

# Algorytmy Numeryczne - Zadanie 2

Patryk Siewruk (246769)

Michał Sławiński (246803)

Kevin Sarfo (246801)

## Operacje na macierzach

Sprawozdanie dotyczy operacji na macierzach z zaimplementowanym algorytmem eliminacji Gaussa w następujących wariantach:

G: bez wyboru elementu podstawowego,

PG: z częściowym wyborem elementu podstawowego,

FG: z pełnym wyborem elementu podstawowego.

Testy poprawnościowe i wydajnościowe zostały przeprowadzone dla trzech typów reprezentujących liczbę rzeczywistą: Float, Double, Fraction (typ własny przechowujący liczbę w postaci ułamka liczb całkowitych).

Konfiguracja sprzętowa: procesor Intel Core i5 6600k 4.8Ghz, 16 GB RAM DDR4,  
system operacyjny Windows 10

Język programowania: **Java (v. 1.8.0\_151)**.

Wszystkie obliczenia przeprowadzono na macierzach o wartościach z przedziału  $[-65536, 65535]$  otrzymanych jako iloraz  $r/2^{16}$  (gdzie  $r$  to pseudolosowa liczba całkowita z przedziału:  $[-2^{16}, 2^{16} - 1]$ ). W celu weryfikacji poprawności hipotez zawartych w treści polecenia zadania, dla każdego z typów wygenerowano pseudolosowo:

-macierz  $A$

-wektor  $X$

-wektor  $B$  (iloczyn  $A * X$ )

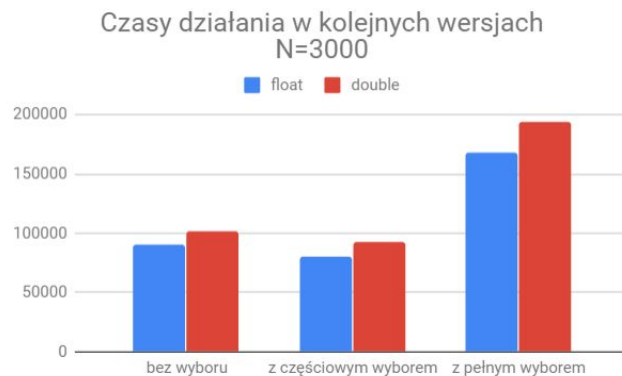
Następnie rozwiązano układ równań  $A * X = B$  korzystając z zaimplementowanych wariantów metody Gaussa.

W celu weryfikacji hipotez zawartych w treści polecenia przeprowadzono testy dla różnych rozmiarów macierzy i wektorów:

- 100, 200, 500, 1000, 2000, 3000 - dla Float i Double
- 10, 20, 50, 100, 200 - dla Fraction

**H1: Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (1, 2, 3) rośnie**

Float i Double



Fraction



Jak widać na załączonych wykresach, najkrótszym czasem działania charakteryzuje się wariant z częściowym wyborem a najdłuższym wariant z pełnym wyborem. Zatem czas działania rośnie kolejno w wersjach (2,1,3).

**H2: Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (1, 2, 3) maleje.**

W celu sprawdzenia poprawności hipotezy przeprowadzono testy dla typów float i double. Dla wszystkich testowanych rozmiarów założenie zostało potwierdzone.

	Float	Double
bez wyboru	0.87705576	0.0000000032739687
z częściowym wyborem	0.0018373015	0.0000000000040787
z pełnym wyborem	0.0011989941	0.0000000000021305

W tabeli wyraźnie widać że kolejne warianty algorytmu eliminacji Gaussa uzyskują coraz to mniejszy błąd. Największa różnicę można zaobserwować pomiędzy wariantem G (bez wyboru) i PG (z częściowym wyborem).

W przypadku typu fraction dla każdego testowanego rozmiaru i wariantu błąd był równy zero.

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

Aby sprawdzić poprawność hipotezy użyto typu własnego Fraction

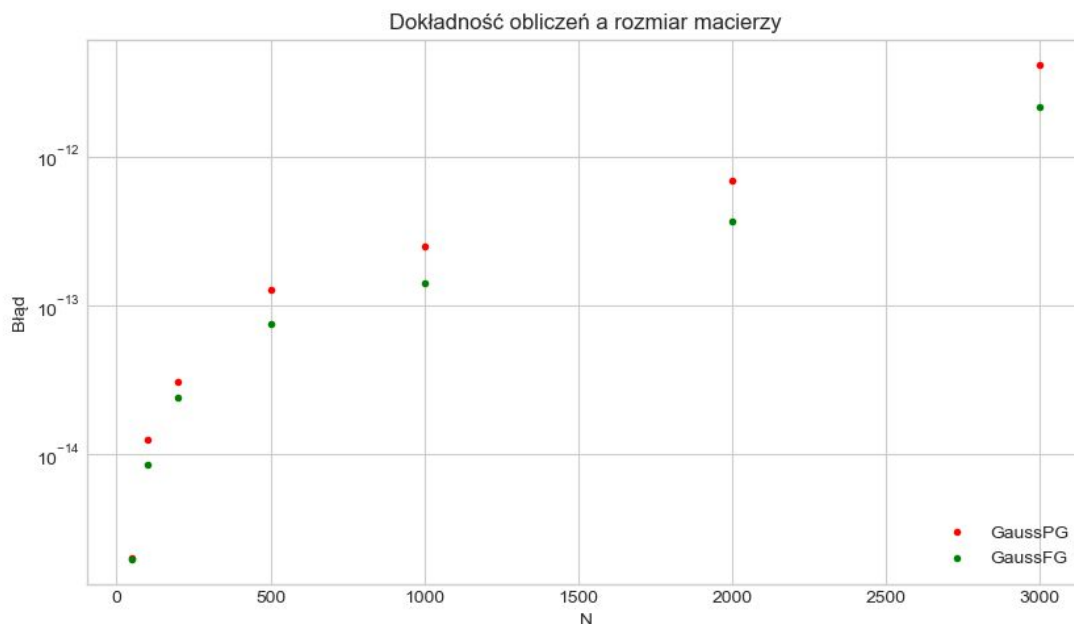
Testy zostały przeprowadzone dla różnych rozmiarów macierzy i wektorów (10,20,50,100,200) i jednoznacznie potwierdziły poprawność hipotezy.

W każdym z przypadków dla wszystkich ww. rozmiarów program zwracał wartości błędów bezwzględnych równe zero w przeciwieństwie do innych typów:

	Float	Double	Fraction
bez wyboru	0.001289829	0.0000000000002895453	0
z częściowym wyborem	0.000015934258	0.0000000000000031007	0
z pełnym wyborem	0.000011564009	0.0000000000000024055	0

**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)?**

#### Double



Dla typu double i wariantów PG (częściowy) oraz FG (pełny) wraz ze wzrostem rozmiaru macierzy, rośnie błąd bezwzględny. Na powyższym wykresie wykazano sposób w jaki zwiększa się błąd.

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

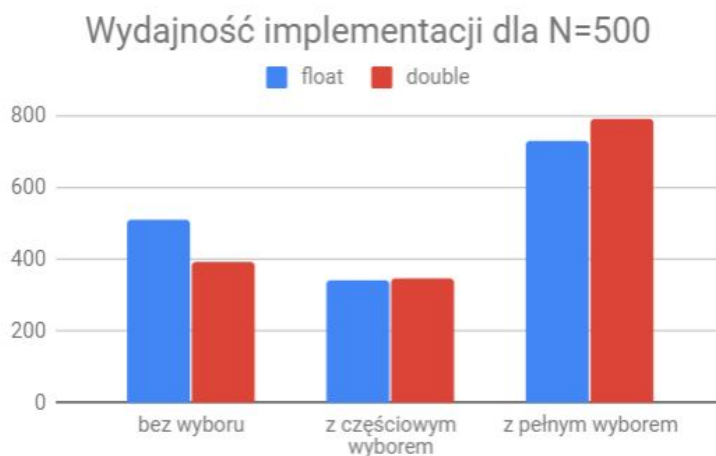
Wariant z pełnym wyborem elementu podstawowego

	100	200	500	1000	2000	3000
Double	9 ms	64 ms	790 ms	6 220 ms	52 957 ms	194 359 ms
Float	8 ms	59 ms	733 ms	5 678 ms	47 063 ms	168 446 ms

	10	20	50	100	200
Fraction	5 ms	47 ms	1 879 ms	44 902 ms	1 198 841 ms

Wyraźnie widać że wraz ze wzrostem rozmiaru macierzy, znacznie wzrasta czas działania algorytmu. Największe wzrosty występują dla typu fraction, następnie double a najmniejsza dla typu float.

### E1: Wydajność implementacji



Dla ułamków czas wykonywania macierzy rozmiaru N = 500 przekroczyłby godzinę.

**Michał Sławiński:** klasa Fraction, klasa RandomGenerator, metody zapisujące wyniki do plików, implementacja eliminacji Gaussa z pełnym wyborem elementu podstawowego.

**Patryk Siewruk:** klasa MyMatrix, klasa MyMath, implementacja generyczności w programie, implementacja eliminacji Gaussa z częściowym wyborem elementu podstawowego, pomoc w sprawozdaniu.

**Kevin Sarfo:** klasa MatrixSet, metody plus, minus, times, absVector w MyMatrix, pomiar czasów działania, implementacja eliminacji Gaussa bez wyboru elementu podstawowego, sprawozdanie