# Metody numeryczne, projekt 2: Układy równań liniowych

Franciszek Fabiński - s197797

6 maja 2025

## 1 Wstep teoretyczny

Tematyką drugiego projektu jest rozwiązywanie układów równań liniowych z wykorzystaniem dwóch metod iteracyjnych (metoda Jacobiego i metoda Gaussa-Seidela) oraz jednej metody bezpośredniej (faktoryzacji LU). Analiza danych metod będzie przeprowadzana na danych wywnioskowanych zgodnie ze schematem instrukcji projektu.

W rzeczywistych problemach takie układy równań często nie będą rozmiaru tutaj analizowanych, lecz znacznie większego, rzędu milionów niewiadomych. W takich przypadkach najmniejsza optymalizacja algorytmu może przenieść się na oszczędzenie wielu godzin, dni czy nawet tygodni obliczeń.

Do zrealizowania projektu wykorzystany został język programowania Python oraz biblioteka NumPy oraz biblioteka matplotlib do wizualizacji wyników. Wszelkie operacje na macierzach i wektorach są realizowane z wykorzystaniem wbudowanych funkcji NumPy, co zapewnia dużą wydajność obliczeń i pozwala skupić się na implementacji algorytmu.

## 2 Formalizm matematyczny i dane testowe

Instrukcja projektu używa stałych zależnych od numeru indeksu autora. W moim przypadku mają one następujące wartości:

- $c = 9$
- $d = 7$
- $e = 7$
- $f = 7$

Macierz $A$ jest macierzą wstęgową o szerokości pasma 5, a wektor $b$ jest wektorem jednostkowym. Macierz $A$ jest generowana w oparciu o następujący wzór:

$$A_{i,j} = \begin{cases} 5 + e = 12 & \text{jeśli } i = j \\ -1 & \text{jeśli } |i - j| < 3 \\ 0 & \text{w przeciwnym razie} \end{cases} \tag{1}$$

Elementy wektora $b$ są opisane wzorem:

$$b_i = \sin(i * (f + 1)) = \sin(i * 8) \tag{2}$$

Macierz $A$ ma rozmiar 1297x1297, a wektor $b$ ma rozmiar 1297x1. Rozmiar macierzy i wektora jest wyznaczany na podstawie wzoru:

$$n = 1200 + 10 * c + d = 1200 + 90 + 7 = 1297 \tag{3}$$

Macierz $A$ ma wymiary $1297 \times 1297$, a wektor $b$ ma wymiary $1297 \times 1$.

Ostatecznie dane wejściowe wyglądają następująco:

$$A = \begin{bmatrix} 12 & -1 & -1 & 0 & 0 & \cdots & 0 & 0 \\ -1 & 12 & -1 & -1 & 0 & \cdots & 0 & 0 \\ -1 & -1 & 12 & -1 & -1 & \cdots & 0 & 0 \\ 0 & -1 & -1 & 12 & -1 & \cdots & 0 & 0 \\ 0 & 0 & -1 & -1 & 12 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 12 & -1 \\ 0 & 0 & 0 & 0 & 0 & \cdots & -1 & 12 \end{bmatrix} \tag{4}$$

$$b = \begin{bmatrix} \sin(1*8) \\ \sin(2*8) \\ \sin(3*8) \\ \sin(4*8) \\ \sin(5*8) \\ \vdots \\ \sin(1297*8) \end{bmatrix} \tag{5}$$

## 2.1 Metoda bezpośrednia

Do metody bezpośredniej użyta została faktoryzacja LU. Faktoryzacja LU polega na rozkładzie macierzy $A$ na iloczyn dwóch macierzy: $L$ i $U$, gdzie $L$ jest macierzą dolnotrójkątną, a $U$ jest macierzą górnotrójkątną.

$$A = LU \tag{6}$$

W celu rozwiązania układu równań $Ax = b$ należy wykonać następujące kroki:

1. Rozwiązać układ równań $Ly = b$ w celu wyznaczenia wektora $y$.
2. Rozwiązać układ równań $Ux = y$ w celu wyznaczenia wektora $x$.

Do rozkładu macierzy $A$ na iloczyn macierzy $L$ i $U$ użyta została metoda Doolittle'a. Rezultatem metody Doolittle'a jest macierz $L$ i $U$, gdzie diagonalna część macierzy $L$ jest równa 1. Metoda ta jest kosztowna obliczeniowo, ponieważ wymaga $O(n^3)$ operacji arytmetycznych.

## 2.2 Metody iteracyjne

Do obydwu metod iteracyjnych macierz $A$ powinna być macierzą diagonalnie dominującą. Macierz $A$ jest diagonalnie dominująca, jeśli dla każdego wiersza $A_{ii}$ zachodzi:

$$|A_{ii}| > \sum_{j \neq i} |A_{ij}| \tag{7}$$

### 2.2.1 Metoda Jacobiego

Metoda Jacobiego jest jedną z najprostszych metod iteracyjnych. Polega na wyznaczeniu wektora $x$ w oparciu o poprzedni wektor $x^{(k)}$.

$$x_i^{(k+1)} = \frac{1}{A_{ii}} \left( b_i - \sum_{j \neq i} A_{ij} x_j^{(k)} \right) \tag{8}$$

Wektor $x^{(k+1)}$ jest wyznaczany na podstawie wektora $x^{(k)}$.

### 2.2.2 Metoda Gaussa-Seidela

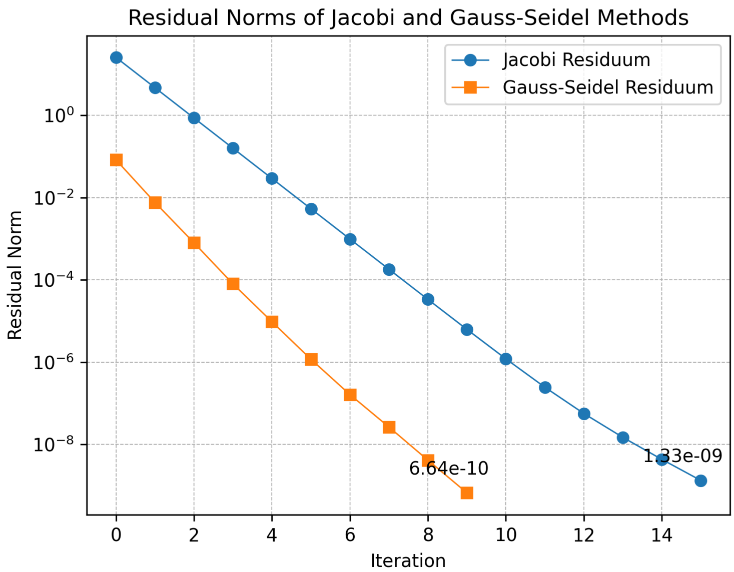
Metoda Gaussa-Seidela polega na wyznaczeniu wektora $x$ w oparciu o poprzedni wektor $x^{(k)}$ oraz aktualny wektor $x^{(k+1)}$.

$$x_i^{(k+1)} = \frac{1}{A_{ii}} \left( b_i - \sum_{j=1}^{i-1} A_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} A_{ij} x_j^{(k)} \right) \tag{9}$$

# 3 Analiza wyników

## 3.1 Poprawnie uwarunkowana macierz

Poniższe wyniki przedstawione są dla macierzy wygenerowanej zgodnie z wzorem (1), wg. którego macierz okazuje się być diagonalnie dominująca.

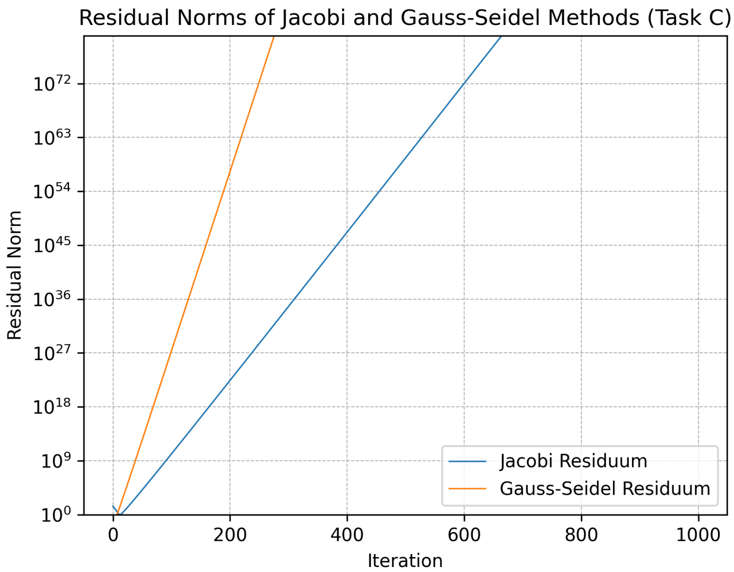



rys. 1: Wykres reszt dla metody Jacobiego i Gaussa-Seidela

Krzywe z rys. 1 przedstawiają zbieżność obu metod iteracyjnych. Metoda Gaussa-Seidela zbiega szybciej niż metoda Jacobiego, przez co potrzebowała ona jedynie 10 iteracji do osiągnięcia wyniku w granicach tolerancji ($10^{-9}$), o 5 iteracji mniej niż metodzie Jacobiego. W tym przypadku metodzie Gaussa-Seidela obliczenie wyniku zajęło **0.06** sekundy, a metodzie Jacobiego **0.09**.

## 3.2 Źle uwarunkowana macierz

W przypadku źle uwarunkowanej macierzy, która nie jest diagonalnie dominująca, zastosowanie metod iteracyjnych skutkuje brakiem ich zbieżności.




rys. 2: Wykres reszt dla metody Jacobiego i Gaussa-Seidela dla źle uwarunkowanej macierzy

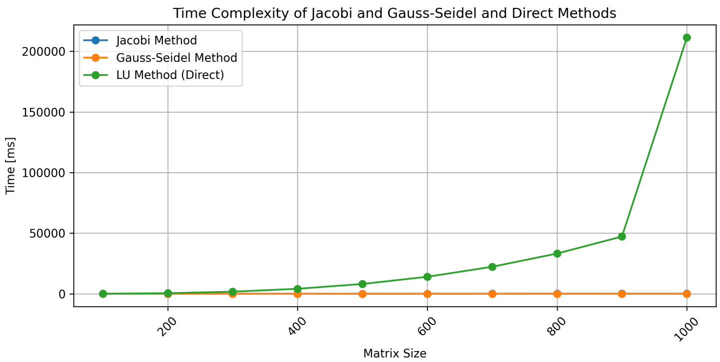
Na rys. 2 przedstawione są krzywe zbieżności dla źle uwarunkowanej macierzy. Jak widać, metoda Gaussa-Seidela rośnie szybciej niż metoda Jacobiego, lecz obydwie rosną wykładniczo. Dalsze iteracje powstrzymał odgórny limit iteracji, który został ustawiony na 1000.

## 3.3 Faktoryzacja LU

W przypadku rozwiązywania układu zawierającego źle uwarunkowaną macierz niemożliwe jest zastosowanie metod iteracyjnych. W takim przypadku zastosować można jedynie metodę bezpośrednią, która w tym przypadku zajęła **195.13** sekundy. Norma residuum wyniosła $1.68 * 10^{-10}$.

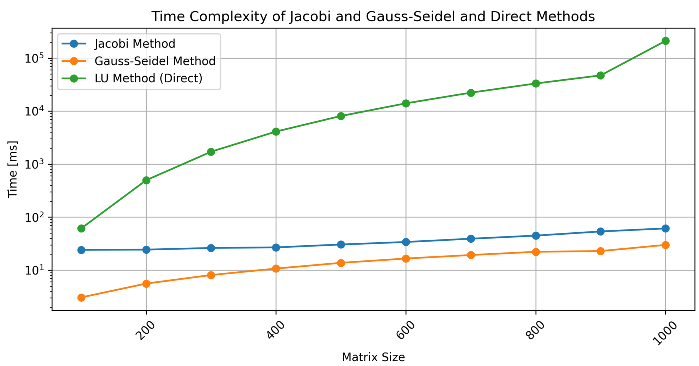
# 4 Porównanie czasów obliczeń

Przedstawione wyniki są czasami obliczeń dla macierzy dobrze uwarunkowanych. Dodatkowo dla pewności poprawnych wyników, dla każdej metody zwiekszono liczbę iteracji do 1000000.




rys. 3: Czasy obliczeń w skali liniowej

Na rys. 4 przedstawione są czasy obliczeń dla każdej z metod w skali logarytmicznej. Jak widać, metoda Jacobiego jest znacznie wolniejsza od metody Gaussa-Seidela. Mimo wszystko obydwie metody iteracyjne wydają się podobnie szybkie w porównaniu do metody bezpośredniej. Metoda bezpośrednia **znacznie** przewyższa czasem obliczeń metody iteracyjne, co widać na rys. 3 posiadającym skalę liniową. Metoda bezpośrednia jest jednak w stanie obliczyć układy równań z macierzami źle uwarunkowanymi, w przeciwieńswtwie do metod iteracyjnych.




rys. 4: Czasy obliczeń w skali logarytmicznej

## 5 Podsumowanie

W projekcie przedstawione zostały trzy metody rozwiązywania układów równań liniowych. Metoda Jacobiego i metoda Gaussa-Seidela są metodami iteracyjnymi, które są stosunkowo proste w implementacji, lecz wymagają diagonalnie dominującej macierzy. W przypadku źle uwarunkowanej macierzy, która nie jest diagonalnie dominująca, metody te nie zbiegają. Użycie metody bezpośredniej jest znacznie wolniejsze, lecz czasami niezbędne.

## References


[1] Geeks for Geeks (2023) *LU Decomposition*, `https://www.geeksforgeeks.org/doolittle-algorithm-lu-decomposition/`.

[2] Yousef Saad *Iterative Methods for Sparse Linear Systems*.