Algorytmy numeryczne

Zadanie 2
Dawid Bińkuś & Oskar Bir & Mateusz Małecki
grupa 1 tester-programista

11 Listopad 2018

1 Operacje na macierzach

Sprawozdanie prezentuje analizę wydajności i poprawności implementacji algorytmu eliminacji Gaussa, dla losowej macierzy kwadratowej A i wektora B w układzie liniowym $A \cdot X = B$.
Zaimplementowano następujące warianty algorytmu:

G: bez wyboru elementu podstawowego,

PG: z częściowym wyborem elementu podstawowego,

FG: z peanym wyborem elementu podstawowego.

Dodatkowo, obliczenia zostały wykonane, używając trzech różny typów reprezentujących liczbę rzeczywistą:

TF: typ pojedynczej precyzji: **float**

TD: typ podwójnej precyzji: **double**

TC: własna implementacja, przechowująca liczbę w postaci ułamka liczb całkowitych: **fraction**

Jako współczynniki macierzy A oraz wektora X zostały wylosowane liczby zmiennoprzecinkowe z przedziału: $\{\frac{-2^{16}}{2^{16}}, \frac{2^{16}-1}{2^{16}}\}$ Następnie wektor B został wyliczony według wzoru $B = A \cdot X$. Macierz A i wektor B zostają podane jako parametry do rozwiązania układu równań, wektor X zaś pozostawiamy jako rozwiązanie wzorcowe, za pomocą którego obliczamy błąd wykonanego algorytmu.

Program do realizacji testów został wykonany w języku *Java*. Typ danych *TC* został zaimplementowany za pomocą wbudowanego typu całkowitego *BigInteger*. Testy zostały wykonane na macierzach o rozmiarze $\{10, 20, ..., 800\}$ (float,double) $\{10, 20..., 150\}$ (fraction) w ilości prób malejącej, wraz z wykonywaniem testów na coraz to większych macierzach. Wyniki zostały zagregowane za pomocą średniej arytmetycznej.

2 Analiza hipotez

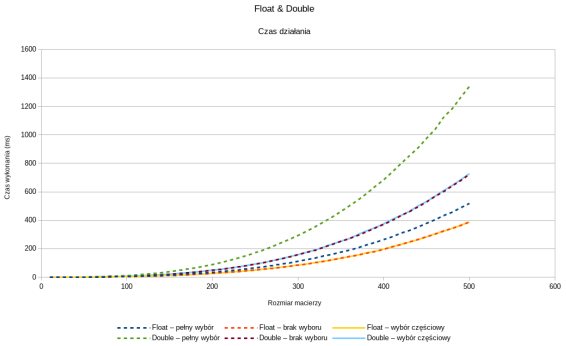
Rozważmy następujące wykresy (Rysunek1) Prezentują one błąd bezwzględny wartości w skali logarytmicznej (chyba że jest podane inaczej), wyliczonej za pomocą wcześniej wspomnianych algorytmów wobec wektora wzorcowego X.
Część z nich prezentuje również czas wykonania algorytmu podany w milisekundach.

2.1 Związek czasu wykonywania z wariantem algorytmu eliminacji Gaussa

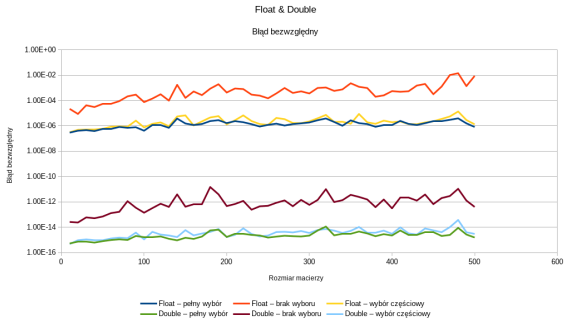
Hipoteza 1 *Dla dowolnego ustalone rozmiaru macierzy czas działania metody Gaussa w kolejnych wersjach (G,PG,FG) rośnie.*

Przeanalizujmy wykresy 1a oraz 1d. Zaprezentowano na nich porównanie czasu wykonania w zależności od wybranego typu danych oraz wariantu algorytmu. Niezależnie od typu danych można zauważyć pewną tendencję.
Po pierwsze, wraz z wzrostem rozmiaru macierzy użytej do rozwiązania układu równań, można zauważyć znaczący wzrost czasu wykonywania dla wariantu FG względem wariantów G i PG. Wykonując operacje na małych macierzach proces rozwiązywania nie jest na tyle skomplikowany by w jakikolwiek sposób wpływać na wydajność. Drugą interesującą rzeczą natomiast jest, brak znaczących różnic między wariantem G i PG. By bardziej uwidocznić to zjawisko przeanalizujmy wykres 1c, który prezentuje różnicę między cześciowym wyborem elementu podstawowego a brakiem wyboru w przypadku typu danych float. Wynikają z niego dwie rzeczy: różnica między wykonywaniem tych dwóch argumentów jest praktycznie nieznacząca (zaledwie kilka milisekund). Drugą rzeczą jest natomiast fakt, że wariant PG zdaje się być nieznacznie wolniejszy od wariantu G (Wraz z wzrostem rozmiaru macierzy okazuje sie w niektorych wypadkach nawet wydajniejszy niz standardowy algorytm Gaussa)

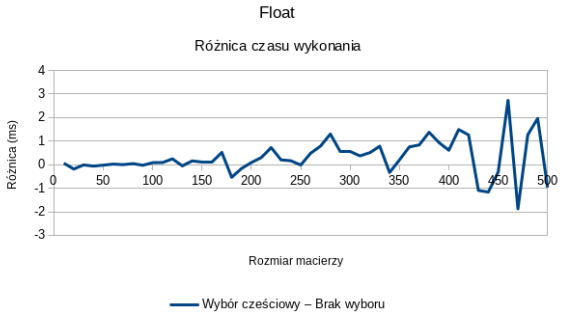
Rysunek 1: Wykresy reprezentujące błąd bezwzględny oraz czas wykonania dla różnych typów danych oraz różnych wariantów algorytmu



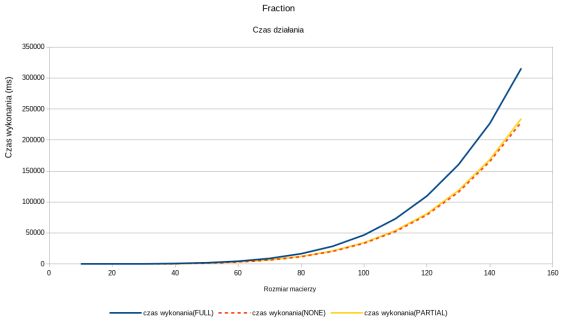
(a)



(b)



(c)



(d)

Wniosek 1 *Hipoteza okazuje się być cześciowo prawidziwa. Wariant FG powoduje znaczny wzrost czasu wykonania dla wszystkich wykorzystanych typów danych. Interesujący natomiast jest brak znaczących różnic między wariantami PG i G, który jest spowodowany niewielkim wpływem na czas wykonania (w wariancie PG sprawdzamy tylko n wartości w przeciwieństwie do $n * n$ w wariancie FG).*

2.2 Związek błędu obliczeń z wariantem algorytmu eliminacji Gaussa

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

Przeanalizujmy wykres 1b. Prezentuje on zestawienie błędu bezwzględnego rozwiązania układu równań względem wektora X dla typów danych float i double. W przypadku obu typów danych najmniej dokładny jest wariant G (różnice między nimi są efektem zastosowanej precyzji w tych typach). W przypadku wariantów PG i FG można zauważyć iż mają one bardzo zbliżoną dokładność z małą przewagą wariantu PG - jego dodatkowym atutem jest większa stabliność wyników. W sytuacjach gdy wariant G i PG mają problemy z bardziej skomplikowanymi równaniami (charakterystyczne wzrosty na wykresie) - PG radzi sobie zdecydowanie lepiej.

Wniosek 2 *Hipoteza okazuje się być prawdziwa. Wariant PG zapewnia największą dokładność i stabilność wyników względem nieco gorszego FG oraz G, w którym różnica błędu sięga nawet dwóch rzędów wielkości. Jednakże biorąc pod uwagę 1 najbardziej opłacalnym w kategorii dokładność/wydajność okazuje się być wariant PG jako iż różnice błędu między nim a pełnym wyborem zdają się być akceptowalne gdy weźmiemy pod uwagę znacznie szybszy czas działania.*

2.3 Poprawność własnej arytmetyki

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

Przeanalizujmy następującą tabelę:
Prezentuje ona wyniki z użyciem własnej arytmetyki dla wszystkich wariantów algorytmu. Obliczenia zostały wykonane na macierzach o rozmiarze n w przedziale $\{10, 20, ..., 150\}$ z racji ograniczeń sprzętowych. Rozpatrując wyniki testów, można dość do wniosku, że użycie własnej artymetyki całkowicie eliminuje błąd obliczeń. Jest to spowodowane charakterystyką typu BigInteger. Teoretycznie, typ ten nie ma żadnego limitu przechowywanych danych - w zależności od potrzeb zajmuje potrzebną ilość pamięci w komputerze. Jedynym ogranicznikiem w przypadku typu BigInteger jest pamięć naszego komputera. Dodając do tego sposób przechowywania ułamków ($\frac{licznik}{mianownik}$) jesteśmy w stanie wykonywać bardzo skomplikowane równania bez utraty precyzji.

Fractial - porównanie							
n	Błąd bezwzględny			Czas wykonania			Ilość prób
Wariant ->	G	PG	FG	G	PG	FG	
10	0	0	0	2.39034797	2.70733242	2.62618003	100
20	0	0	0	26.08455804	28.1878754	37.95241776	25
50	0	0	0	1353.7187555	1418.31329225	1935.74870875	4
100	0	0	0	33360.40558	34285.387315	46668.033684	1
130	0	0	0	115763.860973	118493.372512	160113.679247	1
150	0	0	0	228794.993101	234503.568338	315739.697225	1

Wniosek 3 *Hipoteza 3 okazuje się być prawdziwa. Użycie własnej arytmetyki zapewnia teoretycznie największą możliwą dokładność przechowywanych danych w porównaniu do zwykłych typów zmiennoprzecinkowych. Dzięki temu, wszelkie operacje wykonywane na typie Fraction są wykonywane z dokładnością równą 0*

3 Pytania

3.1 Dokładność obliczeń (typ podwójnej precyzji)

Pytanie 1 *Jak zależy dokładnośc obliczeń (błąd) od rozmiaru macierzy dla dwóch wybranych przez Ciebie wariantów metody Gaussa gdy obliczenia prowadzone są na typie podwójnej precyzji (TD)?*

3.2 Zależność czasu działania algorytmu od rozmiaru macierzy oraz typu

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

4 Wydajność implementacji

Zadanie 1 *Podaj czasy rozwiązania układu równań uzyskane dla macierzy o rozmiarze* 500 *dla* 9 *testowanych wariantów.*

5 Podział pracy

Dawid Bińkuś	**Oskar Bir**	**Mateusz Małecki**
Struktura projektu	Ten też coś robił	A ten to w ogóle bardzo dużo
Przygotowanie sprawozdania	Coś	Coś
Implementacja algorytmu Gaussa w wariantach PG i FG	Coś	Coś