

# UKŁADY RÓWNAŃ LINIOWYCH

## Metody Numeryczne – Projekt nr 2 – Sprawozdanie Wydział Elektroniki, Telekomunikacji i Informatyki, 2022/2023

### Podstawowe metody rozwiązywania równań liniowych

Metody rozwiązywania układów równań liniowych, takie jak metody Jacobiego, Gaussa-Seidla czy faktoryzacja LU, są kluczowe w wielu dziedzinach nauki i techniki. Stąd istotne jest sprawne ich rozwiązywanie. Aby móc korzystać z zaawansowanych metod rozwiązań należy zapoznać się z najprostszymi, które stanowią bazę dla innych. Specyficznie, zastosowaliśmy i porównaliśmy trzy różne metody numeryczne: metodę Jacobiego, metodę Gaussa-Seidla oraz faktoryzację LU. Dwie pierwsze metody nazywane są metodami iteracyjnymi czyli takimi które spełniają pewne warunki i w kolejnych iteracjach wyznaczają coraz dokładniejsze przybliżenia rozwiązania. Metoda faktoryzacji LU to metoda bezpośrednia, która pozwala otrzymać dokładne rozwiązanie. Wymienione algorytmy zostały porównane pod względem rezultatów i zastosowane do rozwiązania układu równań postaci  $\mathbf{Ax} = \mathbf{b}$ .

W realizacji zadania wykorzystałam język *Python*, z bibliotekami *matplotlib*, *time*. Wszystkie operacje na macierzach zostały jednak zaimplementowane samodzielnie.

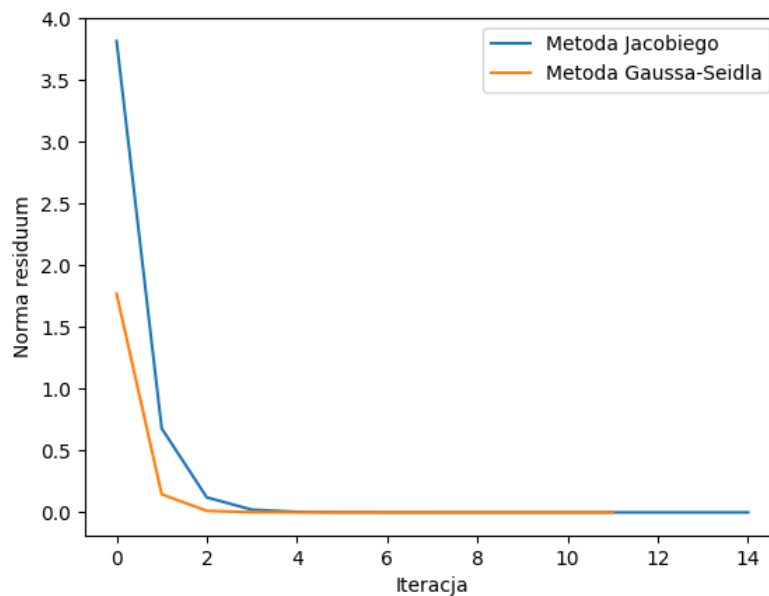
### Zadanie A

Macierz  $A$  która została zaimplementowana jest tzw. Macierzą pasmową o rozmiarze **919 x 919** zawierająca 5 diagonal. Główna z elementami  $a_1 = 13$ , dwie sąsiednie  $a_2$  oraz dwie skrajne diagonale z elementami  $a_3$  ( $a_2 = a_3 = -1$ ). **Wektor  $b$**  ma długość  $N = 919$  i zawiera elementy  $b_n = \sin(2 \cdot n)$

### Zadanie B

Implementacja metod iteracyjnych (metoda Jacobiego i Metoda Gaussa-Seidla) rozwiązywania układów miała na celu porównanie ilości potrzebnych iteracji i czas trwania każdego z tych algorytmów aby otrzymać normę residuum równą  $10^{(-9)}$  Wyniki przedstawiają się następująco:

```
Dla a1 = 5 + 7, a2 = a3 = -1:  
Algorytm Jacobiego:  
Czas trwania: 17.63665223121643  
Liczba iteracji: 15  
Norma residuum: 8.762164104689038e-10  
  
Algorytm Gaussa-Seidla:  
Czas trwania: 7.410661697387695  
Liczba iteracji: 12  
Norma residuum: 2.345548974657906e-10
```



Porównując obie metody istotny jest czas trwania algorytmów. Algorytm Gaussa-Seidla zdecydowanie przewyższał algorytm Jacobiego pod względem szybkości, co widać po czasie trwania (**7.41s do 17.64s**). Metoda Gaussa-Seidla jest bardziej efektywna czasowo dla tego konkretnego układu równań.

Innym kryterium jest liczba iteracji potrzebna do osiągnięcia założonego kryterium zbieżności. Metoda Gaussa-Seidla potrzebuje jedynie **12 iteracji** do **15 iteracji** w przypadku drugiej metody.

Norma residuum, jest miarą dokładności uzyskanego rozwiązania w obu przypadkach była na niskim poziomie, co sugeruje, że oba algorytmy poradziły sobie dobrze z zadaniem. Norma residuum metody Gaussa-Seidla była niższa, co sugeruje, że ta metoda mogła wskazać dokładniejsze rozwiązanie.

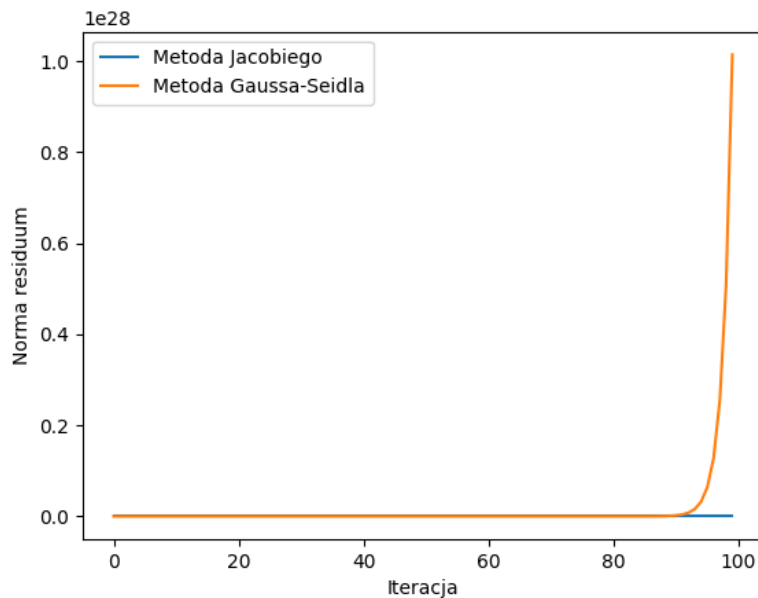
Metoda Gaussa-Seidla okazała się bardziej efektywna zarówno pod względem czasu trwania, liczby iteracji, jak i dokładności rozwiązania dla omawianego układu równań. Warto jednak podkreślić, że wyniki te mogą się różnić w zależności od specyfiki danego układu równań, dlatego zawsze warto rozważyć różne metody i dostosować je do konkretnego problemu.

### Zadanie C

Metody iteracyjne, takie jak metoda Jacobiego i Gaussa-Seidla, mają pewne warunki, które muszą być spełnione, aby metody te mogły zbiec do prawdziwego rozwiązania. Nie spełnienie tych warunków, może spowodować, że metody iteracyjne mogą nie zbiec, czyli, że nie znajdą dokładnego rozwiązania. W takim przypadku, może być konieczne zastosowanie innej metody rozwiązywania układu równań.

```
Dla a1 = 3, a2 = a3 = -1:
Algorytm Jacobiego:
Czas trwania: 35.20334219932556
Liczba iteracji: 100
Maksymalne residuum: 9434488509.66396

Algorytm Gaussa-Seidla:
Czas trwania: 43.23769927024841
Liczba iteracji: 100
Maksymalne residuum: 1.0139834271620832e+28
```



Wyniki pokazują, że żadna z metod nie była w stanie znaleźć rozwiązania układu równań w określonej liczbie iteracji (możliwość zmiany w kodzie), co sugeruje, że metody te mogą nie zbiegać dla tego konkretnego układu równań.

W przypadku metody Jacobiego, po **100 iteracjach**, maksymalne residuum wynosiło **9434488509.66396**, co jest bardzo dużą wartością, sugerującą znacząco inny wynik od rzeczywistego rozwiązania. Metoda Gaussa-Seidla również nie zdołała zbiec w **100 iteracjach**, a maksymalne residuum wynosiło nawet **1.0139834271620832e+28**, co jest znacznie większą wartością niż dla metody Jacobiego.

Żadna z metod nie zdołała efektywnie rozwiązać układu równań w określonej liczbie iteracji, co wskazuje na ich niezbieżność dla tego konkretnego układu.

Wykres metody Jacobiego utrzymuje się na stałym poziomie, czyli nie jest w stanie poprawić obecnego przybliżenia. Model nie poprawia swojej skuteczności dlatego jest płaski. Dla drugiego algorytmu residuum rośnie, czyli to sytuacja, w której algorytm jest niestabilny i zamiast zbiegać, odchodzi od rozwiązania.

## Zadanie D

Metoda faktoryzacji LU to metoda bezpośrednia, co oznacza, że dąży do uzyskania dokładnego rozwiązania, w przeciwieństwie do metod iteracyjnych, które zbliżają się do rozwiązania w nieskończoności. Polega na rozkładzie macierzy układu równań na iloczyn dwóch macierzy: dolnej trójkątnej L i górnej trójkątnej U. Rozkład LU umożliwia przekształcenie pierwotnego układu równań do formy, która jest łatwiejsza do rozwiązania.

```
Metoda faktoryzacji LU:
Czas faktoryzacji LU: 83.37582349777222

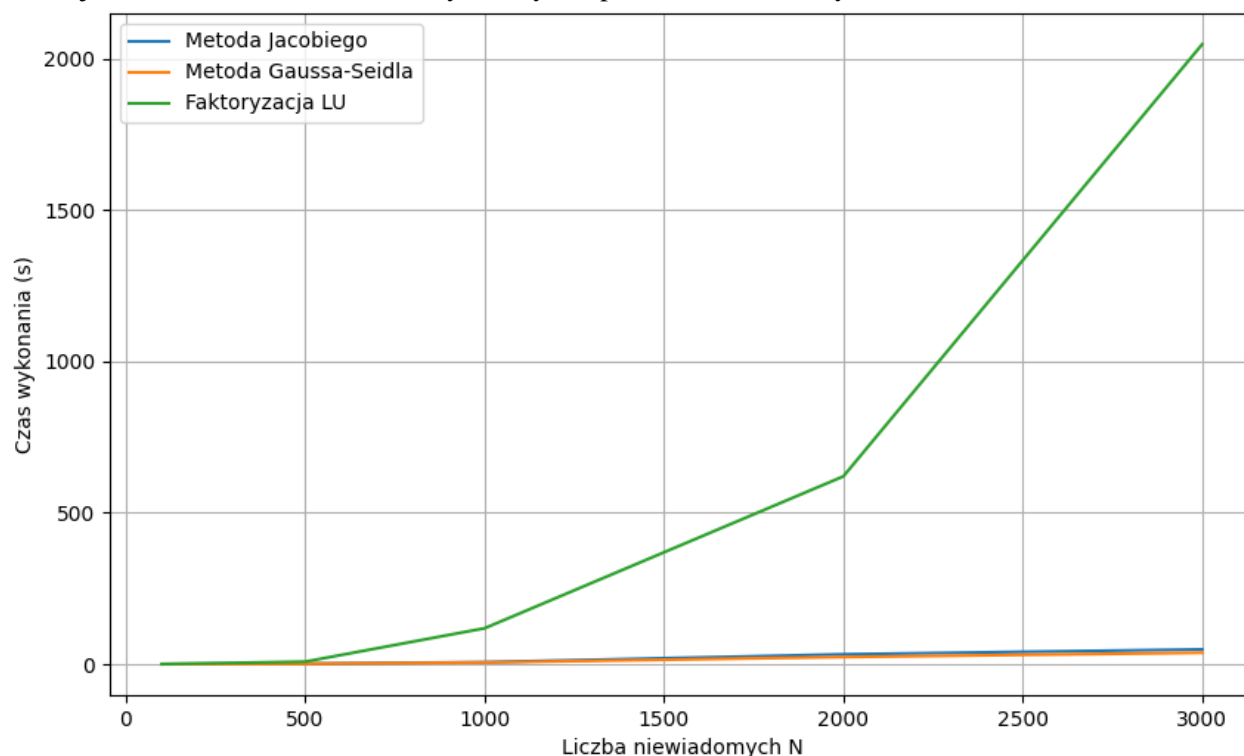
Norma residuum: 8.14214975811311e-13
```

Czas faktoryzacji LU, wynoszący **83,38s**, może wydawać się dość duży. Jednakże, wartości czasu trwania są zależne od wielu czynników, takich jak wielkość układu równań, wydajność komputera, na którym przeprowadzano obliczenia i implementacji samej metody.

Norma residuum jest bliska zeru, co wskazuje na wysoką precyzję uzyskanego rozwiązania.

## Zadanie E

Trzy zaprezentowane algorytmy zostały także przetestowane i porównane pod kątem ich złożoności czasowej w zależności od rozmiaru danych. Wyniki przedstawiono na wykresie.



W przypadku małych układów równań metody iteracyjne oraz metoda bezpośrednia dostarczyły rozwiązanie w podobnym czasie. Przy liczbach niewiadomych 2000, 3000 można zauważyć że metoda bezpośrednia potrzebuje zdecydowanie więcej czasu niż metody iteracyjne. Dla metod iteracyjnych gdy zwiększa się liczba niewiadomych można zauważyć powolną rosnącą różnicę w czasie wykonania tych metod.

|      | M. JACOBIEGO                         | M. GAUSSA-SEIDLA                     | M. LU                                |
|------|--------------------------------------|--------------------------------------|--------------------------------------|
| 100  | Czas trwania:<br>0.06886672973632812 | Czas trwania:<br>0.04290032386779785 | Czas trwania:<br>0.09652829170227051 |
| 500  | Czas trwania:<br>2.2396738529205322  | Czas trwania:<br>1.5321168899536133  | Czas trwania:<br>12.539247512817383  |
| 1000 | Czas trwania:<br>9.531396865844727   | Czas trwania:<br>4.497114896774292   | Czas trwania:<br>101.7416718006134   |
| 2000 | Czas trwania:<br>46.38270401954651   | Czas trwania:<br>32.35386538505554   | Czas trwania:<br>916.3515961170197   |
| 3000 | Czas trwania:<br>73.67983341217041   | Czas trwania:<br>58.75905418395996   | Czas trwania:<br>2044.9113478660583  |

## Zadanie F

Metody iteracyjne Jacobiego i Gaussa-Seidla wykazały różne wyniki w zależności od parametrów układu równań. W przypadku A, metoda Gaussa-Seidla była szybsza i wymagała mniej iteracji niż metoda Jacobiego. W przypadku C, żadna z metod nie zbiegła, co sugeruje, że nie były one odpowiednie dla tego konkretnego układu równań.

Z drugiej strony, metoda faktoryzacji LU wykazała wysoką precyzję rozwiązania, jak wskazuje bardzo niska norma residuum. Czas faktoryzacji był jednak relatywnie duży, co może sugerować, że metoda może być mniej efektywna dla większych układów równań lub na mniej wydajnych komputerach.

Wszystko to pokazuje, jak różne metody mogą dawać różne wyniki w zależności od specyfiki układu równań i podkreśla znaczenie wyboru odpowiedniej metody dla danego problemu.