Podstawy Obliczeń Komputerowych
Laboratorium 1

Kamil Czop 259613
Łukasz Majchrzak 262761
Wtorek 11:15TN - Y02-37c

28 marca 2023

1 Cel ćwiczeń

Celem przeprowadzonych laboratoriów było utworzenie programu rozwiązującego układ równań liniowych z wykorzystaniem dowolnej metody skończonej.

2 Dane

$$\mathbf{A} = \begin{bmatrix} 2 & 1 & 1 & -1 \\ 1 & 1 & 1 & -1 \\ -1 & 2 & -1 & 1 \\ -1 & 2 & -1 & 1 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 3 \\ 4 \\ 10 \\ -4 \end{bmatrix} \qquad \mathbf{AB} = \begin{bmatrix} 2 & 1 & 1 & -1 & 3 \\ 1 & 1 & 1 & -1 & 4 \\ -1 & 2 & -1 & 1 & 10 \\ -1 & 2 & -1 & 1 & 4 \end{bmatrix}$$

3 Wykonanie

3.1 Twierdzenie Kroneckera-Capellego

Bazując na wymaganiach określonych w instrukcji laboratoryjnej wykonano początkowo program określającego z tw. Kroneckera-Capellego do wyznaczenia liczby rozwiązań naszego układu równań. Po przepuszczeniu przez algorytm obliczający rangę macierzy w C++ i gotowej funkcji Matlab rank, dla macierzy **A** uzyskano wartość 4, a dla **AB** także 4. Z tego wynika iż nasz układ równań posiadając równe rangi może posiadać jedno rozwiązanie układu. Posiadając potwierdzenie istnienia rozwiązania możemy rozwiązać układ metodą Gauss'a-Jordana.

3.2 Metoda Gauss'a - Jordana

Algorithm 1 Algorytm postępowania dla metody Gauss'a-Jordana

procedure GAUSSJORDAN
  for $i = 1ton$ **do**
    if $A_{i,i} = 0$ **then**
      Stop
    for $j = 1ton$ **do**
      if $i! = j$ **then**
        $Ratio \leftarrow A_{j,i}/Ai, i$
        for $k = 1ton$ **do**
          $A_{j,k} \leftarrow A_{j,k} - Ratio * Ai, k$
          Next k
      Next j
    Next i
  Stop

Metoda ta stanowi modyfikacje metody Gauss'a gdzie układ równań liniowych jest sprowadzany do macierzy jednostkowej. W początkowej faczie poszerzamy macierz **A** o macierz **B** uzyskując macierz **AB**. Macierz AB przechodzi podaną transformacje.

$$\begin{bmatrix} A_{11} & \dots & A_{1n} & B_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{n1} & \dots & A_{nn} & B_n \end{bmatrix} \quad \rightarrow \begin{bmatrix} 1 & \dots & 0 & B_1^{(n)} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & 1 & B_n^{(n)} \end{bmatrix}$$

Wynikiem operacji na wykorzystywanym zbiorze danych jest macierz jednostkowa o postaci:

$$\mathbf{AB} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 2 \\ 0 & 0 & 1 & 0 & 3 \\ 0 & 0 & 0 & 1 & 4 \end{bmatrix}$$

Co przekłada się na uzyskanie współczynników o wartościach:

$$x_1 = 1 \qquad x_2 = 2 \qquad x_3 = 3 \qquad x_4 = 4$$

4 Kod źródłowy

4.1 Kod funkcji wykonującej metodę gauss'a-jordana

```
std::vector<std::vector<double>> gaussJordan(std::vector<std::vector<double>> A){
    for(long unsigned int i = 0; i < A.size(); i++){
        for(long unsigned int j = 0; j < A.size(); j++){
            if( i!=j ){
                double ratio = A[j][i] / A[i][i];
                for(long unsigned int k = 0; k < A[0].size(); k++){
                    A[j][k] = A[j][k] - ratio * A[i][k];
                }
            }
        }
    }
    for(long unsigned int i = 0; i < A.size(); i++){
        double ratio = 1 / A[i][i];
        A[i][i] = A[i][i]*ratio;
        A[i][A[i].size()-1] = A[i][A[i].size()-1]*ratio;
    }
    return A;
}
```

5 Github, bibliografia, dokumentacja

- https://github.com/Myknakryu/pok-2023
- https://traf-barak.pwr.edu.pl/wp-content/uploads/2020/08/Raport.pdf
- https://www.youtube.com/watch?v=xOLJMKGNivU