Alan Nadolny
Informatyka Praktyczna, II rok, semestr 4
Numer indeksu: 274950

26 kwietnia 2022
Projekt 2

Algorytmy numeryczne

1 Wstęp

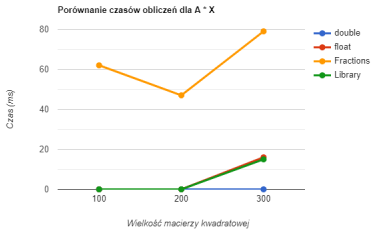
Celem projektu jest porównanie poprawności i wydajności operacji na macierzach oraz rozwiązywania układów równań liniowych metodą Gaussa w wersjach: bez wyboru elementu podstawowego, z wyborem częściowym oraz pełnym. Wszystkie testy przeprowadzane są dla typów Float, Double oraz własnego typu Fractions, przechowującego liczbę w typie BigInteger. Wszystkie obliczenia wykonywane są w języku Java.

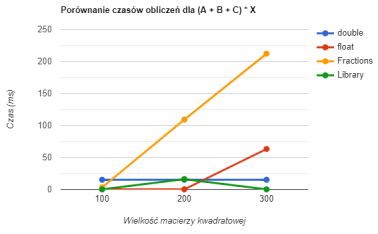
2 Obliczenia

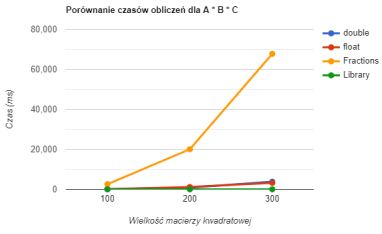
W celu wskazania różnic pomiędzy wynikami operacji policzyłem normy dla macierzy, będących rożnicami błędów bezwzględnych wartości bibliotecznej, a wartości otrzymanej. Do obliczenia normy używałem wzoru: $|X| = \sum_{i=1}^{n} |x_i|$

	100 elementów			200 elementów			300 elementów		
	float	double	fractions	float	double	fractions	float	double	fractions
A * X	3.90e-5	0	1.39e-16	1.92e-3	0	2.91e-16	3.76e-4	0	6.16e-16
(A + B + C) * X	8.05e-5	0	1.94e-16	2.91e-4	0	2.19e-16	6.49e-4	0	4.15e-16
A * (B * C)	3.70e-2	0	8.80e-12	4.30e-1	0	3.39e-10	1.76	0	1.76e-9

Zerowe wyniki normy dla typu double powstały przez to, że użyta biblioteka (ejml) wykonywała obliczenia na typie double, więc wyniki uzyskane z mojej implementacji pokrywały się z wynikami bibliotecznymi.
Wyniki wydajnościowe dla powyższych rezultatów prezentują się następująco:







3 Rozwiązywanie układów równań liniowych

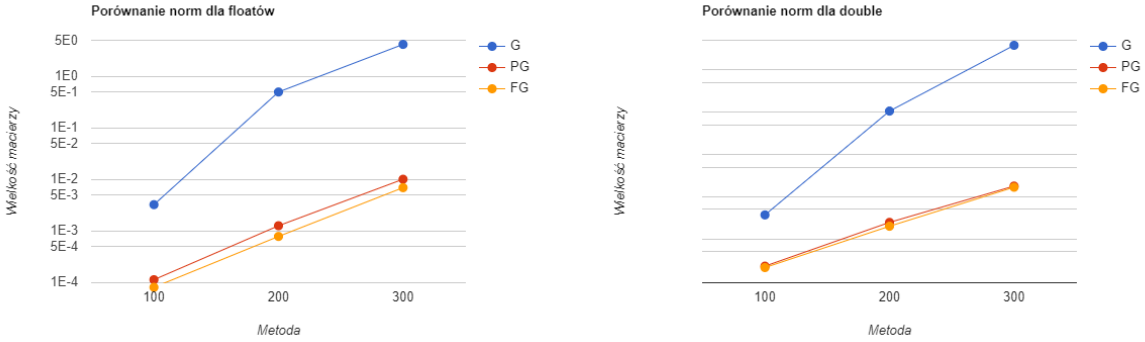
Poniższe tabele przedstawiają normy wyliczone z tego samego wzoru co powyższe, lecz tutaj zestawione są ze sobą metody rozwiązywania układów.

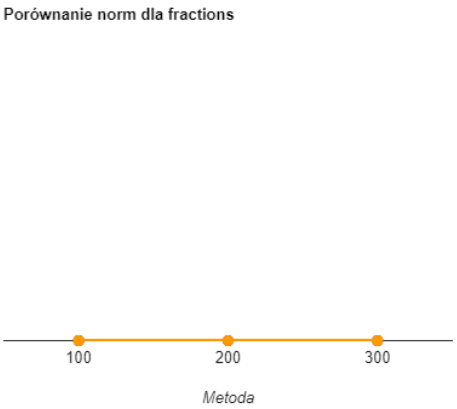
Normy dla 100 elementów	float	double	fractions
Podstawowy Gauss	3.21e-3	3.72e-12	0
Częściowy wybór elementów	1.12e-4	2.29e-13	0
Pełny wybór elementów	7.97e-5	2.12e-13	0

Normy dla 200 elementów	float	double	fractions
Podstawowy Gauss	4.95e-1	1.06e-9	0
Częściowy wybór elementów	1.25e-3	2.48e-12	0
Pełny wybór elementów	7.75e-4	2.01e-12	0

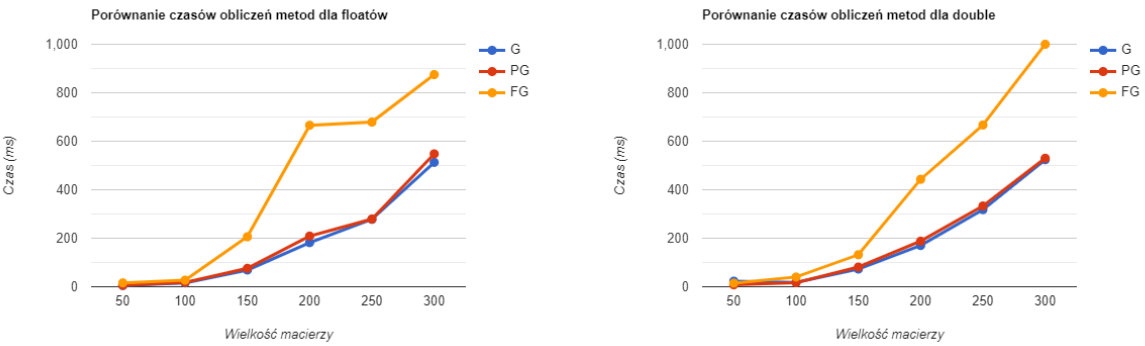
Normy dla 300 elementów	float	double	fractions
Podstawowy Gauss	4.16	3.80e-8	0
Częściowy wybór elementów	1.00e-2	1.79e-11	0
Pełny wybór elementów	6.86e-3	1.68e-11	0

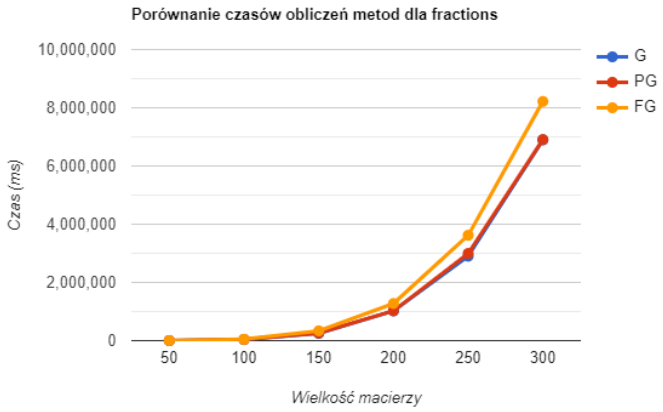
Do powyższych tabel przygotowałem wykresy. Widać na nich, że podstawowa wersja Gaussa znacznie odbiega wynikami od pozostałych metod. W celu wskazania różnic pomiędzy metodami oś OY jest przeskalowana logarytmicznie.





Wyniki wydajnościowe powyższych rozwiązań:





4 Wnioski

H1: Czas w kolejnych wersjach metody Gaussa rośnie, można to zauważyć na powyższych wykresach (przykładowo czasy rozwiązywania dla macierzy 300 x 300 typu double wynosiły kolejno: 524, 530 oraz 1000 ms. Czas dla częściowego wyboru elementu często był bardzo podobny do zwykłej metody Gaussa, czasami udało mu się osiągnąć nawet lekko lepsze wyniki czasowe.
H2: Z tabel, w których porównywane są normy błędów można zauważyć, że błąd maleje. Nie dzieje się tak tylko w klasie fractions, w której błąd jest równy 0.
H3: Użycie własnej arytmetyki zapewnia bezbłędne wyniki. Jednyny błąd, który może wystąpić używając tego typu danych powstaje gdy zwracamy ułamek jako liczbę dziesiętną, ponieważ w momencie dzielenia licznika przez mianownik nastąpi błąd zaokrąglenia.
Q1: Z powyższych tabel można odczytać, że błąd dla typu podwójnej precyzji jest tym mniejszy im bardziej skomplikowana jest wersja metody Gaussa. Przykładowo pomiędzy częściowym wyborem elementów, a pełnym wyborem dla macierzy 300 x 300, błąd zmalał minimalnie 6%, a maksymalnie 24%.
Q2: Przy którymkolwiem z wykonanych wariantów metody Gaussa, czas działania rośnie wraz ze wzrostem wielkości macierzy. Czas też zazwyczaj rośnie jeśli zaczynam korzystać z typu, który zapewnia większą dokładność. W wykonanych przeze mnie testach, raz nie potwierdziła się ta teza, w momencie gdy typ double osiągnął lepsze czasy niż float.
E1: Czasy rozwiązania układu równań dla macierzy 500 x 500 z użyciem typu double we wszystkich wariantach prezentują sie następująco: