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:

,

gdzie:  
– macierz kwadratowa o rozmiarze ,  
 – szukany wektor rozwiązań,

– wektor o rozmiarze .

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

,

gdzie:  
 – macierz diagonalna zawierająca elementy z głównej przekątnej macierzy ,  
 – macierz dolna trójkątna zawierająca elementy macierzy znajdujące się poniżej głównej przekątnej pomnożone przez ,  
 – macierz górna trójkątna zawierająca elementy macierzy znajdujące się powyżej głównej przekątnej pomnożone przez .

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

,

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

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

.

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 na macierze dolnotrójkątną i górnotrójkątną , takie że

,

a następnie, metodą podstawiania wprzód, wyznaczeniu wektora z równania

,

po czym, metodą podstawiania wstecz, wyznaczeniu rozwiązania z równania

.

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:

.

Stworzona została zatem macierz pasmowa o wymiarach z pięcioma niezerowymi przekątnymi:

– główną z elementami ,  
– dwoma sąsiednimi oraz dwoma skrajnymi z elementami   
oraz wektor o długości , którego -ty element miał wartość .

4. Zadanie B

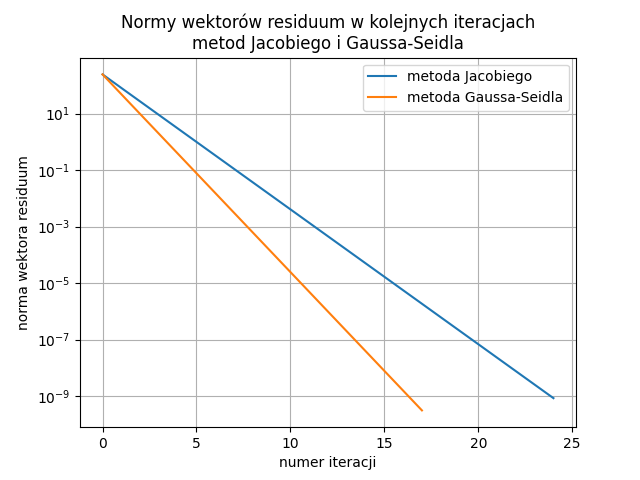
Zaimplementowane zostały opisane we wstępie metody Jacobiego i Gaussa-Seidla rozwiązywania układów równań liniowych. Przyjęto następujący warunek zakończenia iteracji:

a) norma wektora residuum jest nie większa niż 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 iteracjach,
* czas przygotowywania macierzy wyniósł ok. ,
* czas wykonywania iteracji wyniósł ok. ,
* ostateczna norma residuum wyniosła ok. ;
* metodą Gaussa-Seidla:
* układ został rozwiązany po iteracjach,
* czas przygotowywania macierzy wyniósł ok. ,
* czas wykonywania iteracji wyniósł ok. ,
* ostateczna norma residuum wyniosła ok. .

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



5. Zadanie C

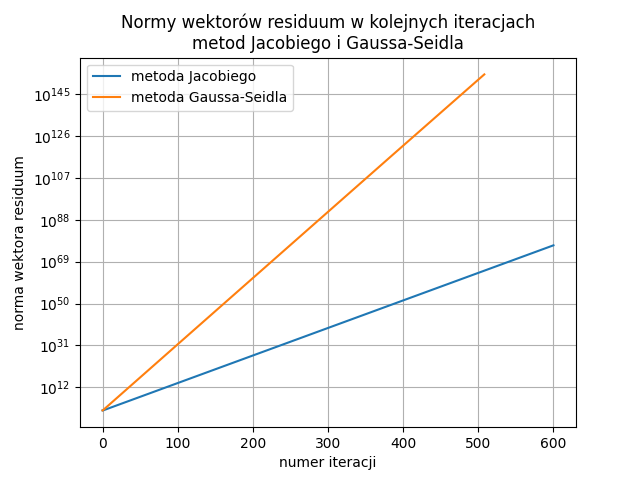
Stworzona została nowa macierz pasmowa o takich samych wymiarach z pięcioma niezerowymi przekątnymi:

– główną z elementami ,  
– dwoma sąsiednimi oraz dwoma skrajnymi z elementami .  
Wektor pozostawiono bez zmian.

Należy zauważyć, że w przeciwieństwie do macierzy 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. ,
* czas wykonywania iteracji wyniósł ok. ,
* ostateczna norma residuum wyniosła ok. ;
* metody Gaussa-Seidla:
* norma residuum przekroczyła dopuszczalny rozmiar zmiennej,
* czas przygotowywania macierzy wyniósł ok. ,
* czas wykonywania iteracji wyniósł ok. ,
* liczba wykonanych iteracji wyniosła .

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



6. Zadanie D

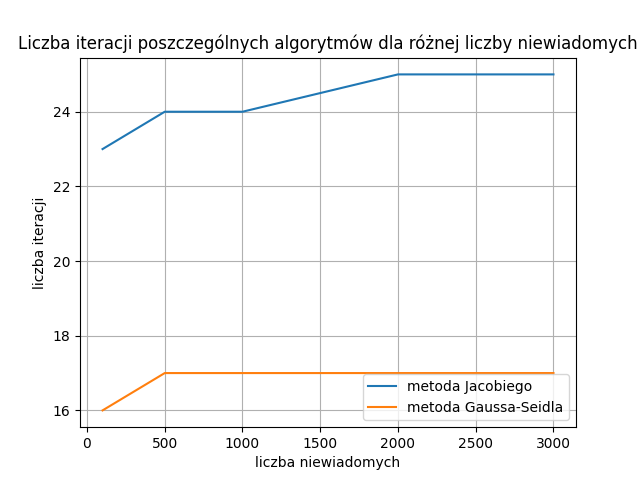
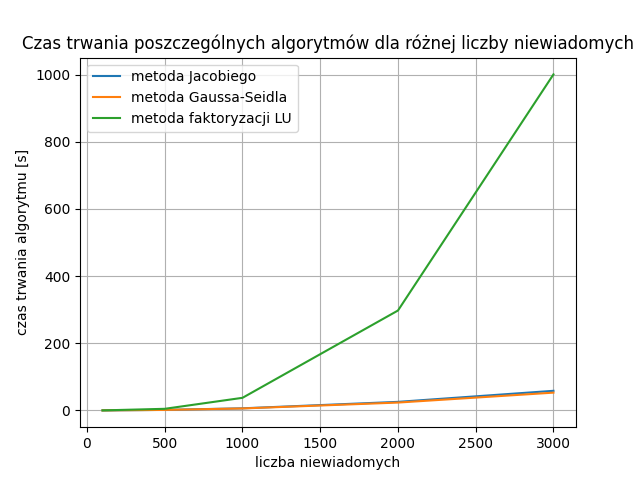
Zaimplementowano opisaną we wstępie metodę faktoryzacji LU. Zastosowana została do rozwiązania układu równań z zadania C:

* czas faktoryzacji macierzy wyniósł ok. ,
* czas wykonywania podstawień wprzód i w tył wyniósł ok. s,
* norma wektora residuum wyniosła ok. .

7. Zadanie E

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

Otrzymane wyniki prezentują się jak na poniższych wykresach.



8. Zadanie F

W przypadku macierzy 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 . 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 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.