MOwNiT - Rozwiązywanie układów równań liniowych metodami bezpośrednimi

Jakub Frączek

5 czerwca 2024

1 Wstęp

Tematem ćwiczenia było zaimplementowanie algorytmu Gaussa otraz Thomasa do rozwiązywania układów równań liniowych, a następnie rozwiązanie 3 ćwiczeń.

1.1 Ćwiczenie 1

Dla macierzy zadanej wzorem:

$$\begin{cases} a_{ij} = 1, & \text{i != 1.} \\ a_{ij} = 1/(i+j-1), & \text{otherwise.} \end{cases} \tag{1}$$

Gdzie $i, j = 1, 2, ...n$
Przyjąć wektor X jako dowolną n-elementowąpermutacją ze zbioru -1, 1 i obliczyć wektor B ze wzoru $A \cdot X = B$. Następnie dla różnych precyzji i rozmiarów macierzy A i wektora B wyliczyć rozwiązanie układu metodą Gaussa i porównać błędy zaokrągleń.

1.2 Ćwiczenie 2

Dla macierzy danej wzorem:

$$\begin{cases} a_{ij} = \frac{2i}{j} & \text{dla } j \geq i \\ a_{ij} = a_{ji} & \text{dla } j < i \end{cases} \qquad \text{dla } i, j = 1, \ldots, n$$

Powtórzyć eksperyment z ćwiczenia 1, porównać różnice w wynikach i policzyć uwarunkowanie obu układów.

1.3 Ćwiczenie 3

Dla macierzy danej wzorem:

$$\begin{cases} a_{i,j} = k & \\ a_{i,j+1} = \frac{1}{i+m} & \\ a_{i+1,j} = \frac{k}{i+m+1} & \text{dla } i > 1 \\ a_{i,j} = 0 & \text{dla } j < i-1 \quad \text{oraz} \quad j > i+1 \end{cases} \qquad i, j = 1, \ldots, n$$

Powtórzyć eksperyment z ćwiczenia 1 i 2, porównać różnice w wynikach i policzyć uwarunkowanie obu układów. Dodatkowo przeprowadzić analizę dla algorytmu `Thomasa` dla macierzy trójdiagonalnej.

2 Dane techniczne

2.1 Hardware

Laptop z procesorem `Intel Core i5-9300H 2.4GHz` oraz `32 GB` pamięci `RAM`.

2.2 Software

Wykorzystany został system `Windows 11 x64` oraz język `Python` w wersji `3.11.8` wraz z bibliotekami:

- `numpy`
- `random`
- `time`

Do stworzenia wykresów wykorzystane zostało narzędzie `Google Sheets`.

3 Sposób obliczania błędu

Błąd obliczony został jako norma euklidesowa różnicy oczekiwanego i otrzymanego wektora.

$$\|\mathbf{v}\|_2 = \sqrt{v_1^2 + v_2^2 + \ldots + v_n^2}$$

4 Wyniki

Podczas przeprowadzania testów wykorzystałem dwie różne precyzje liczby zmiennoprzecinkowej udostępnione przez bibliotekę `numpy`:

- `np.float32`
- `np.float64`

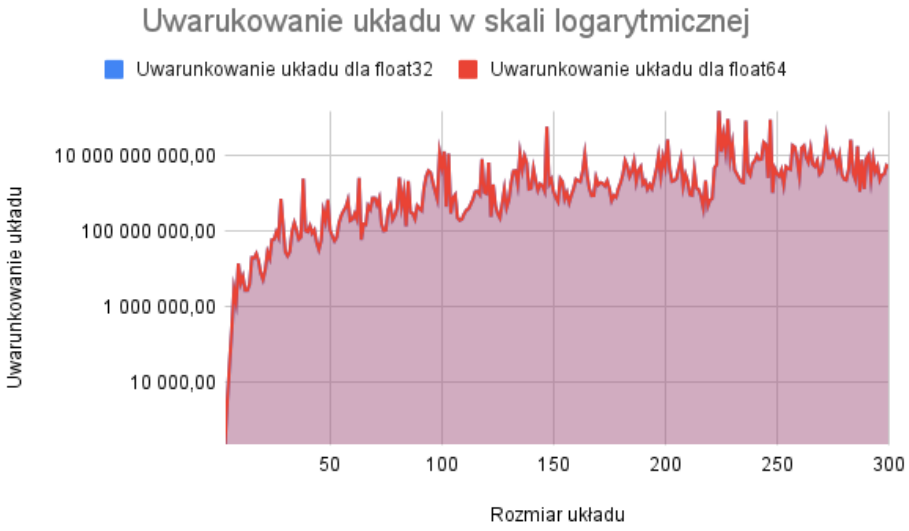
W miejscach, gdzie wykres w skali liniowej nie obrazował dobrze badanej zależności, wykorzystany był wykres w skali logarytminczej.

4.1 Zadanie 1

Jak widać na `wykresie 1` czas wykonania `algorytmu Gaussa` zwiększa się znacznie wraz z zwiększeniem rozmiaru układu. Można zauważyć, że algorym wykonuje się nieznacznie szybciej dla 32 bitowej wersji float'a.

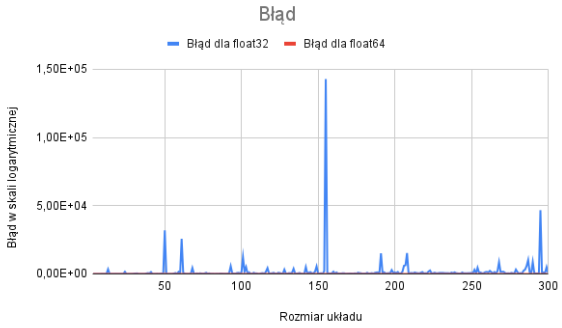
Wykres 1: Wykres czasu działania algorytmu w zależności od przyjetej precyzji)

`Wykres 2` prezentuje uwarunkowanie układu w zależności od jego rozmiaru. Jak widać uwarunkowanie układów jest bardzo złe na co wskazuje współczynnik uwarunkowania rzędu od 10^4 do 10^{10}.

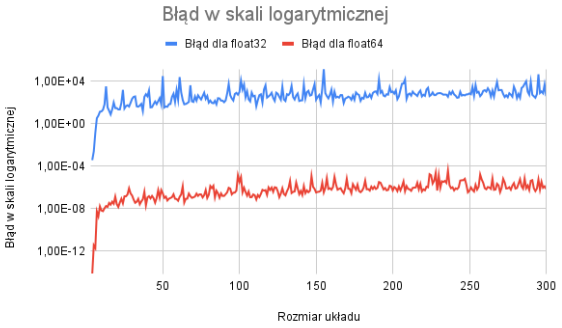


Wykres 2: Wykres uwarunkowania układu w skali logarytmicznej w zależności od przyjętej precyzji)

`Wykres 3` prezentuje wartości otrzymanych błędów dla różnych precyzji, a `Wykres 4` to samo tylko w skali logarytmicznej. Można zauważyć, że wyniki różnią się około 8 - 12 rzędami wartości w skali logarytmicznej, więc jest to bardzo duża różnica. Błędy dla wyników otrzymanych z 64 bitową precyzją są bliskie 0.



Wykres 3: Wykres błędu w zależności od przyjętej precyzji



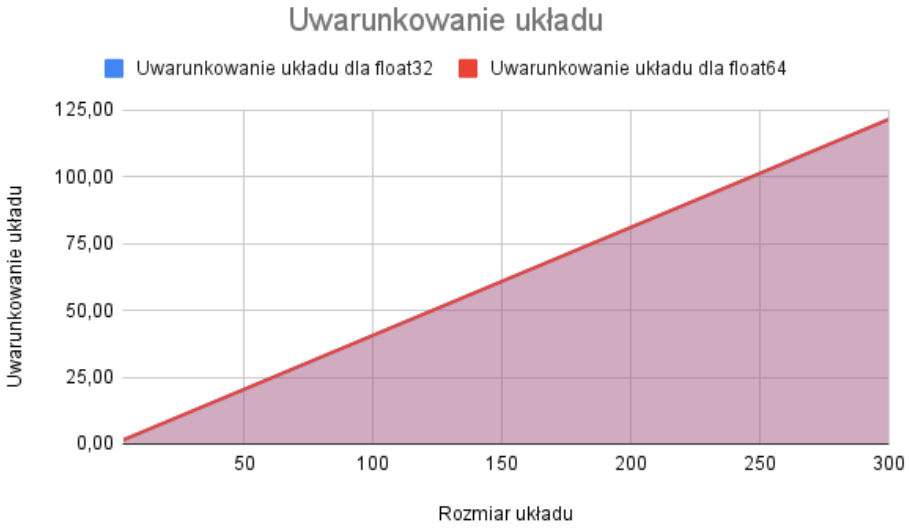
Wykres 4: Wykres błędu w skali logarytmicznej w zależności od przyjętej precyzji

4.2 Zadanie 2

W przypadku zadania 2, wykres czasu działania wygląda niemal identycznie jak w przypadku zadania 1. Można zaobserwować, że algorytm wykonał się nieznacznie szybciej dla `float32` (`Wykres 5`).

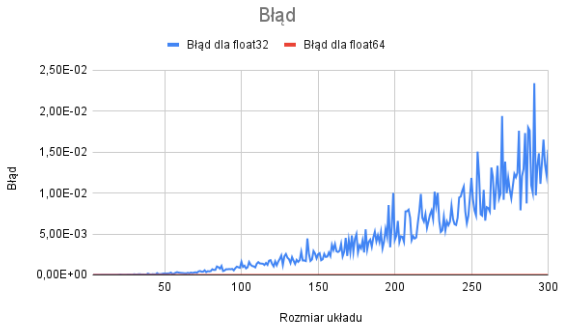
Wykres 5: Wykres czasu działania algorytmu w zależności od przyjętej precyzji)

`Wykres 5` przedstawia uwarunkowanie układu w zależności od jego rozmiaru. Jak widać, w tym przypadku zależność jest liniowa, czyli układ jest coraz gorzej uwarunkowany wraz z wzrostem jego rozmiaru. Warto zauważyć, że jest to znacznie lepsze uwarunkowanie niż w zadaniu 1.

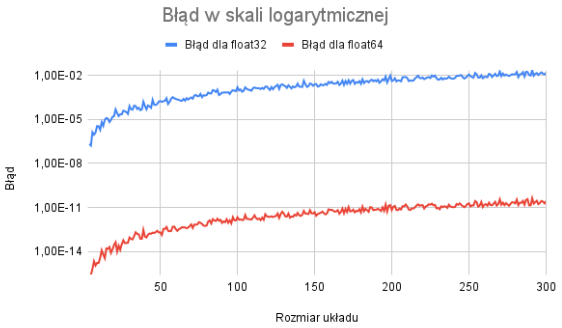


Wykres 6: Wykres uwarunkowania układu w skali logarytmicznej w zależności od przyjętej precyzji

Na `wykresie 7` pokazany został otrzymany błąd przybliżenia, a na `Wykresie 8` ten sam błąd w skali logarytmicznej. Jak widać ponownie 64 bitowa precyzja daje znacznie lepsze wyniki, a błąd jest bliski 0. Na wykresie logarytmicznym błędy różnią się aż o 8 - 12 rzędów wartości.



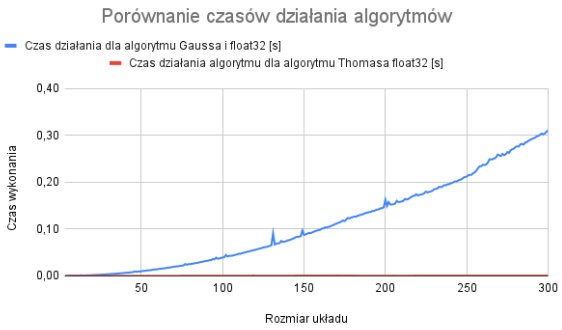
Wykres 7: Wykres błędu w zależności od przyjętej precyzji



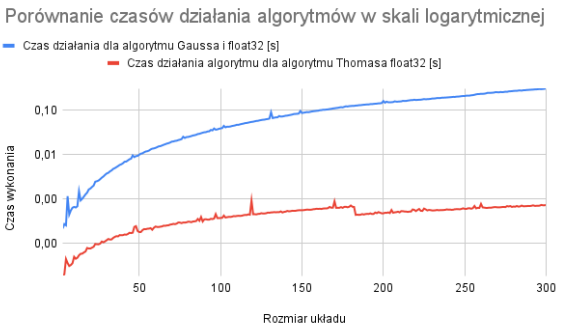
Wykres 8: Wykres błędu w skali logarytmicznej w zależności od przyjętej precyzji

4.3 Zadanie 3a - float32

`Wykres 9` przedstawia porównanie czasów działania algorytmów `Gaussa` oraz `Thomasa`, a na `Wykresie 10` to samo tylko w skali logarytmicznej. Jak widać, algorytm `Thomasa` poradził sobie znacznie lepiej, co jest zgodne z przewidywaniami, tj. złożoność algorytmu `Thomasa` jest liniowa, a `Gaussa` sześcienna.



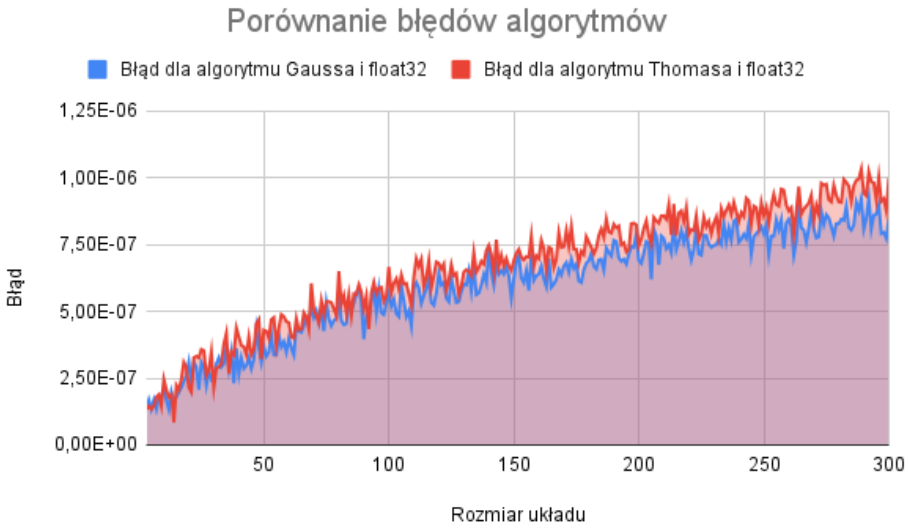
Wykres 9: Wykres czasów działania w zależności od przyjętej precyzji



Wykres 10: Wykres czasów działania w skali logarytmicznej w zależności od przyjętej precyzji

Jak widać na `Wykresie 11` uwarunkowanie układu jest dobre i nie zmienia się wraz ze wzrostem układu, w przeciwieństwie do układu z zadania 2.

Wykres 11: Wykres uwarunkowania układu

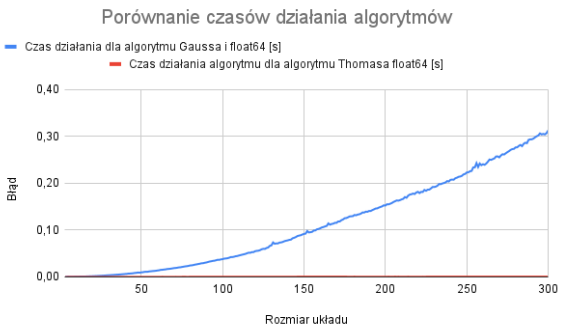


Wykres 12: Wykres błędów w zależności od wybranego algorytmu

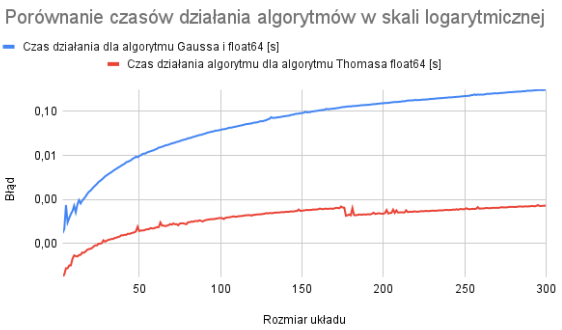
Otrzymane wyniki, jak widać na `wykresie 12` otrzymane wartości błędów są bardzo zbliżone i dość bliskie zeru, czyli można uznać obliczenia za dokładne. Można zauważyć nieznaczną przewagę algorytmu `Gaussa`.

4.4 Zadanie 3a - float64

`Wykres 13` przedstawia porównanie czasów działania algorytmów `Gaussa` oraz `Thomasa`, a na `Wykresie 10` to samo tylko w skali logarytmicznej. Jak widać, ponownie algorytm `Thomasa` poradził sobie znacznie lepiej, co jest zgodne z przewidywaniami, tj. złożoność algorytmu `Thomasa` jest liniowa, a `Gaussa` sześcienna.



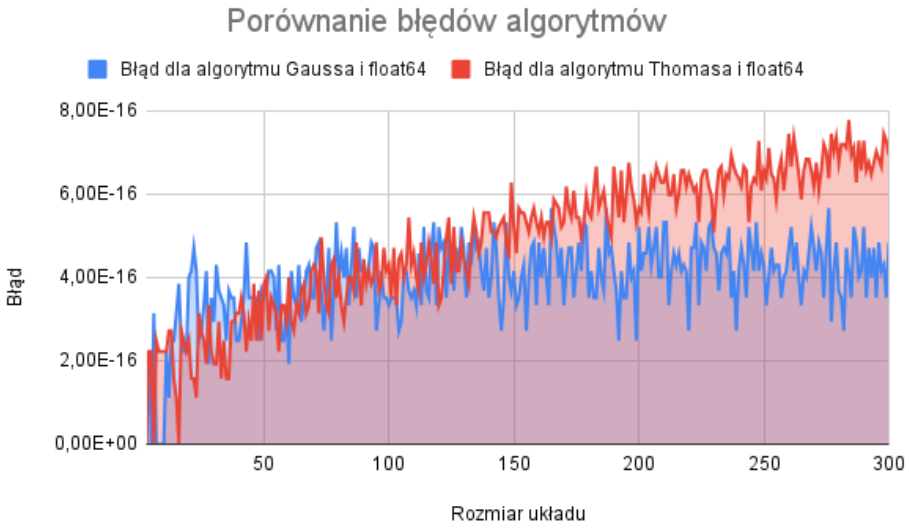
Wykres 13: Wykres czasów działania w zależności od przyjętej precyzji



Wykres 14: Wykres czasów działania w skali logarytmicznej w zależności od przyjętej precyzji

Ponownie, jak widać na `Wykresie 15` uwarunkowanie układu jest dobre i nie zmienia się wraz ze wzrostem układu, w przeciwieństwie do układu z zadania 2.

Wykres 15: Wykres uwarunkowania układu)



Wykres 16: Wykres błędów w zależności od wybranego algorytmu

Otrzymane wyniki, jak widać na `wykresie 16` otrzymane wartości błędów są jeszcze bardziej bliskie zeru niż w przypadku precyzji 32 bitowej. Ponownie błędy dla algorytmów `Thomasa` i `Gaussa` są zbliżone. Dla układu o rozmiarze mniejszym od ok. 125 lepsze przybliżenie daje algorytm `Thomasa`, a dla większych układów algorytm `Gaussa`.

5 Wnioski

1. Na podstawie przeprowadzonych testów można zauważyć, że uwarunkowanie układu ma duży wpływa na dokładnośc otrzymanego przybliżenia.

2. W niektórych przypadkach wyniki dla 64 bitowej precyzji są "o niebo"lepsze od wyników dla 32 bitowej precyzji

3. Algorytm Thomasa jest znacznie szybszy od algorytmu Gaussa, jednek ten drugi jest bardziej uniwersalny.