

# Implementacja algorytmu eliminacji Gaussa oraz testy wydajnościowe i obliczeniowe trzech określonych typów

Grams, Stanisław  
Jeziński, Maciej  
Korczakowski, Juliusz  
MFI UG  
Algorytmy Numeryczne

25 listopada 2018

## 1 Operacje na macierzach

### 1.1 O sprawozdaniu

Sprawozdanie prezentuje analizę wydajności i poprawności implementacji algorytmu eliminacji Gaussa, dla macierzy kwadratowej  $A$  oraz wektora  $B$  w układzie liniowym  $A * X = B$  gdzie współczynniki wektora  $A$  oraz macierzy kwadratowej  $X$  są losowane.

Zakres losowanych współczynników jest równy  $[-1, 1]$ , natomiast każdy z nich został wyznaczony według wzoru  $r/2^{16}$ , gdzie  $r$  jest losową liczbą całkowitą z zakresu  $[-2^{16}, 2^{16}]$ . Wygenerowany w ten sposób wektor  $X$  pozostaje próbą kontrolną natomiast do algorytmu jako parametr przekazujemy wyliczony według powyższego wzoru wektor  $B$ .

### 1.2 O implementacji

Program „*gauss*” został napisany w języku C++ z użyciem bibliotek:

- „*gmp.h*” — pozwalającej zaimplementować wymagany typ całkowity (TC)
- „*pthread.h*” — do celów obsługi wielowątkowości,

Wyniki działania programu zapisywane są do poszczególnych plików *\*.csv*. Zaimplementowano następujące warianty algorytmu Gaussa dla kolejnych typów zmiennych rzeczywistych:

Warianty algorytmu eliminacji

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

- FG: z całkowitym wyborem elementu podstawowego.

Typy zmiennych rzeczywistych

- float – wbudowany typ pojedynczej precyzji
- double – wbudowany typ podwójnej precyzji
- MyType – typ zaimplementowany przez zespół, oparty o bibliotekę „GMP.h” i typ „mpq\_t”

### 1.3 Postawione hipotezy

**Hipoteza 1.** Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (G,PG,FG) rośnie.

**Hipoteza 2.** Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (G, PG, FG) maleje.

**Hipoteza 3.** Użycie własnej arytmetyki na ułamkach zapewnia bezbłędne wyniki niezależnie od wariantu metody Gaussa i rozmiaru macierzy.

### 1.4 Wykresy

Wykresy (w postaci wektorowej) załączono w katalogu „*report/plots/*”

Do narysowania wykresów użyto oprogramowania „*GNU Octave*” wraz z pakietem „*gnuplot*”. Skrypty załączono w katalogu „*scripts/*”.

## 2 Poprawność hipotez

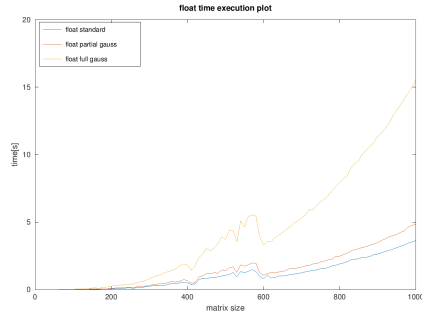
### 2.1 Czas wynonywania

**Hipoteza 1.** Dla dowolnego ustalonego rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (G,PG,FG) rośnie.

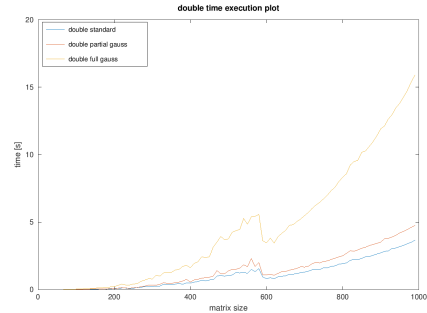
Analiza wykresów przedstawiających dla typów float oraz double (typ własny pomijamy na rzecz trzeciej hipotezy), pozwala na wyciągnięcie następujących wniosków:

1. W przypadku typów float oraz double kolejne typy metody Gaussa wykazują coraz większą czasochłonność jednak największą różnicę widać pomiędzy FG a pozostawłymi.
2. W przypadku typu własnego różnice są minimalne, mimo tendencji potwierdzającej hipotezę zdarzają się momenty w których wykres jej przeczy.
3. W przypadku typu własnego wzrost jest prawie paraboliczny, jednak dla typów float oraz double krzywe wykresów są z początku bardzo nieregularne aby potem zbliżyć się do paraboli.

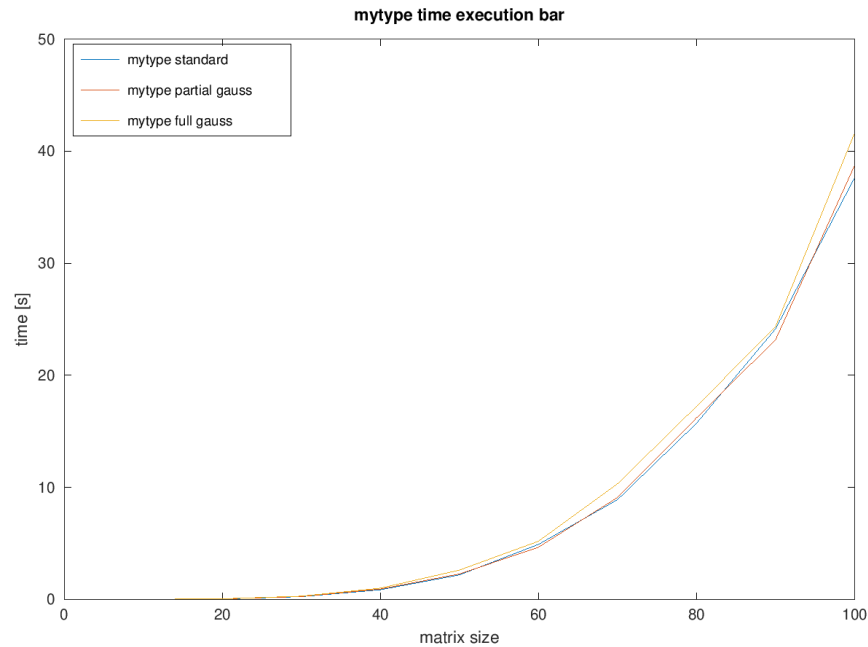
**Wniosek 1.** Hipoteza potwierdza się w pełni dla typów float oraz double, typ własny w większej części wykresu zdaje się jej nie przeczyć, jednak niektórych momentach wykres układu się wbrew hipotezie.



Rysunek 1: Czas dla float



Rysunek 2: Czas dla double



Rysunek 3: Czas dla typu własnego

## 2.2 Błąd wyniku

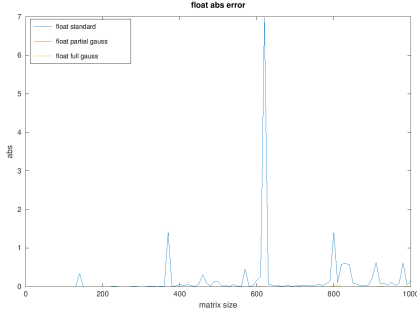
**Hipoteza 2.** Dla dowolnego ustalonego rozmiaru macierzy błąd uzyskanego wyniku metody Gaussa w kolejnych wersjach (G, PG, FG) maleje.

Analiza wykresów przedstawiających błędy obliczeń dla typów float oraz double (typ własny pomijamy na rzecz trzeciej hipotezy), pozwala na wyciągnięcie następujących wniosków:

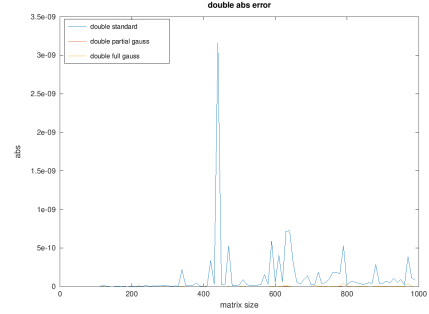
1. Kolejne typy metody Gaussa zmniejszają błędy zarówno względne jak i bezwzględne.
2. Względem standardowej, metody PG i FG mają wyraźnie większą dokładność.
3. Dokładność metody FG jest zdecydowanie najwyższa oraz bliska pełnej.
4. Typ float wykazuje znacznie mniejszą dokładność (kilka rzędów wielkości) od double co jest spowodowane różnicą w precyzji obu typów.

**Wniosek 2.** Hipoteza jest prawdziwa - kolejne typy metody Gaussa zwiększają dokładność

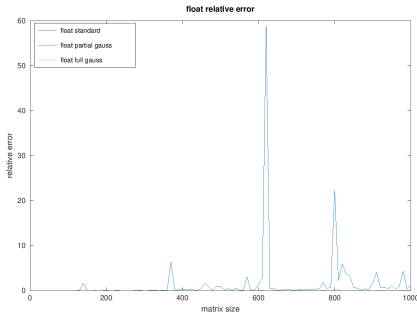
wyników, największą różnicę widać między metodą standardową a pozostałymi.



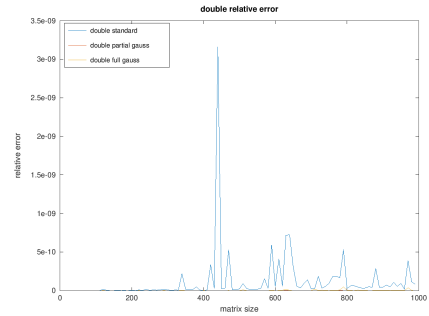
Rysunek 4: Błąd absolutny dla float



Rysunek 5: Błąd absolutny dla double



Rysunek 6: Błąd względny dla float



Rysunek 7: Błąd względny dla double

## 2.3 Bezbieżność własnej artmetyki na ułamkach

**Hipoteza 3.** Użycie własnej arytmetyki na ułamkach zapewnia bezbłędne wyniki niezależnie od wariantu metody Gaussa i rozmiaru macierzy.

Z analizy tabeli (Tabela 1) wynika, że wszystkie błędy obliczeń wykonywanych na własnym typie ułamkowym są równe zeru. Z powyższej obserwacji można wyciągnąć wniosek, że własna artmetyka ułamków przechowująca licznik oraz mianownik skutecznie eliminuje błąd obliczeń. Warto jednak zaznaczyć, że powyższą skuteczność własny typ ułamkowy zawdzięcza typowi zmiennej, w której jest przechowywany licznik oraz mianownik. Nasz typ oparliśmy na bibliotece GMP co daje praktycznie pełną dokładność przy nawet skomplikowanych obliczeniach na niesamowicie małych liczbach.

**Wniosek 3.** Hipoteza jest prawdziwa - poprawnie zaimplementowany typ ułamkowy daje praktycznie pełną dokładność.

		MyType								
matrix size	relative error			absolute error			execution time			
gauss type	standard	partial	full	standard	partial	full	standard	partial	full	
10	0	0	0	0	0	0	0,002927	0,002510	0,003722	
20	0	0	0	0	0	0	0,039439	0,038365	0,047704	
30	0	0	0	0	0	0	0,227981	0,247261	0,249637	
40	0	0	0	0	0	0	0,772091	0,799447	0,832883	
50	0	0	0	0	0	0	1,904712	1,969039	2,218641	
60	0	0	0	0	0	0	3,888714	4,159920	4,450316	
70	0	0	0	0	0	0	6,206715	7,008851	7,800630	
80	0	0	0	0	0	0	12,738509	13,224752	14,033934	
90	0	0	0	0	0	0	20,126191	21,740307	24,022611	
100	0	0	0	0	0	0	31,732314	31,406399	34,099180	
110	0	0	0	0	0	0	42,728936	43,738354	50,380356	
120	0	0	0	0	0	0	65,817100	69,808432	70,863825	
130	0	0	0	0	0	0	86,339258	91,908211	100,786675	
140	0	0	0	0	0	0	121,456511	131,576217	140,215938	
150	0	0	0	0	0	0	165,185310	171,851760	192,351049	

Rysunek 8: Czasy oraz błędy dla typu władnego

### 3 Odpowiedzi na zadane pytania

**Pytanie 1.** 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)?

W przypadku analizowania metody standardowej dla obu wbudowanych typów ułamkowych można zauważyć pewną prawidłowość, dla mniejszych macierzy błędy są znikome jednak począwszy od wielkości nieco mniejszych niż 400 błędy stają się większe jednak po powiększeniu pozostają na podobnym poziomie.

**Pytanie 2.** Jak przy wybranym przez Ciebie wariancie metody Gaussa zależy czas działania algorytmu od rozmiaru macierzy i różnych typów?

Analizuje danych dla metody FG.

W przypadku typu własnego wzrost jest prawie paraboliczny, jednak dla typów wbudowanych do wielkości macierzy rzędu 600 krzywa obrazująca wzrost czasu jest bardzo nieregularna aby po przekroczeniu tej wielkości być zbliżoną do paraboli.

## 4 Wydajność implementacji

**Ćwiczenie 2.** Podaj czasy rozwiązania układu równań uzyskane dla macierzy o rozmiarze 500 dla 9 testowanych wariantów.

W naszym przypadku czas wykonania algorytmu dla macierzy o wielkości 500x500 prezentuje się następująco (czas dla typu własnego zosał wyliczony w przybliżeniu według danych uzyskanych w wyniku działania programu):

Czas wykonania dla macierzy 500x500		
Typ danych	Typ algorytmu	Czas wykonania
Float	G	4816.077736
Double	PG	4958.616576
MyType	FG	5327.405392

Rysunek 9: Czasy wykonania dla macierzy 500x500

## 5 Podział pracy nad projektem

Podział Pracy		
Stanisław Grams	Maciej Jezierski	Juliusz Korczakowski
Implementacja klasy generycznej	Implementacja algorytmu G i PG	Implementacja algorytmu FG
Implementacja własnego typu ułamkowego	Przygotowanie sprawozdania	Agregacja danych i przygotowanie skryptu generującego wykresy
Przygotowanie i przeprowadzenie testów	Analiza wygenerowanych danych	Analiza algorytmów gaussa