

Eliminacja Gaussa i LU faktoryzacja

Jakub Płowiec, Filip Dziurdzia

Zadanie

Zaimplementować poniższe algorytmy dla macierzy o rozmiarze $n = 32$

1. Algorytm eliminacji Gaussa bez pivotingu generujący jedynki na przekątnej
2. Algorytm eliminacji Gaussa z pivotingiem
3. Algorytm LU faktoryzacji bez pivotingu
4. Algorytm LU faktoryzacji z pivotingiem

1. Wprowadzenie

Eliminacja Gaussa oraz faktoryzacja LU to podstawowe metody rozwiązywania układów równań liniowych i dekompozycji macierzy.

Eliminacja Gaussa

1. **Bez pivotingu** – metoda eliminacji zmiennych przez zerowanie współczynników pod główną przekątną, bez zmiany kolejności równań.
2. **Z pivotingiem** – dodatkowo dokonuje zamiany wierszy (pivotingu) w celu uniknięcia błędów numerycznych wynikających z dzielenia przez małe wartości.

Faktoryzacja LU

1. **Bez pivotingu** – dekompozycja macierzy na iloczyn dwóch macierzy trójkątnych (dolnej L i górnej U) bez zamiany wierszy.
2. **Z pivotingiem** – uwzględnia zamiany wierszy, co zwiększa stabilność numeryczną i unika dzielenia przez małe wartości.

Implementacja tych metod dla macierzy o rozmiarze 32×32 pozwala na efektywne rozwiązywanie układów równań liniowych oraz badanie ich stabilności numerycznej.

2.1 Eliminacja Gaussa bez pivotingu

Idea metody

Metoda polega na eliminacji zmiennych przez zerowanie współczynników pod główną przekątną, bez zamiany wierszy.

Algorytm

1. Iteracyjnie dla każdej kolumny k :
 - Dzielimy wiersz k przez element diagonalny, aby na diagonalu znajdowały się jedynki.

- Odejmujemy wielokrotności wiersza k od kolejnych wierszy.
2. Po uzyskaniu macierzy schodkowej w celu otrzymania rozwiązania wykorzystywane jest podstawianie wsteczne.

Złożoność obliczeniowa

Złożoność tej metody wynosi $O(n^3)$.

2.2 Eliminacja Gaussa z pivotingiem

Idea metody

Aby zwiększyć stabilność numeryczną, dokonuje się zamiany wierszy w celu uniknięcia dzielenia przez małe wartości.

Algorytm

1. Iteracyjnie dla każdej kolumny k :
 - Wybieramy największy element w kolumnie (pivot) i zamieniamy wiersze.
 - Dzielimy wiersz k przez element diagonalny.
 - Odejmujemy wielokrotności wiersza k od kolejnych wierszy.
2. Po uzyskaniu macierzy schodkowej w celu otrzymania rozwiązania wykorzystywane jest podstawianie wsteczne.

Złożoność obliczeniowa

Złożoność tej metody wynosi $O(n^3)$, lecz zapewnia większą stabilność numeryczną niż wersja bez pivotingu.

2.3 Faktoryzacja LU bez pivotingu

Idea metody

Dekompozycja macierzy A na dwie macierze: dolną trójkątną L i górną trójkątną U , bez zamiany wierszy.

Po rozkładzie $A = LU$, układ $Ax = b$ zapisujemy jako:

$$LUx = b$$

Podstawiamy:

$$Ux = y$$

Po faktoryzacji rozwiązujemy układ równań w dwóch krokach: najpierw rozwiązujemy układ $Ly = b$, gdzie y jest wektorem pośrednim, a następnie $Ux = y$, aby znaleźć ostateczne rozwiązanie x .

Algorytm

1. Iteracyjnie dla każdej kolumny k :
 - Eliminujemy elementy pod diagonalą i zapisujemy współczynniki w L .
 - Wartości nad diagonalą zapisujemy w U .
2. Rozwiązanie układu:
 - Rozwiązujemy układ $Ly = b$ metodą podstawiania w przód.
 - Rozwiązujemy układ $Ux = y$ metodą podstawiania wstecz.

Złożoność obliczeniowa

Złożoność tej metody wynosi $O(n^3)$.

2.4 Faktoryzacja LU z pivotingiem

Idea metody

Podobna do LU bez pivotingu, ale uwzględnia zamiany wierszy w celu zwiększenia stabilności numerycznej. Po faktoryzacji rozwiązujemy układ równań w dwóch krokach: najpierw rozwiązujemy układ $Ly = b$, gdzie y jest wektorem pośrednim, a następnie $Ux = y$, aby znaleźć ostateczne rozwiązanie x .

Algorytm

1. Iteracyjnie dla każdej kolumny k :
 - Wybieramy największy element w kolumnie (pivot) i zamieniamy wiersze.
 - Eliminujemy elementy pod diagonalą, zapisując współczynniki w L .
 - Wartości nad diagonalą zapisujemy w U .
2. Rozwiązanie układu:
 - Rozwiązujemy układ $Ly = b$ metodą podstawiania w przód.
 - Rozwiązujemy układ $Ux = y$ metodą podstawiania wstecz.

Złożoność obliczeniowa

Złożoność tej metody wynosi $O(n^3)$, ale metoda ta jest bardziej stabilna numerycznie niż wersja bez pivotingu.

Rozwiązanie

Implementację wszystkich algorytmów wykonaliśmy w języku **Python**.

Funkcje Główne

Funkcja: `gauss_ones_no_pivot(A)`

- **Dane wejściowe:**
 - A : Prostokątna macierz o rozmiarze $(n + 1) \times n$, gdzie jest uzupełniona o macierz wyrazów wolnych.
- **Wynik:**

- Wektor \mathbf{x} , będąca zbiorem wyników po sprowadzeniu macierzy do postaci schodkowej i zastosowaniu podstawiania wstecznego.

- **Algorytm:**

1. Normalizacja wiersza – dzieli każdy wiersz przez element główny na diagonalu.
2. Eliminacja – zeruje elementy poniżej elementu głównego przez operacje na wierszach.
3. Rozwiązanie – po uzyskaniu postaci górnotrójkątnej wywołuje eliminację wsteczną.

Funkcja: `gauss_pivot(A)`

- **Dane wejściowe:**

- A : Prostokątna macierz o rozmiarze $(n + 1) \times n$, gdzie jest uzupełniona o wektor wyrazów wolnych.

- **Wynik:**

- Wektor \mathbf{x} , będąca zbiorem wyników po sprowadzeniu macierzy do postaci schodkowej i zastosowaniu podstawiania wstecznego.

- **Algorytm:**

1. Wybór elementu głównego – zamienia wiersze, tak aby na diagonalu był największy element.
2. Normalizacja wiersza – dzieli wiersz przez element główny.
3. Eliminacja – zeruje elementy poniżej elementu głównego przez operacje na wierszach.
4. Rozwiązanie – po uzyskaniu postaci górnotrójkątnej wywołuje eliminację wsteczną.

Funkcja: `lu_decomposition_no_pivot(A, b)`

- **Dane wejściowe:**

- A : Kwadratowa macierz o rozmiarze $n \times n$.
- b : Wektor wyrazów wolnych o rozmiarze n .

- **Wynik:**

- Wektor \mathbf{x} , będący zbiorem wyników.

- **Algorytm:**

1. Faktoryzacja LU – rozkłada macierz A na iloczyn macierzy dolnotrójkątnej L i górnotrójkątnej U , zapisując współczynniki eliminacji w L .
2. Rozwiązanie układu:
 - Przesyłanie wprzód (`forward_substitution`) – rozwiązuje $Ly = b$.
 - Eliminacja wsteczna (`backward_substitution`) – rozwiązuje $Ux = y$.

Funkcja: `lu_decomposition_pivot(A, b)`

- **Dane wejściowe:**

- A : Kwadratowa macierz o rozmiarze $n \times n$.
- b : Wektor wyrazów wolnych o rozmiarze n .

- **Wynik:**

- Wektor \mathbf{x} , będący zbiorem wyników.

- **Algorytm:**

1. Pivoting – zamienia wiersze, aby uniknąć zer na diagonalu i poprawić stabilność obliczeń.
2. Faktoryzacja LU – rozkłada macierz A na iloczyn macierzy dolnotrójkątnej L i górnortrójkątnej U , zapisując współczynniki eliminacji w L .
3. Rozwiązanie układu:
 - Przesyłanie wprzód (**forward_substitution**) – rozwiązuje $Ly = b$.
 - Eliminacja wsteczna (**backward_substitution**) – rozwiązuje $Ux = y$.

Funkcje pomocniczne

Funkcja: **check_solution(A, b, x, tol=1e-6)**

- **Dane wejściowe:**
 - **A**: Kwadratowa macierz o rozmiarze $n \times n$.
 - **b**: Wektor wyrazów wolnych o rozmiarze n .
 - **x**: Wektor otrzymanych rozwiązań do sprawdzenia.
- **Wynik:**
 - Wartość True/ False w zależności od tego, czy rozwiązanie jest prawidłowe.

Funkcja: **backward_substitution(A)**

- **Dane wejściowe:**
 - **A**: Prostokątna macierz o rozmiarze $(n + 1) \times n$ uzupełniona o wektor wyrazów wolnych.
- **Wynik:**
 - Wektor **x**, będący zbiorem wyników.

Funkcja: **forward_substitution(A)**

- **Dane wejściowe:**
 - **A**: Prostokątna macierz o rozmiarze $(n + 1) \times n$ uzupełniona o wektor wyrazów wolnych.
- **Wynik:**
 - Wektor **y**, będący pośrednim rozwiązaniem potrzebnym do rozwiązania układu $Ax = b$ za pomocą podstawiania wstecznego.

2. Wyniki

Obliczenia zostały wykonane dla $n = 32$, które zostało zdefiniowane za pośrednictwem daty urodzenia. Dla takiej wielkości macierzy został zmierzony czas wyliczania rozwiązania dla każdej z metod i prezentuje się ono następująco:

Metoda	Czas [s]
Gauss bez pivotu	0,0120

Metoda	Czas [s]
Gauss z pivotem	0,0130
Dekompozycja LU bez pivotu	0,0020
Dekompozycja LU z pivotem	0,0020

Powyższe wyniki mogą się różnić w zależności od wygenerowanych danych w macierzach, natomiast metoda dekompozycji LU osiąga zazwyczaj najlepsze wyniki czasowe.