

# Sprawozdanie - Projekt 2

## Układy równań liniowych

### Implementacja metod Jacobiego, Gaussa-Seidla i LU

Damian Jankowski s188597

27 kwietnia 2023

## Spis treści

<b>1</b>	<b>Wstęp</b>	<b>1</b>
<b>2</b>	<b>Metody rozwiązywania układów równań liniowych</b>	<b>2</b>
2.1	Metody iteracyjne . . . . .	2
2.1.1	Metoda Jacobiego . . . . .	2
2.1.2	Metoda Gaussa-Seidla . . . . .	2
2.1.3	Podstawienie w przód . . . . .	3
2.1.4	Podstawienie w tył . . . . .	3
2.1.5	Warunek zakończenia . . . . .	3
2.2	Metody bezpośrednie . . . . .	4
2.2.1	Metoda faktoryzacji LU . . . . .	4
<b>3</b>	<b>Implementacja metod i analiza wyników</b>	<b>5</b>
3.1	Zadanie A . . . . .	5
3.2	Zadanie B . . . . .	5
3.3	Zadanie C . . . . .	6
3.4	Zadanie D . . . . .	6
3.5	Zadanie E . . . . .	7
<b>4</b>	<b>Wnioski</b>	<b>8</b>

## 1 Wstęp

Celem projektu było zaimplementowanie metod Jacobiego, Gaussa-Seidla i LU oraz porównanie wydajności, dokładności, jak również czasu ich wykonania.

Każda z metod w różny sposób rozwiązuje pewien układ równań liniowych:

$$Ax = b \quad (1)$$

gdzie:

- $A$  – macierz kwadratowa zawierająca współczynniki układu równań,
- $b$  – wektor wyrazów wolnych,
- $x$  – wektor rozwiązań układu.

Macierz  $A$  została zdefiniowana jako macierz pasmowa o rozmiarze  $997 \times 997$ :

$$A = \begin{bmatrix} a_1 & a_2 & a_3 & 0 & 0 & 0 & 0 & \dots & 0 \\ a_2 & a_1 & a_2 & a_3 & 0 & 0 & 0 & \dots & 0 \\ a_3 & a_2 & a_1 & a_2 & a_3 & 0 & 0 & \dots & 0 \\ 0 & a_3 & a_2 & a_1 & a_2 & a_3 & 0 & \dots & 0 \\ 0 & 0 & a_3 & a_2 & a_1 & a_2 & a_3 & \dots & 0 \\ 0 & 0 & 0 & a_3 & a_2 & a_1 & a_2 & \dots & 0 \\ 0 & 0 & 0 & 0 & a_3 & a_2 & a_1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & a_3 & a_2 & a_1 \end{bmatrix} \quad (2)$$

gdzie:

$$a_1 = 10, \quad a_2 = -1, \quad a_3 = -1$$

Wektor  $b$  długości 997 został zdefiniowany jako:

$$b = \begin{bmatrix} \sin(0 \cdot (f + 1)) \\ \sin(1 \cdot (f + 1)) \\ \sin(2 \cdot (f + 1)) \\ \vdots \\ \sin(996 \cdot (f + 1)) \end{bmatrix} \quad f = 8 \quad (3)$$

## 2 Metody rozwiązywania układów równań liniowych

Istnieje wiele metod rozwiązywania układów równań liniowych. W projekcie zostały zaimplementowane trzy z nich: metoda Jacobiego, Gaussa-Seidla i LU.

Pierwsze dwa należą do grupy metod iteracyjnych, natomiast ostatnia jest metodą bezpośrednią.

### 2.1 Metody iteracyjne

Metody iteracyjne polegają na wyznaczeniu kolejnych przybliżeń rozwiązania układu równań liniowych.

Korzystają one z tzw. macierzy: trójkątnej dolnej (*Lower*)  $L$ , górnej (*Upper*)  $U$  oraz diagonalnej  $D$ , które spełniają warunek:

$$A = L + U + D \quad (4)$$

Warunkiem zakończenia iteracji jest osiągnięcie zadanej dokładności lub maksymalnej liczby iteracji.

#### 2.1.1 Metoda Jacobiego

Metoda Jacobiego polega na wyznaczeniu kolejnych przybliżeń rozwiązania układu równań liniowych zgodnie ze wzorem:

$$x^{(k+1)} = -D^{-1}(L + U)x^{(k)} + D^{-1}b \quad (5)$$

gdzie:

- $x^{(k)}$  - wektor przybliżenia rozwiązania w  $k$ -tej iteracji

#### 2.1.2 Metoda Gaussa-Seidla

Metoda Gaussa-Seidla podobnie jak metoda Jacobiego polega na wyznaczeniu kolejnych przybliżeń, jednakże zgodnie z tym wzorem:

$$x^{(k+1)} = -(D + L)^{-1}Ux^{(k)} + (D + L)^{-1}b \quad (6)$$

Problemem tej metody jest konieczność wyznaczenia macierzy  $(D + L)^{-1}$ , czego powinno się unikać z racji dużej złożoności obliczeniowej.

Z tego powodu zamiast wyznaczać odwrotności macierzy  $D + L$ , stosuje się tzw. podstawienie w przód (ang. *forward substitution*).

### 2.1.3 Podstawienie w przód

Metoda podstawienia w przód polega na wyznaczeniu kolejnych wartości wektora rozwiązań  $\mathbf{x}$  układu równań  $\mathbf{L}\mathbf{x} = \mathbf{b}$ , w następujący sposób:

$$\begin{aligned}x_1 &= \frac{b_1}{l_{11}} \\x_2 &= \frac{b_2 - l_{21}x_1}{l_{22}} \\x_i &= \frac{b_i - \sum_{j=1}^{i-1} l_{ij}x_j}{l_{ii}} \\&\vdots \\x_n &= \frac{b_n - \sum_{j=1}^{n-1} l_{nj}x_j}{l_{nn}}\end{aligned}$$

Koniecznym jest by macierz  $\mathbf{L}$  była macierzą trójkątną dolną, np. suma macierzy  $\mathbf{D} + \mathbf{L}$  spełnia ten warunek.

### 2.1.4 Podstawienie w tył

Metoda podstawienia w tył podobnie jak metoda podstawienia w przód polega na wyznaczeniu kolejnych wartości wektora rozwiązań  $\mathbf{x}$  tym razem układu równań  $\mathbf{U}\mathbf{x} = \mathbf{b}$ . Jednakże w tym przypadku koniecznym jest by macierz  $\mathbf{U}$  była macierzą trójkątną górną. Kolejne kroki wyglądają następująco:

$$\begin{aligned}x_n &= \frac{b_n}{u_{nn}} \\x_{n-1} &= \frac{b_{n-1} - u_{n-1,n}x_n}{u_{n-1,n-1}} \\x_i &= \frac{b_i - \sum_{j=i+1}^n u_{ij}x_j}{u_{ii}} \\&\vdots \\x_1 &= \frac{b_1 - \sum_{j=2}^n u_{1j}x_j}{u_{11}}\end{aligned}$$

### 2.1.5 Warunek zakończenia

By sprawdzić czy osiągnięto zadaną dokładność należy przy każdej iteracji sprawdzać czy norma tzw. wektora residuum (błędu rezydualnego)  $\mathbf{res}$  jest mniejsza od zadanej wartości, np.  $10^{-9}$ .

Wektor residuum jest zdefiniowany następująco:

$$\mathbf{res}^{(k)} = \mathbf{A}\mathbf{x}^{(k)} - \mathbf{b} \quad (7)$$

W idealnej sytuacji powinien być równy wektorowi zerowemu.

Normę wektora residuum możemy obliczyć ze wzoru na normę euklidesową:

$$\|\mathbf{res}^{(k)}\| = \sqrt{\sum_{i=1}^n (\mathbf{res}_i^{(k)})^2} \quad (8)$$

Tym sposobem możemy z góry określić dokładność rozwiązania.

Czasem może dojść do sytuacji, w której metoda nie zbiega do rozwiązania, np. norma wektora residuum rośnie do nieskończoności. W takim przypadku należy sprawdzić czy macierz  $\mathbf{A}$  jest zdominowana przez przekątną, czyli czy zachodzi nierówność:

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad \text{dla} \quad i = 1, 2, \dots, n \quad (9)$$

Dodatkowo można określić limit ilości iteracji, po których metoda powinna zakończyć działanie, nawet jeśli nie osiągnęła zadanego poziomu dokładności. Do wykonania zadań został wykorzystany limit 1000 iteracji.

## 2.2 Metody bezpośrednie

Metody bezpośrednie polegają na wyznaczeniu rozwiązania układu równań bezpośrednio z macierzy współczynników  $\mathbf{A}$ . Odznaczają się one dużą dokładnością, jednakże są czasochłonne i zasobożerne.

### 2.2.1 Metoda faktoryzacji LU

Metoda faktoryzacji LU polega na rozkładzie macierzy współczynników  $\mathbf{A}$  na iloczyn macierzy  $\mathbf{L}$  i  $\mathbf{U}$ :

$$\mathbf{A} = \mathbf{LU} \quad (10)$$

gdzie:

- $\mathbf{L}$  - macierz trójkątna dolna
- $\mathbf{U}$  - macierz trójkątna górna

Wtedy układ równań  $\mathbf{Ax} = \mathbf{b}$  można zapisać jako:

$$\mathbf{LUx} = \mathbf{b}$$

By przejść dalej konieczne jest wyznaczenie potrzebnych macierzy pomocniczych.

1. Na początku tworzy się macierz  $\mathbf{L}$ , która jest macierzą jednostkową, czyli taką, której elementy na głównej przekątnej są równe 1. Natomiast macierz  $\mathbf{U}$  to kopia macierzy  $\mathbf{A}$ .

Faktoryzację LU opisać można tym kodem w języku C:

```
for (int k = 0; k < n - 1; k++) {
    for (int j = k + 1; j < n; j++) {
        L[j][k] = U[j][k] / U[k][k];
        for (int i = k; i < n; i++) {
            U[j][i] = U[j][i] - L[j][k] * U[k][i];
        }
    }
}
```

2. Następnie metodą podstawiania w przód wyznacza się wektor  $\mathbf{y}$ , który jest rozwiązaniem układu równań  $\mathbf{Ly} = \mathbf{b}$ .
3. Ostatnim krokiem jest wyznaczenie wektora rozwiązań  $\mathbf{x}$ , który jest rozwiązaniem układu równań  $\mathbf{Ux} = \mathbf{y}$ . Z racji, że macierz  $\mathbf{U}$  jest macierzą trójkątną górną, to wyznaczenie wektora  $\mathbf{x}$  jest możliwe metodą podstawiania w tył.

Jak widać złożoność obliczeniowa tej metody wynosi  $O(n^3)$ , dlatego nie jest zalecana do rozwiązywania układów równań o dużych rozmiarach.

### 3 Implementacja metod i analiza wyników

Implementację metod przedstawionych w poprzednim rozdziale wykonano w języku C. Do pomiaru czasu wykorzystano funkcję `clock()` z biblioteki `time.h`, natomiast do wyznaczenia sinusa wykorzystano bibliotekę `math.h`.

W pliku `main.c` znajduje się funkcja `main()`, która odpowiada za wykonywanie zadań. Plik `metody.c` zawiera implementacje metod iteracyjnych oraz bezpośredniej LU, natomiast plik `funkcje.c` – wszystkie implementacje potrzebnych operacji na macierzach i wektorach.

#### 3.1 Zadanie A

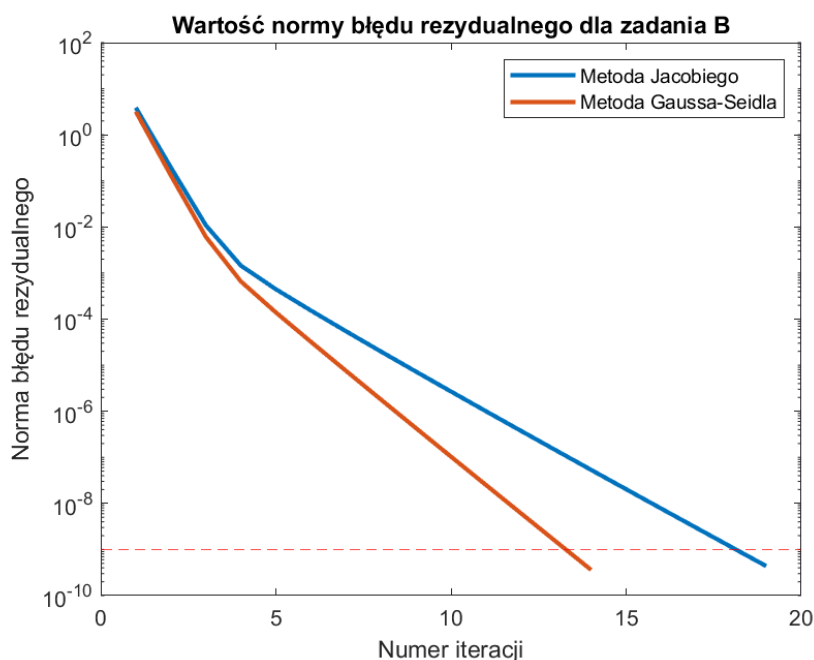
Zadanie A polegało na zaimplementowaniu układu równań przedstawionego we wstępie.

#### 3.2 Zadanie B

Zadanie B polegało na zaimplementowaniu metod iteracyjnych Jacobiego i Gaussa-Seidla oraz sprawdzeniu ilości iteracji potrzebnych do rozwiązania układu równań z zadania A. Dodatkowo należało zmierzyć czas potrzebny na rozwiązanie układu równań przez obie metody, tj. czas potrzebny by norma wektora residuum była mniejsza od  $10^{-9}$ . Wyniki zostały przedstawione poniżej.

Metoda	Czas	Ilość iteracji
Jacobi	0.132s	18
Gauss-Seidel	0.105s	14

Tabela 1: Wyniki pomiarów czasu i ilości iteracji dla zadania A



Rysunek 1: Wykres przedstawiający wartość normy wektora residuum od iteracji dla metod iteracyjnych

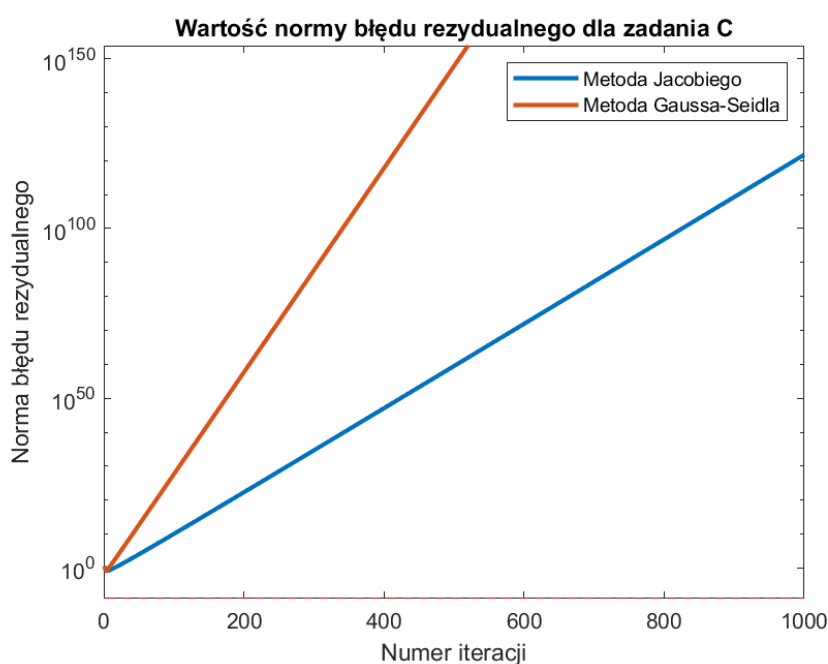
Jak widać na wykresie oraz w tabeli, metoda Gaussa-Seidla zbiega szybciej do rozwiązania niż metoda Jacobiego. Jest to spowodowane tym, że w metodzie Gaussa-Seidla wykorzystywane są już nowe wartości, natomiast w metodzie Jacobiego wykorzystywane są wartości z poprzedniej iteracji.

### 3.3 Zadanie C

Zadanie C podobnie jak B polegało na porównaniu działania metod iteracyjnych. Natomiast trzeba było dokonać pewnej zmiany w budowie macierzy  $A$ . Wartość na głównej przekątnej macierzy  $a_{11}$  została zastąpiona wartością 3. Dla takiego układu równań wyniki prezentują się następująco:

Metoda	Czas	Ilość iteracji
Jacobi	7.324s	1000
Gauss-Seidel	3.812s	521

Tabela 2: Wyniki pomiarów czasu i ilości iteracji dla zadania C



Rysunek 2: Wykres przedstawiający wartość normy wektora residuum od iteracji dla metod iteracyjnych

Obie metody nie zbiegły się do rozwiązania. Metoda Gaussa-Seidla zakończyła się po 521 iteracjach z racji, że wartość normy wektora błędu rezydualnego doszła do nieskończoności.

Powodem takiego zachowania jest to, że macierz  $A$  nie jest macierzą diagonalnie dominującą (9).

### 3.4 Zadanie D

W zadaniu D trzeba było zaimplementować metodę faktoryzacji LU i rozwiązać ten sam zmodyfikowany układ z zadania C, którego metody iteracyjne nie zdołały rozwiązać. Rezultaty przedstawia poniższa tabela.

Metoda	Czas	Wartość normy błędu rezydualnego
Faktoryzacja LU	1.14s	$2.414325 \cdot 10^{-13}$

Tabela 3: Wyniki pomiaru czasu oraz normy błędu dla zadania D

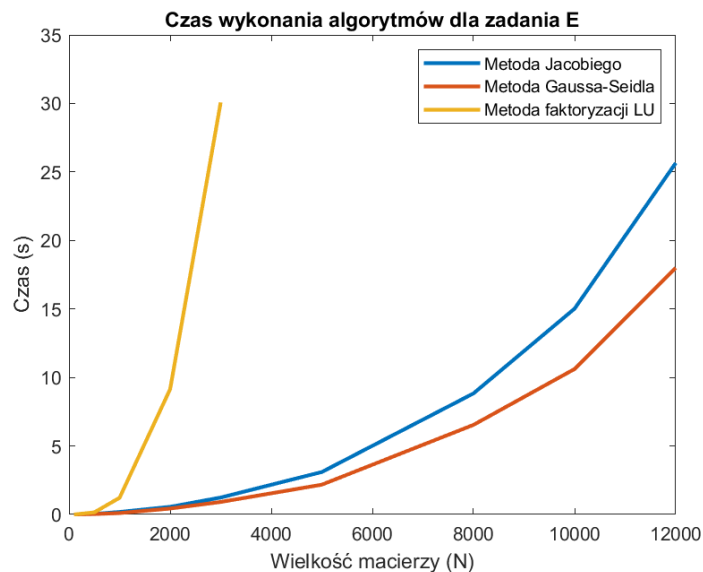
Wartość normy błędu rezydualnego jest bardzo mała, co oznacza, że metoda faktoryzacji LU poprawnie rozwiązała układ równań zadanego w zadaniu C.

### 3.5 Zadanie E

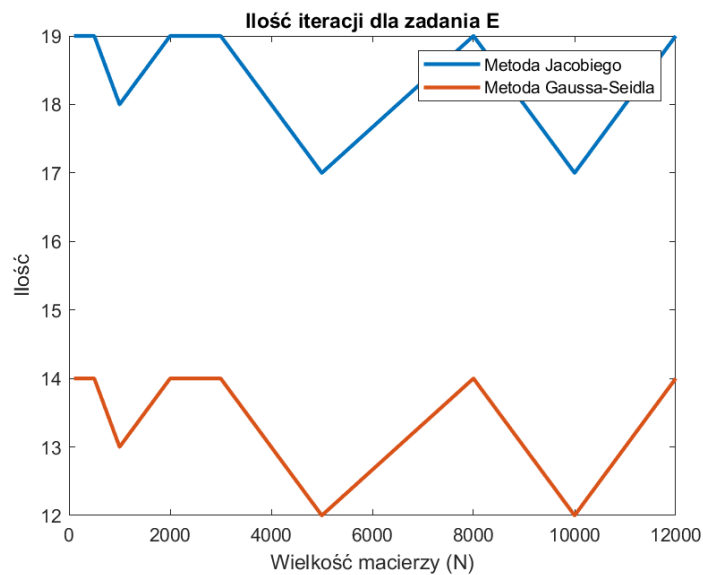
Zadanie E miało na celu porównanie czasu działania i ilości potrzebnych iteracji do rozwiązania układu z zadania A dla różnych rozmiarów macierzy  $A$ .

Do wykonania zadania przyjęto następujące wartości  $N$ :

- $N_1 = 100$
- $N_2 = 500$
- $N_3 = 1000$
- $N_4 = 2000$
- $N_5 = 3000$
- $N_6 = 5000$
- $N_7 = 8000$
- $N_8 = 10000$
- $N_9 = 12000$



Rysunek 3: Wykres przedstawiający czas działania metod od rozmiaru macierzy  $A$



Rysunek 4: Wykres przedstawiający ilość iteracji metod iteracyjnych od rozmiaru macierzy  $A$

Jak widać na wykresach, metoda faktoryzacji LU jest najwolniejsza (dla  $N > 3000$  zdecydowałem o przerwaniu obliczeń dla tej metody, ponieważ czas wykonania był zbyt długi), natomiast metoda Gaussa-Seidla jest najszybsza. Potrzebowała ona również mniejszej ilości iteracji niż metoda Jacobiego, aby osiągnąć odpowiednio dokładne rozwiązanie.

Co ciekawe ilość iteracji potrzebnych do rozwiązania układu równań dla obu metod nie rosła wraz z rozmiarem macierzy  $A$ . Wynika to z tego, że wartości w macierzy  $A$  oraz w wektorze  $b$  nie zmieniają się.

## 4 Wnioski

Wyniki testów pokazały, że metoda Gaussa-Seidla działa szybciej niż metoda Jacobiego w przypadku rozwiązywania danego układu równań.

Metoda faktoryzacji LU jest zdecydowanie wolniejsza od metod iteracyjnych, natomiast radzi sobie z każdym układem równań, niezależnie od tego czy macierz  $A$  jest diagonalnie dominująca czy nie.

Warto jednak zauważyć, że metody iteracyjne również mają swoje zalety, takie jak mniejsze wymagania pamięciowe, co może prowadzić do lepszych osiągnięć w niektórych przypadkach.

Ostatecznie, wybór metody rozwiązania liniowego układu równań zależy od specyficznych wymagań i ograniczeń problemu, takich jak dokładność, czas obliczeń, pamięć, struktura macierzy itp.

## Źródła

- [1] Kurs e-nauczanie Metody Numeryczne (Informatyka) - 2023  
*Instrukcja do laboratorium 3*  
*Wykład 3 i 4*
- [2] AGH, *Metoda Jacobiego oraz Gaussa-Seidla*  
<http://sendzimir.metal.agh.edu.pl/~im4/metnum/dyd/pm/iteracyjne.htm>
- [3] Wikipedia, *Metoda Gaussa-Seidla*  
[https://pl.wikipedia.org/wiki/Metoda\\_Gaussa-Seidla](https://pl.wikipedia.org/wiki/Metoda_Gaussa-Seidla)
- [4] Wikipedia, *Faktoryzacja LU*  
[https://pl.wikipedia.org/wiki/Faktoryzacja\\_LU](https://pl.wikipedia.org/wiki/Faktoryzacja_LU)
- [5] Jerzy Wałaszek, *Metody numeryczne - Rozkład LU i układ równań liniowych*  
[https://eduinf.waw.pl/inf/alg/008\\_nm/0026.php](https://eduinf.waw.pl/inf/alg/008_nm/0026.php)
- [6] Wikipedia, *Macierz diagonalnie dominująca*  
[https://pl.wikipedia.org/wiki/Macierz\\_diagonalnie\\_dominuj%C4%85ca](https://pl.wikipedia.org/wiki/Macierz_diagonalnie_dominuj%C4%85ca)