Rozwiązywanie równań i układów równań nieliniowych cz. III

Seweryn Tasior, WI, grupa 5

22.05.2025

1 Wprowadzenie

1.1 Zagadnienie

Celem ćwiczenia jest porównanie stabilności numerycznej, czasu wykonywania i zużywanej pamięci rozwiązywania układów równań liniowych metodą eliminacji Gaussa i Thomasa.

1.2 Dane techniczne

Programy zostały napisane w języku Python w wersji 3.11.5. Dodatkowo, do narysowania wykresów i tabel zostały użyte biblioteki Pandas i matplotlib. Pomocniczo do obliczeń zastosowano funkcjonalności biblioteki numpy. Do inwersji macierzy zastosowano funkcje linalg.inv, a do liczenia iloczynów macierzy np.dot. Maksymalną zużytą pamięć przez funkcję uzyskano za pomocą tracemalloc, a czas wykonywania otrzymano poprzez bibliotekę timeit

Zadania programistyczne wykonano na laptopie Lenovo IdeaPad Gaming 3 15ACH6. Urządzenie posiada 6-rdzeniowy procesor o taktowaniu 4,4 GHz. Korzystano przy tym z systemu operacyjnego Windows 11.

2 Realizacja ćwiczenia

W ćwiczeniu zaimplementowano macierze kwadratowe A o rozmiarach nxn, według podanego wzoru:

$$a_{i,j} = \begin{cases} -3i - 6 & \text{dla } j = i \\ i & \text{dla } j = i + 1 \\ \frac{3}{i} & \text{dla } j = i - 1,\ i > 1 \\ 0 & \text{dla } j < i - 1 \text{ oraz } j > i + 1 \end{cases} \qquad i, j = 1, \ldots, n$$

Zmienna n w obliczeniach przyjmowała następujące zakresy wartości:

- $n \in \{1, 2, .., 20, 30, 50\}$, dla precyzji typu float
- $n \in \{1, 2, .., 20, 30, 50, 80, 100, 200\}$, dla precyzji typu double

Następnie dla każdego n wyliczono wektor x jako dowolną n-elementową permutację ze zbioru $\{1, -1\}$, według powyższych wzorów utworzono macierz A, obliczono wektor b jako iloczyn A i x. Otrzymane b wraz z macierzą A, wykorzystano do obliczenia metodą Gaussa wektora y.

Do wyliczenia rozwiązania metodą Thomasa użyto trzech wektorów reprezentujących podprzekątną, przekątną i nadprzekątną. Na ich podstawie uzyskano wektor y.

W metodzie Gaussa do poprawienia stabilności numerycznej użyto algorytmu **częściowego pivotowania**. Polega on na tym, że w każdej kolumnie i (odpowiadającej i-temu krokowi eliminacji) algorytm szuka elementu o największej wartości bezwzględnej wśród elementów znajdujących się w wierszach od i do $n - 1$.Następnie zamienia wiersz i z wierszem zawierającym ten największy element.

Aby sprawdzić zaburzenia rozwiązań układów, porównano oba wektory x i y (*zadany* i *otrzymany*) za pomocą normy maksimum:

$$\sigma(x, y) = \max_{i=1,\ldots,n} \{|x_i - y_i|\}$$

gdzie:

- x_i – i-ta współrzędna zadanego wektora $\mathbf{x}$

- y_i – i-ta współrzędna obliczonego wektora $\mathbf{y}$

Do oceny wrażliwości rozwiązania układu na małe zaburzenia w danych wejściowych wykorzystano **współczynnik uwarunkowania**. Mówi on, jak bardzo błędy zaokrągleń w obliczeniach komputerowych mogą zostać "wzmocnione"i wpłynąć na dokładność uzyskanego rozwiązania. Jeśli jest on bliski 1, to oznacza, że jest dobrze uwarunkowany i jest mało podatny na błędy. W przypadku, gdy jest on duży, to macierz jest nazywana źle uwarunkowaną, więc nawet małe błędy w danych mogą pogorszyć znacznie wyniki. Współczynnik można dostać z macierzy A na podstawie wzoru:

$$\kappa(A) = \|A^{-1}\| \cdot \|A\|$$

Przyjęto normę wzorem:

$$\|A\| = \max_{1 \le i \le n} \sum_{j=1}^{n} |a_{i,j}|$$

gdzie:

- A – macierz kwadratowa
- n – rozmiar macierzy
- $a_{i,j}$ – element macierzy o współrzędnych i, j

Przeprowadzono eksperymenty dla dwóch różnych precyzji liczbowych: float i double. Wyniki przedstawiono w tabelach.

3 Wyniki i analiza

3.1 Porównanie norm maksimum

Tabela 1: Błędy dla układu w zależności od wielkości macierzy A, precyzja float

n	$\kappa(A)$	$\sigma(x,y)$ (Thomas)	$\sigma(x,y)$ (Gauss)
2	1.65×10^{0}	0.00×10^{0}	0.00×10^{0}
3	1.98×10^{0}	0.00×10^{0}	0.00×10^{0}
4	2.35×10^{0}	0.00×10^{0}	0.00×10^{0}
5	2.81×10^{0}	1.19×10^{-7}	1.19×10^{-7}
6	3.29×10^{0}	0.00×10^{0}	0.00×10^{0}
7	3.77×10^{0}	1.19×10^{-7}	1.19×10^{-7}
8	4.26×10^{0}	1.19×10^{-7}	1.19×10^{-7}
9	4.75×10^{0}	1.19×10^{-7}	1.19×10^{-7}
10	5.24×10^{0}	1.19×10^{-7}	1.19×10^{-7}
11	5.73×10^{0}	1.19×10^{-7}	1.19×10^{-7}
12	6.22×10^{0}	1.19×10^{-7}	1.19×10^{-7}
13	6.71×10^{0}	1.19×10^{-7}	1.19×10^{-7}
14	7.21×10^{0}	1.19×10^{-7}	1.19×10^{-7}
15	7.70×10^{0}	1.19×10^{-7}	1.19×10^{-7}
16	8.19×10^{0}	1.19×10^{-7}	1.19×10^{-7}
17	8.68×10^{0}	1.19×10^{-7}	1.19×10^{-7}
18	9.18×10^{0}	1.19×10^{-7}	1.19×10^{-7}
19	9.67×10^{0}	1.19×10^{-7}	1.19×10^{-7}
20	1.02×10^{1}	1.19×10^{-7}	1.19×10^{-7}
30	1.51×10^{1}	1.19×10^{-7}	1.19×10^{-7}
50	2.50×10^{1}	1.19×10^{-7}	1.19×10^{-7}

Tabela 2: Błędy dla układu w zależności od wielkości macierzy A, precyzja double

n	$\kappa(A)$	$\sigma(x,y)$ (Thomas)	$\sigma(x,y)$ (Gauss)
2	1.65×10^{0}	0.00×10^{0}	0.00×10^{0}
3	1.98×10^{0}	0.00×10^{0}	0.00×10^{0}
4	2.35×10^{0}	0.00×10^{0}	0.00×10^{0}
5	2.81×10^{0}	2.22×10^{-16}	2.22×10^{-16}
6	3.29×10^{0}	0.00×10^{0}	0.00×10^{0}
7	3.77×10^{0}	1.11×10^{-16}	1.11×10^{-16}
8	4.26×10^{0}	2.22×10^{-16}	2.22×10^{-16}
9	4.75×10^{0}	2.22×10^{-16}	2.22×10^{-16}
10	5.24×10^{0}	2.22×10^{-16}	2.22×10^{-16}
11	5.73×10^{0}	2.22×10^{-16}	2.22×10^{-16}
12	6.22×10^{0}	2.22×10^{-16}	2.22×10^{-16}
13	6.71×10^{0}	2.22×10^{-16}	2.22×10^{-16}
14	7.21×10^{0}	2.22×10^{-16}	2.22×10^{-16}
15	7.70×10^{0}	2.22×10^{-16}	2.22×10^{-16}
16	8.19×10^{0}	2.22×10^{-16}	2.22×10^{-16}
17	8.68×10^{0}	2.22×10^{-16}	2.22×10^{-16}
18	9.18×10^{0}	2.22×10^{-16}	2.22×10^{-16}
19	9.67×10^{0}	2.22×10^{-16}	2.22×10^{-16}
20	1.02×10^{1}	2.22×10^{-16}	2.22×10^{-16}
30	1.51×10^{1}	2.22×10^{-16}	2.22×10^{-16}
50	2.50×10^{1}	2.22×10^{-16}	2.22×10^{-16}
80	3.98×10^{1}	4.44×10^{-16}	4.44×10^{-16}
100	4.97×10^{1}	3.33×10^{-16}	3.33×10^{-16}
200	9.92×10^{1}	2.22×10^{-16}	2.22×10^{-16}
400	1.98×10^{2}	4.44×10^{-16}	4.44×10^{-16}

- Metoda Thomasa i Gaussa osiąga dokładność rzędu 10^{-16} (dla zmiennych `double`) i 10^{-7} (dla zmiennych `float`) co odpowiada precyzji maszynowej.
- Metoda Gaussa ma dobrą stabilność numeryczną ze względu na częściowy pivoting.

- Obydwie metody dają dokładne wyniki. Nawet dla dużego układu ($n = 400$), błąd metody Thomasa i Gaussa nie przekracza 4.44×10^{-16}

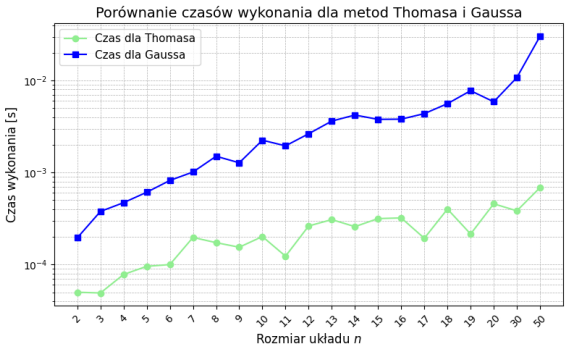
3.2 Porównanie czasów wykonania

Tabela 3: Czas wykonania w zależności od wielkości macierzy A, precyzja float

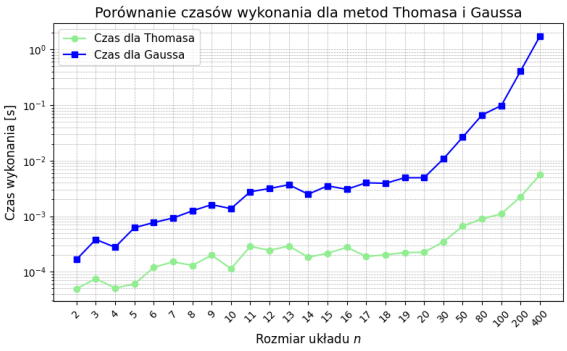
n	Thomas [s]	Gauss [s]
2	5.89×10^{-5}	1.80×10^{-4}
3	4.70×10^{-5}	2.01×10^{-4}
4	8.13×10^{-5}	3.52×10^{-4}
5	5.81×10^{-5}	3.55×10^{-4}
6	6.87×10^{-5}	4.94×10^{-4}
7	8.22×10^{-5}	6.18×10^{-4}
8	8.95×10^{-5}	1.10×10^{-3}
9	1.04×10^{-4}	9.92×10^{-4}
10	1.10×10^{-4}	1.35×10^{-3}
11	1.25×10^{-4}	1.43×10^{-3}
12	1.33×10^{-4}	1.76×10^{-3}
13	1.59×10^{-4}	2.14×10^{-3}
14	1.56×10^{-4}	2.18×10^{-3}
15	1.75×10^{-4}	2.80×10^{-3}
16	1.86×10^{-4}	3.05×10^{-3}
17	2.23×10^{-4}	3.38×10^{-3}
18	2.08×10^{-4}	3.80×10^{-3}
19	2.23×10^{-4}	4.18×10^{-3}
20	2.30×10^{-4}	4.71×10^{-3}
30	3.52×10^{-4}	1.01×10^{-2}
50	6.28×10^{-4}	2.70×10^{-2}

Tabela 4: Czas wykonania w zależności od wielkości macierzy A, precyzja double

n	Thomas [s]	Gauss [s]
2	3.10×10^{-5}	1.92×10^{-4}
3	3.80×10^{-5}	1.78×10^{-4}
4	4.71×10^{-5}	3.74×10^{-4}
5	6.01×10^{-5}	3.59×10^{-4}
6	6.94×10^{-5}	5.04×10^{-4}
7	8.08×10^{-5}	6.34×10^{-4}
8	8.90×10^{-5}	8.44×10^{-4}
9	1.04×10^{-4}	9.99×10^{-4}
10	1.12×10^{-4}	1.20×10^{-3}
11	1.26×10^{-4}	1.43×10^{-3}
12	1.40×10^{-4}	1.77×10^{-3}
13	1.46×10^{-4}	1.88×10^{-3}
14	1.56×10^{-4}	2.16×10^{-3}
15	1.68×10^{-4}	2.51×10^{-3}
16	1.78×10^{-4}	3.20×10^{-3}
17	2.09×10^{-4}	3.14×10^{-3}
18	2.02×10^{-4}	3.48×10^{-3}
19	2.14×10^{-4}	3.85×10^{-3}
20	2.20×10^{-4}	5.17×10^{-3}
30	3.60×10^{-4}	1.00×10^{-2}
50	5.54×10^{-4}	2.46×10^{-2}
80	8.77×10^{-4}	6.22×10^{-2}
100	1.11×10^{-3}	9.83×10^{-2}
200	2.24×10^{-3}	4.07×10^{-1}
400	4.96×10^{-3}	1.67×10^{0}



a) Wykres dla precyzji float



b) Wykres dla precyzji double

Rysunek 1: Wykresy czasów wykonania w zależności od n dla różnych precyzji

Tabela 5: Porównanie czasów wykonania dla n=50

Precyzja	**Thomas [s]**	**Gauss [s]**	**Stosunek (Gauss/Thomas)**
Double	6.28×10^{-4}	2.70×10^{-2}	≈ 43
Float	5.54×10^{-4}	2.46×10^{-2}	≈ 44

- Metoda Thomasa wykazuje liniowy wzrost czasu wykonania $O(n)$

- Metoda Gaussa wykazuje wykładniczy wzrost $O(n^3)$ dla większych n
- Dla $n = 400$, metoda Thomasa jest ponad 300 razy szybsza od metody Gaussa

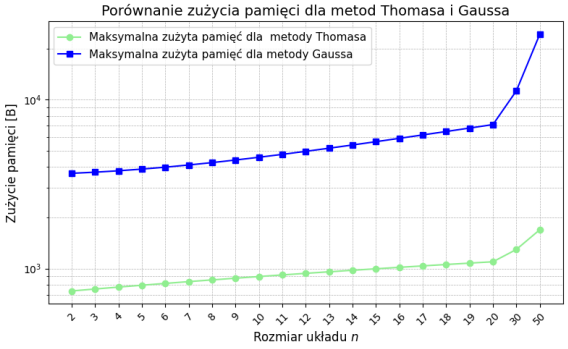
3.3 Porównanie zużytej pamięci

Tabela 6: Zużycie pamięci w zależności od wielkości macierzy A, precyzja float

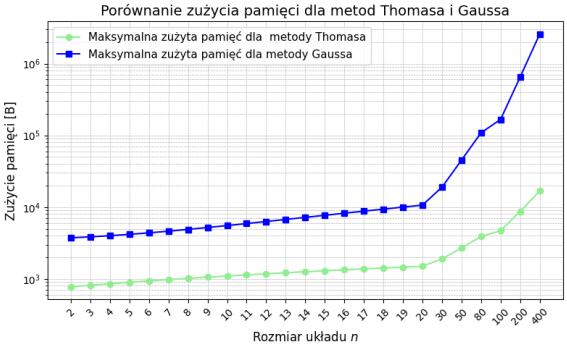
n	Thomas [B]	Gauss [B]
2	7.52×10^2	4.30×10^3
3	7.56×10^2	3.72×10^3
4	7.76×10^2	3.79×10^3
5	7.96×10^2	3.88×10^3
6	8.16×10^2	3.98×10^3
7	8.36×10^2	4.10×10^3
8	8.56×10^2	4.24×10^3
9	8.76×10^2	4.39×10^3
10	8.96×10^2	4.56×10^3
11	9.16×10^2	4.74×10^3
12	9.36×10^2	4.94×10^3
13	9.56×10^2	5.16×10^3
14	9.76×10^2	5.39×10^3
15	9.96×10^2	5.64×10^3
16	1.02×10^3	5.90×10^3
17	1.04×10^3	6.18×10^3
18	1.06×10^3	6.48×10^3
19	1.08×10^3	6.79×10^3
20	1.10×10^3	7.12×10^3
30	1.30×10^3	1.13×10^4
50	1.70×10^3	2.44×10^4

Tabela 7: Zużycie pamięci w zależności od wielkości macierzy A, precyzja double

n	Thomas [B]	Gauss [B]
2	7.76×10^2	4.25×10^3
3	8.16×10^2	3.85×10^3
4	8.56×10^2	3.99×10^3
5	8.96×10^2	4.17×10^3
6	9.36×10^2	4.38×10^3
7	9.76×10^2	4.62×10^3
8	1.02×10^3	4.89×10^3
9	1.06×10^3	5.19×10^3
10	1.10×10^3	5.53×10^3
11	1.14×10^3	5.90×10^3
12	1.18×10^3	6.30×10^3
13	1.22×10^3	6.73×10^3
14	1.26×10^3	7.19×10^3
15	1.30×10^3	7.69×10^3
16	1.34×10^3	8.22×10^3
17	1.38×10^3	8.78×10^3
18	1.42×10^3	9.37×10^3
19	1.46×10^3	9.99×10^3
20	1.50×10^3	1.06×10^4
30	1.90×10^3	1.90×10^4
50	2.70×10^3	4.52×10^4
80	3.90×10^3	1.09×10^5
100	4.70×10^3	1.67×10^5
200	8.70×10^3	6.50×10^5
400	1.68×10^4	2.58×10^6



a) Wykres dla precyzji float



b) Wykres dla precyzji double

Rysunek 2: Wykresy maksymalnej zużytej pamięci w zależności od n dla różnych precyzji

Tabela 8: Porównanie zużycia pamięci dla n=50

Precyzja	**Thomas [B]**	**Gauss [B]**	**Stosunek (Gauss/Thomas)**
Float	2.12×10^3	3.08×10^4	≈ 14.5
Double	2.92×10^3	4.10×10^4	≈ 19.3

- Metoda Thomasa wymaga przechowywania tylko trzech wektorów (a, b, c) o rozmiarze n, co skutkuje liniowym zużyciem pamięci $O(n)$
- Metoda Gaussa wymaga przechowywania całej macierzy $n \times n$, co daje kwadratowe zużycie pamięci $O(n^2)$
- Dla $n = 400$, metoda Thomasa zużywa około 13 kB pamięci, podczas gdy metoda Gaussa wymaga około 2.57 MB

4 Wnioski ogólne

1. **Przewaga algorytmu specjalizowanego:**
   - **Dokładność:** Metoda Thomasa i Gaussa osiągają błądy rzędu precyzji maszynowej.
   - **Wydajność:** Dla $n = 400$ metoda Thomasa jest 337 razy szybsza (złożoność $O(n)$ vs $O(n^3)$)
   - **Oszczędność pamięci:** Dla $n = 400$ metoda Thomasa zużywa 193 razy mniej pamięci (16.8 kB vs 2.58 MB)
2. **Wpływ rozmiaru układu:** Różnica w wydajności między metodami rośnie wykładniczo ze wzrostem n
3. **Wpływ precyzji:**
   - Obliczenia w double są nieoczekiwanie szybsze (o 10-13%) niż w float
   - Dokładność metody Thomasa odpowiada precyzji maszynowej (10^{-7} dla float, 10^{-16} dla double)
4. **Stabilność:** Obydwie metody zachowują wysoką dokładność nawet przy rosnącym współczynniku uwarunkowania macierzy. W przypadku Gaussa jest to możliwe ze względu na częściowy pivoting.