Akademia Górniczo-Hutnicza w Krakowie

Wydział Fizyki i Informatyki Stosowanej

# Metody Numeryczne

## Laboratorium 00: Metoda Eliminacji Gaussa

*Andrzej Świętek*

27.02.2024

# Contents



# 1 Wstęp teoretyczny

Metoda eliminacji Gaussa jest jedną z podstawowych technik rozwiązywania układów równań liniowych. Jest to algorytm iteracyjny, który przekształca układ równań liniowych w równoważny układ, w którym jedna zmienna jest eliminowana w każdym kroku, aż do uzyskania postaci trójkątnej macierzy współczynników. Następnie, stosując algorytm substitucji wstecznej, możemy znaleźć wartości zmiennych.

## 1.1 Metoda eliminacji Gaussa

Rozważmy układ trzech równań liniowych:

$$\begin{cases} -1x & +2y & +z & = -1 \\ 1x & -3y & -2z & = -1 \\ 3x & -1y & -1z & = 4 \end{cases} \implies \begin{bmatrix} -1 & 2 & 1 \\ 1 & -3 & -2 \\ 3 & -1 & -1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -1 \\ -1 \\ 4 \end{bmatrix}$$

W celu rozwiązania tego układu równań za pomocą metody eliminacji Gaussa, wykonujemy następujące kroki:

1. **Zerowanie pierwszej kolumny macierzy**

   Pierwszym krokiem metody eliminacji Gaussa jest utworzenie macierzy rozszerzonej dla rozpatrywanego układu równań, a następnie wykorzystanie pierwszego wiersza do wyzerowania współczynników przy zmiennej $x$ w kolejnych wierszach.
   Operacja ta polega na wybraniu odpowiedniego współczynnika (najczęściej nazywanego "pivotal element") w pierwszej kolumnie i wykorzystaniu go do wyzerowania odpowiednich współczynników w kolejnych wierszach.

Aby wyzerować współczynniki przy zmiennej $x$ w kolejnych wierszach, należy znaleźć współczynnik, który po odpowiedniej operacji (dodawania lub odejmowania) spowoduje, że po pomnożeniu przez niego pierwszego wiersza i dodaniu (lub odjęciu) od kolejnych wierszy, współczynniki przy zmiennej $x$ w tych wierszach staną się zerami.

$$\left[\begin{array}{ccc|c} -1 & 2 & 1 & -1 \\ 1 & -3 & -2 & -1 \\ 3 & -1 & -1 & 4 \end{array}\right] \quad \begin{array}{l} \\ +1 \cdot w_1 \\ +3 \cdot w_1 \end{array}$$

2. **Zerowanie zmiennej $y$:**

W drugim kroku metody eliminacji Gaussa koncentrujemy się na wyzerowaniu współczynników zmiennej $y$ w kolejnych wierszach macierzy.

Po wyzerowaniu kolumny $y$ w pierwszym wierszu (który już został przetworzony podczas pierwszego kroku eliminacji), skupiamy się na wyzerowaniu współczynników tej zmiennej w pozostałych wierszach. Wybieramy odpowiedni współczynnik w drugiej kolumnie, który po odpowiedniej operacji

$$\left[\begin{array}{ccc|c} -1 & 2 & 1 & -1 \\ 0 & -1 & -1 & -2 \\ 0 & 5 & 2 & 1 \end{array}\right] \quad \begin{array}{l} \\ \\ +5 \cdot w_1 \end{array}$$

3. **Otrzymana Macierz trójkątna**

Po zakończeniu procesu eliminacji Gaussa uzyskujemy macierz trójkątną górną, w której wszystkie współczynniki poniżej głównej przekątnej są równe zero. Jest to macierz, w której każdy kolejny wiersz ma coraz więcej zerowych współczynników po lewej stronie głównej przekątnej. Dzięki temu macierz trójkątna górna umożliwia łatwe rozwiązanie układu równań za pomocą substytucji wstecznej.

$$\left[\begin{array}{ccc|c} -1 & 2 & 1 & -1 \\ 0 & -1 & -1 & -2 \\ 0 & 0 & -3 & 9 \end{array}\right] \quad \Longrightarrow \; z = 3$$

4. **Wyliczenie wartości zmiennych:**
   Korzystając z otrzymanych równań, możemy wyliczyć wartości zmiennych. Wartość $z$ już znamy ($z = 1$). Podstawiając $z = 1$ do drugiego równania, otrzymujemy $y = 3$. Następnie, podstawiając znane wartości $y = 3$ i $z = 1$ do pierwszego równania, otrzymujemy $x = 2$.

$$\begin{aligned} 0 \cdot x + (-1) \cdot y + (-1) \cdot z &= -2 \\ -y - 3 &= -2 \\ y &= -1 \end{aligned}$$

$$\begin{aligned} (-1) \cdot x + 2 \cdot y + 1 \cdot z &= -1 \\ -x - 2 + 3 &= -1 \\ x &= 2 \end{aligned}$$

# 2 Implementacja

W samej implementacji możemy wyróżnić dwa główne etapy algorytmu.

## 2.1 Eliminacja i tworzenie macierzy trójkątnej

### 2.1.1 Wersja synchroniczna

1. Algorytm iteruje kolejno przez wiersze macierzy (zewnętrzna pętla), a w każdym wierszu przegląda elementy od i-tego indeksu, co skutkuje iteracją po diagonali macierzy.

2. Dla każdego elementu wyliczany jest współczynnik, który służy do wyzerowania odpowiedniej komórki w i-tym wierszu.

3. Po znalezieniu tego współczynnika stosujemy operację zmiany dla każdego elementu danego wiersza, uwzględniając obliczony współczynnik.

Operacja ta ma na celu stworzenie macierzy trójkątnej górnej, gdzie wszystkie elementy poniżej głównej przekątnej są równoważone do zera.

```
for (int i = 0; i < n-1; i++) {
    for (int j = i+1; j < n; j++) {
        double factor = mat[j][i] / mat[i][i];
        for (int k = 0; k < n+1; k++) {
            mat[j][k] -= factor * mat[i][k];
        }
    }
}
```

Listing 1: Tworzenie macierzy trójkątnej

### 2.1.2 Wersja zrównoleglona

Wersja zrównoleglona tego algorytmu wykorzystuje OpenMP do równoczesnego przetwarzania wielu wierszy macierzy przez wiele wątków. W każdej iteracji zewnętrznej pętli, w której iterujemy po wierszach, instrukcje wewnętrznej pętli wykonywane są równolegle przez wiele wątków, co przyspiesza proces przetwarzania.

```
for (i = 0; i < n - 1; i++) {
    #pragma omp parallel for private(j, k) shared(mat)
    for (j = i + 1; j < n; j++) {
        double factor = mat[j][i] / mat[i][i];
        for (k = 0; k < n + 1; k++) {
            mat[j][k] -= factor * mat[i][k];
        }
    }
}
```

Listing 2: Tworzenie macierzy trójkątnej - OMP

- Wersja zrównoleglona wykorzystuje dyrektywę $pragmaompparallelfor$, aby równolegle wykonywać pętlę wewnętrzną dla różnych wartości j.

- Dyrektywa $private(j, k)$ określa, że zmienne $j$ i $k$ będą prywatne dla każdego wątku, a $shared(mat)$ wskazuje, że macierz $mat$ jest współdzielona między wszystkie wątki. Dzięki temu każdy wątek może równolegle przetwarzać odpowiednie wiersze macierzy, co przyspiesza obliczenia.

Oczywistym kandydatem do zrównoleglenia w tym przypadku jest pętla wewnętrzna, w której iterujemy po elementach macierzy dla danego wiersza. W tej pętli obliczany jest współczynnik factor i aktualizowane są elementy macierzy.

Przykładowo, jeśli mamy macierz o rozmiarze 3x3: Pętla wewnętrzna iteruje przez kolejne wiersze macierzy, zaczynając od drugiego wiersza (bo pierwszy zostaje pominięty przez pętlę zewnętrzną).
Dla każdego wiersza $j$, współczynnik factor jest obliczany jako $mat[j][i]/mat[i][i]$, gdzie i to aktualny indeks kolumny.
Następnie elementy macierzy $mat[j][k]$ (gdzie k to kolejne kolumny) są aktualizowane poprzez odjęcie $factor * mat[i][k]$.

Wersja zrównoleglona równocześnie oblicza wartości factor i aktualizuje elementy macierzy dla różnych wierszy $j$.
Wątki równoległe wykonują obliczenia dla różnych wierszy, co pozwala na zrównoleglenie obliczeń i przyspieszenie procesu. Jednak należy zauważyć, że istnieje ryzyko wystąpienia wyścigów danych lub innych problemów związanych z dostępem do pamięci współdzielonej, dlatego konieczne jest odpowiednie zarządzanie dostępem do danych współdzielonych w programie równoległym.

## 2.2 Wyznaczanie wyniku i podstawianie

W tej sekcji algorytmu, po uzyskaniu macierzy trójkątnej, przechodzimy do wyznaczenia rozwiązania układu równań poprzez podstawienie. Iterujemy od ostatniego wiersza do pierwszego (wstecz) macierzy trójkątnej.

Dla każdego wiersza $i$, najpierw obliczamy wartość $wynik[i]$ jako wyraz wolny tego wiersza. Następnie przeprowadzamy operację podstawiania, wykorzystując już obliczone wartości $wynik[j]$ dla $j > i$. Dla każdej kolumny j większej niż $i$, mnożymy współczynnik $mat[i][j]$ przez odpowiednią wartość $wynik[j]$ i odejmujemy od $wynik[i]$.

Na koniec, wynik dzielimy przez współczynnik główny $mat[i][i]$, aby otrzymać ostateczną wartość zmiennej.

Powyższy opis odpowiada implementacji w języku C++ przedstawionej poniżej.

```
1    for ( int i = n-1; i >= 0; i-- )
2    {
3        wynik[i] = mat[i][n];
4        for ( int j = i+1; j < n; j++ )
5            if( i != j ) {
6                wynik[i] = wynik[i] - mat[i][j] * wynik[j];
7            }
8
9        wynik[i] /=  mat[i][i];
10   }
```

Listing 3: Wyznaczanie wyniku

# 3 Wnioski

Wyniki działania programu są zgodne z wynikami wyliczonymi ręcznie zgodnie z teorią, ponieważ metoda ta jest zaimplementowana bezpośrednio.

## 3.1 Skuteczność

Metoda eliminacji Gaussa jest skuteczną metodą rozwiązywania układów równań liniowych. Jest stosowana powszechnie ze względu na swoją prostotę i efektywność.

## 3.2 Stabilność numeryczna

Metoda eliminacji Gaussa może być podatna na błędy numeryczne, szczególnie gdy współczynniki macierzy są bliskie zeru lub gdy występują duże różnice w wartościach współczynników.

Konieczne może być zastosowanie dodatkowych technik, takich jak skalowanie, aby zapobiec utracie dokładności numerycznej.

## 3.3 Złożoność obliczeniowa:

Złożoność obliczeniowa metody eliminacji Gaussa wynosi około $O(n^3)$ gdzie n to liczba równań.

### 3.4 Wymagania dotyczące pamięci:

Metoda eliminacji Gaussa może wymagać dużych ilości pamięci, szczególnie dla dużych macierzy.
W przypadku ograniczonej dostępności pamięci może to być istotne ograniczenie.

### 3.5 Implementacja i optymalizacje:

Istnieją różne warianty i optymalizacje metody eliminacji Gaussa, które mogą być stosowane w zależności od konkretnych przypadków.

Optymalizacje te mogą obejmować eliminację rekalkulacji, eliminację zerowych kolumn czy też wykorzystanie obliczeń równoległych.

### 3.6 Podsumowanie

Metoda eliminacji Gaussa jest skuteczną i wszechstronną metodą rozwiązywania układów równań liniowych. Pomimo pewnych ograniczeń związanych z wymaganiami dotyczącymi pamięci i potencjalnymi problemami związanymi z błędami numerycznymi, jest powszechnie stosowana w różnych dziedzinach nauki i techniki, dzięki swojej prostocie i skuteczności. Zastosowanie odpowiednich technik optymalizacji może zwiększyć wydajność metody eliminacji Gaussa i umożliwić jej zastosowanie w rozwiązywaniu bardziej skomplikowanych problemów numerycznych. Dodatkowym atutem tej metody jest łatwość i przejrzystość. Sama implementacja bardzo prosta i intuicyjna pozwalając na szybką implementacje wrazie potrzeby.