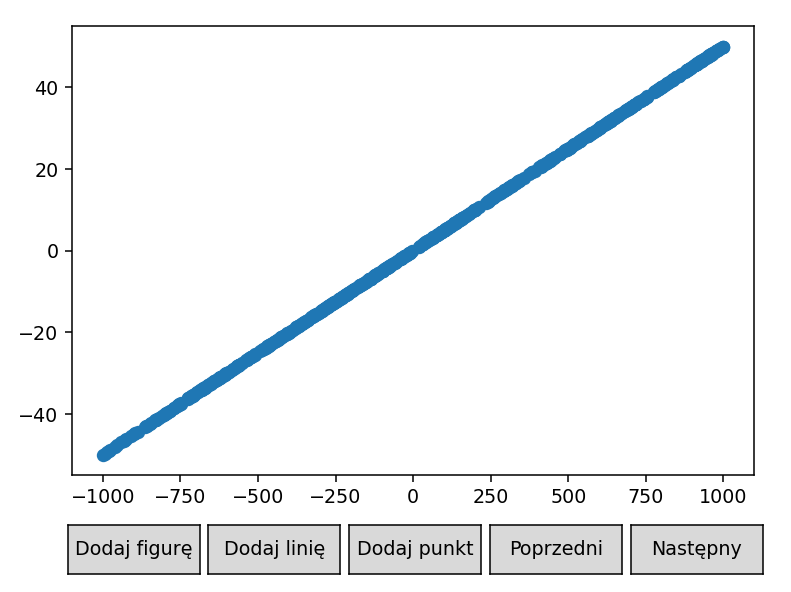
Wykres 5.D.4

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 2X2 NUMPY I 3X3 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)



Wykres 5.D.5

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 2X2 NUMPY I 2X2 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)



Wykres 5.D.6

PUNKTY SKLASYFIKOWANE INACZEJ DLA WYZNACZNIKA 3X3 WŁASNEJ IMPLEMENTACJI I 2X2 WŁASNEJ IMPLEMENTACJI (**ZESTAW D**)

6. Analiza wyników dla różnych tolerancji zera

Dla **zestawu A**, wszędzie te same wyniki niezależnie od użytego wyznacznika ani tolerancji

Dla **zestawu B**, zauważamy różnice jedynie przy zmianie wyznacznika, jednak nie przy zmianie tolerancji dla zera

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wyznacznik** | **tolerancja** | **po lewej** | **na prostej** | **po prawej** |
| numpy 3x3 | 10^-18 | 50152 | 0 | 49848 |
| numpy 3x3 | 10^-14 | 50152 | 0 | 49848 |
| numpy 3x3 | 10^-10 | 50152 | 0 | 49848 |
| numpy 3x3 | 10^-6 | 50152 | 0 | 49848 |
| numpy 2x2 | 10^-18 | 50146 | 12 | 49842 |
| numpy 2x2 | 10^-14 | 50146 | 12 | 49842 |
| numpy 2x2 | 10^-10 | 50146 | 12 | 49842 |
| numpy 2x2 | 10^-6 | 50146 | 12 | 49842 |
| mój 3x3 | 10^-18 | 50152 | 0 | 49848 |
| mój 3x3 | 10^-14 | 50152 | 0 | 49848 |
| mój 3x3 | 10^-10 | 50152 | 0 | 49848 |
| mój 3x3 | 10^-6 | 50152 | 0 | 49848 |
| mój 2x2 | 10^-18 | 50146 | 12 | 49842 |
| mój 2x2 | 10^-14 | 50146 | 12 | 49842 |
| mój 2x2 | 10^-10 | 50146 | 12 | 49842 |
| mój 2x2 | 10^-6 | 50146 | 12 | 49842 |

Tabela 6.1

Dla **zestawu C**, ponownie brak różnic, te same wyniki niezależnie od wyznacznika ani tolerancji

Dla **zestawu D**, duże różnice, przy większej tolerancji (w moim przypadku 1e-10) dla zera wszystkie punkty zaczynają pojawiać się na danej prostej

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **wyznacznik** | **tolerancja** | **po lewej** | **na prostej** | **po prawej** |
| numpy 3x3 | 10^-18 | 444 | 92 | 464 |
| numpy 3x3 | 10^-14 | 91 | 734 | 175 |
| numpy 3x3 | 10^-10 | 0 | 1000 | 0 |
| numpy 3x3 | 10^-6 | 0 | 1000 | 0 |
| numpy 2x2 | 10^-18 | 168 | 675 | 157 |
| numpy 2x2 | 10^-14 | 158 | 686 | 156 |
| numpy 2x2 | 10^-10 | 0 | 1000 | 0 |
| numpy 2x2 | 10^-6 | 0 | 1000 | 0 |
| mój 3x3 | 10^-18 | 354 | 231 | 415 |
| mój 3x3 | 10^-14 | 70 | 753 | 177 |
| mój 3x3 | 10^-10 | 0 | 1000 | 0 |
| mój 3x3 | 10^-6 | 0 | 1000 | 0 |
| mój 2x2 | 10^-18 | 122 | 730 | 148 |
| mój 2x2 | 10^-14 | 116 | 739 | 145 |
| mój 2x2 | 10^-10 | 0 | 1000 | 0 |
| mój 2x2 | 10^-6 | 0 | 1000 | 0 |

Tabela 6.2

7. Wnioski

* Odnosząc się do powyższych danych, różnica w klasyfikacji położenia punktu względem odcinka w zależności od metody obliczania wyznacznika jest znacząca.
* Różne tolerancje dla zera jeszcze bardziej zwiększają różnice. Co ciekawe dzieje się to tylko w niektórych przypadkach. Dla zestawów A i C takich różnic nie było niezależnie od sposobu wyliczania wyznacznika ani tolerancji dla zera. Co może prowadzić do wniosku że różnice występują tylko bardzo blisko linii z której pomocą dokonywaliśmy klasyfikowania.
* Na poprawność klasyfikacji jak widać ma też wpływ wielkość liczb. Zestawy A i B różniły się tylko wielkością liczb co doprowadziło do znaczących różnic zaobserwowanych w zestawie B i braku takich w zestawie A.
* Najwięcej różnic pojawiło się jednak w zestawie D. W teorii każdy punkt powinien znaleźć się na prostej, jednak tak się nie stało, a wręcz przy bardzo małej tolerancji dla zera punkty rzadko pojawiały się na prostej.
* Dla najmniejsze tolerancji 10^-18 najlepszym sposobem wyliczania wyznacznika okazuje się 2x2 (mała różnica w punktach na prostej dla wersji numpy i z własnej implementacji).
* Dla tolerancji rzędu 10^-14 wszystkie wyznaczniki poradziły już sobie porównywalnie, chociaż wyznacznik numpy’owy 2x2 najgorzej ze wszystkich.
* Kiedy tolerancja dla zera zwiększa się do 10^-10 wszystkie punkty pojawiają się na prostej niezależnie od metody wyznaczania wyznacznika.
* Ciekawość poprowadziła mnie to sprawdzenia również wyników dla tolerancji dla zera 10^-12, przy której to już oba wyznaczniki 3x3 okazały się lepsze od tych 2x2
* Powyższe wyniki prowadzą do wniosków, iż dla różnych tolerancji dla zera i zestawów danych wyniki są zgoła inne co świadczy o tym że wyboru wyznacznika zawsze trzeba dokonywać w zależności od zadania (tj. zestawu danych i danej tolerancji).
* Podsumowując nie ma jednego najlepszego sposobu, który działałby w każdej sytuacji lepiej od innych.