Alan Nadolny, Mateusz Stapaj
Informatyka Praktyczna, II rok, semestr 4
Numery indeksów: 274950, 275036

15 maja 2022
Projekt 3

Algorytmy numeryczne

1 Wstęp

Celem tego projektu było sprawdzenie poprawności metod iteracyjnych dla macierzy rzadkich oraz ich standardowej implementacji. Dokładność była porównywana przy użyciu 3 typów liczbowych: Float, Double oraz Fractions (bazującej na typie BigInteger). Wyniki były porównywane względem implementacji z biblioteki EJML. Wszystkie operacje wykonywane były w języku Java.

2 Prawidłowość implementacji oraz stosowania metod iteracyjnych

W postawionym problemie możliwe jest stosowanie metod iteracyjnych.
Przykładowa macierz (3 agentów):

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{2}{3} & -\frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{2}{3} & -\frac{2}{3} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & -\frac{2}{3} & 0 & 0 \\ -\frac{1}{3} & 0 & 0 & 0 & 0 & 1 & -\frac{1}{3} & 0 & -\frac{1}{3} & 0 \\ 0 & -\frac{2}{3} & 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \frac{2}{3} & 0 & -\frac{2}{3} \\ 0 & 0 & 0 & 0 & -\frac{2}{3} & 0 & 0 & 0 & \frac{2}{3} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Możliwość rozwiązania układu równań zweryfikowaliśmy metodą Monte Carlo (Ilość symulacji: 100000). Poniższe wyniki pokazują poprawność wyliczeń, ale nie dają prawdziwych wartości, ponieważ Monte Carlo jest symulacją.

	Double		Float		Fraction	
	GJ	GS	GJ	GS	GJ	GS
10 elementów, 100 iteracji, 100 symulacji	6.39E-2	6.39E-2	7.60E-2	7.60E-2	4.59E-2	4.58E-2
20 elementów, 1000 iteracji, 1000 symulacji	3.38E-2	3.38E-2	4.59E-2	4.59E-2	3.40E-2	3.39E-2
30 elementów, 2000 iteracji, 2000 symulacji	2.28E-2	2.28E-2	2.59E-2	2.59E-2	-	-
40 elementów, 3000 iteracji, 3000 symulacji	2.29E-2	2.29E-2	2.31E-2	2.31E-2	-	-

	1E3 iteracji, 1E3 symulacji		2E3 iteracji, 1E4 symulacji		3E3 iteracji, 1E5 symulacji	
	GJ	GS	GJ	GS	GJ	GS
20 elementów	4.42E-2	4.32E-2	1.36E-3	1.19E-3	9.59E-4	9.49E-4
30 elementów	4.09E-2	4.09E-2	9.52E-3	9.52E-3	3.9E-3	3.89E-3
40 elementów	2.97E-2	2.99E-2	1.18E-2	1.12E-2	3.62E-3	3.62E-3

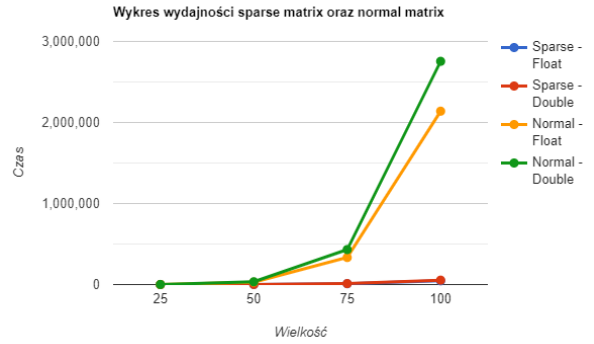
3 Wyniki rozwiązań układów

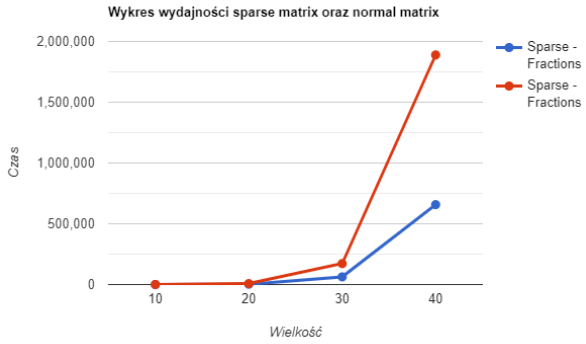
3.1 PG dla normal matrix oraz sparse matrix

We metodach iteracyjnych wykonaliśmy 1000 iteracji. Wszystkie normy zostały policzone ze wzoru: $||A||_{\infty}$ $\max_{1 \le i \le n} = \sum_{i=1}^{n} |a_{ij}|$

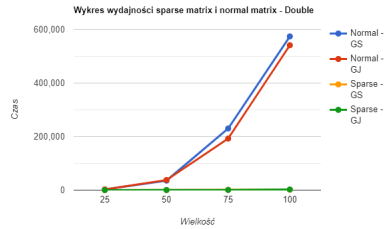
	Double		Float	
	Normal Matrix	Sparse Matrix	Normal Matrix	Sparse Matrix
25	1.11E-16	1.11E-16,	1.55E-6	1.55E-6
50	2.15E-16	2.15E-16	6.02E-6	6.02E-6
75	3.33E-16	3.33E-16	6.20E-6	6.20E-6
100	2.22E-15	2.22E-15	1.52E-5	1.52E-5

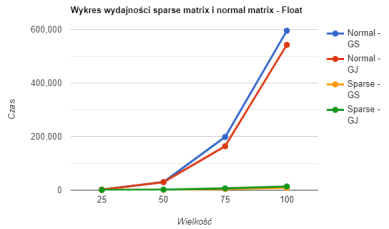
	Fraction	
	Normal Matrix	Sparse Matrix
10	1.19E-16	1.19E-16
20	1.48E-16	1.48E-16
30	2.07E-16	2.07E-16
40	4.65E-16	4.65E-16

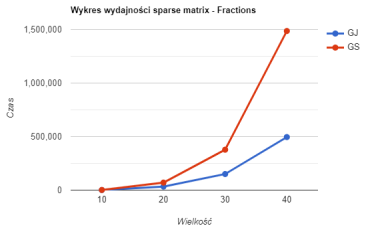




3.2 GJ i GS dla normal matrix oraz sparse matrix

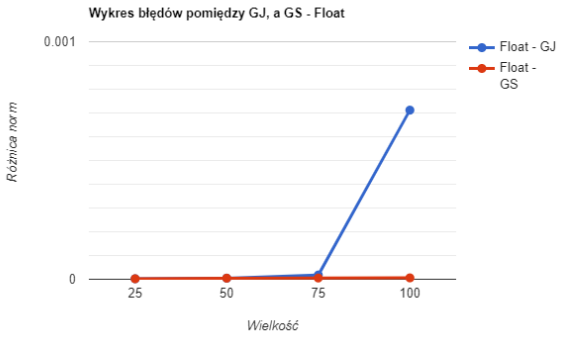


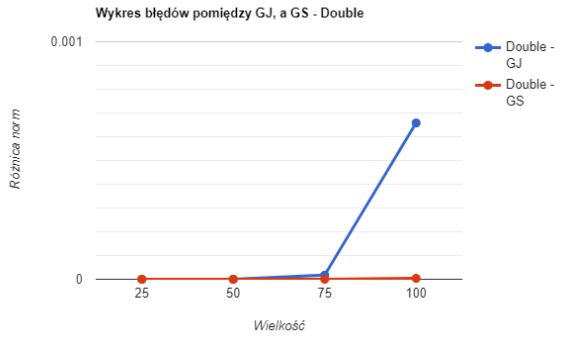




	Double				Float			
	Normal matrix		Sparse matrix		Normal matrix		Sparse matrix	
	GJ	GS	GJ	GS	GJ	GS	GJ	GS
25	1.67E-15	1.66E-15	1.67E-15	1.66E-15	1.07E-6	1.07E-6	1.07E-6	1.07E-6
50	5.30E-9	1.58E-13	5.30E-9	1.58E-13	2.32E-6	2.29E-6	2.32E-6	2.29E-6
75	8.70E-6	8.93E-9	8.70E-6	8.93E-9	8.92E-6	2.68E-6	8.92E-6	2.68E-6
100	3.29E-4	2.01E-6	3.29E-4	2.01E-6	3.56E-4	2.93E-6	3.56E-4	2.93E-6

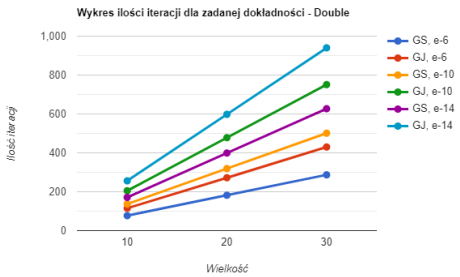
	Fractions - Sparse Matrix (100 iteracji)	
	GJ	GS
10	2.61E-7	2.59E-9
20	1.55E-2	6.69E-3
30	1.45E-1	2.48E-2
40	4.02E-1	1.14E-1





4 Wnioski

Optymalna metoda względem rozmiaru planszy: W celu szybszego otrzymania wyników lepiej używać metod iteracyjnych, które okazały się szybsze, niż Gauss z częściowym wyborem elementów. Natomiast metody iteracyjne dawały znacznie mniej dokładne wyniki. Było to spowodowane tym, że przyjęliśmy dosyć małą liczbę (1000) iteracji. W celu otrzymania bardziej dokładnych wyników powinniśmy używac Gaussa z częściowym wyborem elementów.
Optymalna metoda względem dokładność obliczeń: W przypadku Gaussa z częściowym wyborem elementów nie wybieramy dokładności jaką chcemy otrzymać. W tym wypadku jedyny wpływ na dokładność ma typ danych, którym się posługujemy. Porównując metody iteracyjne znacznie lepiej wypada metoda Gaussa-Seidla, która zwraca wynik z zadaną dokładnością znacznie szybciej.



Optymalna metoda względem typu liczbowego: W każdej metodzie najlepiej wypada wybór typu Double. Powoduje on znacznie mniejsze błedy w obliczeniach niż typ Float, a czas obliczeń jest lepszy, niż w typie Fraction o kilka rzędów wielkości.