Metody numeryczne – Projekt nr 2

Układy równań liniowych

Dar-Na

1. Wstep

Głównym celem projektu było zaimplementowanie metod iteracyjnych (Jacobiego, Gaussa-Seidla) i bezpośrednich (faktoryzacja LU) rozwiązywania układów równań liniowych. Do zrealizowania zadania wykorzystałem język Python wraz z biblioteką matplotlib do wyświetlania wykresu, time do mierzenia czasu obliczania oraz math do wykonania matematycznych obliczeń (sin).

2. Analiza zadania

• <u>Zadanie A</u> – Dla numeru indeksu 187726 otrzymujemy wartości: a1 = 12, a2 = a3 = -1 dla macierzy A oraz jej wielkość N = 926. Wektor b również ma długość N = 926, natomiast n–ty element ma wartość sin(n ·(7 + 1)).

• <u>Zadanie B</u> – Dla układu równań z podpunktu A wyniki dla metody Jacobiego i Gaussa–Seidla przedstawiają się następująco:

```
Jacobi method
time: 4.488175392150879
iterations: 15

Gauss-Seild method
time: 3.06767201423645
iterations: 11
```

Iteracja była przeprowadzana, dopóki norma z wektora residuum była większa niż 10^{-9}. Możemy zauważyć, że metoda Gaussa-Seidla potrzebuje mniejszej liczby iteracji oraz około 1.5 razy szybciej wykonuje obliczenia niż metoda Jacobiego.

• <u>Zadanie C</u> – Tworzymy kolejny układ równań, dla którego a1 = 3, a2 = a3 = −1 i N = 926, natomiast wektor b pozostaje bez zmian. Otrzymujemy następujący rezultat:

```
Traceback (most recent call last):
  File "D:\PG\Sem 4\MN\Project\Uklady rownan liniowych\main.py", line 25, in <module>
    timeJacobi = jacobi(matrixC, vectorB)
  File "D:\PG\Sem 4\MN\Project\Uklady rownan liniowych\iterativeMethods.py", line 25, in jacobi
    if norm(res) < pow(10, -9):
  File "D:\PG\Sem 4\MN\Project\Uklady rownan liniowych\vector_methods.py", line 46, in norm
    count += el ** 2
OverflowError: (34, 'Result too large')
```

```
Traceback (most recent call last):
  File "D:\PG\Sem 4\MN\Project\Uklady rownan liniowych\main.py", line 26, in <module>
    timeGauss = gaussaSeilda(matrixC, vectorB)
  File "D:\PG\Sem 4\MN\Project\Uklady rownan liniowych\iterativeMethods.py", line 60, in gaussaSeilda
    if norm(res) < pow(10, -9):
  File "D:\PG\Sem 4\MN\Project\Uklady rownan liniowych\vector_methods.py", line 46, in norm
    count += el ** 2
OverflowError: (34, 'Result too large')
```

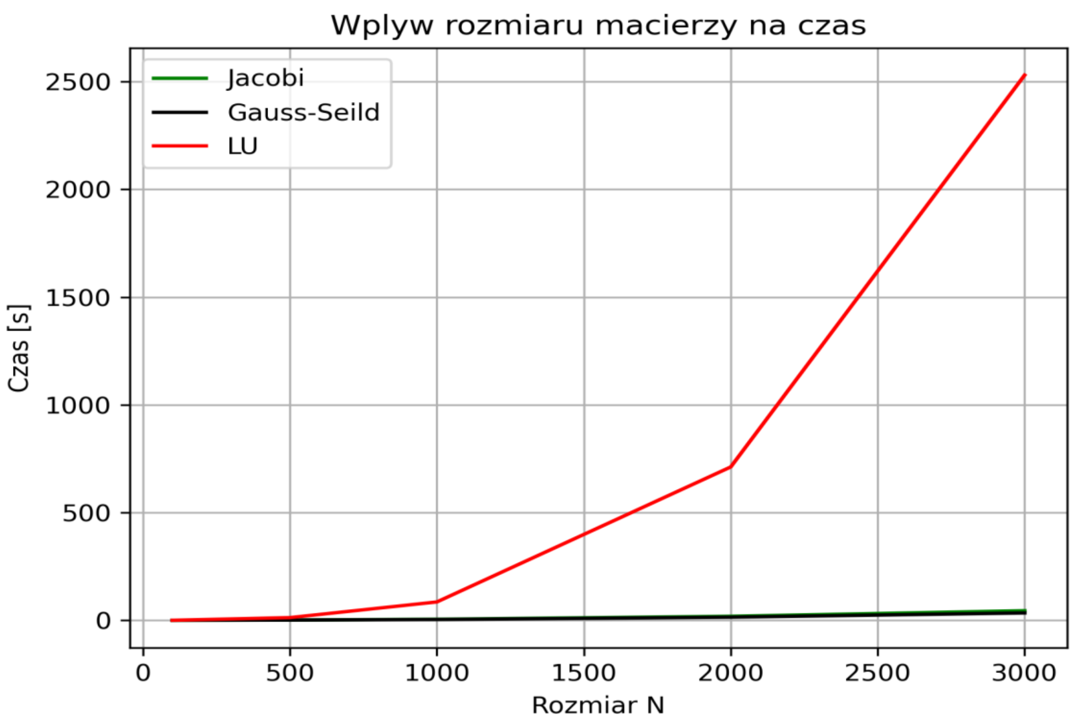
Podczas liczenia normy z wektora residuum za pomocą powyższych metod iteracyjncych kończyły się pythonowym błędem OverflowError - Result too large (po 1266 iteracjach metodą Jacobiego i 522 metodą Gaussa-Seidla). Można na podstawie tego wywnioskować, że dla powyższych wartości metody iteracyjne nie zbiegają się.

• <u>Zadanie D</u> – Po zaimplementowaniu metody bezpośredniej rozwiązania układów równań liniowych: metodę faktoryzacji LU, stosujemy ją dla przypadku C. Oto rezultat:

```
LU method
time: 0.08178138732910156
Residuum norm: 7.282698899171515e-16
```

Wynik jest bliski 0, zatem metoda charakteryzuje się wysoką dokładnością obliczeń.

• <u>Zadanie E</u> – W ramach tego zadania utworzono wykres zależności czasu od liczby niewiadomych dla powyższych metod. Zależność została przedstawiona na poniższym wykresie:



Dla każdego z algorytmów czas trwania rośnie wraz ze wzrostem rozmiaru macierzy, czyli zwiększaniem liczby niewiadomych N. Metoda Gaussa-Seidla jest szybsza niż metoda Jacobiego dla N większych niż ok.1500. Natomiast możemy zauważyć, że metoda faktoryzacji LU jest znacznie wolniejsza od metod iteracyjnych.

• Zadanie F – Czas wykonywania obliczeń dla każdej z powyższych metod wzrasta wraz ze wzrostem liczby niewiadomych. Metoda Gaussa-Seidla oraz Jacobiego są znacznie szybsze od metody faktoryzacji LU, zwłaszcza dla większych N(przy 3000 niewiadomych czas wykonywania wynosi więcej niż pół godziny). Jednak nie wszystkie układy równań można rozwiązać poprzez metody iteracyjne, co miało miejsce w zadaniu C. Metoda faktoryzacji LU pomimo swojego długiego czasu działania, znajduje rozwiązanie.
Z tego można wnioskować że metody iteracyjne nie są tak dokładne, ale wykonują się szybciej. Znajdują się jednak przypadki, gdzie metody te się nie zbiegają. Wtedy pomimo dłuższego czasu działania należy zastosować metodą bezpośrednią.