Projekt nr.2 - Metody Numeryczne

Martin Kuczyński

12 kwietnia 2021

1 Wstęp

1.1 Opis problemu

Celem projektu było zaimplementowanie trzech metod rozwiązywania układów równań liniowych.

- Metoda LU
- Metoda Jacobiego
- Metoda Gaussa-Seidla

Każda z tych metod została przetsetowana pod względem dokładności rozwiązań oraz czasu ich wykonania. Taki zabieg pokazuje dla jakich wartości warto używać każdej z tych metod, co w przyszłościowym procesie implementacji innych programów może pomóc programiście, matematykowi czy też komukolwiek innemu zainteresowanemu dokonać poprawnego wyboru metody rozwiązywania takich równań.

1.2 Narzędzia

Projekt został zaimplementowany w języku **python** z pomocą zewnętrznej bilblioteki **numpy**. Jest ona niezbędna, ze względu na wbudowane metody tworzenia macierzy oraz operacji na nich, takich jak mnożenie, norma oraz wiele innych.

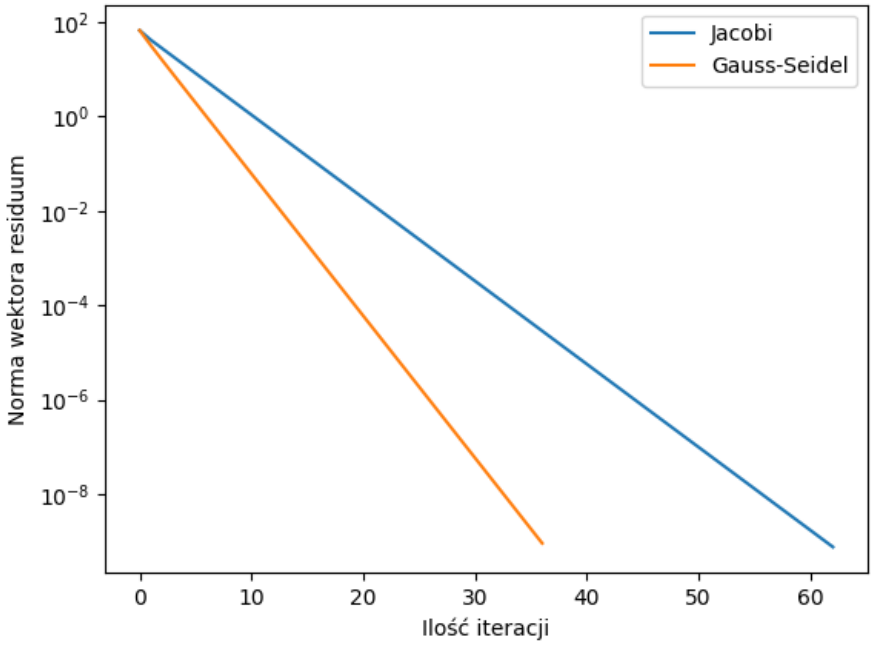
2 Obserwacje

2.1 Zadanie B

W zadaniu B oba algorytmy zwracają wynik zbliżony do teoretycznego, a norma wektora residuum jest mniejsza niż zadane 10^{-9}.

```
Task B: Starting Jacobi method computing
Jacobi method did 62 iterations and residuum =7.54029935135512e-10
Jacobi took 0.6517260074615479 seconds
Task B: Starting Gauss_Seidel method computing
Gauss_Seidl method did 36 iterations and residuum = 9.041493441701794e-10
Gauss_Seidl took 0.3789184093475342 seconds
```

Algorytm Jacobiego wykonał zadanie w **62 iteracji**, które zajęły ok. **0.65s**, a zmiana normy wektora residuum w zależności od iteracji prezentuje się następująco.
Z kolei algorytm Gaussa-Seidla wykonał zadanie w **36 iteracji**, które zajęły ok.**0,37s**, a zmiana normy wektora residuum w zależności od iteracji dla tej metody wygląda następująco.

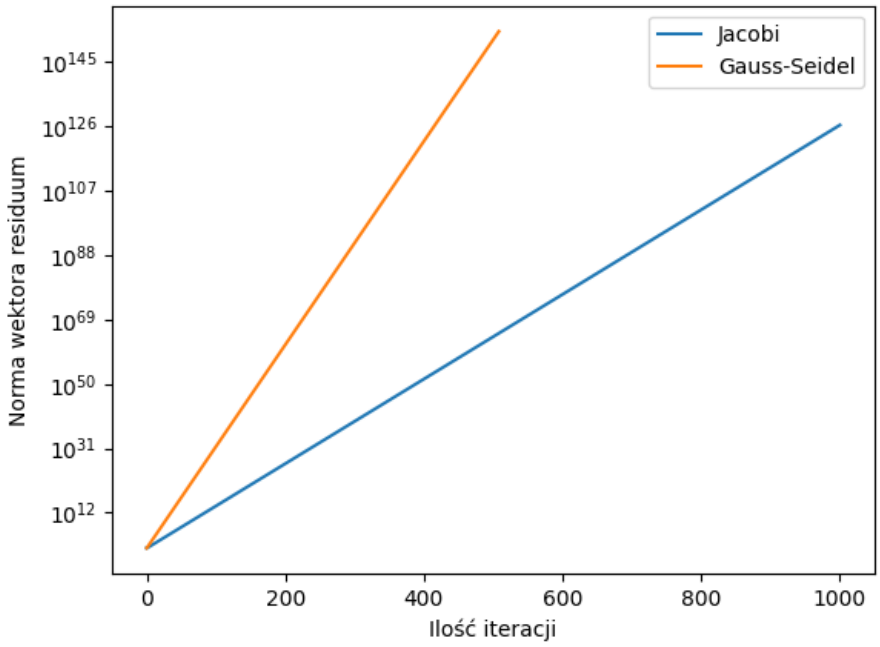


Jak widać metoda Gaussa-Seidla wykonuje się około 2 razy szybciej, co prawdopodobnie wynika z tego, że do obliczania kolejnych rozwiązań korzystamy nie tylko z rozwiązania w poprzedniej iteracji, ale także z aktualnie wykonwyanej iteracji.

2.2 Zadanie C

W programie zostało zaimplementowane założenie, że w przypadku metod iteracyjnych dojście do progu 1000 iteracji powinno zakończyć wywoływanie tejże metody. Na dodatek w momencie osiągnięcia normy wektora residuum rzędu **inf** powoduje także wyjście z metody.

```
Task C ------------------------------
Task C: Starting Jacobi method computing
Jacobi method did 1000 iterations and ended with residuum =2.587140432086766e+126
Jacobi took 10.44653034210205 seconds
Task C: Starting Gauss_Seidel method computing
Gauss_Seidl method did 1000 iterations and ended with residuum = inf
Gauss_Seidl took 10.294887781143188 seconds
```



W przypadku macierzy, której wartości na głównej diagonali wynoszą 3, oba algorytmy zawiodły zwracając w/w wartości.
Jak wiadomo zarówno metoda Gaussa-Seidla oraz Jacobiego są metodami zwracającymi sukces przy pewnych założeniach co do macierzy **A**. Problem związany z nierozwiązaniem układów równań z zadania C jest spowodowany zmianą wartości elementów na głównej diagonali tejże macierzy. Wydaje mi się, że niepowodzenie zaimplementowanych metod iteracyjnych jest stricte związane z faktem, że macierz **A** z podpunktu C nie jest diagonalnie dominująca, a przynajmniej jest mniej dominująca niż ta z podpunktu B.

2.3 Zadanie D

Algorytm LU nie jest algorytmem iteracyjnym. Jest to metoda faktoryzacji macierzy **A** na macierze L oraz U, a następnie rozwiązania dwóch układów równań za pomocą podstawiania wprzód oraz wstecz. W taki też sposób metoda LU została zaimplementowana.

```
Task D -----------------------------
Task D: Starting LU method computing
LU method ended with residuum = 6.620801338097519e-13
LU took 2.6351194381713867 seconds
```

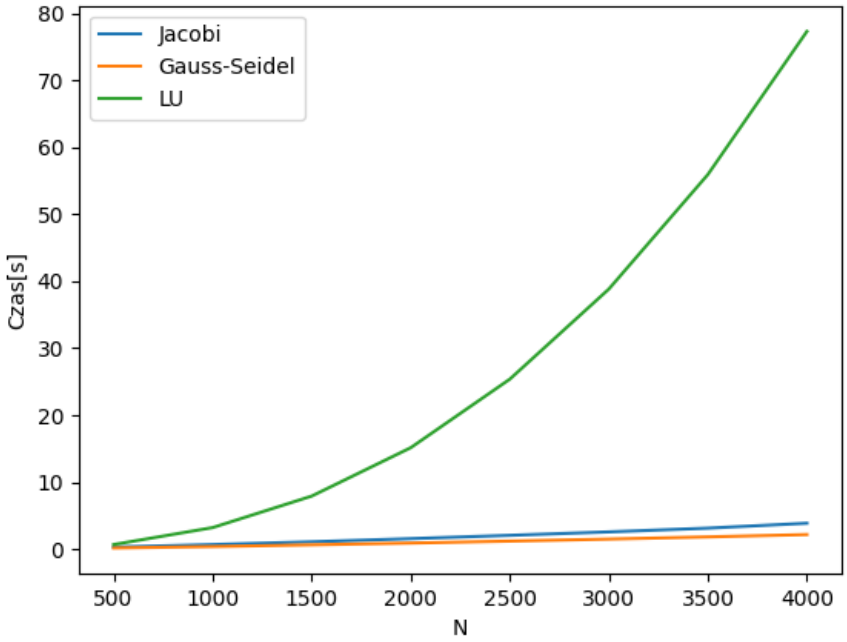
Dla układu równań liniowych z podpunktu C(dla którego metody iteracyjne nie działały) norma residuum rozwiązania jest rzędu 10^{-13}. Czas wykonania metody LU dla tak dobrego wyniku wynosi ok. **2.63s**, co jest o wiele większym wymiarem czasowym w porównaniu do metod iteracyjnych.

2.4 Zadanie E

Porównanie użyteczności trzech metod rozwiązywania układów równań liniowych wymaga sprawdzenia ich czasu wykonywania względem pewnej wielkości danych. Zakładamy także, że metody iteracyjne rozwiążą układ poprawnie, tj. norma wektora residuum będzie mniejsza niż 10^{-9}. W tym celu został przygotwany zbiór testowy macierzy z podpunktu A o rozmiarach $N = [500, 1000, 1500, ..., 4000]$

```
Task E: Jacobi --------
Task E: Starting Jacobi method for 500
Task E: Finished Jacobi method for 500 with time 0.30232787132263184
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 500
Task E: Finished Gauss-Seidel method for 500 with time 0.17303895950317383
Task E: LU --------
Task E: Starting LU method for 500
Task E: Finished LU method for 500 with time 0.5602588653564453
Task E: Jacobi --------
Task E: Starting Jacobi method for 1000
Task E: Finished Jacobi method for 1000 with time 0.6564128398895264
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 1000
Task E: Finished Gauss-Seidel method for 1000 with time 0.37110304832458496
Task E: LU --------
Task E: Starting LU method for 1000
Task E: Finished LU method for 1000 with time 2.5996594429016113
Task E: Jacobi --------
Task E: Starting Jacobi method for 1500
Task E: Finished Jacobi method for 1500 with time 1.016819953918457
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 1500
Task E: Finished Gauss-Seidel method for 1500 with time 0.6123034954071045
Task E: LU --------
Task E: Starting LU method for 1500
Task E: Finished LU method for 1500 with time 6.488328218460083
Task E: Jacobi --------
Task E: Starting Jacobi method for 2000
Task E: Finished Jacobi method for 2000 with time 1.461449146270752
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 2000
Task E: Finished Gauss-Seidel method for 2000 with time 0.8792409896850586
Task E: LU --------
Task E: Starting LU method for 2000
Task E: Finished LU method for 2000 with time 12.422227144241333
```

```
Task E: Jacobi --------
Task E: Starting Jacobi method for 2500
Task E: Finished Jacobi method for 2500 with time 1.9088196754455566
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 2500
Task E: Finished Gauss-Seidel method for 2500 with time 1.12353515625
Task E: LU --------
Task E: Starting LU method for 2500
Task E: Finished LU method for 2500 with time 21.12784242630005
Task E: Jacobi --------
Task E: Starting Jacobi method for 3000
Task E: Finished Jacobi method for 3000 with time 2.373073101043701
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 3000
Task E: Finished Gauss-Seidel method for 3000 with time 1.3844635486602783
Task E: LU --------
Task E: Starting LU method for 3000
Task E: Finished LU method for 3000 with time 33.06022763252258
Task E: Jacobi --------
Task E: Starting Jacobi method for 3500
Task E: Finished Jacobi method for 3500 with time 2.9182701110839844
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 3500
Task E: Finished Gauss-Seidel method for 3500 with time 1.7030761241912842
Task E: LU --------
Task E: Starting LU method for 3500
Task E: Finished LU method for 3500 with time 48.07080698013306
Task E: Jacobi --------
Task E: Starting Jacobi method for 4000
Task E: Finished Jacobi method for 4000 with time 3.618114948272705
Task E: Gauss-Seidel --------
Task E: Starting Gauss-Seidel method for 4000
Task E: Finished Gauss-Seidel method for 4000 with time 2.0582334995269775
Task E: LU --------
Task E: Starting LU method for 4000
Task E: Finished LU method for 4000 with time 66.76947498321533
```



Widzimy, zę metody iteracyjne dla wysokich wielkości macierzy są bardzo opłacalne, ponieważ dają wyniki zadowalające w czasie niebotycznie mniejszym niż metoda LU. Z kolei przy niewielkich wartościach N warto zastanowić się nad priorytetem obliczeń. Mianowicie metody iteracyjne nie oddają tak dokładnych wyników jak metoda LU, a czas jaki potencjalnie musimy dołożyc, aby rozwiązać układ równań tą metodą jest stosunkowo niewielki. Zatem czasami bardziej opłaca się zastosować metodę nieiteracyjną, aczkolwiek im większe dane posiadamy, tym bardziej opłaca się skorzystać z metody Jacobiego czy też Gaussa-Seidla. Warto także dodać że wielkim plusem metody LU jest to, że niezależnie od właściwości macierzy $\mathbf{A}$ wynik uzyskany za jej pomocą będzie na pewno prawidłowy z dosyć dużą dokładnością.