

Algorytmy numeryczne - Projekt 2

Mateusz Soroka 250999

Bartłomiej Skopiński 246830

Grupa 1 - aplikacje internetowe i bazy danych

7 listopad 2018

Opracowanie dotyczy operacji na macierzach z zaimplementowanym algorytmem eliminacji Gaussa w trzech wariantach:

1. bez wyboru elementu podstawowego,
2. z częściowym wyborem elementu podstawowego
3. z pełnym wyborem elementu podstawowego.

Stopień złożoności obliczeniowej dla poszczególnych wariantów rośnie, od 1 - najmniej złożonego, do 3-najbardziej złożonego.

Program został napisany w języku Java (wersja 8.0.181), testy przeprowadzono na komputerze MacBook Air wyposażonym w procesor Intel Core i5 (1,8 GHz), pamięć 8 GB 1600MHz DDR3 z wersją systemu Mojave 10.14.1. Do testów poprawnościowych i wydajnościowych wykorzystano dwa typy zmiennoprzecinkowe float oraz double.

Testy zostały przeprowadzone dla macierzy o rozmiarze $N \times N$ i wektora o rozmiarze N gdzie $N=2000$. Współczynniki wylosowanych macierzy to pseudolosowe liczby zmiennoprzecinkowe otrzymane jako iloraz liczb z przedziału $[-65536, 65536]$ oraz liczby 65536. Dla macierzy A i wektora B rozwiązano układ równań liniowych: $A * X = B$, wektor X zachowano jako rozwiązanie wzorcowe do obliczenia błędów bezwzględnych otrzymanych rozwiązań.

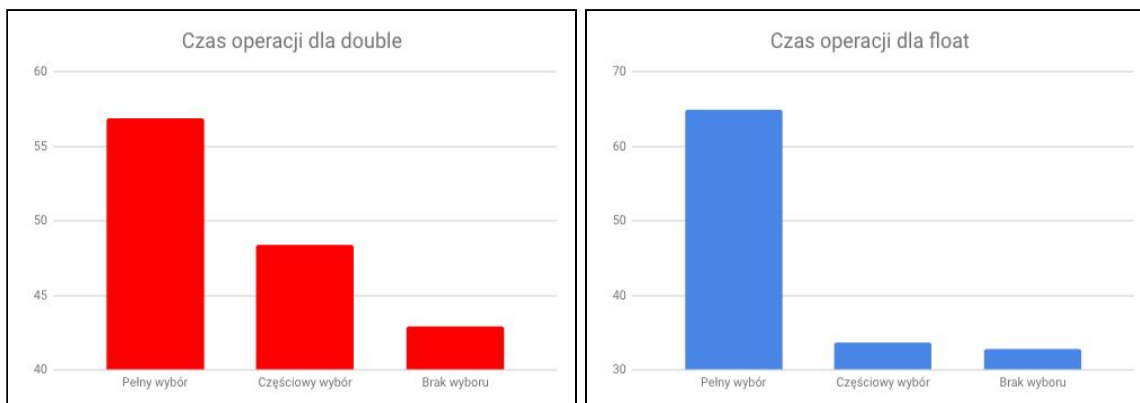
Aby dowieść, że wszystkie warianty algorytmu eliminacji Gaussa zostały zaimplementowane poprawnie, porównano wyniki dla macierzy o rozmiarze $N \times N$ i wektora o rozmiarze N , gdzie $N=2$ z wynikami podanymi przez wolframalpha, wykorzystano typ zmiennoprzecinkowy float. Poniżej wyniki (a - brak wyboru, b - wybór częściowy, c - wybór pełny):

$$\text{Macierz: } \begin{pmatrix} -0.83000183 & -0.270813 \\ -0.92918396 & -0.947052 \end{pmatrix} \quad \text{Wektor: } \begin{pmatrix} 0.5124512 \\ -0.59062195 \end{pmatrix}$$

$$\text{a: } \begin{pmatrix} -1.2074139 \\ 1.8082762 \end{pmatrix} \quad \text{b: } \begin{pmatrix} -1.2074143 \\ 1.8082765 \end{pmatrix} \quad \text{c: } \begin{pmatrix} -1.2074142 \\ 1.8082764 \end{pmatrix} \quad \text{wolfram: } \begin{pmatrix} -1.2074141 \\ 1.8082764 \end{pmatrix}$$

Jak więc widać wyniki są identyczne, ale nie są sobie równe. To dlaczego tak się dzieje zostanie wytłumaczone w dalszej części sprawozdania.

Testy dla $N=2000$



Błędy bezwzględne

	double	float
Pełny wybór	0,0000014179509406631	0,00030708313
Częściowy wybór	0,000055073435719066	0,0038013458
Brak wyboru	0,107671248273206	0,21160973

Zarówno dla typu double jak i float, czasy wykonywanych operacji maleją wraz ze zmianą metody na mniej złożoną. Najdłużej zajmują obliczenia dla metody z pełnym wyborem elementu, najkrócej dla metody bez wyboru elementu. Z kolei błędy bezwzględne rosną wraz ze zmianą metody na mniej złożoną. Najmniejszy błąd bezwzględny występuje przy wariancie z pełnym wyborem elementu, największy zaś przy wariancie z brakiem wyboru elementu.

Jak więc widać precyzja wyniku jest zależna od wybranej metody algorytmu eliminacji Gaussa, stąd również rozbieżności w wynikach obliczeń dla zaimplementowanych wariantów dla macierzy $N \times N$ i wektora N w porównaniu z wynikiem z wolframalpha, o których była mowa we wcześniejszej części sprawozdania.

Przeprowadzono również testy dla macierzy o rozmiarze $N \times N$ oraz wektora o rozmiarze N , gdzie $N=2500$, aby sprawdzić jak zmienia się błąd bezwzględny w zależności od rozmiaru macierzy dla typu zmiennoprzecinkowego double oraz jak zmienia się czas działania algorytmu w zależności od rozmiaru macierzy i zastosowanego typu zmiennoprzecinkowego.

Błędy bezwzględne dla typu double

Wariant	$N=2000$	$N=2500$	Zmiana
Pełny wybór	0,0000014179509406	0,0000035329660938	+149,16%
Częściowy wybór	0,0000550734357190	0,0000853564498726	+54,99%
Brak wyboru	0,107671248273206	0,658993695022758	+512,04%

Jak więc widać zmiana rozmiaru macierzy na większą sprawia, że błąd bezwzględny rośnie.

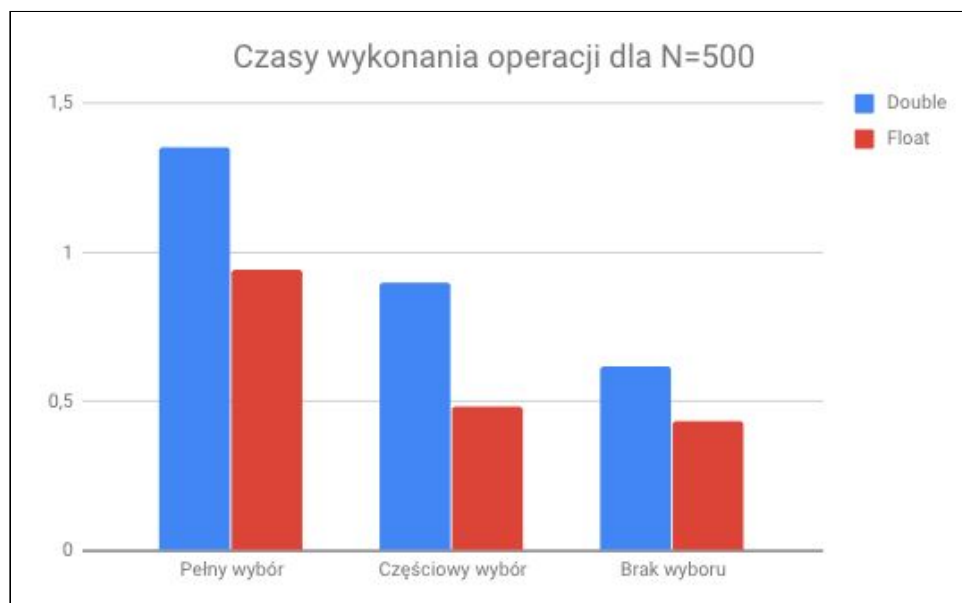
Czasy wykonywania operacji dla typów float i double dla rozmiarów $N=2000$ oraz $N=2500$

	Wariant	$N=2000$	$N=2500$	Zmiana
Double	Pełny wybór	56,915s	111,062s	+95,14%
	Częściowy wybór	48,413s	91.303s	+88,59%
	Brak wyboru	42,914s	90.708s	+111,37%
Float	Pełny wybór	64,907s	138.616s	+113,56%
	Częściowy wybór	33,717s	75.503s	+123,93%
	Brak wyboru	32,84s	74.76s	+127,65%

Zmiana rozmiaru macierzy ma ogromny wpływ na czas wykonywania obliczeń, niezależnie od wybranej metody czy typu zmiennoprzecinkowego.

Wydajność implementacji

Do sprawdzenia wydajności implementacji wszystkich wariantów algorytmu eliminacji Gaussa, dla obu typów zmiennoprzecinkowych float i double przeprowadzono test na macierzy $N \times N$ i wektorze N , gdzie $N=500$.



Zakres prac członków zespołu

Mateusz Soroka	Bartłomiej Skopiński
Klasa Randomizer do generowania losowych macierzy	Matematyczne opracowanie wszystkich wariantów algorytmu
Szkielet klasy MyMatrix i wdrożenie generyczności	Implementacja wariantu z częściowym wyborem elementu
Implementacja wariantu bez wyboru i pomoc w implementacji częściowego wyboru elementu	Implementacja wariantu z pełnym wyborem elementu
Przeprowadzenie testów i zredagowanie sprawozdania	Obliczanie wyniku z wektorem wzorcowym i liczenie błędów bezwzględnych