# Sprawozdanie z ćwiczenia pierwszego

## Algorytmy Geometryczne

Iwo Szczepaniak, Windows 11 – 4,2GHz

# 1) Wstęp

Celem ćwiczenia jest losowe wygenerowanie zadanych punktów, ich wizualizacja i sprawdzenie położenia względem prostej AB(na lewo, na prawo, na prostej), dzięki wykorzystaniu wyznacznika macierzy 2x2 lub 3x3(we własnej implementacji oraz w wersji bibliotecznej numpy) z ustaloną tolerancją odchylenia od 0.
Wizualizacja i zliczanie punktów było badane pod dwoma kątami –zmiennej tolerancji ze stałą metodą(w czterech wersjach) i zmiennej metody ze stałą tolerancją(w czterech wersjach), co umożliwiło dokładną analizę wpływu tych czynników na wyniki.

# 2) Słowniczek

### Metody wyliczania położenia punktu:

V1 -> wyznacznik macierzy 3x3, własna implementacja
V1 biblioteczne -> wyznacznik macierzy 3x3, biblioteka numpy
V2 -> wyznacznik macierzy 2x2, własna implementacja
V2 biblioteczne -> wyznacznik macierzy 2x2, biblioteka numpy

### Tolerancja dla zera:

1 -> 1e-10
2 -> 1e-14
3 -> 1e-16
4 -> 1e-18

### Zastosowane programy:

Jupyter – wizualizacja punktów
PyCharm – zliczanie punktów
Excel – tworzenie tabel i wykresów

# 3) Generowanie, wizualizacja i zliczanie punktów

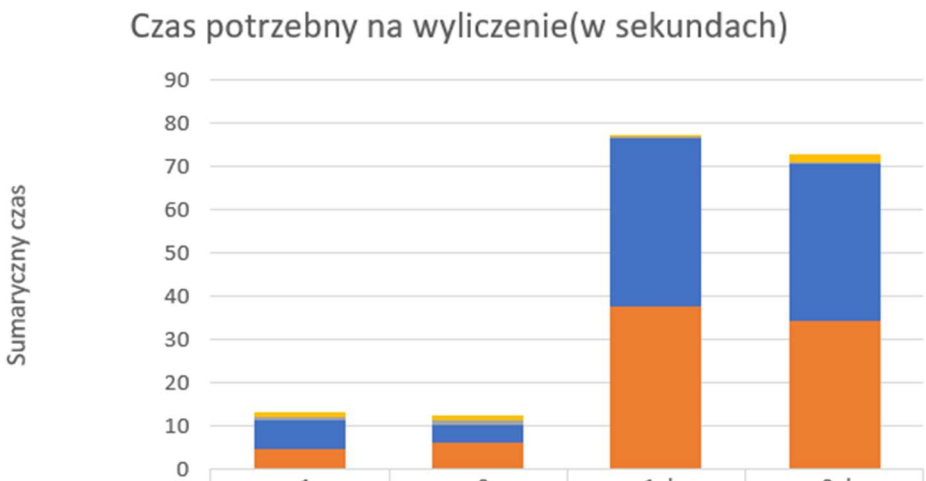
W dwóch pierwszych podpunktach punkty są generowane całkowicie losowo - obie zmienne są losowane osobno. W trzecim podpunkcie losowany jest kąt na okręgu(wykorzystywany w obu zmiennych), a w czwartym podpunkcie losowana jest jedna zmienna(x) i na podstawie wzoru prostej AB wyliczana jest druga zmienna.

```python
# podpunkt a, analogicznie podpunkt b
for i in range(10 ** 5):
    c = (random.uniform(-1000.0, 1000.0), random.uniform(-1000.0, 1000.0))
# podpunkt c
r = 100
for i in range(1000):
    alfa = random.uniform(0, 2 * pi)
    c = (r * sin(alfa), r * cos(alfa))
# podpunkt d
for _ in range(1000):
    x = random.uniform(-1000.0, 1000.0)
    c = (x, x / 20 + 1 / 20)
```

Generowanie losowych punktów

Ze względu na czytelność, zdecydowano, że każde z zadań wyświetlane jest osobno. W celu wybrania podpunktu zadania należy wprowadzić literę podpunktu jako string do zmiennej task.
W wizualizacji  każda z tolerancji lub metod zaznaczona jest innym kolorem(w celu rozróżnienia poszczególnych scen zastosowano różne kolory dla prawej i lewej strony, punkty na linii zawsze oznaczane są czarnym).
Wyniki w tabelach uzyskano wykorzystując tę samą metodę generowania punktów, ale wykorzystywana jedynie do otrzymania liczności danego zbioru. W celu zmniejszenia niedokładności pojedynczych losowań - powstawania mało reprezentatywnych danych - zdecydowano się na wyliczenie średniej z 10 pomiarów dla poszczególnych wartości tolerancji.

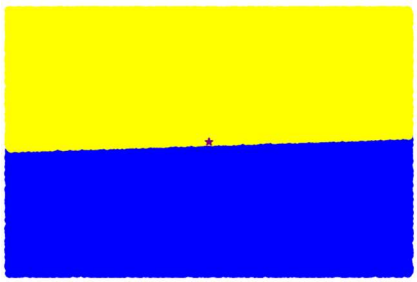
# 4) Czasochłonność




| | v1 | v2 | v1_b | v2_b |
|---|---|---|---|---|
| Czas w zadaniu d | 0.98 | 0.98 | 0.45 | 1.8 |
| Czas w zadaniu c | 0.97 | 1.21 | 0.37 | 0.33 |
| Czas w zadaniu b | 6.68 | 3.87 | 38.83 | 36.55 |
| Czas w zadaniu a | 4.68 | 6.4 | 37.59 | 34.25 |

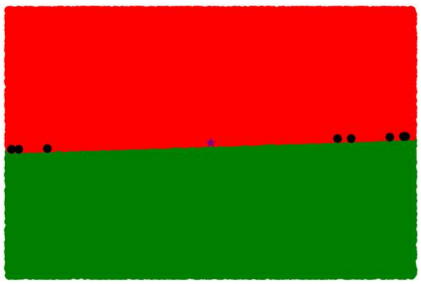
Analizując czasochłonność metod, jasno widać, że biblioteczne funkcje znaczenie odstają pod kątem wydajności od tych implementowanych samodzielnie.
Przyjęta metoda wyliczania 10 powtórzeń sprawiła znaczne wydłużenie czasu i tym większe spotęgowanie efektu.

# 5) Wizualizacja zadań a, b i c



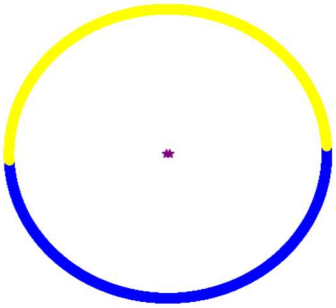
Zad b metoda v1

Zad b metoda v1 biblioteczna

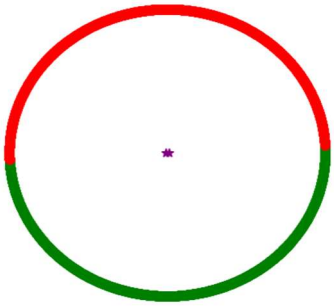


Zad b metoda v2

Zad b metoda v2 biblioteczna



Zad c metoda v1



Zad c metoda v2

Zdecydowano się nie umieszczać zadania a z powodu dużego podobieństwa do zadania b, a znacznie mniej interesujących wyników. Podobnie w zadaniu c – dalsze wizualizacje wyglądają identycznie i nie przekazują żadnych nowych informacji.

# 6) Dane do zadań a, b i c

| V1_biblio | | | | | |
|---|---|---|---|---|---|
| | | | | | |
| Zadanie a | | | | | |
| | Tolerancja | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
| | Prawo | 50025.9 | 50025.9 | 50025.9 | 50025.9 |
| | Lewo | 49974.1 | 49974.1 | 49974.1 | 49974.1 |
| | Na Linii | 0 | 0 | 0 | 0 |
| | | | | | |
| | | | | | |
| Zadanie b | | | | | |
| | Tolerancja | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
| | Prawo | 49991.9 | 49991.9 | 49991.9 | 49991.9 |
| | Lewo | 50008.1 | 50008.1 | 50008.1 | 50008.1 |
| | Na Linii | 0 | 0 | 0 | 0 |
| | | | | | |
| | | | | | |
| Zadanie c | | | | | |
| | Tolerancja | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
| | Prawo | 505.9 | 505.9 | 505.9 | 505.9 |
| | Lewo | 494.1 | 494.1 | 494.1 | 494.1 |
| | Na Linii | 0 | 0 | 0 | 0 |

W tych trzech zadaniach nie zauważono żadnej różnicy w wynikach dla wybranych tolerancji dla zera(na ilustracji przykładowo metoda v1 biblioteczna).

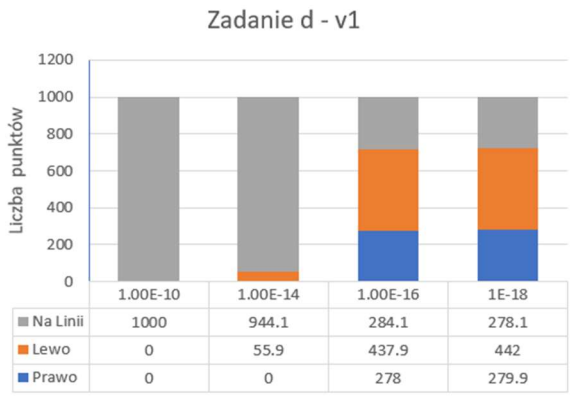
| Tolerancja | | 1E-16 | | | |
|---|---|---|---|---|---|
| | | | | | |
| Zadanie a | | | | | |
| | Metoda | v1 | v1 biblio | v2 | v2 biblio |
| | Prawo | 49969.9 | 49969.9 | 49969.9 | 49969.9 |
| | Lewo | 50030.1 | 50030.1 | 50030.1 | 50030.1 |
| | Na Linii | 0 | 0 | 0 | 0 |
| | | | | | |
| | | | | | |
| Zadanie b | | | | | |
| | Metoda | v1 | v1 biblio | v2 | v2 biblio |
| | Prawo | 49978.5 | 49978.5 | 49974.8 | 49974.3 |
| | Lewo | 50021.5 | 50021.5 | 50018.1 | 50018.9 |
| | Na Linii | 0 | 0 | 7.1 | 6.8 |
| | | | | | |
| | | | | | |
| Zadanie c | | | | | |
| | Metoda | v1 | v1 biblio | v2 | v2 biblio |
| | Prawo | 499.1 | 499.1 | 499.1 | 499.1 |
| | Lewo | 500.9 | 500.9 | 500.9 | 500.9 |
| | Na Linii | 0 | 0 | 0 | 0 |

Tabela z wynikami dla przykładowej tolerancji 1e-16

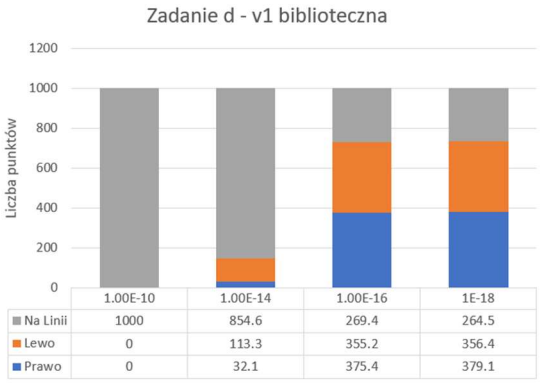
Nie zauważono też większej różnicy między metodami v1 a v2 oraz między funkcjami bibliotecznymi a tymi implementowanych samodzielnie.

**Wnioski:** Prawdopodobną przyczyną braku różnicy w wynikach jest brak punktów na linii lub bardzo blisko niej przy tak dużym rozrzucie punktów w tych zadaniach.

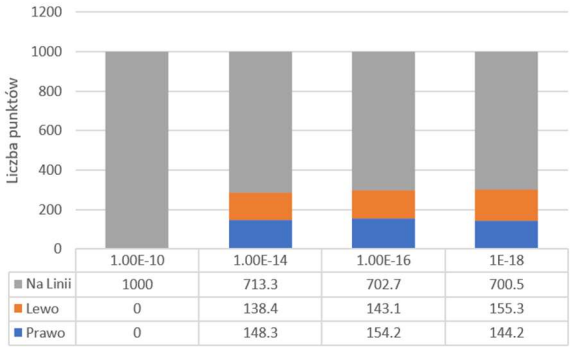
# 7) Zadanie d



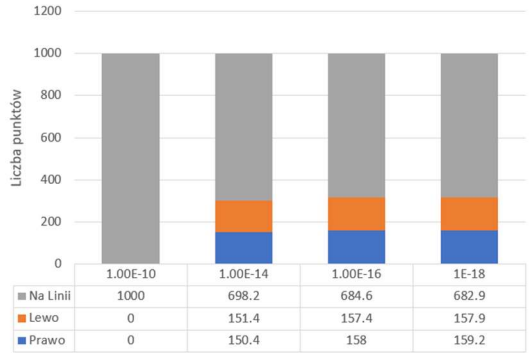

| | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
|---|---|---|---|---|
| Na Linii | 1000 | 944.1 | 284.1 | 278.1 |
| Lewo | 0 | 55.9 | 437.9 | 442 |
| Prawo | 0 | 0 | 278 | 279.9 |




| | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
|---|---|---|---|---|
| Na Linii | 1000 | 854.6 | 269.4 | 264.5 |
| Lewo | 0 | 113.3 | 355.2 | 356.4 |
| Prawo | 0 | 32.1 | 375.4 | 379.1 |




| | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
|---|---|---|---|---|
| Na Linii | 1000 | 713.3 | 702.7 | 700.5 |
| Lewo | 0 | 138.4 | 143.1 | 155.3 |
| Prawo | 0 | 148.3 | 154.2 | 144.2 |




| | 1.00E-10 | 1.00E-14 | 1.00E-16 | 1E-18 |
|---|---|---|---|---|
| Na Linii | 1000 | 698.2 | 684.6 | 682.9 |
| Lewo | 0 | 151.4 | 157.4 | 157.9 |
| Prawo | 0 | 150.4 | 158 | 159.2 |

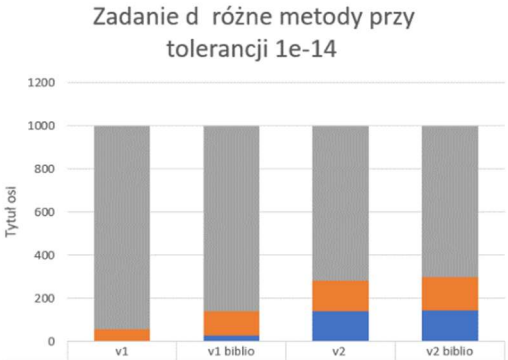
Wykresy przedstawiające podział punktów w zadaniu d w zależności od metody i implementacji

W zadaniu d widać znaczące różnice w wyliczaniu położenia punktów w zależności od metody i od tego jak była implementowana. Dla najmniejszej tolerancji obie metody wzorcowo przydzielają punkty do kategorii „na linii”. Dla większej „dokładności” 1e-14 mniej pomyłek zwraca funkcja v1. Jednak ilość błędnie przydzielonych przez nią punktów szybko rośnie między 1e-14 a 1e-16. Dlatego w następnych dwóch dokładnościach lepiej wypada metoda v2, której ilość błędnie przydzielonych punktów zdaje się być na podobnym poziomie w trzech ostatnich dokładnościach.

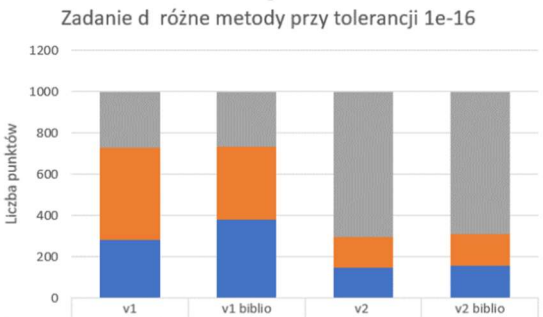
We wszystkich przypadkach, gdzie pojawiają się błędnie przydzielone punkty - funkcje biblioteczne fałszują wynik o około 10% więcej punktów, niż w przypadku implementacji własnej.

Analizując różnice między różnymi metodami dla konkretnych tolerancji również można zaobserwować lepsze wyniki v1 przy 1e-14, a gorsze przy dwóch „dokładniejszych” tolerancjach.

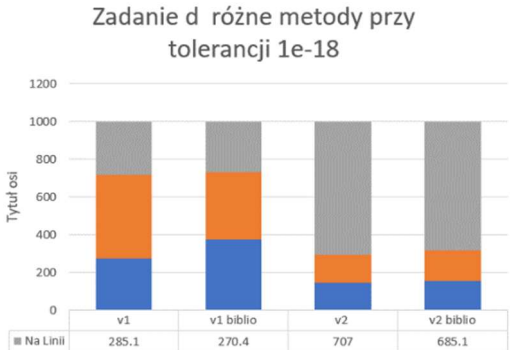

| | v1 | v1 biblio | v2 | v2 biblio |
|---|---|---|---|---|
| Na Linii | 1000 | 1000 | 1000 | 1000 |
| Lewo | 0 | 0 | 0 | 0 |
| Prawo | 0 | 0 | 0 | 0 |




| | v1 | v1 biblio | v2 | v2 biblio |
|---|---|---|---|---|
| Na Linii | 941.1 | 858.5 | 718.5 | 699 |
| Lewo | 58.8 | 111.9 | 139.3 | 154.9 |
| Prawo | 0.1 | 29.6 | 142.2 | 146.1 |

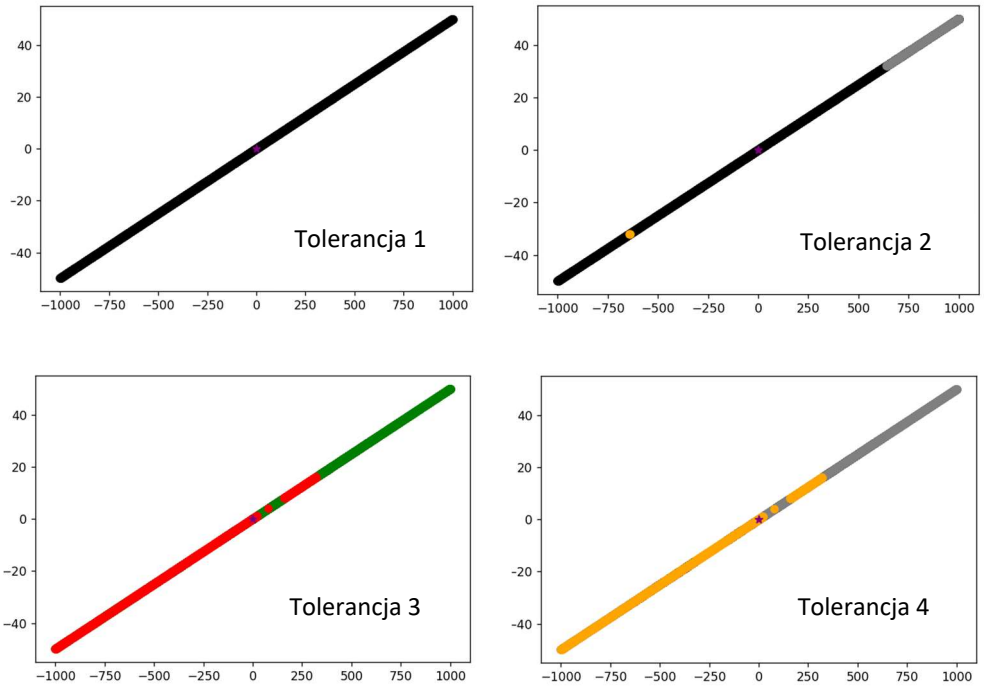



| | v1 | v1 biblio | v2 | v2 biblio |
|---|---|---|---|---|
| Na Linii | 270.5 | 265 | 702.9 | 687.7 |
| Lewo | 445.7 | 354.1 | 148 | 153.1 |
| Prawo | 283.8 | 380.9 | 149.1 | 159.2 |

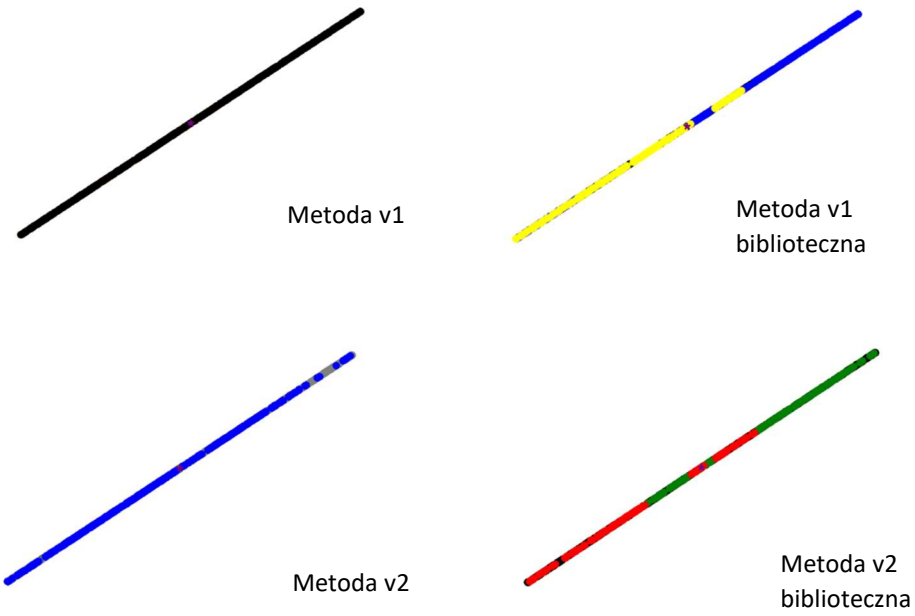



| | v1 | v1 biblio | v2 | v2 biblio |
|---|---|---|---|---|
| Na Linii | 285.1 | 270.4 | 707 | 685.1 |
| Lewo | 442 | 352.9 | 146.7 | 158.5 |
| Prawo | 272.9 | 376.7 | 146.3 | 156.4 |

# 8) Wizualizacja zadania d




Różnice między tolerancjami w obrębie jednej metody (podobne wyniki dla wszystkich metod – tu v2)




Różnice między metodami w obrębie jednej metody (podobne wyniki dla wszystkich tolerancji – tu 3)

Co interesujące, punkty błędnie przydzielane zdają się kumulować w grupy.

# 9) Wnioski

W pierwszych trzech zadaniach występuję bardzo małe zagęszczenie punktów przy prostej, stąd różnice w podziale punktów de facto nie występują(z małym wyjątkiem dla v1/v2 w zadaniu b, gdzie jednak odchylenia nie przekraczają 0,5 punktu). Dla zadania d różnice są widoczne, gdyż wszystkie punkty leżą na linii AB. Przy bardzo niskiej tolerancji znacząca część punktów jest wykrywana jako leżące po lewej bądź prawej stronie.

Istnieją dwie istotne granice w dokładności obliczeniowej – pierwsza między 1e-10 a 1e-14, gdzie z linii prostej w zadaniu d zaczynają być wyłapywane pierwsze punkty jako nienależące do niej oraz druga - między 1e-14 a 1e-16, gdzie metoda v1 staje się mniej dokładna niż v2. To może sugerować, że pierwsza granica stanowi miejsce bliskie dokładności obliczeniowej - „ucinanie" liczb po przecinku zaczyna być istotnym problemem. Pojawienie się drugiej granicy prawdopodobnie ilustruje różnice obliczeniową między dodawaniem a mnożeniem uciętych liczb(v1 – więcej mnożeń, v2 – więcej dodawań).

Metody biblioteczne są mniej dokładne(o około 10%) i znacznie wolniejsze(parokrotna różnica) niż te własnoręcznie zaimplementowane.