



Implementacja Eliminacji Gaussa-Jordana w różnych wariantach

IMPLEMENTACJA SEKWENCYJNA, WSPÓŁBIEŻNA ORAZ RÓWNOLEGŁA

Cel projektu

- ▶ Celem mojego projektu jest **implementacja oraz analiza wydajności metody eliminacji Gaussa-Jordana w różnych wariantach**, aby sprawdzić, która metoda zrównoleglenia jest najbardziej efektywna w zależności od wielkości problemu i liczby wątków.

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & \dots & a_{1n} \\ a_{21} & a_{22} & a_{23} & a_{24} & \dots & a_{2n} \\ a_{31} & a_{32} & a_{33} & a_{34} & \dots & a_{3n} \\ a_{41} & a_{42} & a_{43} & a_{44} & \dots & a_{4n} \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & a_{n4} & \dots & a_{nn} \end{bmatrix} \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ \vdots \\ x_n \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ \vdots \\ b_n \end{bmatrix}$$

Wstęp teoretyczny

- ▶ **Eliminacja Gaussa-Jordana – co to jest?**

Eliminacja Gaussa-Jordana to metoda przekształcenia macierzy układu równań liniowych do postaci jednostkowej, co pozwala na łatwe wyznaczenie rozwiązań.

- ▶ **Charakterystyka algorytmu:**

1. Normalizacja wiersza głównego – dzielenie wiersza przez wartość osiowego elementu, aby uzyskać wartość 1 w przekątnej macierzy.

2. Eliminacja elementów pozostałych w kolumnie – zerowanie wszystkich innych elementów w kolumnie osiowego elementu.

3. Powtórzenie operacji dla kolejnych kolumn aż do uzyskania macierzy jednostkowej w przypadku macierzy kwadratowej lub jej rozszerzonej wersji dla układów równań.

- ▶ **Złożoność czasowa:**

$O(n^3)$ – oznacza, że dla dużych układów równań obliczenia są bardzo kosztowne czasowo.

Obszar stosowalności

► Implementacja eliminacji Gaussa-Jordana znajdzie zastosowanie w:

- Obliczeniach inżynierskich wymagających rozwiązania dużych układów równań (np. Matematyka stosowana czy uczenie maszynowe)
- Przetwarzanie sygnałów – filtrowanie, analiza danych
- Grafice komputerowej, np. w procesach renderowania

Oczekuje się, że implementacje te znacząco skrócą czas wykonywania operacji. Eksperymentalne wyniki pokażą, jaka liczba wątków daje optymalne przyspieszenie w zależności od architektury sprzętowej.

Koncepcja zrównoleglenia

- ◆ Sekwencyjna (C++) – pętla for przetwarzająca wiersze jeden po drugim
- ◆ Współbieżna (std::thread) – przypisanie wątków do operacji na różnych wierszach
- ◆ OpenMP – automatyczne rozdzielanie operacji wierszowych na wiele rdzeni

Wstępne pomiary dla implementacji sekwencyjnej

Testy wydajnościowe (dla sekwencyjnej wersji w C++):

Wymiar (n)	Czas obliczeń (sekundy)
10	0.0001
50	0.0023
100	0.0124
1000	1.5231
10 000	305.431

Obserwacje:

- Czas rośnie **w tempie n^3**
- Konieczność **optymalizacji za pomocą równoległości**

Założenia projektowe

Zakres działań obejmuje:

- ◆ Implementację **trzech wersji algorytmu**
- ◆ **Pomiary czasów wykonania** dla różnych rozmiarów macierzy
- ◆ **Analizę skalowalności** w zależności od liczby wątków
- ◆ **Wykresy i porównanie wydajności** implementacji

Cel końcowy:

- ✓ Odpowiedź na pytanie: **Która metoda zrównoleglenia jest najbardziej efektywna?**
- ✓ **Określenie optymalnej liczby wątków** dla różnych przypadków



Dziękuję za uwagę

Wykonał: Dawid Garncarek