Projekt 2 – Układy równań liniowych

Emilia Pieśnikowska

May 16, 2022

1 Wprowadzenie

Celem projektu jest zaimplementowanie rozwiazywania równań liniowych za pomoca wybranych algorytmów. Metody użyte do rozwiazywania tych równań to metody iteracyjne Jacobiego i Gaussa-Seidla oraz bezpośrednia Gaussa. Do implementacji wykorzystałam jezyk Python.

2 Opisy oraz obserwacje poszczegółych zadań

2.1 Zadanie A

Dla numeru indeksu 180112 zmienne wynosiły a1 = 6, N = 912. Macierz A wygladała nastepujaco

$$\mathrm{A} = \begin{bmatrix} 6 & -1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 \\ -1 & 6 & -1 & -1 & 0 & \dots & 0 & 0 & 0 \\ -1 & -1 & 6 & -1 & -1 & \dots & 0 & 0 & 0 \\ 0 & -1 & -1 & 6 & -1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 6 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & 6 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & -1 & 6 \end{bmatrix}$$

Wektor b = sin(n*(f+1)), co dla $f = 0$ daje nastepujacy wektor:

$$\mathbf{b} = \begin{bmatrix} 0.0 & 0.842 & 0.909 & 0.141 & \dots & -0.882 & -0.873 & -0.062 \end{bmatrix}$$

2.2 Zadanie B

W zadaniu B zaimplementowane zostały dwie metody rozwiazywania równań - metoda Jacobiego oraz Gaussa-Seidla. Wyniki zliczeń iteracji oraz czasu trwania algorytmu przy rozwiazaniach równań o podanych parametrach wygladaja nastepujaco:

```
Jacobi
number of iterations:  43
time: 10.994608163833618
Gauss-Seidel
number of iterations:  36
time: 9.940430879592896
```

Możemy zauważyć, że metoda Gaussa-Seidla potrzebuje mniejszej liczby iteracji oraz odrobine szybciej wykonuje obliczenia niż metoda Jacobiego.

2.3 Zadanie C

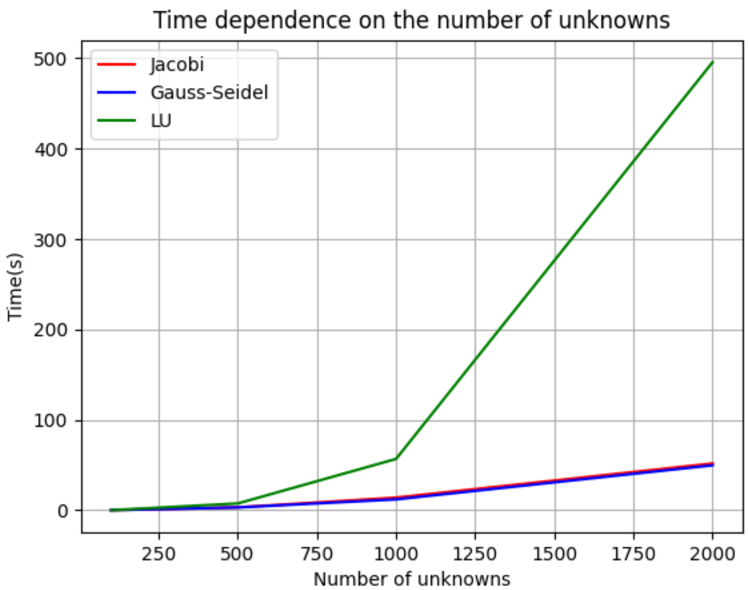
Zadanie polegało na utworzeniu układu o zmienionej wartości a1 na 3, i przetestowanie obu metod iteracyjnych z pozostawieniem tego samego wektora b. Program jednak wyrzucał bład przepełnienia OverflowError, z czego wnioskuje, że dla wektora A o zmienionym współczynniku a1 metody iteracyjne nie zbiegaja sie.

2.4 Zadanie D

W tym zadaniu rozwiazano powyższy układ równań za pomoca metody bezpośredniej: faktoryzacji LU. Po zaimplementowaniu metody bezpośredniej rozwiazania układów równań liniowych metode faktoryzacji LU, stosujemy ja dla przypadku C. W tym przypadku norma z residuum wyniosła około $2.73 * 10^{-12}$. Jest to wynik bliski 0, co oznacza wysoka dokładność wykonanych obliczeń.

2.5 Zadanie E

Stworzyłam wykres zależności czasu trwania poszczególnych algorytmów od liczby niewiadomych N = 100, 500, 1000, 2000(niestety możliwości komputera nie pozwolily mi na wiecej niż 2000) dla przypadku z punktu A.



Czas wykonywania algorytmów w zależności od liczby nie wiadomych rośnie dla każdego z algorytmów, jednak widać, że dla metody LU dzieje sie to szybciej.

3 Zadanie F - wnioski

Mimo znacznie szybszego wzrostu czasu trwania w przypadku metody faktoryzacji, sprawdzi sie ona w wiekszej ilości przypadków. Wyciagajac wnioski z zadania C możemy zauważyć, że metody iteracyjne, mimo tego że sa szybsze, nie wykonuja sie tak dokładnie jak metoda faktoryzacji LU.Metoda faktoryzacji LU pomimo swojego długiego czasu działania, znajdzie rozwiazanie. Dlatego w przypadkach gdzie metody iteracyjne sie nie zbiegaja, warto zastosować metode bezpośrednia.