11.11.2019

Algorytmy Numeryczne
Zadanie 2. Rozwiązywanie układów równań liniowych metodą eliminacji Gaussa

Agnieszka Harłozińska
253994

## 1. Wprowadzenie

Celem zadania było zdefiniowanie klasy parametryzowanej typem MyMatrix reprezentującą macierz nad ciałem liczb rzeczywistych oraz
implementacja algorytmu eliminacji Gaussa w trzech wariantach:

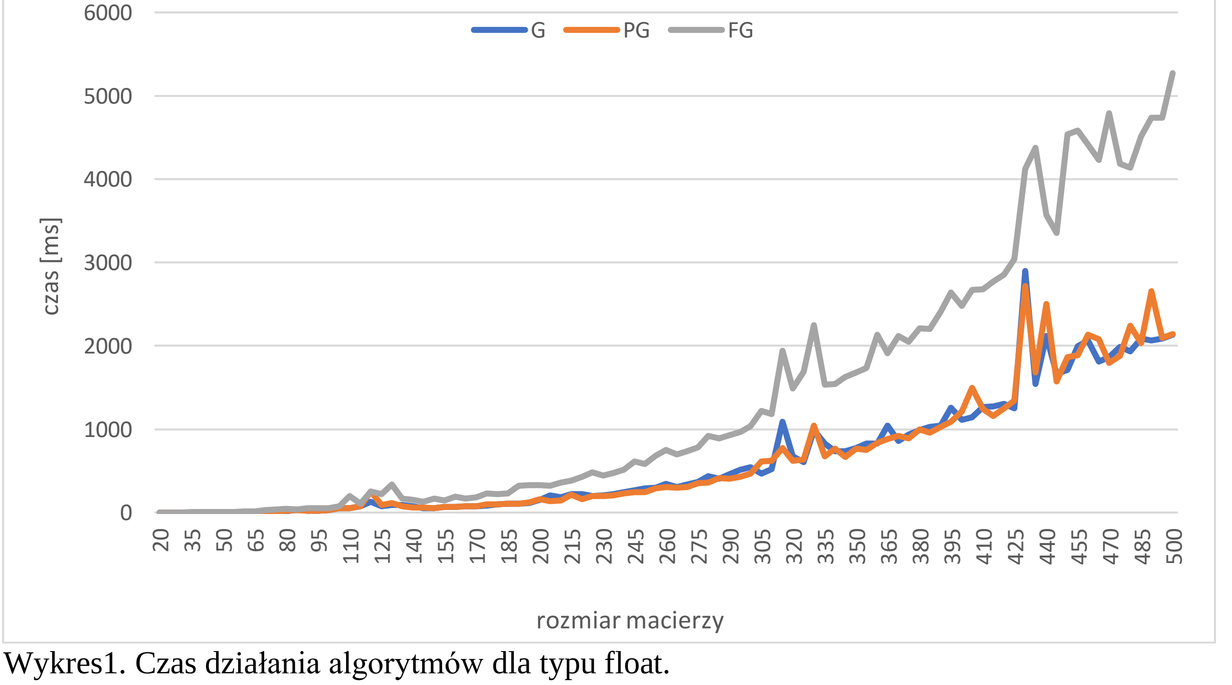
- G: bez wyboru elementu podstawowego,
- PG: z częściowym wyborem elementu podstawowego,
- FG: z pełnym wyborem elementu podstawowego.

## 2. Podejście do zagadnienia

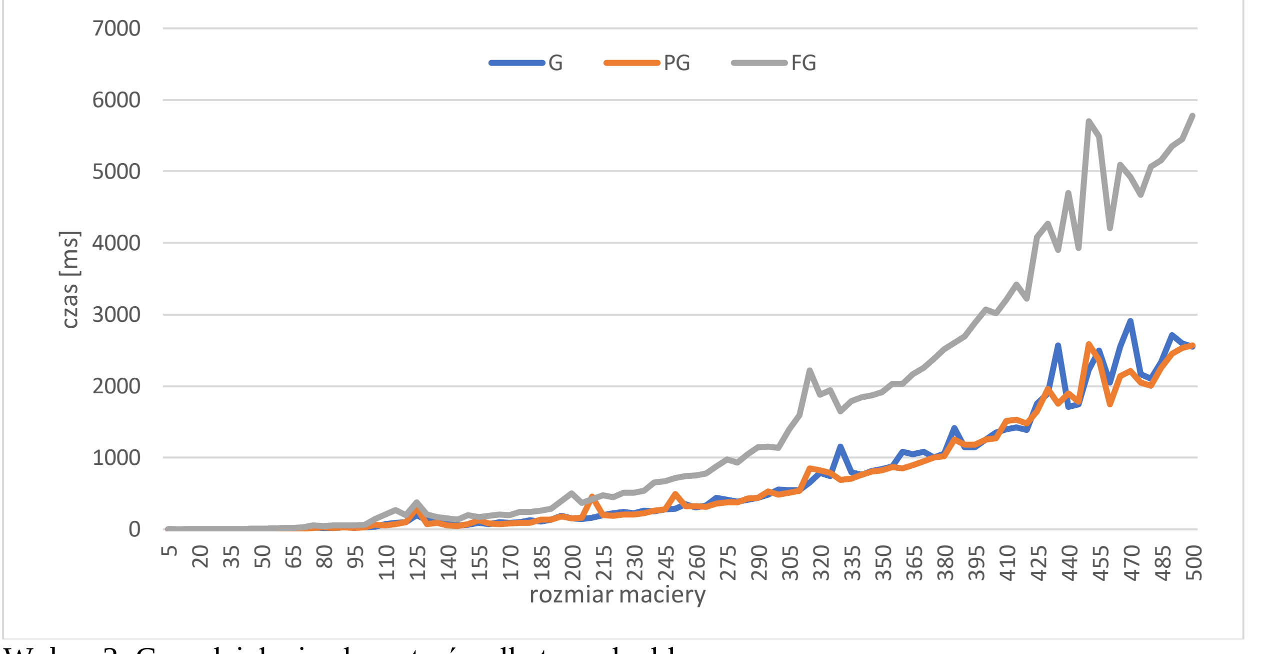
- Zadanie zostało wykonane przy użyciu języka C# w technologii .NET.
- Wykorzystano dodatkowo bibliotekę Miscellaneous Utility Library (MiscUtil) wspomagającą operacje na typach generycznych.
- Testy przeprowadzono dla trzech różnych typów reprezentujących liczbę rzeczywistą: float, double oraz własnego typu ułamkowego zaimplementowanego w klasie Fraction.

## 3. Analiza problemu

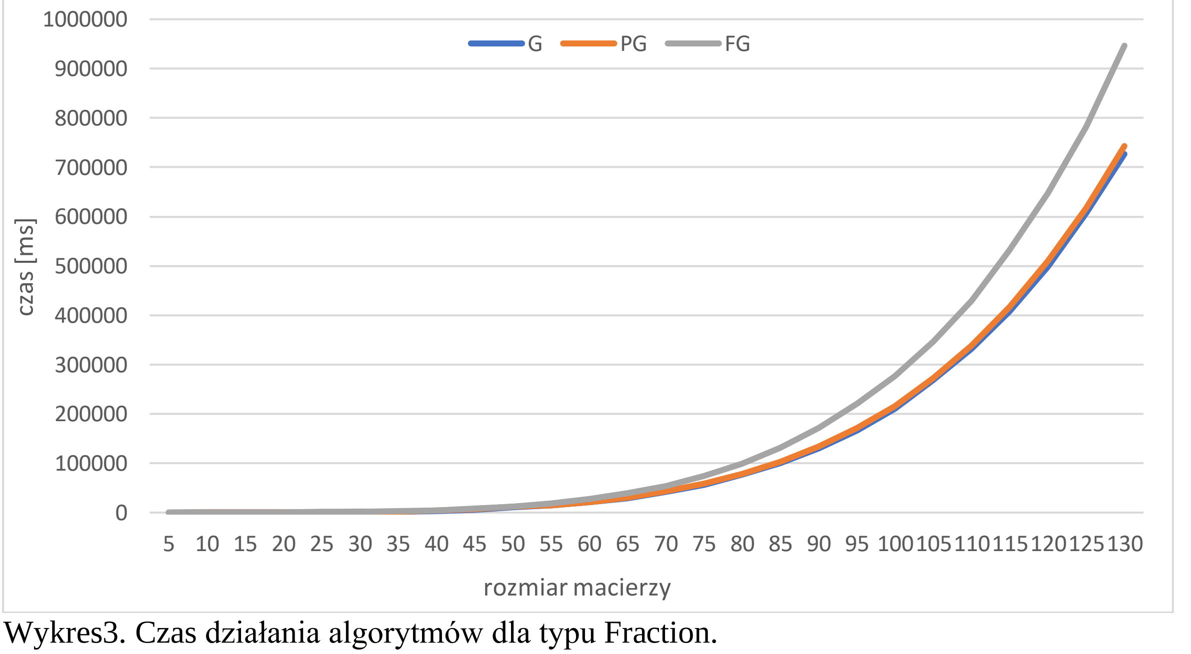
H1: Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (G, PG, FG) rośnie.
Wyraźnie różnice w czasie wykonywania dostrzegalne są dla większych rozmiarów macierzy.
Ciężko jest zauważyć różnicę pomiędzy G i PG (przykładowo dla macierzy o rozmiarze 150 badając typ float czasy dla wersji G i PG wyniosły kolejno 51,15 ms i 52,66 ms). Czas działania w wersji FG zdecydowanie jest większy od pozostałych.




Wykres1. Czas działania algorytmów dla typu float.

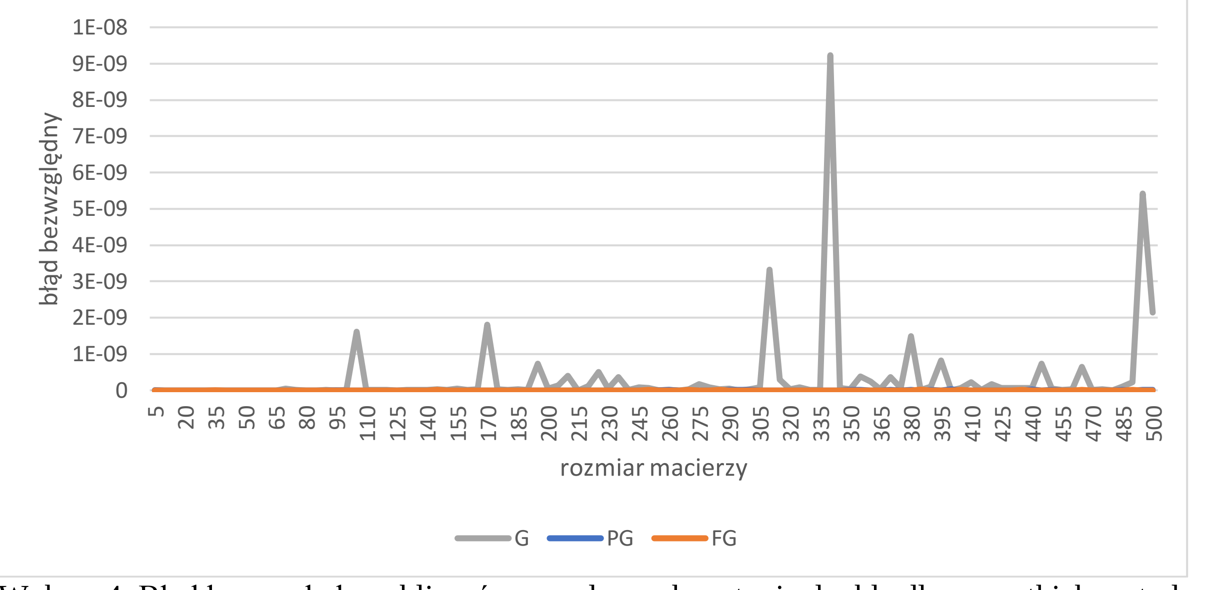



Wykres2. Czas działania algorytmów dla typu double.

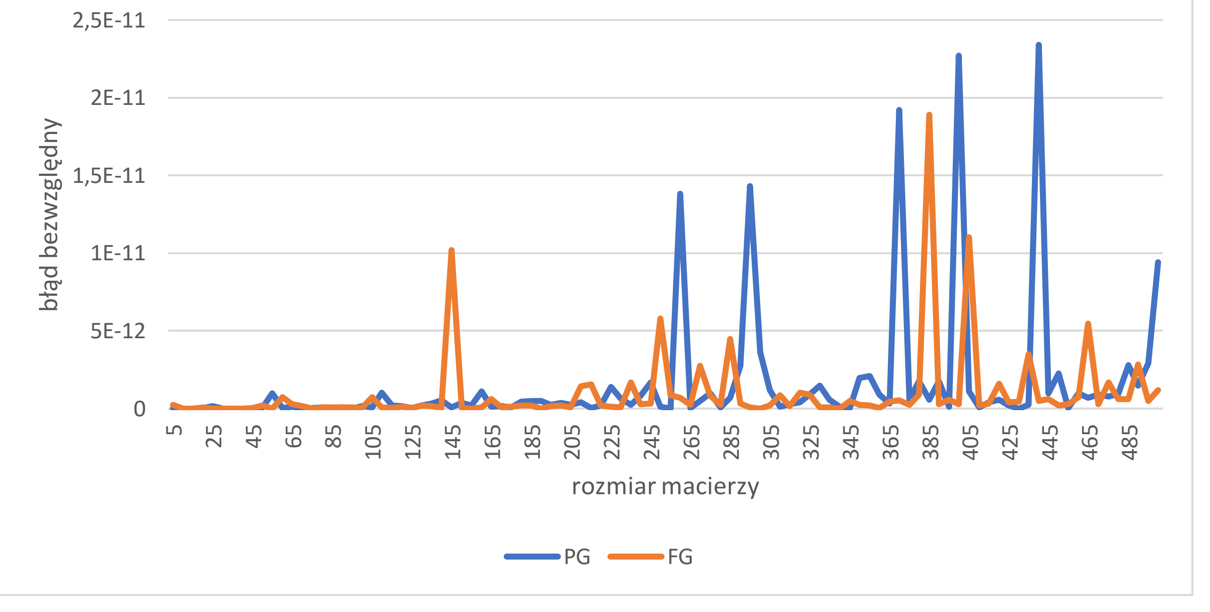



Wykres3. Czas działania algorytmów dla typu Fraction.

H2: Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (G, PG, FG) maleje.
Na wykresie 4 dotyczącym typu double bez wątpienia można stwierdzić, że metoda G generuje największy błąd. Ciężko rozstrzygnąć kwestię błędów między metodami PG i FG (nawet na wykresie 5, który przedstawia tylko te dwie metody). Powołując się na średnią błędu z badań na rozmiarach macierzy 5-500 , która dla PG jest rzędu $10^{-12}$, dla FG $10^{-13}$ można potwierdzić poprawność hipotezy H2. Dla typu float zachodzi analogiczna sytuacja.




Wykres 4. Błąd bezwzględny obliczeń prowadzonych na typie double dla wszystkich metod.

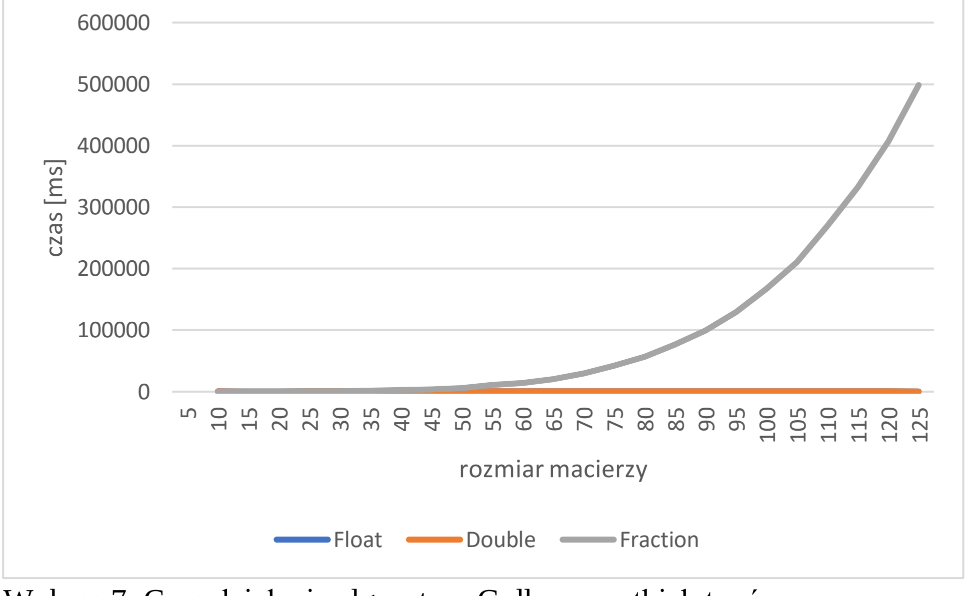



Wykres 5. Błąd bezwzględny obliczeń prowadzonych na typie double dla PG i FG.

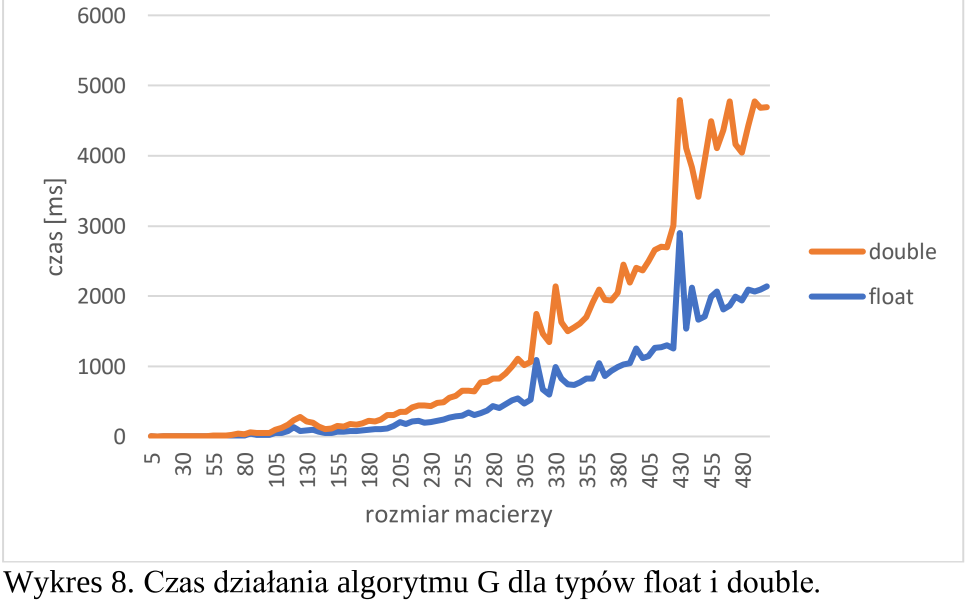
H3: Użycie własnej arytmetyki na ułamkach zapewnia bezbłędne wyniki niezależnie od wariantu metody Gaussa i rozmiaru macierzy.
Własna arytmetyka na ułamkach zapewnia bezbłędne wyniki ze względu na pomięcie zaokrągleń, ponieważ są stosowane tylko operacje na liczbach całkowitych.

Q1: Jak zależy dokładność obliczeń od rozmiaru macierzy dla dwóch wybranych przez Ciebie wariantów metody Gaussa gdy obliczenia prowadzone są na typie podwójnej precyzji (TD)?
Na wykresie 5 przedstawiono błąd bezwzględny dla metod z częściowym i pełnym wyborem, w obu przypadkach błędy ulegają wahaniom, jednak dla większych rozmiarów macierzy (powyżej 250) tendencje do wahań są znacznie większe. Wahania te utrzymują się w graniach rzędów $10^{-11} \div 10^{-15}$.

Q2: Jak przy wybranym przez Ciebie wariancie metody Gaussa zależy czas działania algorytmu od rozmiaru macierzy i różnych typów?
Na wykresach 7 i 8 rozpatrzono metodę bez wyboru elementu podstawowego. Działania na typie Fraction są zdecydowanie najbardziej czasochłonne, przykładowo dla macierzy o rozmiarze 100 czasy działania na typach float, double i Fraction kolejno wynoszą 21,86 ms, 31,61 ms i 129894 ms. Przytłaczającą różnicę pomiędzy typami wbudowanymi a własną implementacją typu ułamkowego można dostrzec na wykresie dla macierzy o rozmiarach większym niż 60. Ograniczając się do obserwacji wydajności obliczeń na typach float i double (wykres 8) można stwierdzić, że operacje na typie float są szybsze. Istotną rozbieżność można zauważyć od rozmiaru macierzy równego 200.




Wykres 7. Czas działania algorytmu G dla wszystkich typów.




Wykres 8. Czas działania algorytmu G dla typów float i double.

## 4. Wydajność implementacji

E1: Podaj czasy rozwiązania układu równań uzyskane dla macierzy o rozmiarze 500
dla 9 testowanych wariantów.
Dla typu Fraction podano czas obliczeń dla macierzy o rozmiarze 130 (ze względu na zbyt długi czas trwania testów dla większych rozmiarów)

| Typ danych | Wariant algorytmu | Czas rozwiązania [ms] |
|---|---|---|
| float | G | 2137,033 |
| | PG | 2142,660 |
| | FG | 5271,362 |
| double | G | 2555,008 |
| | PG | 2565,953 |
| | FG | 5778,396 |
| Fraction (130x130) | G | 726782,6 |
| | PG | 743026,0 |
| | FG | 946467,9 |

Tabela 1. Zestawienie wydajności implementacji

## 5. Konfiguracja sprzętowa

Procesor: Intel Core i5-4210U 4x1.70GHz