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.

1. **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

1. **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.

Obraz zawierający tekst

Opis wygenerowany automatycznie

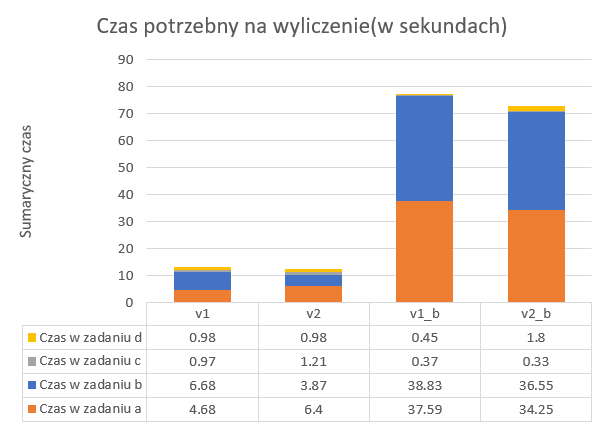
Generowanie losowych punktów

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.

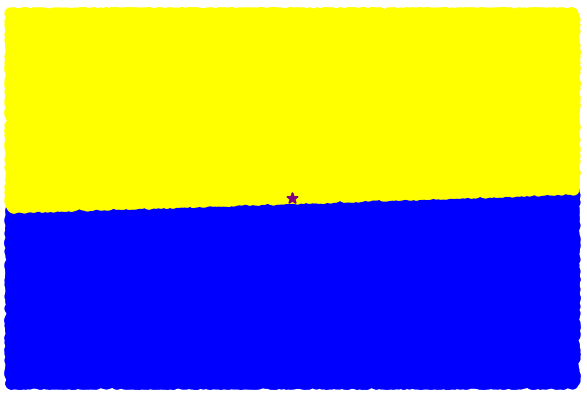
1. **Czasochłonność**

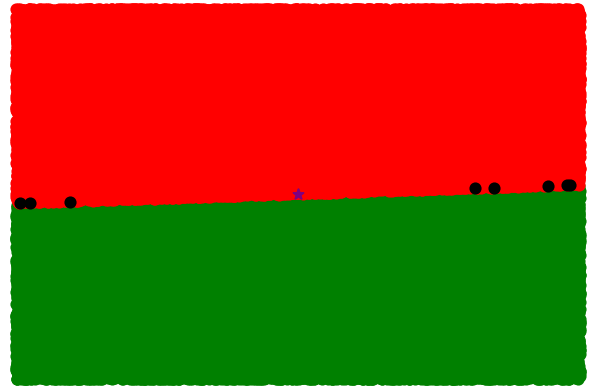
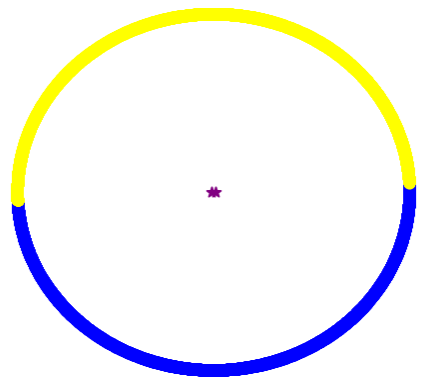
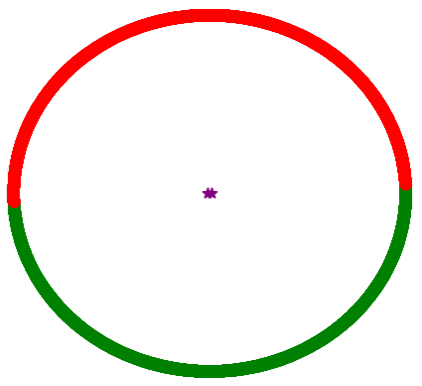
****

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.

1. **Wizualizacja zadań a, b i c**

****

****

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.

Zad c metoda v1

Zad c metoda v2

Zad b metoda v2

Zad b metoda v1

Zad b metoda v2 biblioteczna

Zad b metoda v1 biblioteczna

1. **Dane do zadań a, b i c**

Obraz zawierający stół

Opis wygenerowany automatycznie

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

Obraz zawierający stół

Opis wygenerowany automatycznie

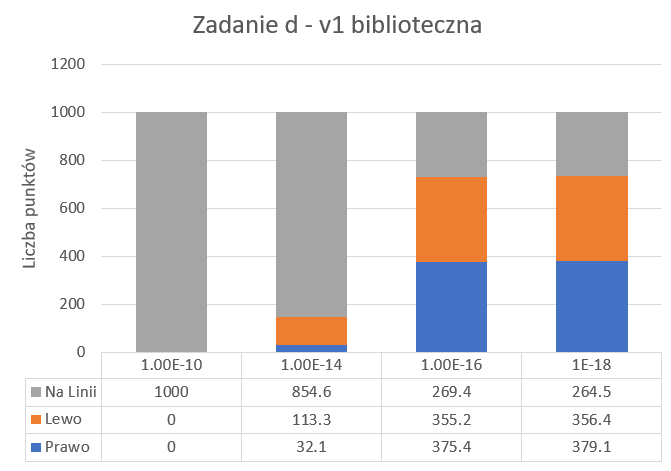
Tabela z wynikami dla przykładowej tolerancji 1e-16

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.

1. **Zadanie d**

Obraz zawierający stół

Opis wygenerowany automatycznie 

Obraz zawierający stół

Opis wygenerowany automatycznie Obraz zawierający stół

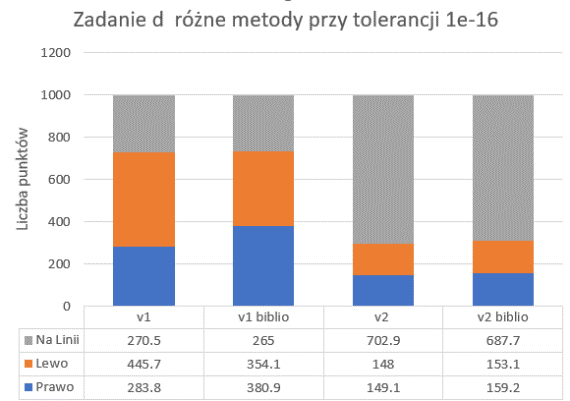
Opis wygenerowany automatycznie

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

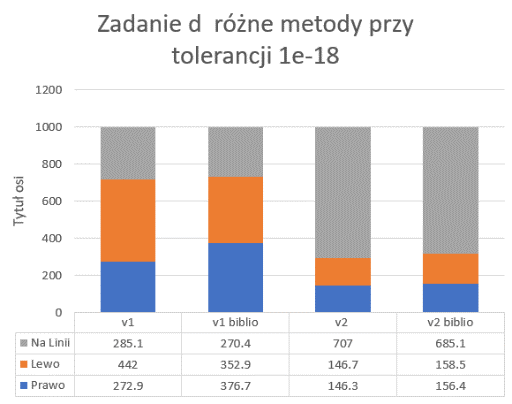
W zadaniu d widać znaczące różnice w wyliczaniu położenia punktów w zależności od metody i od tego jak była implementowana. Dla najmniejszej tolerancji obie metody wzorcowo przydzielają punkty do kategorii „na linii”. Dla większej „dokładności” 1e-14 mniej pomyłek zwraca funkcja v1. Jednak ilość błędnie 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.

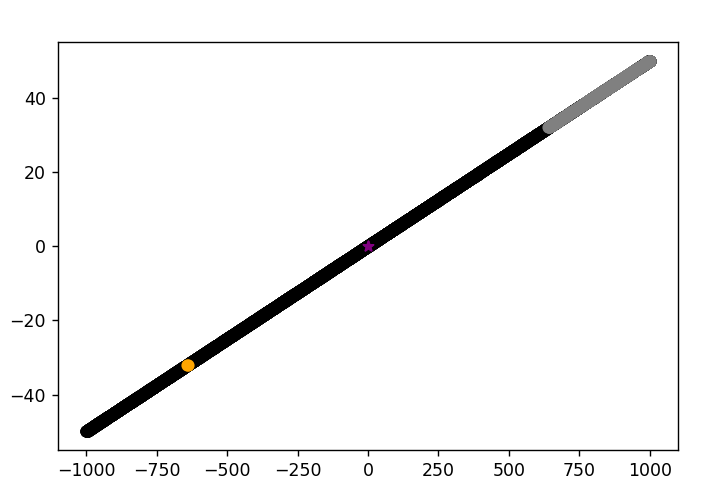
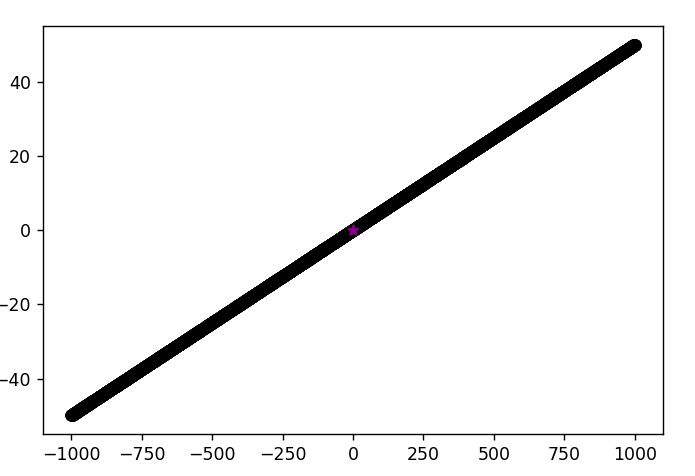
Analizując różnice między różnymi metodami dla konkretnych tolerancji również można zaobserwować lepsze wyniki v1 przy 1e-14, a gorsze przy dwóch „dokładniejszych” tolerancjach.

Obraz zawierający stół

Opis wygenerowany automatycznieObraz zawierający stół

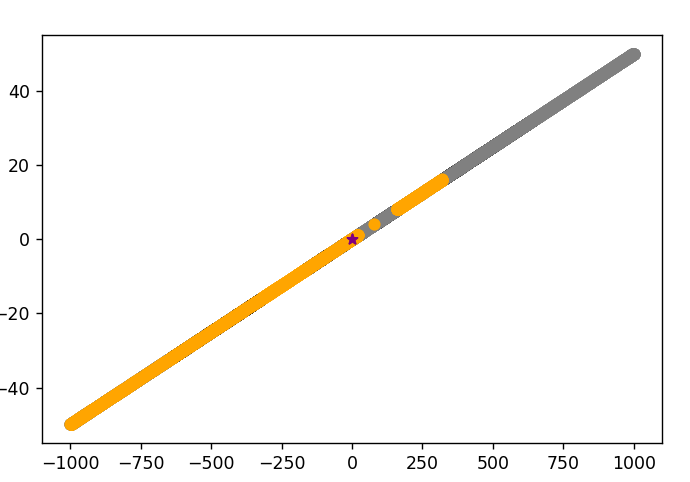
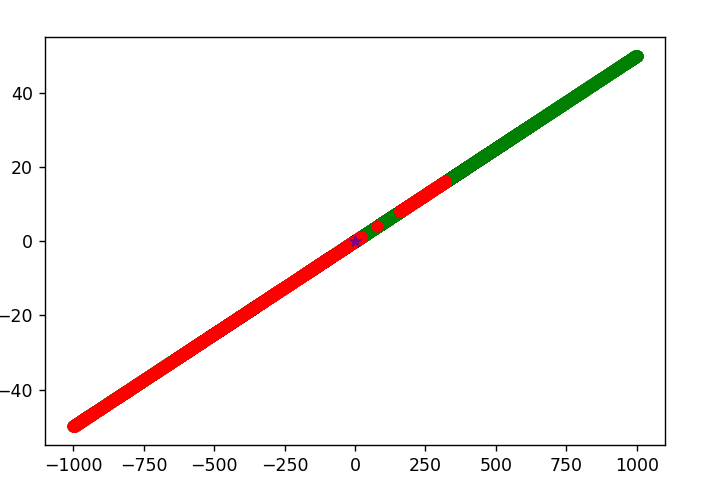
Opis wygenerowany automatycznie

1. Wizualizacja zadania d



Tolerancja 2

Tolerancja 1

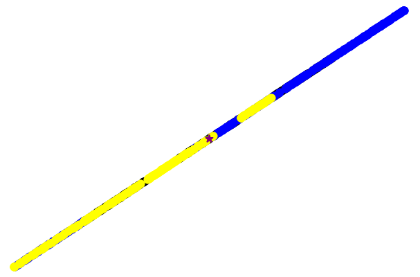
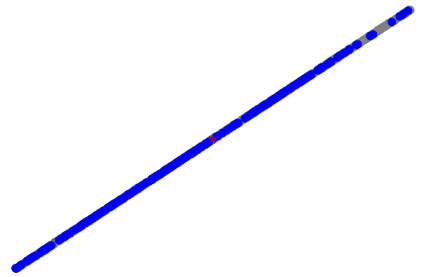
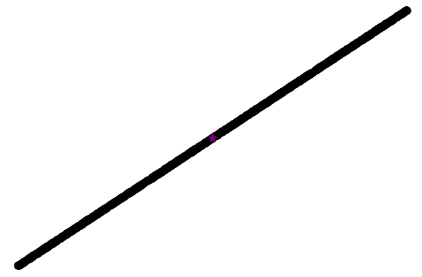


Tolerancja 4

Tolerancja 3

Różnice między tolerancjami w obrębie jednej metody

(podobne wyniki dla wszystkich metod – tu v2)



Obraz zawierający narzędzie, łopata, miotła

Opis wygenerowany automatycznie

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

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

Metoda v2 biblioteczna

Metoda v1

Metoda v2

Metoda v1 biblioteczna

1. 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.