

Obliczenia Naukowe - Laboratorium 5

Układy równań liniowych z macierzami blokowymi

Krzysztof Zająć

6 stycznia 2026

1 Wstęp i sformułowanie problemu

Celem laboratorium było opracowanie i implementacja efektywnych algorytmów rozwiązywania układów równań liniowych $Ax = b$, gdzie $A \in \mathbb{R}^{n \times n}$ jest macierzą rzadką o specyficznej strukturze blokowej.

Macierz A zdefiniowana jest następująco:

$$A = \begin{pmatrix} A_1 & C_1 & 0 & \cdots & 0 \\ B_2 & A_2 & C_2 & \cdots & 0 \\ 0 & B_3 & A_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & C_{v-1} \\ 0 & \cdots & 0 & B_v & A_v \end{pmatrix}, \quad (1)$$

gdzie n jest podzielne przez l (rozmiar bloku), a $v = n/l$ to liczba bloków wzdłuż diagonali. Bloki wewnętrzne charakteryzują się następującą strukturą:

- $A_k \in \mathbb{R}^{l \times l}$ – macierz gęsta,
- $B_k \in \mathbb{R}^{l \times l}$ – macierz rzadka, posiadająca niezerowe elementy wyłącznie w pierwszym wierszu oraz w ostatniej kolumnie,
- $C_k \in \mathbb{R}^{l \times l}$ – macierz diagonalna.

Standardowa eliminacja Gaussa posiada złożoność obliczeniową $\mathcal{O}(n^3)$. Ze względu na duży rozmiar n oraz rzadkość macierzy, wymagane było opracowanie algorytmów o złożoności liniowej $\mathcal{O}(n)$, uwzględniających strukturę (1).

2 Analiza teoretyczna i algorytmy

Zastosowano dwa podejścia do rozwiązywania problemu: metodę eliminacji Gaussa oraz rozkład LU. W obu przypadkach zmodyfikowano standardowe procedury, ograniczając operacje wyłącznie do elementów niezerowych bloków A_k, B_k, C_k .

2.1 Własności strukturalne i złożoność

Dla ustalonego rozmiaru bloku l , liczba operacji zmiennoprzecinkowych dla pojedynczego kroku eliminacji (redukcja bloku B_{k+1} przy użyciu A_k) jest stała i wynosi $\mathcal{O}(l^3)$ lub $\mathcal{O}(l^2)$ w zależności od optymalizacji. Całkowita złożoność obliczeniowa wynosi zatem:

$$T(n) \approx v \cdot \mathcal{O}(l^\alpha) = \frac{n}{l} \cdot C_l = \mathcal{O}(n), \quad (2)$$

gdzie C_l jest stałą zależną od l . Złożoność pamięciowa również wynosi $\mathcal{O}(n)$, gdyż przechowujemy jedynie v bloków A_k, B_k, C_k o rozmiarze $l \times l$.

2.2 Metoda Eliminacji Gaussa

Algorytm przebiega w $v - 1$ krokach. W k -tym kroku ($k = 1, \dots, v - 1$):

- Wykonujemy eliminację wewnątrz bloku A_k (z ewentualnym częściowym wyborem elementu głównego w kolumnie).
- Eliminujemy poddiagonalny blok B_{k+1} mnożąc $k + 1$ -szy wiersz blokowy przez odpowiednie mnożniki wyznaczone z A_k .
- Operacja ta modyfikuje blok A_{k+1} (fill-in), ale nie narusza rzadkości pozostazej części macierzy.

Ze względu na postać C_k (diagonalna), fill-in w A_{k+1} jest ograniczony.

2.3 Rozkład LU

Wyznaczono rozkład $A = LU$, gdzie L jest macierzą dolną trójkątną z jedynkami na diagonali, a U górną trójkątną.

$$L_{ii} = 1, \quad U_{ij} = 0 \text{ dla } i > j. \quad (3)$$

Macierze L i U są przechowywane w strukturach macierzy A, B, C ("in-place"). Po wyznaczeniu rozkładu, rozwiązywanie układu $Ax = b$ sprowadza się do dwóch podstawień:

$$Ly = b \quad (\text{w przód}), \quad Ux = y \quad (\text{w tył}). \quad (4)$$

3 Wyniki eksperymentów

3.1 Poprawność numeryczna

Weryfikację przeprowadzono na danych syntetycznych, gdzie $b = A \cdot x_{dokl}$, a $x_{dokl} = (1, \dots, 1)^T$. Błąd względny zdefiniowano jako:

$$\varepsilon = \frac{\|x_{obl} - x_{dokl}\|_2}{\|x_{dokl}\|_2}. \quad (5)$$

Poniższa tabela przedstawia wyniki eksperymentów dla różnych rozmiarów macierzy n oraz różnych wskaźników uwarunkowania macierzy wewnętrznych $cond(A_k)$. Obliczenia wykonano z użyciem wyboru elementu głównego (pivoting).

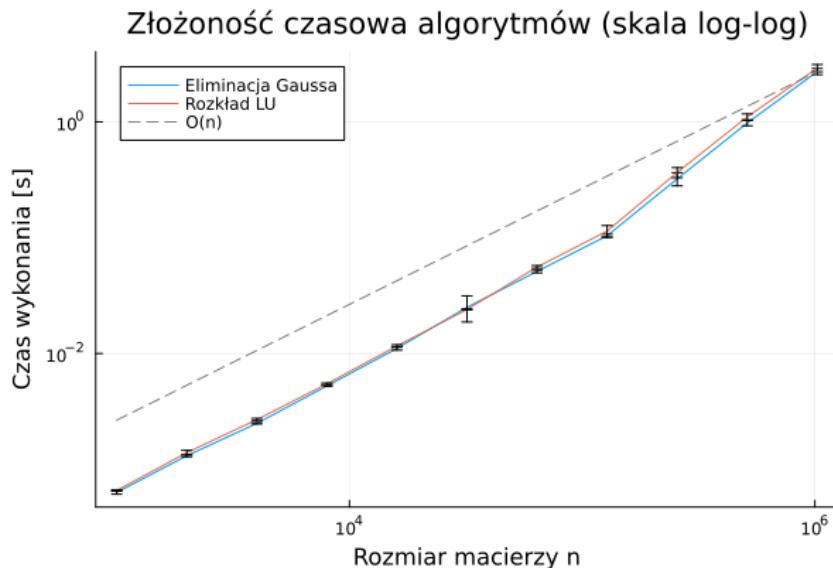
Tabela 1: Błąd względny rozwiązyania dla różnych n i $cond(A_k)$ (blok $l = 4$).

| n | $cond(A_k)$ | Błąd (Eliminacja Gaussa) | Błąd (Rozkład LU) |
|------|-------------|--------------------------|--------------------------|
| 100 | 1.0 | 2.0592×10^{-16} | 2.2699×10^{-16} |
| 100 | 10.0 | 5.0840×10^{-16} | 4.9539×10^{-16} |
| 100 | 1000.0 | 2.9858×10^{-14} | 2.9255×10^{-14} |
| 1000 | 1.0 | 2.4838×10^{-16} | 2.3433×10^{-16} |
| 1000 | 10.0 | 4.1869×10^{-16} | 4.2007×10^{-16} |
| 1000 | 1000.0 | 3.3596×10^{-14} | 3.2694×10^{-14} |
| 5000 | 1.0 | 2.3535×10^{-16} | 2.3091×10^{-16} |
| 5000 | 10.0 | 4.1710×10^{-16} | 4.1858×10^{-16} |
| 5000 | 1000.0 | 3.4627×10^{-14} | 3.4393×10^{-14} |

Metody z wyborem elementu głównego (pivoting) wykazują dużą stabilność numeryczną. Dla macierzy dobrze uwarunkowanych ($cond \approx 1$) błąd jest na poziomie precyzyji maszynowej ($\approx 10^{-16}$). Wraz ze wzrostem uwarunkowania błąd rośnie, co jest zachowaniem oczekiwany.

3.2 Złożoność czasowa

Zmierzono czas wykonania algorytmów w funkcji rozmiaru macierzy n .



Rysunek 1: Zależność czasu obliczeń od rozmiaru macierzy n (skala log-log).

Wyniki potwierdzają teoretyczną liniową złożoność $\mathcal{O}(n)$. Czas wykonania rośnie proporcjonalnie do wzrostu n .

3.3 Porównanie pamięciowe

Standardowe przechowywanie macierzy $n \times n$ wymagałoby $n^2 \cdot 8$ bajtów. Dla $n = 100\,000$, macierz gęsta zajęłaby ok. 74.5 GB RAM. Zastosowana reprezentacja blokowa wymaga pamięci rzędu $3 \cdot (n/l) \cdot l^2 \cdot 8$ bajtów. Dla $l = 4$ i $n = 100\,000$ jest to ok. 2.4 MB, co stanowi redukcję.

4 Wnioski

1. Zaprojektowane algorytmy poprawnie wykorzystują strukturę macierzy blokowej, osiągając złożoność czasową $\mathcal{O}(n)$ oraz pamięciową $\mathcal{O}(n)$. Umożliwia to rozwiązywanie układów o rozmiarach $n > 10^6$, niemożliwych do przetworzenia standardowymi metodami gęstymi ($\mathcal{O}(n^3)$).
2. Częściowy wybór elementu głównego jest kluczowy dla zachowania stabilności numerycznej, szczególnie dla dużych n , gdzie błędy zaokrągleń mogą się kumulować.
3. Rozkład LU jest nieznacznie wolniejszy od jednorazowej eliminacji Gaussa ze względu na narzut zarządzania strukturą, jednak jego koszt amortyzuje się przy wielokrotnym rozwiązywaniu układu dla tej samej macierzy A i różnych wektorów b .