

Metody numeryczne

Projekt 2 – Układy równań liniowych

Mateusz Kowalczyk, s188717

1. Wstęp

Celem projektu była implementacja oraz porównanie metod rozwiązywania układów równań liniowych danych w postaci macierzowej:

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

gdzie:

\mathbf{A} – macierz kwadratowa o rozmiarze $N \times N$,

\mathbf{x} – szukany wektor rozwiązań,

\mathbf{b} – wektor o rozmiarze $N \times 1$.

Pierwszą z zaimplementowanych metod była iteracyjna **metoda Jacobiego** pozwalająca na wyznaczenie w k iteracjach rozwiązanie przybliżonego

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

gdzie:

\mathbf{D} – macierz diagonalna zawierająca elementy z głównej przekątnej macierzy \mathbf{A} ,

\mathbf{L} – macierz dolna trójkątna zawierająca elementy macierzy \mathbf{A} znajdujące się poniżej głównej przekątnej pomnożone przez -1 ,

\mathbf{U} – macierz górna trójkątna zawierająca elementy macierzy \mathbf{A} znajdujące się powyżej głównej przekątnej pomnożone przez -1 .

Iteracje wykonuje się do spełnienia warunku stopu, np. do momentu, aż norma wektora residuum, danego w k -tej iteracji wzorem

$$\mathbf{res}^{(k)} = \mathbf{Ax}^{(k)} - \mathbf{b},$$

osiągnie zadowalająco niską wartość. Oznacza to, iż uzyskany wektor rozwiązań przybliży dostatecznie rozwiązanie dokładne. W niniejszym projekcie używana była norma 2.

Drugą zaimplementowaną metodą była iteracyjna **metoda Gaussa-Seidla**. W jej przypadku rozwiązanie przybliżone w k -tej iteracji przyjmuje wartość

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

W tym przypadku również warunek zakończenia iteracji może dotyczyć normy wektora residuum.

Trzecią metodą zaimplementowaną w ramach projektu była bezpośrednia **metoda faktoryzacji LU**. Polega ona na rozkładzie macierzy \mathbf{A} na macierze dolnotrójkątną \mathbf{L} i górnortrójkątną \mathbf{U} , takie że

$$\mathbf{A} = \mathbf{LU},$$

a następnie, metodą podstawiania wprzód, wyznaczeniu wektora \mathbf{y} z równania

$$\mathbf{L}\mathbf{y} = \mathbf{b},$$

po czym, metodą podstawiania wstecz, wyznaczeniu rozwiązania \mathbf{x} z równania

$$\mathbf{U}\mathbf{x} = \mathbf{y}.$$

2. Oprogramowanie

Do wykonania projektu użyty został język programowania Python. Możliwość tworzenia wykresów zapewniła biblioteka Matplotlib. Do tworzenia kodu wykorzystano zintegrowane środowisko programistyczne PyCharm.

3. Zadanie A

Dla numeru indeksu 188717 otrzymane zostały następujące wartości:

$$c = 1, d = 7, e = 7, f = 8,$$

$$N = 917,$$

$$a_1 = 12,$$

$$a_2 = a_3 = -1.$$

Stworzona została zatem macierz pasmowa \mathbf{A} o wymiarach 917×917 z pięcioma niezerowymi przekątnymi:

– główną z elementami 12,

– dwoma sąsiednimi oraz dwoma skrajnymi z elementami -1

oraz wektor \mathbf{b} o długości 917, którego n -ty element miał wartość $\sin(9n)$.

4. Zadanie B

Zaimplementowane zostały opisane we wstępie metody Jacobiego i Gaussa-Seidla rozwiązywania układów równań liniowych. Przyjęto następujący warunek zakończenia iteracji:

a) norma wektora residuum jest nie większa niż 10^{-9} lub

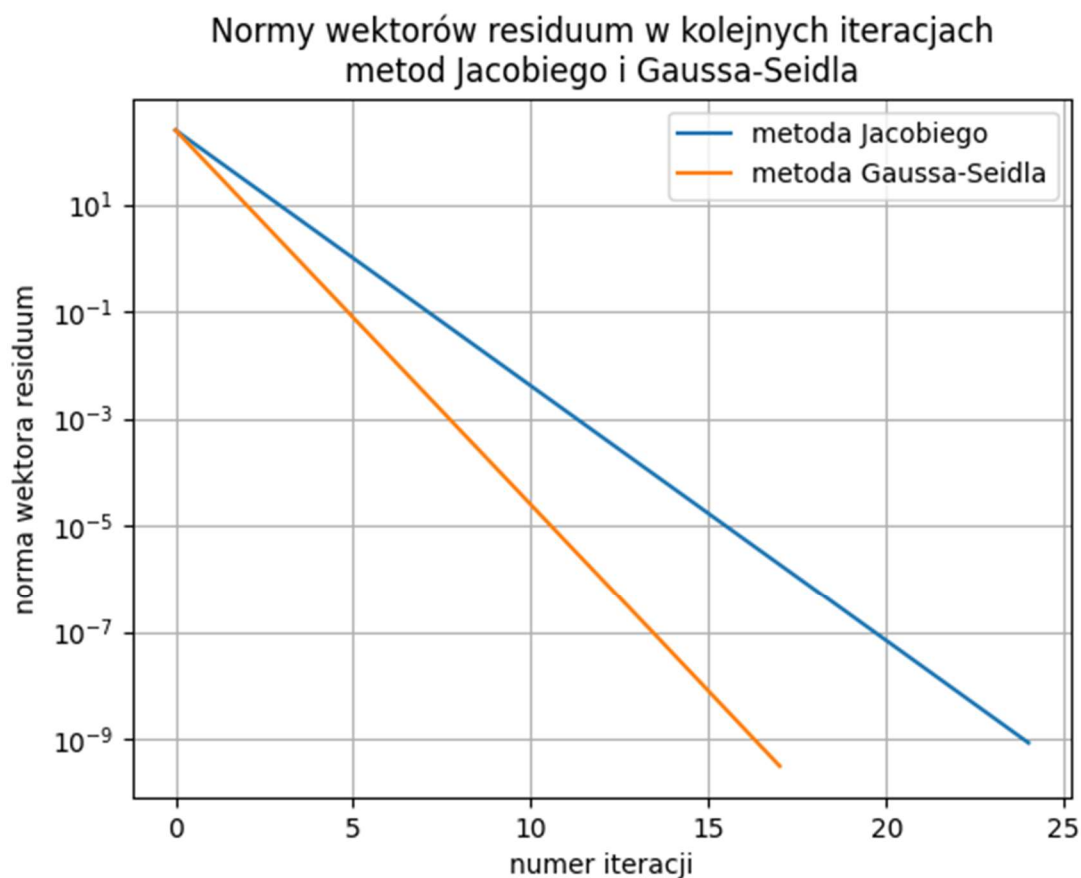
b) norma wektora residuum przekracza dopuszczalny rozmiar zmiennej (Python uznaje ją za nieskończoną) lub

c) wykonano już 600 iteracji.

W przypadku układu równań z tego podpunktu obiema metodami iteracyjnymi udało się spełnić warunek a i otrzymano następujący rezultat:

- metodą Jacobiego:
 - układ został rozwiązany po 24 iteracjach,
 - czas przygotowywania macierzy wyniósł ok. 1,73 s,
 - czas wykonywania iteracji wyniósł ok. 3,58 s,
 - ostateczna norma residuum wyniosła ok. $8,5 \cdot 10^{-10}$;
- metodą Gaussa-Seidla:
 - układ został rozwiązany po 17 iteracjach,
 - czas przygotowywania macierzy wyniósł ok. 1,67 s,
 - czas wykonywania iteracji wyniósł ok. 3,20 s,
 - ostateczna norma residuum wyniosła ok. $3,14 \cdot 10^{-10}$.

Na poniższym wykresie przedstawiona została norma wektora residuum w zależności od liczby wykonanych iteracji dla powyższych dwóch metod.



5. Zadanie C

Stworzona została nowa macierz pasmowa **A** o takich samych wymiarach z pięcioma niezerowymi przekątnymi:

- główną z elementami 3,
- dwoma sąsiednimi oraz dwoma skrajnymi z elementami -1 .

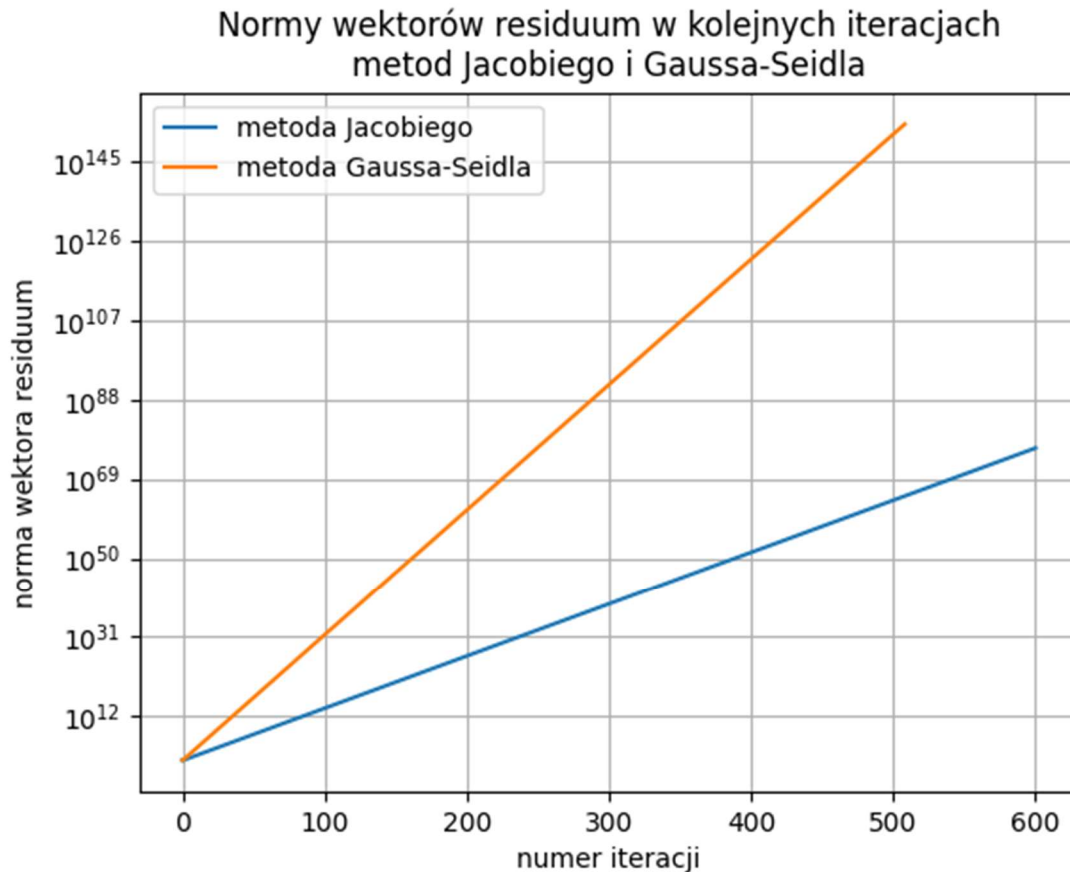
Wektor **b** pozostawiono bez zmian.

Należy zauważyć, że w przeciwieństwie do macierzy **A** z poprzedniego zadania, ta nie jest diagonalnie dominująca, a więc nie spełnia warunków zbieżności metod Jacobiego oraz Gaussa-Seidla. Spodziewanym wynikiem działania programu było zatem uzyskiwanie coraz większych wartości normy residuum. Okazało się, iż w przypadku:

- metody Jacobiego:
 - osiągnięta została maksymalna liczba 600 iteracji,
 - czas przygotowywania macierzy wyniósł ok. 1,69 s,
 - czas wykonywania iteracji wyniósł ok. 1 min 29,01 s,
 - ostateczna norma residuum wyniosła ok. $2,65 \cdot 10^{76}$;
- metody Gaussa-Seidla:
 - norma residuum przekroczyła dopuszczalny rozmiar zmiennej,
 - czas przygotowywania macierzy wyniósł ok. 1,67 s,

- o czas wykonywania iteracji wyniósł ok. 1 min 37,25 s,
- o liczba wykonanych iteracji wyniosła 509.

Norma wektora residuum w zależności od liczby wykonanych iteracji dla powyższych dwóch metod została przedstawiona na poniższym wykresie.



6. Zadanie D

Zaimplementowano opisaną we wstępie metodę faktoryzacji LU. Zastosowana została do rozwiązania układu równań z zadania C:

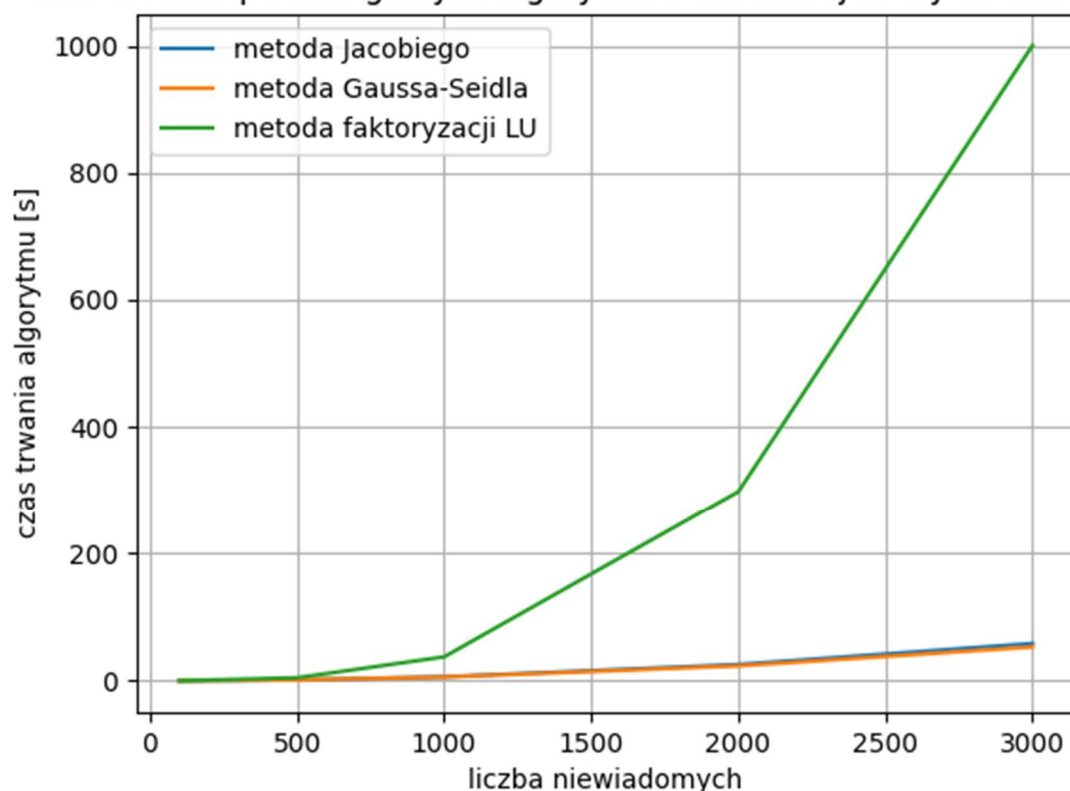
- o czas faktoryzacji macierzy wyniósł ok. 29,57 s,
- o czas wykonywania podstawień wprzód i w tył wyniósł ok. 0,08 s,
- o norma wektora residuum wyniosła ok. $2,35 \cdot 10^{-13}$.

7. Zadanie E

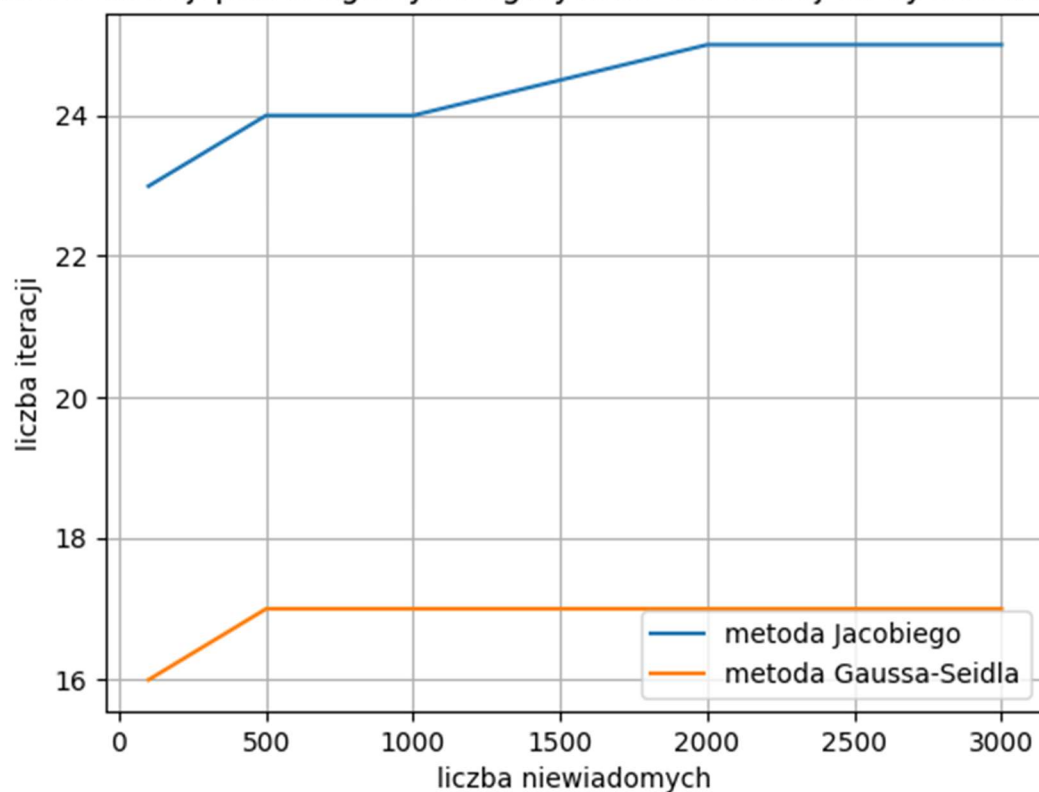
W tym punkcie porównany został czas działania wszystkich trzech algorytmów dla różnej liczby niewiadomych N , a więc dla różnych rozmiarów macierzy. Dodatkowo porównano liczbę iteracji wykonywanych przez algorytmy iteracyjne wymaganą do uzyskania dostatecznie dokładnego wyniku. Macierze **A** i **b** zbudowane zostały zgodnie z opisem podanym w zadaniu A.

Otrzymane wyniki prezentują się jak na poniższych wykresach.

Czas trwania poszczególnych algorytmów dla różnej liczby niewiadomych



Liczba iteracji poszczególnych algorytmów dla różnej liczby niewiadomych



8. Zadanie F

W przypadku macierzy **A** z zadania A, spełniającej warunek zbieżności metod Jacobiego i Gaussa-Seidla, obie metody użyte do rozwiązania układu równań w zadaniu B uzyskały w czasie kilku sekund wynik z dokładnością (w sensie normy residuum) rzędu 10^{-10} . Czas przygotowywania macierzy w obu metodach był podobny. Jednakże metodą Gaussa-Seidla w mniejszej liczbie iteracji oraz krótszym czasie otrzymano wystarczająco dobre przybliżenie rozwiązania.

W zadaniu C, kiedy macierz **A** nie spełniała warunku zbieżności badanych metod iteracyjnych, metoda Gaussa-Seidla, bardziej efektywna poprzednio, tym razem powodowała znacznie szybszy wzrost normy wektora residuum. Metoda faktoryzacji LU w zadaniu D poradziła sobie z tym układem, zatem jest bardziej uniwersalna.

Na wykresach przedstawionych w zadaniu E łatwo dostrzec, iż za uniwersalność metody faktoryzacji LU należy zapłacić gorszą złożonością obliczeniową – czas jej wykonania rośnie wraz ze wzrostem liczby niewiadomych znacząco szybciej niż czas rozwiązywania układów metodami iteracyjnymi. Porównanie tych prowadzi do obserwacji, iż metoda Jacobiego wymagała w każdym przypadku większej liczby iteracji. Wzrost liczby iteracji każdej z tych metod osobno okazał się jednak niewielki przy zwiększaniu liczby niewiadomych.