

Algorytmy Numeryczne – Zadanie 2

Macierze – Eliminacja Gaussa

Kasper Cisewski 253902

Marcel Dajnowicz 253971

1. Wstęp

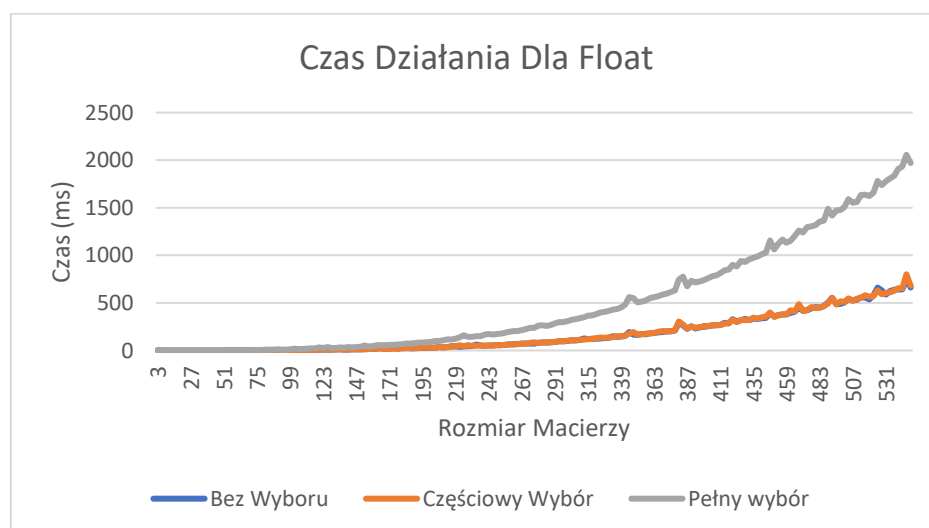
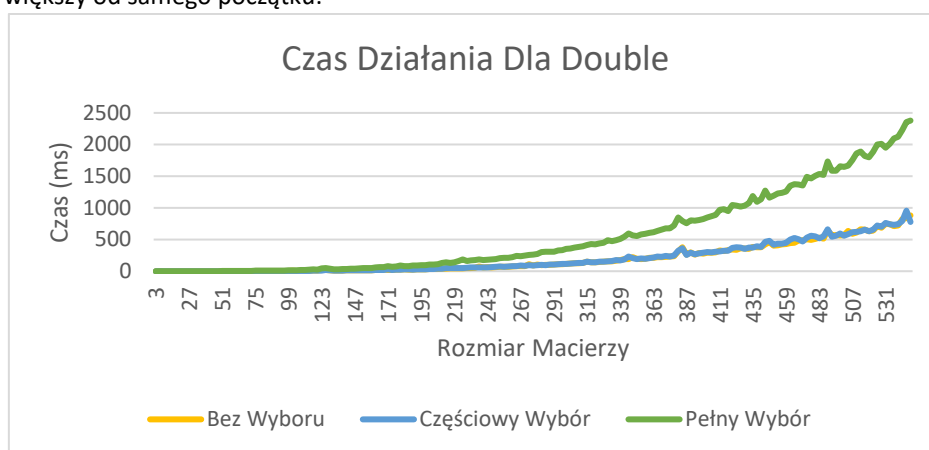
Zadaniem było zdefiniowanie generyczną klasę `MojaMacierz`, która reprezentuje macierz nad ciałem liczb rzeczywistych. Zaimplementowanie algorytmu eliminacji Gaussa w wariantach bez wyboru, częściowym i pełnym elementu podstawowego. Następnie dla macierzy kwadratowej A i wektora B rozwiązać układ równań liniowych. Testy przeprowadziliśmy używając typu `float`, `double` i własnego bezstratnego typu ułamkowego.

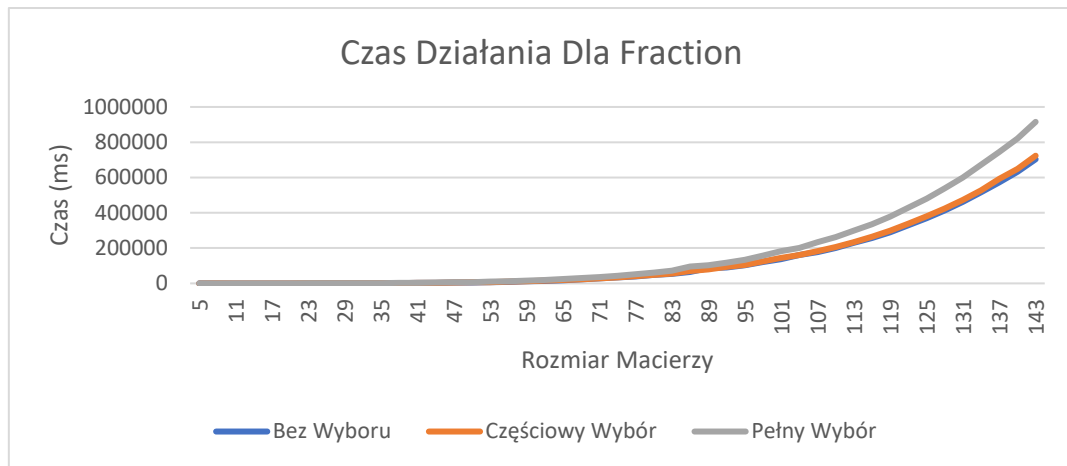
2. Poprawność

Poprawność algorytmów, które zostały przez nas napisane sprawdzaliśmy na macierzy 3×3 mnożąc wektor z ustalonymi liczbami. Dowiedliśmy poprawnie działający algorytm poprzez sprawdzenie wyników wyliczonych ręcznie i przez komputer.

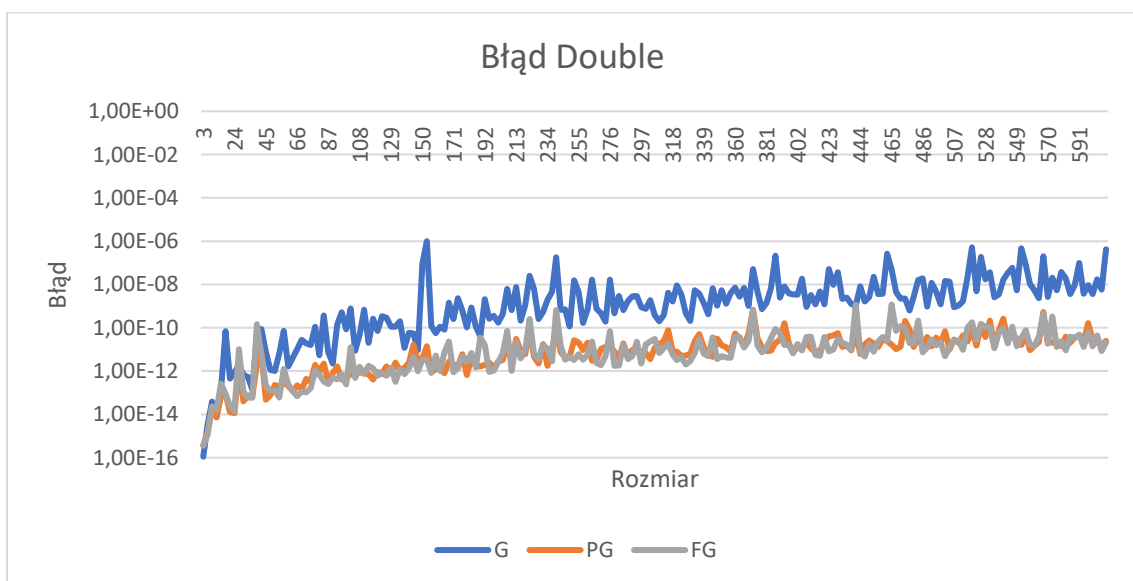
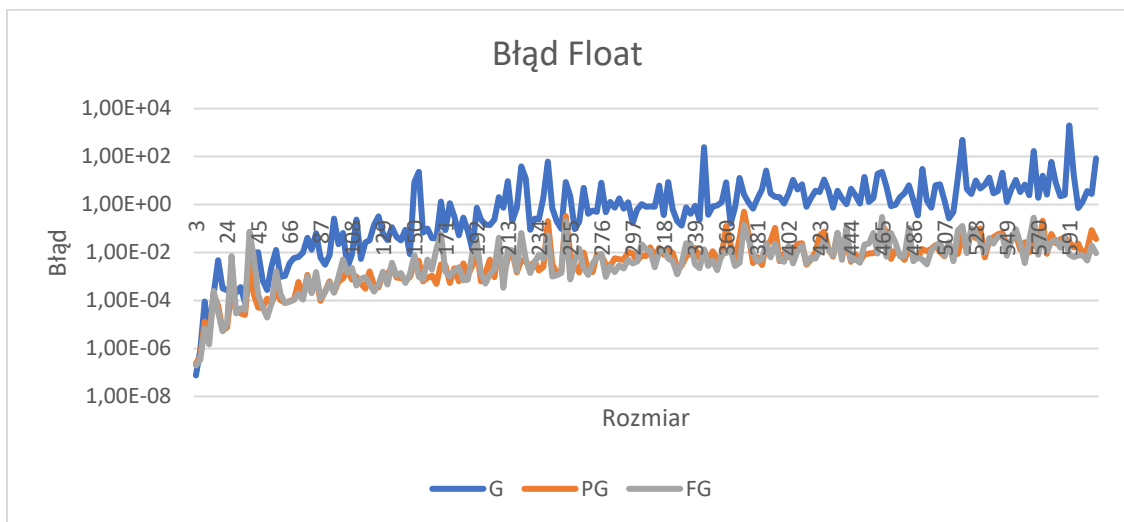
3. Hipotezy.

H1: Jak możemy zauważyć, każda następna, większa macierz wymaga coraz więcej czasu do policzenia. W przypadku `float` i `double`, czas znacząco rośnie dopiero od pewnego rozmiaru. Przy ułamku, czas jest o wiele większy od samego początku.





H2: Błąd liczony jest sumując kolejne różnice między elementami wektora X' oraz wektora wynikowego X wyliczone poprzez metodę eliminacji Gaussa $A \cdot X = B$.

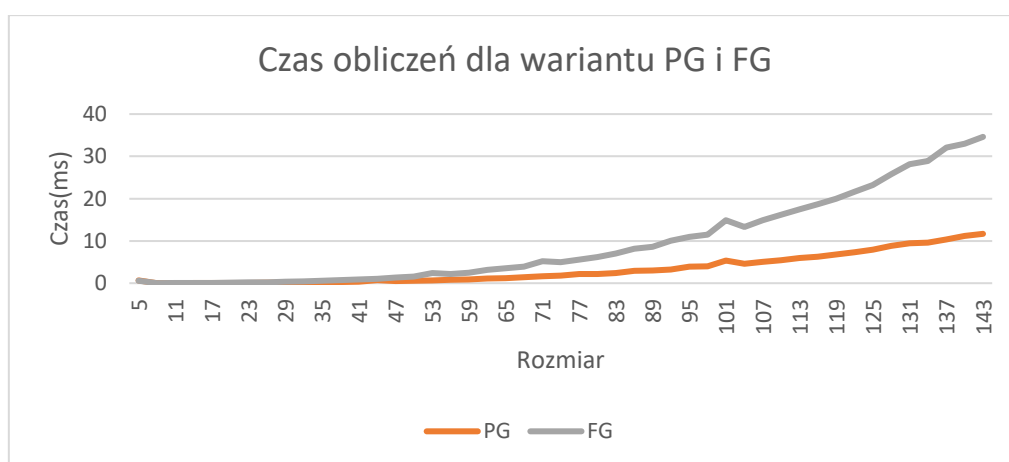


Po głębszej analizie danych, we trzech wariantach float'a, najmniejszy błąd otrzymywaliśmy z wariantu **FG**, który osiągnął najmniejszy błąd w 58% przypadków. W double, najmniejszy błąd również otrzymywał **FG** w 54% macierzy.

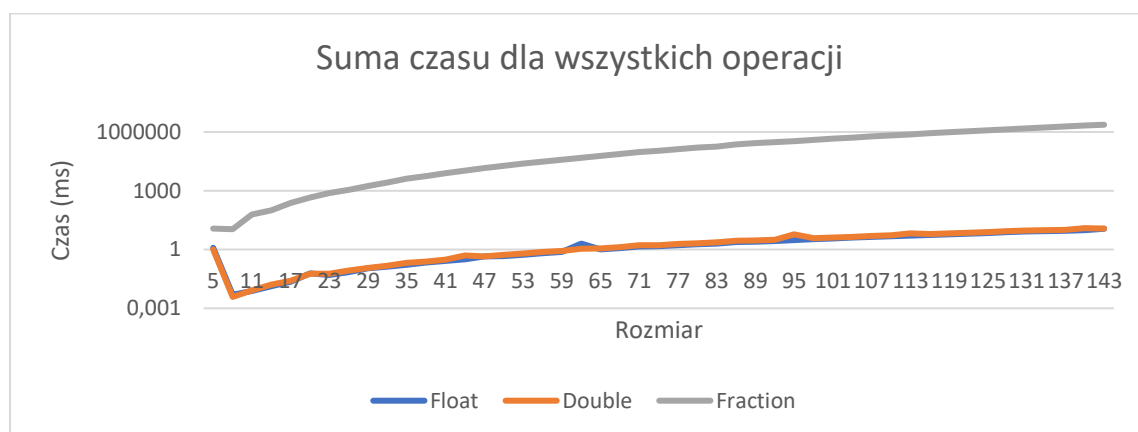
H3: Błąd przy użyciu własnej arytmetyki na ułamkach zapewnia bezbłędne wyniki poprzez liczenie na dwóch liczbach całkowitych, które nie podlegają zaokrąglaniu przy tym, że nie zachodzi żadna konwersja na wbudowane typy zmiennoprzecinkowe.

4. Pytania

Q1: Porównując wariant PG i FG zauważyć możemy wzrost wymaganego czasu potrzebnego do obliczeń wraz ze wzrostem rozmiaru macierzy. FG rośnie o wiele szybciej przez bardziej skomplikowane kalkulacje potrzebne w każdej iteracji.



Q2: Na poniższym wykresie zauważać możemy dużą różnicę między typem ułamkowym a float'em czy double'em dla wariantu bez wyboru elementu podstawowego. Float i double są na całej szerokości dość podobne ale po głębszej analizie float jest **41 razy szybszy** na 46 razy. Czas zwiększa się proporcjonalnie dla rozmiaru macierzy dla każdego typu.



5. Wyniki

Podsumowane wyniki dla wszystkich testów dla macierzy od 3 do 158 liczone co 3.

1.0 Tabela z pomiarami czasu dla poszczególnych metod liczenia i typów.

	Float	Double	Fraction
<i>Bez wyboru elementu podstawowego</i>	133,5902 ms	152,9772 ms	6220148 ms
<i>Z częściowym wyborem elementu podstawowego</i>	138,8284 ms	152,5962 ms	6383701 ms
<i>Z pełnym wyborem elementu podstawowego</i>	413,2433 ms	446,6439 ms	8099539,814 ms

6. Konfiguracja

Wszystkie test zostały wykonane na podzespołach:

- Procesor i5-8300H
- 20 GB Ram

7. Podział Pracy

Kasper	Marcel
Klasa Fraction	Klasa Matrix
Klasa Test	Napisanie Sprawozdania