

UKŁADY RÓWNAŃ LINIOWYCH

Spis treści

1. Wstęp	1
1.1. Używane metody	1
1.1.1. Metoda Jacobiego	1
1.1.2. Metoda Gaussa-Seidla	1
1.1.3. Metoda LU	2
2. Badane układy równań	2
2.1. Główna przekątna wypełniona ósemkami	2
2.2. Główna przekątna wypełniona trójkami	3
3. Porównanie wydajności metod	4
4. Wnioski	4

1. Wstęp

Celem niniejszego sprawozdania jest przedstawienie wyników analizy metod rozwiązywania układów równań liniowych. Są to metody iteracyjne - Jacobiego oraz Gaussa-Seidla oraz metoda bezpośrednia faktoryzacji LU.

Kod napisany w celu realizacji badań jest niskopoziomowy (wykorzystano język C++) i nie używa żadnych zewnętrznych bibliotek do realizacji obliczeń. Przechowywanie macierzy odbywa się za pomocą specjalnie stworzonej klasy `Matrix`, będącej niejako nakładką na tablicę dynamiczną, ułatwiającą realizację wszelkich operacji macierzowych. Podstawowe operacje matematyczne, takie jak pierwiastek z liczby, są wykonywane za pomocą wbudowanej biblioteki `math.h`.

1.1. Używane metody

1.1.1. Metoda Jacobiego

Metoda Jacobiego jest znaną i prostą w implementacji metodą rozwiązywania układów równań liniowych. Jej celem jest uzyskanie odpowiednio dokładnego przybliżenia wektora x , będącego rozwiązaniem równania. Jest on początkowo inicjalizowany zerami, a następnie aktualizowany w każdej iteracji zgodnie ze wzorem:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j \neq i} a_{i,j} * x_j^{(k)} \right), i = 1, 2, \dots, n$$

A jest tzw. macierzą systemową reprezentującą środowisko problemu, np. obwód elektryczny czy geometrię pomieszczenia, natomiast b jest wektorem pobudzenia (impuls elektryczny, fala akustyczna itp.).

1.1.2. Metoda Gaussa-Seidla

Metoda Gaussa-Seidla, również jest metodą iteracyjną, co więcej jest ona podobna to opisanej powyżej metody Jacobiego. Główną różnicą jest wykorzystywanie we wzorze na przybliżenie wektora x najnowszych danych (z aktualnie trwającej iteracji), co pozwala na skrócenie czasu obliczeń. Wygląda on następująco:

$$x_i^{(k+1)} = \frac{1}{a_{i,i}} \left(b_i - \sum_{j=1}^{i-1} a_{i,j} * x_j^{(k+1)} - \sum_{j=i+1}^n a_{i,j} * x_j^{(k)} \right), i = 1, 2, \dots, n$$

1.1.3. Metoda LU

Ostatnią z badanych metod jest bezpośrednia metoda faktoryzacji LU (z ang. *Lower Upper*).

W metodzie tej wykorzystuje się macierze trójkątne - dolną i górną, takie że $LUx = b$. Następnie, należy rozwiązać dwa układy równań $Ly = b$ oraz $Ux = y$, odpowiednio za pomocą podstawienia wprzód i wstecz.

2. Badane układy równań

W obydwu badanych układach, macierz A jest macierzą pasmową kwadratową o wymiarach 957 na 957, z kolei wektor b jest wypełniony wartościami według wzoru $b_N = \sin(4N)$.

2.1. Główna przekątna wypełniona ósemkami

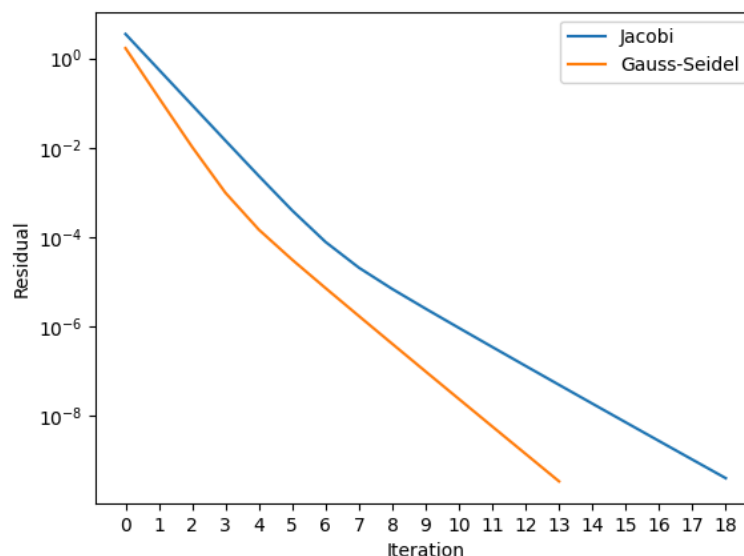
Układ równań prezentuje się następująco:

$$\begin{pmatrix} 8 & -1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -1 & 8 & -1 & -1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 8 & -1 & -1 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 8 & -1 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 8 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 8 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & 8 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & -1 & 8 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & -1 & -1 & 8 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & -1 & 8 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \\ x_{n-4} \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} -0.757 \\ 0.989 \\ -0.537 \\ -0.288 \\ 0.913 \\ \vdots \\ -0.948 \\ 0.860 \\ -0.176 \\ -0.630 \\ 1.000 \end{pmatrix}$$

Wszystkie metody zbiegły się. Poniższa tabela prezentuje normy residuum i czasy wykonania dla poszczególnych metod oraz w przypadku metod iteracyjnych - liczbę iteracji niezbędnych dla uzyskania pożądanego wyniku.

Metoda	Liczba iteracji	Czas obliczeń [s]	Norma residuum
Jacobi	19	0.59	4.042741e-010
Gauss-Seidel	14	0.60	3.440521e-010
Faktoryzacja LU	nie dotyczy	2.04	2.113118e-015

Tabela 1: Wyniki poszczególnych metod dla pierwszego układu równań



Rysunek 2: Zmiana normy błędu rezydualnego w kolejnych iteracjach

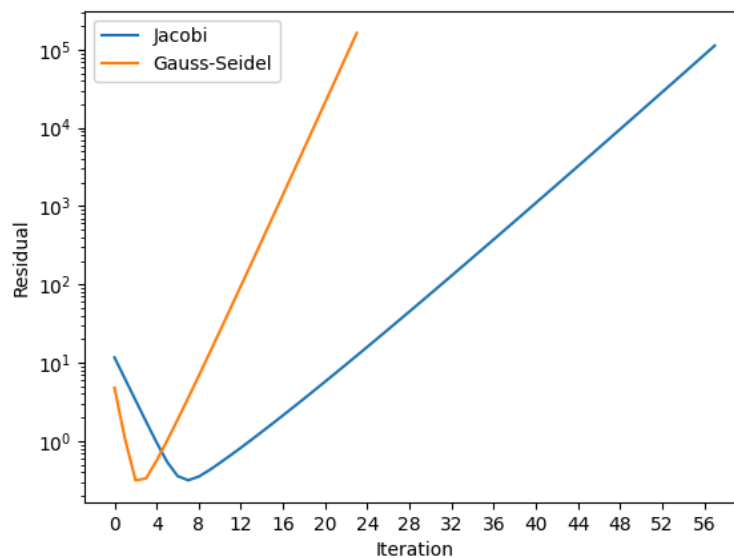
2.2. Główna przekątna wypełniona trójkami

Układ równań jest podobny do poprzedniego, jedyną różnicę stanowi główna przekątna, na której zamiast ósemek są trójki.

$$\begin{pmatrix} 3 & -1 & -1 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & -1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 \\ -1 & -1 & 3 & -1 & -1 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & -1 & 3 & -1 & \dots & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & -1 & 3 & \dots & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \dots & 3 & -1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & 3 & -1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \dots & -1 & -1 & 3 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & -1 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & -1 & -1 & 3 \end{pmatrix} \times \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ \vdots \\ x_{n-4} \\ x_{n-3} \\ x_{n-2} \\ x_{n-1} \\ x_n \end{pmatrix} = \begin{pmatrix} -0.757 \\ 0.989 \\ -0.537 \\ -0.288 \\ 0.913 \\ \vdots \\ -0.948 \\ 0.860 \\ -0.176 \\ -0.630 \\ 1.000 \end{pmatrix}$$

Metoda	Liczba iteracji	Czas obliczeń [s]	Norma residuum
Jacobi	58	-	1.124340e+005
Gauss-Seidel	24	-	1.626362e+005
Faktoryzacja LU	nie dotyczy	2.27	2.661103e-013

Tabela 2: Wyniki poszczególnych metod dla drugiego układu równań



Rysunek 4: Zmiana normy błędu rezydualnego w kolejnych iteracjach

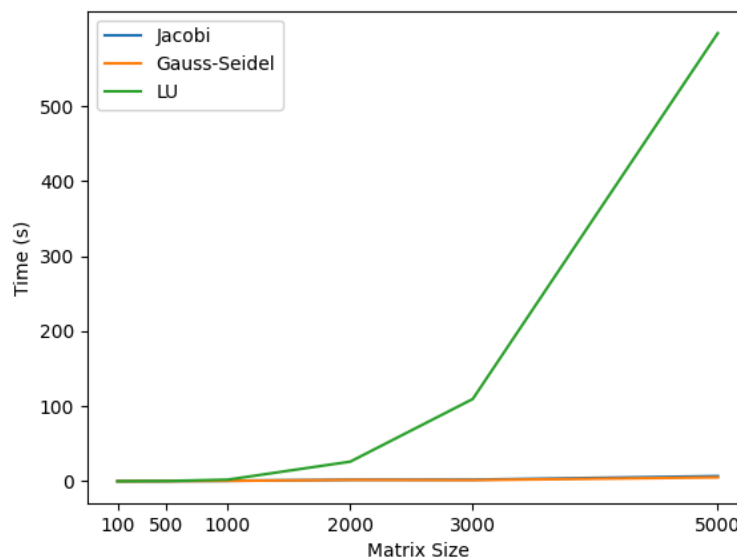
Metody iteracyjne nie zbiegły się, natomiast metoda LU pozwoliła uzyskać satysfakcjonujące rozwiązanie w ciągu niecałych trzech sekund.

3. Porównanie wydajności metod

Uruchomiono wszystkie trzy metody na następujących rozmiarach macierzy A : {100, 500, 1000, 2000, 3000, 5000}. Wyniki prezentują się następująco:

Rozmiar macierzy A	Jacobi	Gauss-Seidel	Faktoryzacja LU
100	0.031251	0.015625	0.015627
500	0.159247	0.140627	0.24986
1000	0.524673	0.514816	2.064534
2000	1.92472	1.754856	26.07829
3000	2.040683	1.480004	109.596
5000	6.778541	5.086763	597.3155

Tabela 3: Czasy wykonywania poszczególnych metod w sekundach dla danego rozmiaru macierzy



Rysunek 5: Zależność czasu wykonywania od rozmiaru macierzy A

4. Wnioski

Na podstawie pomiarów czasu i dokładności obliczeń widać wyraźnie, że dla małych danych lepiej sprawdza się metoda bezpośrednia, natomiast dla dużych ($N \geq 1000$) macierzy odpowiednia wydaje się być metoda Gaussa-Seidla. Dopiero w przypadku braku zbieżności należałoby użyć metody LU.

Ponadto, istnieje wiele bibliotek takich jak NumPy pozwalających przyspieszyć obliczenia poprzez zastosowanie bardziej zaawansowanych metod i efektywniejszego zarządzania pamięcią, np. przez użycie macierzy rzadkich, z których warto korzystać wykonując obliczenia na dużym zbiorze danych.