Porównanie metod wyznaczania rozwiązań układów równań liniowych

Jan Kaczerski 193237

---

1. Wprowadzanie

Celem projektu jest implementacja oraz porównanie trzech metod rozwiązywania równań liniowych, dwóch iteracyjnych (Jacobiego i Gaussa-Seidela) oraz jednej bezpośredniej (faktoryzacja LU). Dla uproszczenia zaimplementowane przeze mnie algorytmy działają na macierzach w formacie pełnym, co zwiększa czas wykonywania obliczeń oraz zapotrzebowanie na pamięć w porównaniu do macierzy w formacie rzadkim.

W badanych przeze mnie układach równań

Ax = b

Macierze miały następujące postaci:

- **A** – macierz pasmowa zgodna ze schematem (1) o rozmiarze NxN, N=937
  - $\mathbf{A_1}$: a1 = 7, a2 = a3 = -1
  - $\mathbf{A_2}$: a1 = 3, a2 = a3 = -1
- **b** – wektor o długości N=937, którego n-ty element ma wartość sin(4*n)
- **x** – wektor szukanych rozwiązań o długości N=937

$$\mathbf{A} = \begin{bmatrix} a1 & a2 & a3 & 0 & 0 & 0 & 0 & \dots & 0 \\ a2 & a1 & a2 & a3 & 0 & 0 & 0 & \dots & 0 \\ a3 & a2 & a1 & a2 & a3 & 0 & 0 & \dots & 0 \\ 0 & a3 & a2 & a1 & a2 & a3 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & a3 & a2 & a1 \end{bmatrix} \tag{1}$$

2. Podstawy teoretyczne

Metody bezpośrednie pozwalają na uzyskanie wyniku po określonej liczbie operacji, natomiast iteracyjne polegają na wyznaczaniu kolejnych przybliżeń rozwiązań, aż do uzyskania akceptowalnego błędu. Metody iteracyjne są szybsze, jednak nie zawsze kolejne przybliżenia będą zbiegać się do zera, a więc nie zawsze otrzymany wynik będzie poprawny.

1. Pierwszą badaną przeze mnie metodą jest **faktoryzacja LU**, polegająca na rozłożeniu macierzy A na dwie macierze trójkątne L i U, takie, że:

   A = LU

   Równanie to podstawiamy do równania macierzowego i tworzymy wektor pomocniczy y:

   LUx = b
   y = Ux
   Ly = b
   Ux = y

   Korzystając z faktu, że macierze L i U są trójkątne, wektory y oraz x znajdujemy za pomocą podstawienia w tył i podstawienia w przód.

2. Drugą metodą jest **metoda iteracyjna Jacobiego**, polegająca na obliczeniu k-tego przybliżenia i-tego elementu wektora x na podstawie poprzedniego przybliżenia, przyjmując początkowe wartości wektora x równe 1. I-ty wyraz wektora w (k+1)-szym przybliżeniu określany jest jako:

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)})/a_{ii}$$

b_i – i-ty wyraz wektora b
a_{ij} – wyraz w i-tym wierszu i j-tej kolumnie macierzy a

3. Ostatnią badaną przez mnie metodą jest **metoda iteracyjna Gaussa-Seidela**. Różni się ona od metody Jacobiego tym, że do wyliczenia k-tego przybliżenia używa aktualnych wartości wektora x, zamiast poprzednich, co pozwala szybciej wykorzystać nowo obliczone wartości. I-ty wyraz wektora w (k+1)-szym przybliżeniu określany jest jako:

$$x_i^{(k+1)} = (b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)})/a_{ii}$$

b_i – i-ty wyraz wektora b
a_{ij} – wyraz w i-tym wierszu i j-tej kolumnie macierzy a

4. Poprawność rozwiązań weryfikowana była poprzez obliczenie **błędu rezydualnego** równania macierzowego, a następnie obliczeniu jego **normy Frobeniusa**:

$$r = Ax - b$$

$$\|\mathbf{r}\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |\mathbf{r}_{ij}|^2 \right)^{1/2}$$

3. Implementacja

Algorytmy zaimplementowane zostały w języku C++ z użyciem bibliotek ctime (do mierzenia czasu wykonywania algorytmów) oraz cmath (do wykonywania obliczeń). Wynik równań uznawany był za poprawny, jeżeli norma wektora rezydualnego nie przekraczała 1E-9. Maksymalna ilość iteracji metod iteracyjnych ustalona została na 100, natomiast metoda faktoryzacji LU nie miała czasowego warunku stopu.

Do wykonania wykresów użyłem biblioteki open-source matplotlibcpp.

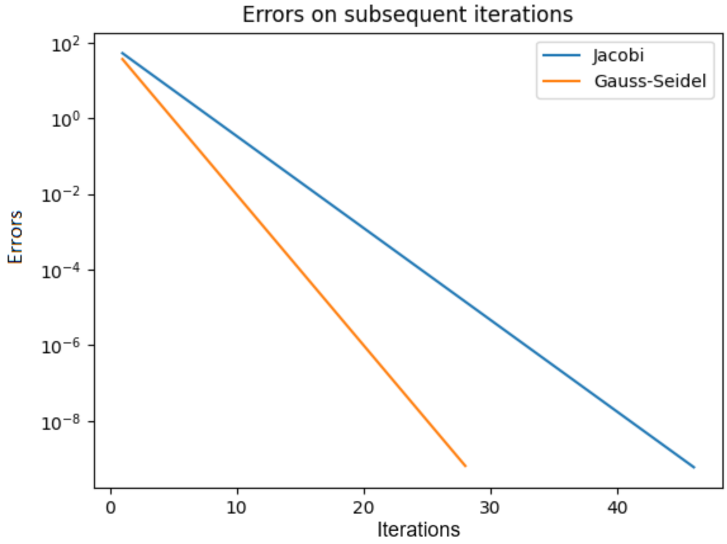
4. Analiza zadań projektowych

- **Zadanie A**
  Dla numeru indeksu 193237, macierz A_1 uzupełniona została wartościami a1 = 5 + 2, a2 = -1, a3 = -1. Rozmiar macierzy oraz długości wektorów x i b N = 937. N-ty wyraz wektora b ma postać sin(n * (3+1)).

- **Zadanie B**
  Dla układu równań $A_1x=b$ obliczonego za pomocą metod iteracyjnych otrzymane rezultaty prezentują się następująco:

Metoda	Iteracje	Czas	Norma błędu
Jacobi	46	0,229s	5,99793E-10
Gauss-Seidel	28	0,163s	6,50992E-10

Natomiast wartości normy błędu rezydualnego dla kolejnych iteracji obu metod przedstawia poniższy wykres (y w skali logarytmicznej):



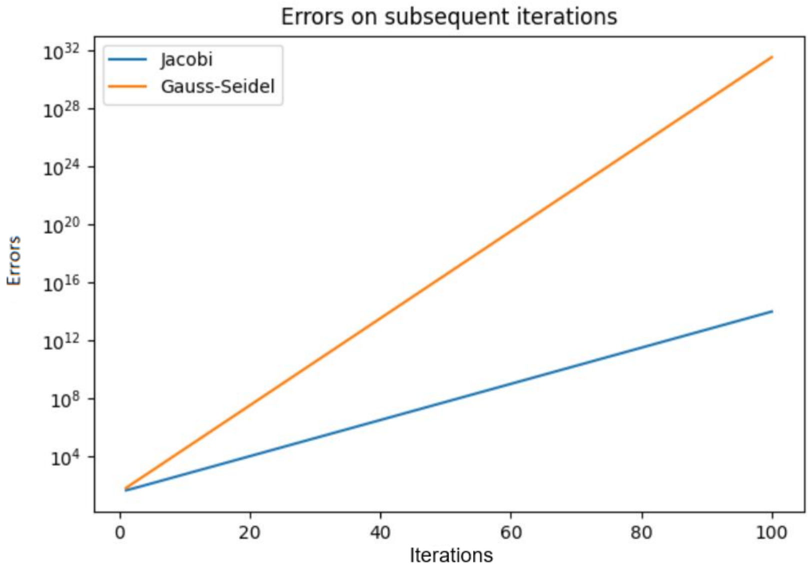
Jak można zobaczyć na powyższym wykresie i w powyższej tabeli, metoda Gaussa-Seidela jest szybsza o około półtora razy od metody Jacobiego. Jest to spowodowane wykorzystaniem przez szybszą metodę nowo obliczonych przybliżeń jeszcze w tej samej iteracji.

- **Zadanie C**

  Dla układu równań $A_2x=b$ rezultaty metod iteracyjnych prezentują się następująco:

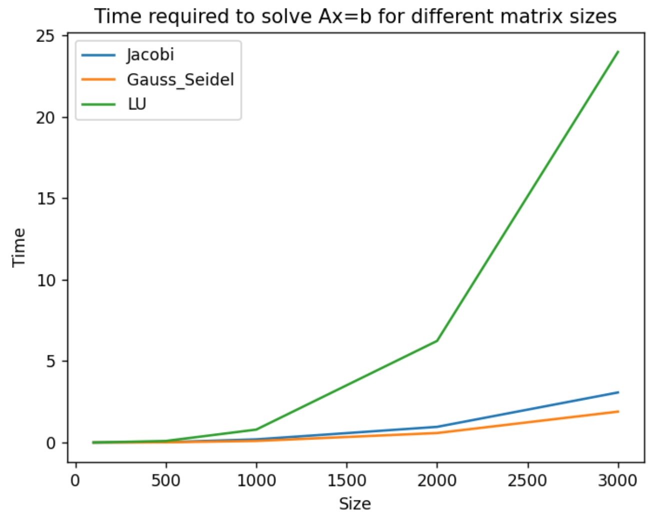
Metoda	Iteracje	Norma błędu
Jacobi	100	1,3648E+13
Gauss-Seidel	100	3,49051E+31

Przy wykorzystaniu macierzy A_2, normy błędów rezydualnych osiągają ogromne wartości, a algorytmy kończą pracę poprzez osiągnięcie limitu maksymalnych iteracji. Jest to przypadek, w którym kolejne przybliżenia wartości wektora x tak naprawdę oddalają go od oczekiwanego rozwiązania – norma nie dąży do zera, a więc metody iteracyjne nie zbiegają się. Zależność tą prezentuje poniższy wykres wartości norm błędów rezydualnych w kolejnych iteracjach (oś y w skali logarytmicznej).



- **Zadanie D**
  Norma błędu rezydualnego rozwiązania $A_2x=b$ metodą bezpośrednią (faktoryzacja LU) wyniosła 6,37641E-13. Jest to wartość na tyle bliska zeru, że rozwiązanie uznane zostało za poprawne, co świadczy o przewadze metod bezpośrednich nad iteracyjnymi pod względem pewności otrzymania prawidłowego wyniku.

- **Zadanie E**
  Wszystkie badane metody zostały przetestowane i porównane dla różnych rozmiarów macierzy w układzie równań. Macierze miały wartości identyczne jak w punkcie Zadanie A, lecz inne rozmiary. Wszystkie algorytmy uzyskały poprawne wyniki dla każdego równania, a zmierzone czasy pracy prezentuje poniższy wykres wraz z tabelą.



Rozmiar N	Gauss-Seidel	Jacobi	Faktoryzacja LU
100	~0 s	~0 s	0,002 s
500	0,013 s	0,021 s	0,095 s
1000	0,098 s	0,193 s	0,794 s
2000	0,583 s	0,962 s	6,239 s
3000	1,897 s	3,074 s	23,985 s

  Testy wykazały, że niezależnie od rozmiaru macierzy, metody można uszeregować od najszybszej do najwolniejszej: Gauss-Seidel, Jacobi, Faktoryzacja LU. Im większy jest rozmiar macierzy, tym różnica czasów wykonywania algorytmów jest coraz większa - dla największej zbadanej macierzy (N = 3000), faktoryzacja LU zajęła 8 razy więcej czasu niż metoda Gaussa-Seidela.

- **Zadanie F (Podsumowanie)**
  Przebadane przeze mnie metody iteracyjne okazały się szybsze niż metoda bezpośrednia, a wraz ze wzrostem rozmiaru macierzy, otrzymane przyspieszenie rośnie. Metoda Gaussa-Seidela jest najszybsza, a metoda faktoryzacji LU najwolniejsza, jednak znajduje ona swoje zastosowanie. Metody iteracyjne nie zawsze zbiegają się i są w stanie odnaleźć właściwe rozwiązanie, natomiast rozwiązania uzyskane przez metody bezpośrednie zawsze będzie prawidłowe (punkty C i D).
  W przypadku gdy wiemy, że metody iteracyjne się zbiegną, najlepszą z badanych metod będzie metoda Gaussa-Seidela (najszybsza), a w przeciwnym wypadku, faktoryzacja LU (jedyna dająca poprawną odpowiedź).