

## Metody Numeryczne

### Projekt 2 – Układ równań liniowych

## Wstęp

Celem tego projektu była implementacja metod rozwiązywania układów równań liniowych - metod iteracyjnych (Jacobiego i Gaussa-Seidla) oraz bezpośredniej z wykorzystaniem faktoryzacji LU. Testy poszczególnych metod będą przeprowadzane na układach równań, które powstają w wyniku dyskretyzacji równań różniczkowych i są powszechnie stosowane w takich zagadnieniach jak: elektronika, elektrodynamika, mechanika, termodynamika, dynamika płynów i wiele innych.

W praktyce najczęściej stosuje się tak zwany rzadki format przechowywania macierzy (który przechowuje tylko wartości niezerowe i ich położenie w macierzy), ponieważ zdecydowana większość elementów ma wartość 0. Jednak ze względu na prostotę, w zadaniu zdecydowałam się na wykorzystywanie tzw. formatu pełnego (przechowujący wszystkie wartości, również 0), który może być stosowany w problemach zawierających nie więcej niż kilka tysięcy niewiadomych.

## Wektor residuum

Metody iteracyjne przeważnie prowadzą do wyniku przybliżonego. Przy wykorzystaniu tych metod należy zastanowić się jakie przybliżenie wyniku jest dla nas zadowalające. Jako, że nie znamy dokładnego wyniku to nie możemy policzyć błędu rzeczywistego, ale możemy zbadać normę euklidesowa wektora residuum, dzięki któremu w każdej iteracji możemy obliczyć błąd jaki wynosi wektor  $x(k)$ .



## Układ równań

Pierwszym zadaniem jest stworzenie układu równań o następujących parametrach:

$$\mathbf{Ax} = \mathbf{b}$$

gdzie:

- Macierz A (macierz reprezentująca badana dziedzinę) o wymiarach NxN zawiera więc pięć diagonal - główna z elementami a1, dwie sąsiednie z elementami a2 i dwie skrajne diagonale z elementami a3.

$$a1 = 5 + 6 = 11$$

$$a2 = a3 = -1 \quad N = 909$$

- Wektor b (wektor pobudzenia) jest wektorem długości N, którego n-ty element ma wartość  $\sin(n \cdot (f + 1))$ .

- Wektor x to wektor rozwiązań'.

Macierz  $[\mathbf{A}|\mathbf{b}]$  prezentuję się następująco:

$$[\mathbf{A}|\mathbf{b}] = \left( \begin{array}{cccccccccc|c} 11 & -1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & \sin(1 * (5 + 1)) \\ -1 & 11 & -1 & -1 & 0 & 0 & 0 & \dots & 0 & \sin(2 * (5 + 1)) \\ -1 & -1 & 11 & -1 & -1 & 0 & 0 & \dots & 0 & \sin(3 * (5 + 1)) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & -1 & -1 & 11 & \sin(n * (5 + 1)) \end{array} \right)$$

## Metody iteracyjne rozwiązywania układów równań liniowych: Jacobi vs Gauss-Seidel – zadanie b

Zaimplementowanie wymienionych wcześniej metod iteracyjnych, dla macierzy zaprezentowanej wyżej, pozwoliło na uzyskanie następujących wyników:



	Gauss – Seidel	Jacobi
Liczba iteracji	15	21
Czas wykonania	0,162	0,224

Jako kryterium stopu przyjąłam normę z residuum mniejszą lub równą  $10^{-9}$ .

## Wnioski

Jak widzimy wykorzystanie metody Gaussa-Seidla zamiast metody Jacobiego pozwala nam uzyskać wynik w mniejszej liczbie iteracji, a co za tym idzie też w krótszym czasie. Wiadomo jednak, że metodę Jacobiego można zrównoleglić, więc uważam, że gdyby do obliczeń użyć większej ilości watorów to czas obliczeń byłby znacznie lepszy niż w przypadku metody Gaussa-Seidla. Wiadomo również, że zrównoleglenie nie wpłynęłoby pozytywnie na liczbę iteracji, po których wynik będzie zadowalająco blisko rzeczywistemu.

## Metody iteracyjne dla innego układu równań – zadanie C

Tak jak poprzednio tworzymy układ równań liniowych tyle, że jako dane wejściowe ustalamy inne parametry:

$$a_1 = 3$$

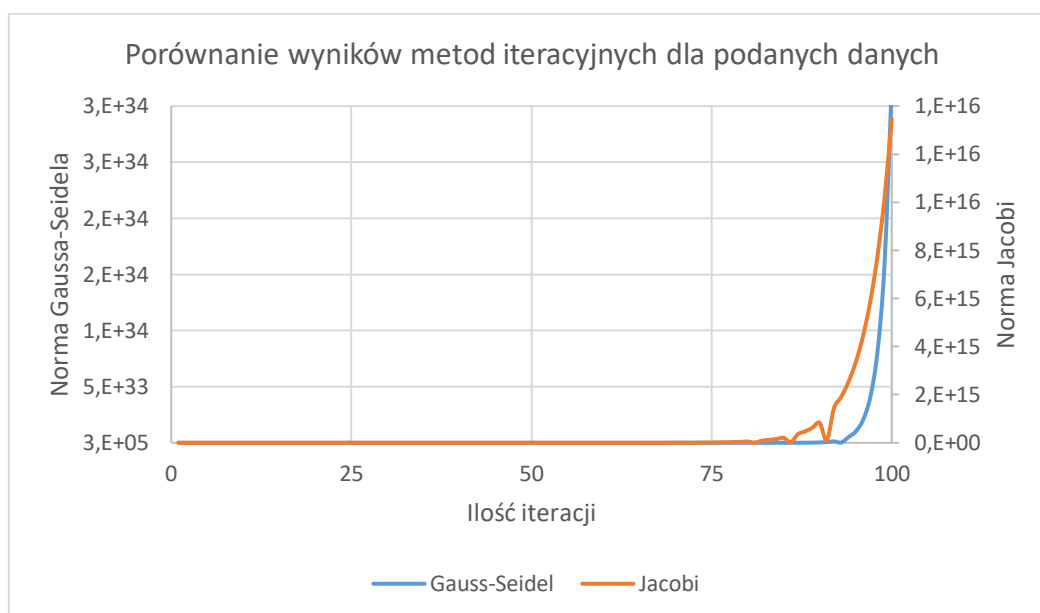
$$a_2 = a_3 = -1$$

$$N = 966$$

Teraz nasza macierz prezentuje się następująco:

$$[\mathbf{A}|\mathbf{b}] = \left( \begin{array}{cccccccc|c} 3 & -1 & -1 & 0 & 0 & 0 & 0 & \dots & 0 & \sin(1 * (5 + 1)) \\ -1 & 3 & -1 & -1 & 0 & 0 & 0 & \dots & 0 & \sin(2 * (5 + 1)) \\ -1 & -1 & 3 & -1 & -1 & 0 & 0 & \dots & 0 & \sin(3 * (5 + 1)) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & -1 & -1 & 3 & \sin(n * (5 + 1)) \end{array} \right)$$





Po zwracanych przez algorytmy normy z kolejnych wektorów residuum, które wzrastają przy każdej iteracji, widać że w tym przypadku metody iteracyjne nie zbiegają się.

Jest to spowodowane tym, że macierz nie jest diagonalnie dominująca.

## Faktoryzacja LU – zadanie D

Niestety metody iteracyjne dla macierzy z poprzedniej sekcji nie pozwoliły uzyskać żadnego rezultatu. W takim wypadku warto użyć jakiejś bezpośredniej metody rozwiązywania układów równań liniowych

Wyniki jakie uzyskałam dzięki skorzystaniu z tej metody to:

**Norma z residuum:  $9.11462e-13$**

**Czas wykonania: 2,448s**

Dla przypadku w którym metody iteracyjne nie dawały rozwiązania metoda bezpośrednia - faktoryzacji LU poprawnie rozwiązuje układ równań. Również osiągnięta przez nią wartość normy z wektora residuum jest mała. W tym przypadku jest to optymalna metoda.



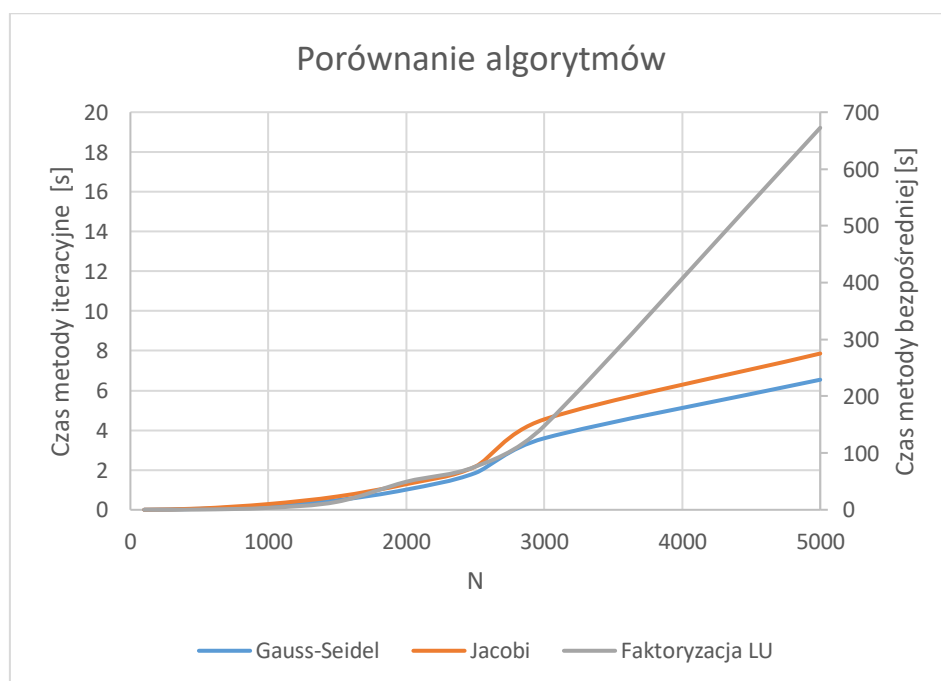
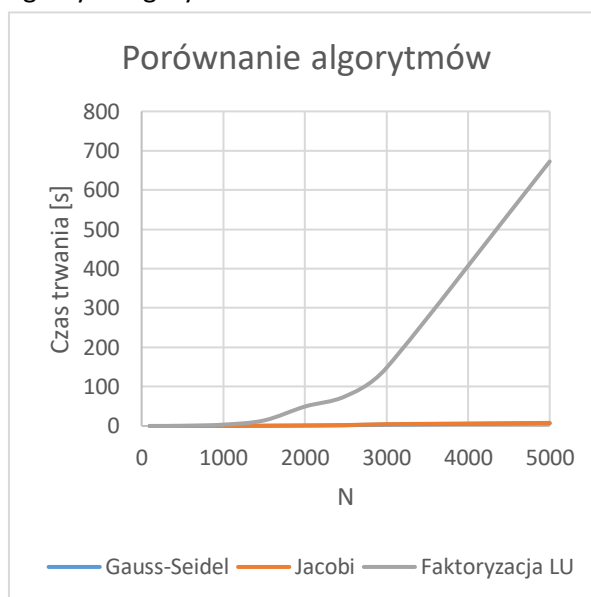
## Porównanie algorytmów

W celu porównania zachowania wcześniej wykorzystywanych metod stworzyłam 8 układów równań dla różnej liczby niewiadomych:

$$N = [100, 500, 1000, 1500, 2000, 2500, 3000, 5000]$$

Poniżej przedstawiam czasy obliczeń poszczególnych algorytmów:

N	Gauss-Seidel	Jacobi	Faktoryzacja LU
100	0,002	0,002	0,001
500	0,049	0,063	0,415
1000	0,211	0,288	3,408
1500	0,484	0,669	14,127
2000	1,015	1,29	49,382
2500	1,851	2,174	75,951
3000	3,598	4,549	147,722
5000	6,543	7,861	672,56



Metody bezpośrednie rozwiązywania układów równań charakteryzują się mniejszą złożonością czasową, kosztem złożoności obliczeniowej.

Wykres pokazuje, że metoda Gaussa-Seidela, chociaż nieznacznie różni się od metody Jacobiego jest o metodą zauważalnie szybsza, szczególnie dla układów równań zawierających wiele zmiennych. Należy również pamiętać, że operujemy na jednym wątku. Rezultaty myłyby inne, gdyby zrównoleglić metodę Jacobiego.

Metoda faktoryzacji LU przestaje być opłacalna dla macierzy o dużych rozmiarach ( $N > 1000$ ).

## *Zadanie F podsumowanie:*

Metody iteracyjne górują na metodami bezpośrednimi pod względem złożoności czasowej, lecz jedynie dla mniejszej ilości danych.

Niestety metody iteracyjne rozwiązywania układów równań, nie zawsze zwracają rozwiązanie. Gdy tak się dzieje, można zastosować metodę bezpośredniego rozwiązywania układu.

Wraz ze wzrostem ilości niewiadomych w układzie czas wykonywania obliczeń przez metodę Jacobiego wzrasta wykładniczo. Czas metody Gaussa-Seidela wzrasta wyraźnie wolniej. Jest to spowodowane wykorzystywaniem przez metodę Gaussa-Seidela wartości obliczonych w aktualnie wykonywanej iteracji.

