Sprawozdanie laboratorium lista 5
Obliczenia naukowe

Zofia Tarchalska, indeks: 279699

1 Wstęp

Celem tej listy było zaimplementowanie trzech poniższych algorytmów o złożoności $O(n)$, gdzie n to rozmiar macierzy $n \times n$.

- funkcja rozwiązująca układ $Ax = b$ metodą eliminacji Gaussa
- funkcja wyznaczająca rozkład LU macierzy A metodą eliminacji Gaussa
- funkcja rozwiązująca układ $Ax = b$ jeśli już wcześniej został wyznaczony rozkład LU

Wszystkie te algorytmy mają działać dla macierzy A o specyficznej postaci, która została dokładnie opisana w poleceniu do zadania. Dodatkowo należało zaprogramować wersję podstawową oraz wersję z wybieraniem elementu głównego.

2 Metoda eliminacji Gaussa

Główną ideą metody eliminacji Gaussa jest doprowadzenie macierzy do postaci, w której pod przekątną znajdują się same zera. Uzyskujemy to poprzez mnożenie kolejnych wierszy macierzy przez odpowiednie czynniki i odejmowanie ich od wierszy następujących po nich. Poprawny schemat:
Eliminujemy zmienną x_k z wierszy od $k + 1$ do n. Mnożymy $k - te$ równanie przez

$$l_{ik} = \frac{a_{ik}}{a_{kk}} \quad \text{dla } i \in \{k + 1, ..., n\}$$

Kiedy jednak na przekątnej w miejscu a_{kk} będzie 0 metoda może powodować błąd numeryczny, ponieważ nastąpi dzielenie przez 0. W przeciwnym wypadku, po wykonaniu odpowiednich kroków kolejno dla wszystkich wierszy macierzy, otrzymujemy macierz górnotrójkątną.

2.1 Wariant pierwszy - bez wyboru elementu głównego

Ten wariant nie zabezpiecza nas przed opisanym powyżej błędem numerycznym.

2.2 Złożoność obliczeniowa

W niezoptymalizowanej wersji algorytmu złożoność wynosiłaby $O(n^3)$, ponieważ mamy 3 pętle `for` przechodzące po elementach macierzy. Dla każdego wiersza w macierzy wyznaczamy czynnik przez który przemnażamy ten wiersz po czym odejmujemy go od kolejnych wierszy (musimy przejść po wszytskich elementach w danym wierszu, a macierz ma wymiary $n \times n$). Jednak wykorzystując blokową strukturę macierzy A da się sprowadzić ten algorytm do złożoności $O(n)$. Wiemy, że rozmiar pojedynczego bloku ma jakiś określony rozmiar. Oznaczmy go przez l. Zatem pod przekątną macierzy jest maksymalnie l elementów, które są niezerowe. Dodatkowo każdy wiersz, o indeksie większym niż 1 ma od lewej same zera, następnie l leżących po sobie nierzerowych wartości i znów zera (chyba, że np. $k + l$ daje już indeks ostatniej kolumny, wtedy nie ma zar od końca). Możemy zatem wykonywać ten algorytm tylko dla wierszy i elementów, dla których ma to sens (oszczędzimy sobie przetwarzania ogromnej liczby samych zer). Możemy zatem odejmować tylko l wierszy i aktualizować w nich l wartości. Ponieważ l jest znacząco mniejsze niż n, złożoność spada do $O(n)$.

2.3 Pseudokod

Zawarte w pseudokodzie metody `get_bottom_row` oraz `get_last_column` zwracają odpowiednio ostatni wiersz i ostatnią kolumnę, w której pojawia się niezerowy element. Dzięki temu dokonujemy optymalizowania opisanego we wcześniejszym paragrafie i ograniczamy nasze obliczenia jedynie do niezerowych elementów.

```
procedure GaussElimination(A, b)
    n ← size(A)
    for k ← 1 to n − 1 do
        for i ← k + 1 to GET_BOTTOM_ROW(A,k) do
            if A[k, k] = 0 then
                error "Zero on diagonal at (k, k)"
            end if
            m ← A[i, k]/A[k, k]
            A[i, k] ← 0
            for j ← k + 1 to GET_LAST_COLUMN(A,k) do
                A[i, j] ← A[i, j] − m · A[k, j]
            end for
            b[i] ← b[i] − m · b[k]
        end for
    end for
    x ← 0 ∈ ℝ^n
    x[n] ← b[n]/A[n, n]
    for i ← n − 1 downto 1 do
        s ← 0
        for j ← i + 1 to GET_LAST_COLUMN(A,i) do
            s ← s + A[i, j] · x[j]
        end for
        x[i] ← (b[i] − s)/A[i, i]
    end for
    return x
end procedure
```

2.4 Wariant drugi - z częściowym wyborem elementu głównego

Ten wariant ma zabezpieczać nas przed potencjalnym dzieleniem przez 0 gdy na przekątnej macierzy taka warotść się znajduje. Biorąc pod uwagę, że obliczenia wykonujemy na komputerze, liczby bardzo zbliżone do 0 również są tymi, które mogą nam zwrócić błąd.
Element główny to nic innego jak wybrana wartość, którą będziemy używać aby wyzerować pozostałe w kolumnie.