Dawid Łydka gr. 4, 184440

Metody numeryczne – Projekt 2
Rozwiązywanie układów równań liniowych

1. Cel projektu, informacje wstępne

Celem projektu było zaimplementowanie dwóch metod iteracyjnych (Jacobiego i Gaussa-Seidla) i metody bezpośredniej (faktoryzacji LU) do rozwiązywania układów równań liniowych. Projekt został zaimplementowany w języku *Python.* Podczas projektowania aplikacji zastosowane uniwersalne podejście w konstrucji macierzy i operacji na nich w celu ułatwienia zastosowania do różnych problemów czy dalszemu rozwojowi opracowanych metod.

Układ równań liniowych miał następującą postać:

$$\mathbf{Ax = b}$$

gdzie **A** jest daną macierzą systemową, **b** danym wektorem wyrazów wolnych, a

x jest wektorem rozwiązań.

2. Zadanie A

Macierz systemowa **A** wykorzystana w testach była rozmiaru 940 x 940. Macierz zawierała więc pięć diagonali - główna z elementami równymi 9, dwie sąsiednie i dwie skrajne diagonale z elementami równymi -1. **b** był wektorem o długości N, którego n−ty element miał wartość sin(5 * n).

3. Zadanie B

Celem zadania było zaimplementowanie dwóch metod iteracyjnych do rozwiązywania układów równań liniowych: metodę Jacobiego oraz metodę Gaussa-Seidla. Owe metody miały posłużyć do rozwiązania układu równań określonego w podpunkcie A. Wyniki reprezentuje poniższa tabela:

Metoda:	Liczba iteracji	Czas [s]
Jacobi	20	2.92
Gauss-Seidl	14	2.1

Liczba operacji oraz czas uzyskany przez obie metody jest związany z wymaganym warunkiem stopu, którym była wartość normy z wektora residuum mniejsza niż 10^{-9}. Metoda Gaussa-Seidla potrzebowała mniejszej liczby iteracji oraz osiągnęła warunek stopu w czasie o około 30% krótszym., co wynika z implementacji obu algorytmów gdzie Gauss-Seidl korzysta na bieżąco obliczonych wartości.

4. Zadanie C

W zadaniu należało zmodyfikować układ równań z podpunktu A poprzez zmianę wartości na głównej diagonali, która w tym momencie miała wynosić 3. Uruchomienie omawianych metod iteracyjnych tj. Jacobiego oraz Gaussa-Seidla powoduje nieotrzymanie wyniku z uwagi na wartość normy błędu rezydualnego dążącej do nieskończoności, a nie zgodnie z oczekiwaniami, do 0. Problem ten obnażyło wykroczenie wartości normy z wektora residuum poza zakres obsługiwany przez język *Python.* Podsumowując, metoda Jacobiego oraz metoda Gaussa-Seidla w tym przypadku nie zbiegają się.

5. Zadanie D

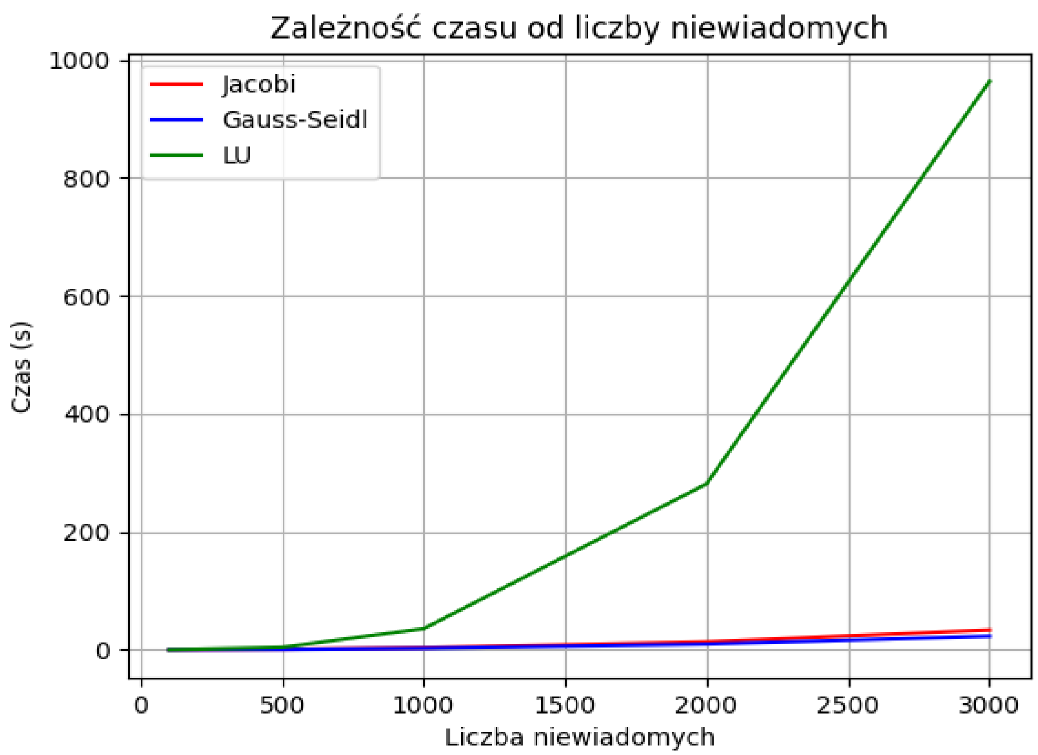
Zadanie polegało na przygotowaniu algorytmu rozwiązującego dany układ równań metodą bezpośrednią - faktoryzacji LU. Faktoryzacja LU polega na rozbiciu macierzy **A** na dwie macierze: trójkątną dolną **L** i trójkątną górną **U**. Następnie należy rozwiązać układ równań **L**y = b za pomocą podstawiania wprzód, a później obliczony wektor y wykorzystać do rozwiązania układu równań **U**x = y za pomocą podstawiania wstecz.

```
Zad D
Faktoryzacja LU
Norma z residuum:  9.59254397395839e-13
Czas:  28.81513524055481
```

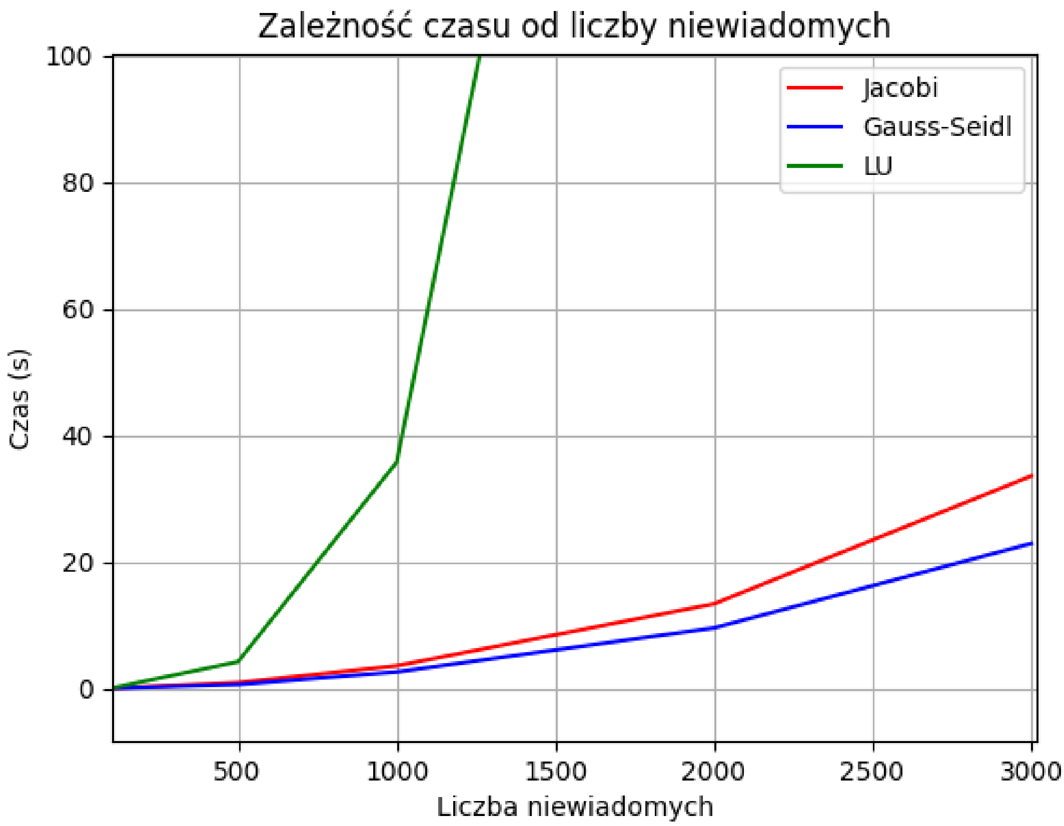
Powyższy zrzut ekranu obrazuje wartość normy z residuum oraz czas wykonania metody faktoryzacji LU dla układu równań z zadania C. Jak widać metoda bezpośrednia doprowadziła do otrzymania wyniku. Jest to wyższość nad omawianymi algorytmami iteracyjnymi, których z kolei niezaprzeczalnym atutem jest szybkość działania w porównaniu do metod bezpośrednich takich jak faktoryzacja LU. Uzyskana norma z residuum rzędu 10^{-13} jest na tyle znikoma, że można uznać ją za pomijalną.

6. Zadanie E

Celem zadania było bezpośrednie porównanie trzech metod, będących przedmiotem badania w projekcie: dwóch metod iteracyjnych – Jacobiego oraz Gaussa-Seidla jak i metody bezpośredniej - faktoryzacji LU. Porównanie polegało na zastosowaniu wszystkich trzech metod do rozwiązania układu równań określonego w podpunkcie A, z tą różnicą, że rozmiar N macierzy systemowej jak i wektora wyrazów wolnych przybierał wartości: 100, 500, 1000, 2000 i 3000. Czasy działania przedstawiono na wykresach:



Powyższy wykresy pokazuje czasy w sekundach jakie potrzebował każdy z algorytmów do osiągnięcia wyniku. Metoda faktoryzacji LU cechuje się znacznie gorszym czasem od swoich konkurentów. Dla macierzy 3000 x 3000 jest to niemal 1000 sekund co daje około 15 minut pracy komputera. Przy tym samych rozmiarze problemu metody iteracyjne oscylują na poziomie 30 sekund. Różnica ma swoje potwierdzenie w złożoności rozpatrywanych algorytmów. Dla metod Jacobiego i Gaussa-Seidla złożność to $O(n^2)$ gdzie dla faktoryzacji LU jest to $O(n^3)$



Wykres zamieszczony powyżej jest przybliżeniem wcześniej omawianego wykresu. Został umieszczony w celu porównania metod iteracyjnych. Jak można zauważyć metoda Gaussa-Seidla jest wydajniejsza od metody Jacobiego. Różnice nie są znaczące dla małej liczby niewiadomych, ale już przy kilku tysiącach elementów w każdym z wymiarów macierzy różnice są coraz wyraźniejsze. Wyższość metody Gaussa-Seidla polega na wykorzystywaniu bardziej aktualnych danych w obliczeniach. Metody te nie różnią się trudnością w implementacji stąd warto korzystać z metody Gaussa-Seidla. Podsumowując każda z metod ma swoje zalety i wady. Metody iteracyjne są z założenia szybkie ale nie zawsze doprowadzają nas do wyniku. Kluczem jest wybór odpowiedniej metody do danego problemu.