Algorytmy numeryczne
Zadanie 2

Iwona Jaśtak | Arkadiusz Dąbrowski
Tester programista - grupa 1

12.11.2018

Sprawozdanie dotyczy implementacji algorytmu eliminacji Gaussa w wariantach:

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

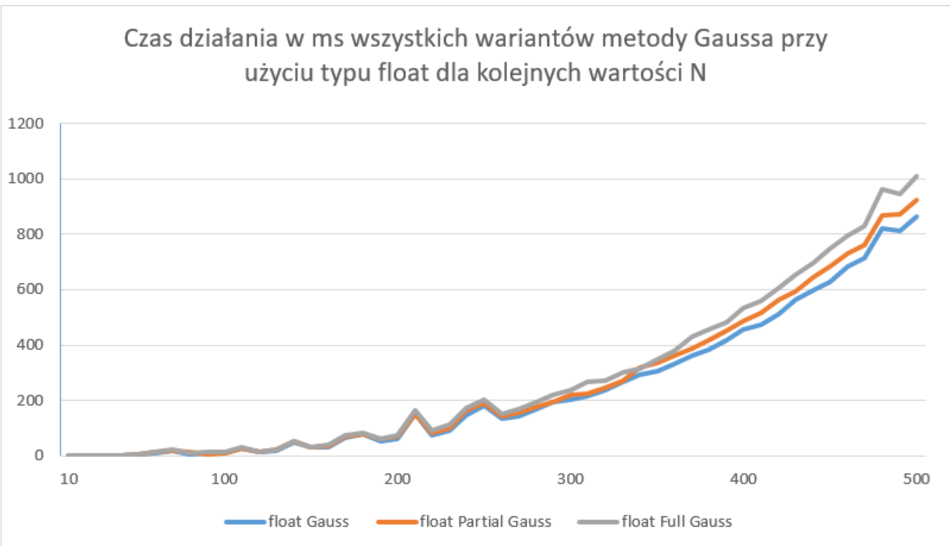
przy pomocy następujących typów:

- TF: typu pojedynczej precyzji - $float$,
- TD: typu podwójnej precyzji - *double*,
- TC: własnego typu, który przechowuje liczbę w postaci ułamka liczb całkowitych.

Program implementujący w/w warianty został napisany w języku *Java*, jak również wszelkie testy w nim przeprowadzone. Najpierw wylosowano macierz A oraz wektor X z przedziału $[-1, 1)$, następnie został wyliczony wektor B ze wzoru $B = A \cdot X$. Macierz A oraz wektor B podajemy jako parametry do rozwiązania układu równań.

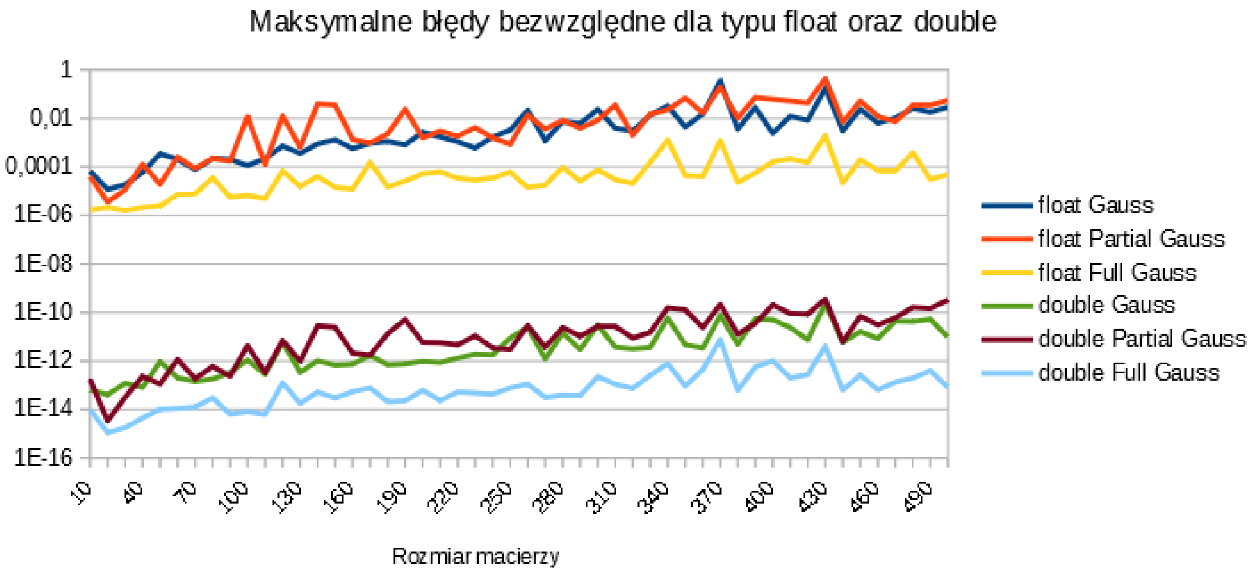
Hipotezy i pytania

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



Powyższa hipoteza została przetestowana dla typu $float$. Wyniki tych testów zostałe umieszczone na powyższym wykresie, z którego wprost wynika, że dla kolejnych wariantów algorytmu Gaussa czas działania jest większy dla tych samych N co bezpośrednio potwierdza tę hipotezę.

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



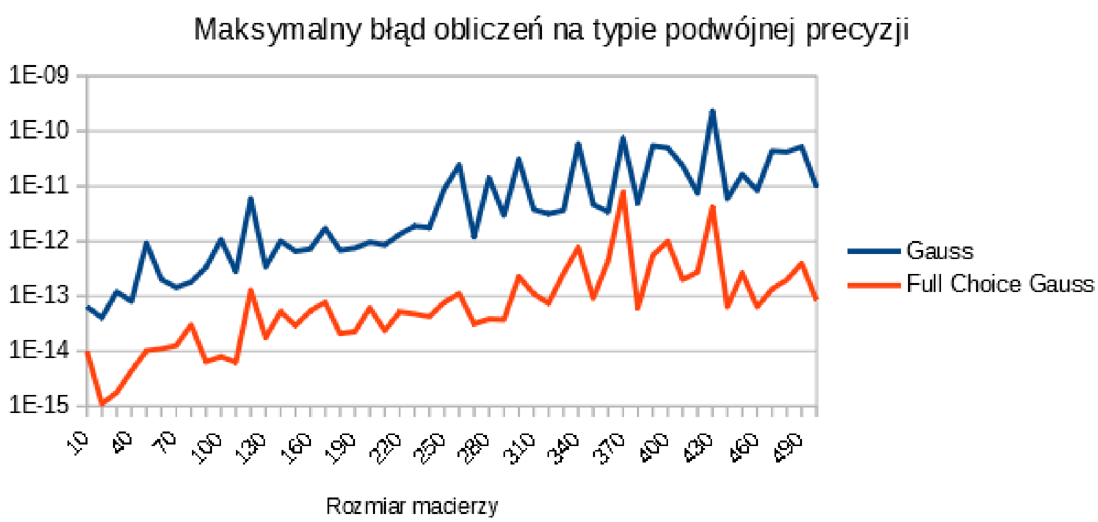
Wykres przedstawia maksymalne błędy bezwzględne dla metod Gaussa we wszystkich wersjach dla typów *float* oraz *double*. Maksymalny rozmiar macierzy to $N = 500$, iteracje zaczynały się od $i = 10$, które w kolejnych krokach były inkrementowane o 10. Pomiary dla typu ułamkowego zostały pominięte, ponieważ jak wynika z hipotezy *H3* wyniki w tym typie są bezbłędne.
Błędy dla metod bez wyboru elementu podstawowego oraz z częściowym wyborem utrzymują się na podobnym poziomie dla obu typów z nieznaczną korzyścią dla pierwszej wersji, natomiast Gauss z pełnym wyborem elementu podstawowego dla obu typów uzyskuje najlepsze wyniki.

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

```
N=10 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=20 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=30 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=40 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=50 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=60 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=70 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=80 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=90 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
N=100 Gauss: 0/1 Partial Gauss: 0/1 Full Gauss: 0/1
```

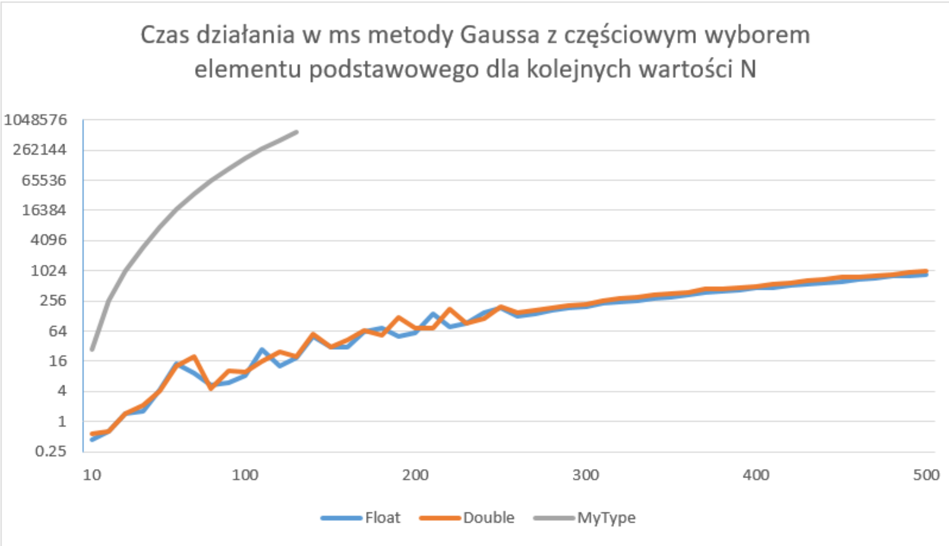
Zrzut ekranu obok prezentuje maksymalne błędy bezwzględne dla własnego typu ułamkowego (TC) dla wszystkich trzech implementacji algorytmu eliminacji Gaussa oraz dla różnych wielkości macierzy, gdzie w uproszczeniu 0/1 oznacza 0. Możemy zauważyć, że wyniki dla każdej kombinacji są bezbłędne.

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)?



W testach wzięto pod uwagę warianty Gaussa bez wyboru oraz z pełnym wyborem elementu podstawowego. Maksymalny rozmiar macierzy to $N = 500$, iteracje zaczynały się od $i = 10$, które w kolejnych krokach były inkrementowane o 10. Dla obu wariantów metody Gaussa można zauważyć tendencję wzrostową maksymalnego błędu bezwzględnego.

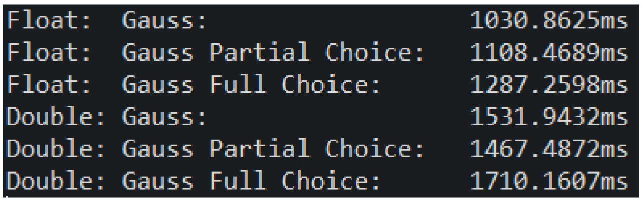
Q2: Jak przy wybranym przez Ciebie wariancie metody Gaussa zależy czas działania algorytmu od rozmiaru macierzy i różnych typów?



Aby odpowiedzieć na powyższe pytanie użyta została metoda Gaussa z częściowym wyborem elementu podstawowego. Testy zostały przeprowadzone dla wszystkich rozpatrywanych typów. Maksymalny rozmiar macierzy wynosił $N = 500$, a iteracje zaczynały się od $i = 10$, które w kolejnych krokach było inkrementowane o 10. Dla odpowiedniej przejrzystości wyniki przedstawione zostały na wykresie ze skalą logarytmiczną.
Jak można łatwo zauważyć dla każdego testowanego typu czas działania algorytmu rośnie wykładniczo. Dla typów float i double tempo wzrostu jest zbliżone, jednak dla własnego typu ułamkowego (TC) czas rośnie bardzo gwałtownie, już dla macierzy o rozmiarze $N = 130$ czas wykonania wynosi aż 9.83 min, a dla macierzy tylko o 10 mniejszej czas wyniósł 6.75 min.

Wydajność

E1: Podaj czasy rozwiązania układu równań uzyskane dla macierzy o rozmiarze 500 dla 9 testowanych wariantów.



```
Float:  Gauss:                  1030.8625ms
Float:  Gauss Partial Choice:   1108.4689ms
Float:  Gauss Full Choice:      1287.2598ms
Double: Gauss:                  1531.9432ms
Double: Gauss Partial Choice:   1467.4872ms
Double: Gauss Full Choice:      1710.1607ms
```

Dany zrzut ekranu przedstawia czasy wykonania poszczególnych wersji algorytmu dla każdego z typów. Zważywszy na ograniczenia sprzętowe oraz czasowe niemożliwe było wykonanie w/w testów dla własnego typu ułamkowego, gdyż szacowany czas wynosiłby około kilkanastu lub kilkadziesięciu godzin. Można tą tendencję zauważyć na wykresie *Q2*, gdzie czas dla tego konkretnego przypadku bardzo gwałtownie rośnie.

Podział pracy

Iwona Jaśtak	Arkadiusz Dąbrowski
klasa Randomizer - losowanie macierzy oraz wektorów	klasa MyType oraz interfejs definiujący operacje
obliczanie wektora B	szkielet i wstępna implementacja klasy generycznej MyMatrix
algorytm eliminacji Gaussa bez wyboru elementu podstawowego	algorytm eliminacji Gaussa z częściowym oraz pełnym wyborem elementu podstawowego
testowanie maksymalnych błędów bezwzględnych	testowanie czasu działania oraz wydajności implementacji
przygotowanie sprawozdania	