

## Algorytmy Numeryczne - zadanie 2

## Rozwiązanie układów równań liniowych metodą eliminacji Gaussa

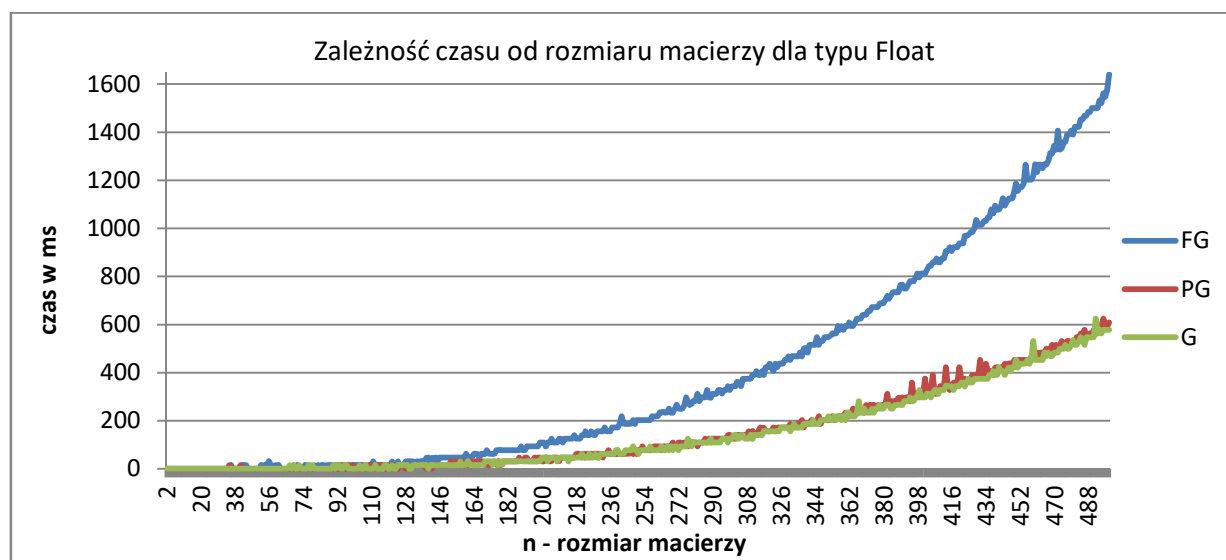
Zadanie polegało na zaimplementowaniu algorytmu Gaussa-Jordana dla macierzy różnych typów i rozmiarów. Algorytm miał być zaimplementowany w 3 wariantach - (P) bez wyboru elementu podstawowego, (PG) z częściowym, (FG) pełnym wyborem, dla różnych typów danych – (TF) Float, (TD) Double, (TC) własnego typu - Ułamek

W celu porównania działania algorytmów napisałem program w języku Java, którego kod źródłowy jest dołączony do sprawozdania. Testy sprawdzające czas wykonania były uruchamiane na komputerze z procesorem Intel Core i5-8250U, 12GB RAMu, systemem Windows 10 Home. Aby otrzymać jak najdokładniejsze wyniki prędkość taktowania procesora została ustawiona sztywno na 1.6GHz. Otrzymane dane znajdują się na poniższych wykresach.

**H1: Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (G, PG, FG) rośnie.**

Z przeprowadzonych przeze mnie prób wynika, że czas FG jest wyraźnie większy od G oraz PG.

Czasy G i PG są bardzo zbliżone, przy czym PG jest minimalnie dłuższy.

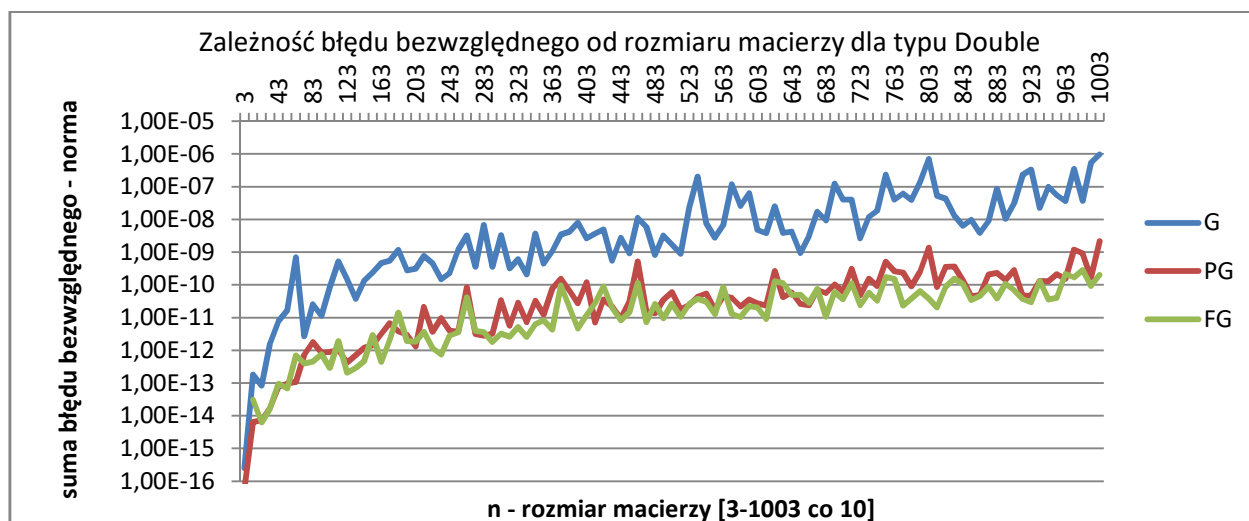


rysunek 1

**H2: Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (G, PG, FG) maleje.**

Z przeprowadzonych przeze mnie testów wynika, że błędy dla G są wyraźnie większe niż PG i FG.

Błędy PG i FG są zbliżone, przy czym dla PG są minimalnie większe niż dla FG.



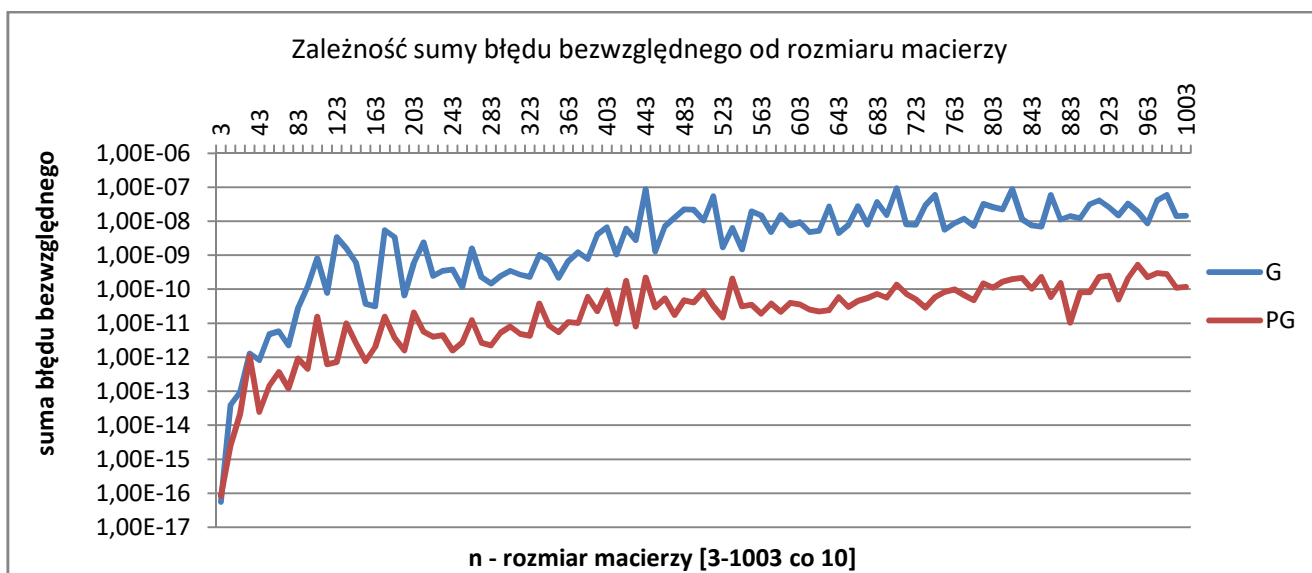
rysunek 2

**H3: Użycie własnej arytmetyki na ułamkach zapewnia bezbłędne wyniki niezależnie od wariantu metody Gaussa i rozmiaru macierzy.**

Dla testowanych przeze mnie danych we wszystkich wariantach metody Gaussa otrzymałem bezbłędne wyniki. Z powodu dużo większego czasu potrzebnego do obliczeń ograniczyłem rozmiar macierzy do wielkości od 2 do 50.

**Q1: Jak zależy dokładność obliczeń (błąd) od rozmiaru macierzy dla dwóch wybranych przez Ciebie wariantów metody Gaussa gdy obliczenia prowadzone są na typie podwójnej precyzji (TD)?**

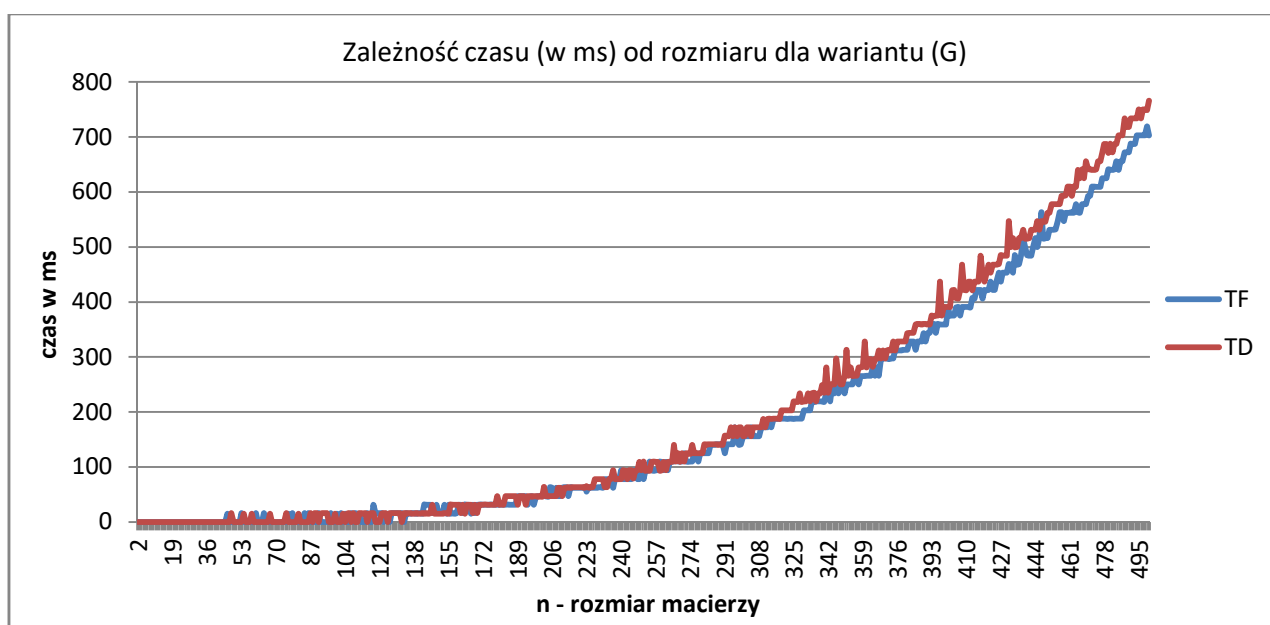
Zgodnie z otrzymanymi wynikami, błąd rośnie wraz ze wzrostem macierzy. Metoda PG daje dokładniejsze wyniki niż metoda G dla typu Double.



rysunek 3

**Q2: Jak przy wybranym przez Ciebie wariantcie metody Gaussa zależy czas działania algorytmu od rozmiaru macierzy i różnych typów?**

Na podstawie testów, obliczenia na typie Double i Float trwają praktycznie tyle samo, jednak dla większych macierzy obliczenia na typie Double są niewiele wolniejsze niż na typie Float.

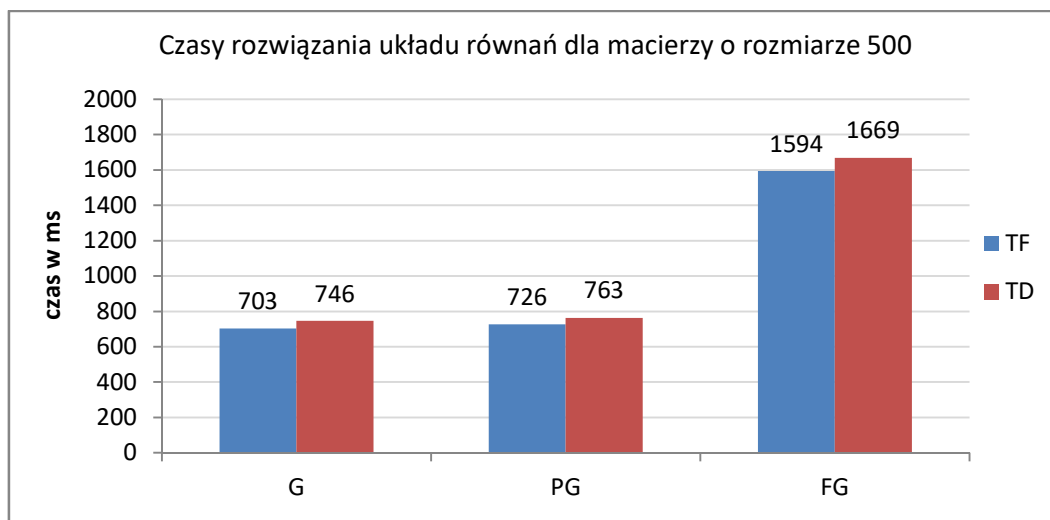


rysunek 4

**E1: Podaj czasy rozwiązania układu równań uzyskane dla macierzy o rozmiarze 500 dla wszystkich testowanych wariantów.**  
Ze względu na bardzo długi czas wykonania nie testowałem macierzy o rozmiarze 500 w wariancie TC – z własnym typem  
‘Ułamek’.

Otrzymane średnie czasów (w ms) dla 6 pozostałych wariantów:

	Float (TF)	Double (TD)
<b>gauss (G)</b>	703ms	746ms
<b>gaussCzesciowy (PG)</b>	726ms	763ms
<b>gaussPelny (FG)</b>	1594ms	1669ms



rysunek 5

Dowód poprawności implementacji mnożenia macierzy:

Macierz A \* wektor X = wektor B

Macierz A:		
-0.4842681884765625	0.5939788818359375	0.53973388671875
-0.2791748046875	-0.4540863037109375	0.848052978515625
-0.3098297119140625	-0.4558868408203125	-0.685760498046875
Wektor X:		
-0.444976806640625	0.69561767578125	0.760955810546875
Wektor B:		
1.0393839585594833	0.4536866955459118	-0.7010893444530666

W moim programie przemnożenie macierzy A przez wektor X dało wynik wektora B dla typu Double.