

Sprawozdanie nr 2

Rozwiązywanie równań liniowych

Seweryn Tasior, WI, grupa 5

17.06.2025

1 Wprowadzenie

Układy równań liniowych to fundamentalne narzędzie w matematyce obliczeniowej, wykorzystywane w modelowaniu fizycznym, analizie struktur i optymalizacji. Rozwiązuje się je metodami bezpośrednimi (eliminacja Gaussa, Thomas) lub iteracyjnymi (algorytm Jacobiego), różniącymi się złożonością obliczeniową i pamięciową. Wybór metody zależy od rozmiaru układu, struktury macierzy i wymaganej dokładności. Dzięki postępowi w algorytmach numerycznych możliwe jest efektywne rozwiązywanie nawet bardzo dużych układów, co jest kluczowe w nowoczesnych symulacjach i obliczeniach inżynierskich.

Celem eksperymentów było porównanie stabilności numerycznej, dokładności, szybkości działania oraz zużycia pamięci dla trzech metod rozwiązywania układów równań liniowych:

- eliminacji Gaussa,
- metody Thomasa,
- metody Jacobiego.

1.1 Warunki techniczne

Programy wykorzystane w eksperymentach zostały napisane w języku Python w wersji 3.11.5. Dodatkowo, do narysowania wykresów i tabel zostały użyte biblioteki Pandas i matplotlib. Pomocniczo do wykonywania obliczeń zastosowano funkcjonalności biblioteki Numpy. 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 ramach ćwiczenia przeprowadzono serię symulacji dla różnych rozmiarów macierzy. Dla każdego rozmiaru n , najpierw wyliczono wektor x jako dowolną n -elementową permutację ze zbioru $\{1, -1\}$. Następnie, zgodnie z później zdefiniowanymi wzorami testowymi (2.4), utworzono macierze A_k o rozmiarze $n \times n$. Kolejno, obliczono wektor prawej strony b jako iloczyn macierzy A i wektora x , tj. $b = A \cdot x$. Otrzymany wektor b wraz z macierzą A wykorzystano do obliczenia wektora rozwiązań y przy użyciu metody Gaussa, Thomasa oraz Jacobiego.

Wartości n zostały ustalone w zakresie: $n \in \{1, 2, \dots, 1000\}$.

2.1 Metoda Gaussa

W celu poprawy stabilności numerycznej zastosowano algorytm **częściowego pivotowania**. Polega on na tym, że w każdym i -tym kroku eliminacji algorytm wyszukuje w kolumnie i element o największej wartości bezwzględnej spośród elementów w wierszach od i do n . Następnie zamienia wiersz i z wierszem zawierającym ten największy element.

2.2 Metoda Thomasa

Do wyliczenia rozwiązania metodą Thomasa wykorzystano trzy wektory reprezentujące podprzekątną, przekątną i nadprzekątną. Na podstawie tych wektorów oraz wektora b , uzyskano wektor y .

2.3 Metoda Jacobiego

Zaimplementowano algorytm Jacobiego, wykorzystując bibliotekę NumPy dla optymalizacji obliczeń macierzowych. Wprowadzono szereg usprawnień umożliwiających precyzyjniejszy pomiar czasu wykonania iteracji oraz zwiększono limit maksymalnej liczby iteracji do 100000, co zapewnia możliwość zbadania zbieżności również dla trudniejszych przypadków układów równań.

Kolejno wyznaczono wartości promieni spektralnych dla różnych n , co umożliwiło znalezienie przypadków takich, które gwarantują zbieżność metody.

Podczas obliczeń, oprócz wektora rozwiązania y , wyliczono liczbę iteracji oraz średni czas wykonania pojedynczej iteracji dla dwóch różnych kryteriów zatrzymania: **kryterium przyrostowego** oraz **kryterium residualnego**.

Podczas korzystania z kryteriów stopu, użyto 3 różnych wartości tolerancji $\epsilon \in \{10^{-3}, 10^{-7}, 10^{-15}\}$

Użyto dwóch wektorów początkowych: wektora zerowego i wektora jako losową permutację $\{-100, 100\}$

2.4 Macierze testowe

W ramach ćwiczenia zaimplementowano kwadratowe macierze testowe A_k o rozmiarach $n \times n$, zdefiniowane następującymi wzorami:

2.4.1 Macierz A_1 : Macierz bardzo źle uwarunkowana

Macierz ta jest bardzo źle uwarunkowana, co czyni ją silnie podatną na błędy numeryczne podczas obliczeń. Została użyta w sekcji 3.1.1.

$$(A_1)_{i,j} = \begin{cases} 1 & \text{dla } i = j \\ \frac{1}{i+j-1} & \text{dla } i \neq j \end{cases}$$

2.4.2 Macierz A_2 : Macierz dobrze uwarunkowana

Macierz A_2 jest dobrze uwarunkowana, co zapewnia stabilność rozwiązania nawet dla dużych wartości n . Została użyta w sekcji 3.1.1.

$$(A_2)_{i,j} = \begin{cases} \frac{2i}{j} & \text{dla } j \geq i \\ (A_2)_{j,i} & \text{dla } j < i \end{cases}$$

—

2.4.3 Macierz A_3 : Macierz trójdzielna

Macierz A_3 została wykorzystana do porównania i analizy metod Thomasa i Gaussa. Została użyta w sekcjach 3.1.2, 3.2.1 i 3.3.1.

$$(A_3)_{i,j} = \begin{cases} -3i - 6 & \text{dla } j = i \\ i & \text{dla } j = i + 1 \\ \frac{i}{3} & \text{dla } j = i - 1, i > 1 \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$

2.4.4 Macierz A_4 : Macierz z dominującą przekątną

Macierz A_4 charakteryzuje się dominacją przekątną. Wykorzystano do analizy metody Jacobiego. Została użyta w sekcjach 3.1.3, 3.2.2 i 3.3.2.

$$(A_4)_{i,j} = \begin{cases} 8 & \text{dla } j = i \\ \frac{1}{|i-j|+5} & \text{dla } i \neq j \end{cases}$$

2.5 Zaburzenia wyników

2.5.1 Norma maksimum

W celu oceny zaburzeń rozwiązań układów, porównano wektory x (zadany) i y (otrzymany) za pomocą **normy maksimum**:

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

gdzie:

- x_i – i -ta współrzędna danego wektora \mathbf{x}
- y_i – i -ta współrzędna obliczonego wektora \mathbf{y}

2.5.2 Współczynnik uwarunkowania

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:

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

gdzie:

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

2.5.3 Promień spektralny

W metodzie Jacobiego kluczowe jest określenie zbieżności układu dla danego wymiaru macierzy n . Do oceny zbieżności metody wykorzystuje się promień spektralny macierzy iteracji. Aby wyznaczyć ten parametr, należy najpierw skonstruować macierz iteracji M_J , którą obliczono według wzoru:

$$M_J = -D^{-1}(A - D) \quad (1)$$

Gdzie:

- D – macierz diagonalna (zawierająca tylko elementy z przekątnej A),
- $A - D$ – macierz A bez elementów diagonalnych
- D^{-1} – macierz odwrotna do D

Na podstawie uzyskanych wyników, sporządzono wykresy, porównujące wyniki interpolacji. Wartości błędów średnich i maksymalnych zostały przedstawione w tabelach.

3 Wyniki i analiza

3.1 Dokładność rozwiązań

3.1.1 Wpływ złego uwarunkowania macierzy na dokładność przy metodzie Gaussa

Tabela 1: Błędy dla układu 2.4.1 w zależności od wielkości macierzy A_1 , precyzja float

| n | $\kappa(A)$ | $\sigma(x, y)$ |
|-----|-----------------------|-----------------------|
| 2 | 1.80×10^1 | 1.79×10^{-7} |
| 3 | 8.64×10^2 | 1.79×10^{-7} |
| 4 | 3.79×10^4 | 3.07×10^{-4} |
| 5 | 1.44×10^6 | 2.02×10^{-3} |
| 6 | 5.79×10^7 | 2.83×10^{-2} |
| 7 | 4.27×10^9 | 1.55×10^0 |
| 8 | 1.03×10^{10} | 5.79×10^0 |
| 9 | 3.23×10^{10} | 1.88×10^1 |
| 10 | 1.55×10^{10} | 1.11×10^0 |
| 12 | 1.26×10^{10} | 3.68×10^0 |
| 14 | 1.84×10^{10} | 7.97×10^0 |
| 16 | 6.39×10^{10} | 1.65×10^1 |
| 18 | 7.65×10^{10} | 2.41×10^1 |
| 20 | 1.71×10^{11} | 3.91×10^1 |

Tabela 2: Błędy dla układu 2.4.2 w zależności od wielkości macierzy A_2 , precyzja float

| n | $\kappa(A)$ | $\sigma(x, y)$ |
|-----|--------------------|-----------------------|
| 2 | 3.00×10^0 | 0.00×10^0 |
| 3 | 8.67×10^0 | 1.43×10^{-7} |
| 4 | 1.65×10^1 | 7.15×10^{-8} |
| 5 | 2.68×10^1 | 2.36×10^{-7} |
| 6 | 3.97×10^1 | 6.25×10^{-7} |
| 10 | 1.15×10^2 | 1.59×10^{-6} |
| 12 | 1.67×10^2 | 3.03×10^{-6} |
| 14 | 2.29×10^2 | 3.26×10^{-6} |
| 16 | 3.01×10^2 | 3.12×10^{-6} |
| 18 | 3.82×10^2 | 2.74×10^{-6} |
| 20 | 4.73×10^2 | 4.96×10^{-6} |
| 30 | 1.07×10^3 | 1.29×10^{-5} |
| 50 | 3.00×10^3 | 1.58×10^{-5} |
| 80 | 7.71×10^3 | 6.40×10^{-5} |
| 100 | 1.21×10^4 | 7.21×10^{-5} |
| 200 | 4.84×10^4 | 4.04×10^{-4} |

Dla **Macierzy A_1 (źle uwarunkowanej)**, błąd względny $\sigma(x, y)$ gwałtownie rośnie wraz ze wzrostem rozmiaru macierzy n . Już dla $n = 7$ błąd osiąga wartość rzędu 10^0 , co oznacza szybką utratę dokładności. Dla większych wartości n , uzyskane rozwiązanie staje się całkowicie niedokładne.

Natomiast dla **Macierzy A_2 (dobrze uwarunkowanej)**, błąd $\sigma(x, y)$ pozostaje na znacznie niższym poziomie (od 10^{-7} do 10^{-4}), nawet dla znacznie większych rozmiarów n . Wskaźnik uwarunkowania $\kappa(A)$ dla macierzy A_2 rośnie znacznie wolniej, co potwierdza jej stabilność numeryczną.

Podsumowując, **złe uwarunkowanie macierzy dramatycznie obniża dokładność numerycznych rozwiązań** uzyskanych metodą Gaussa, szczególnie przy ograniczonej precyzji float. Macierze dobrze uwarunkowane zapewniają natomiast satysfakcjonującą precyzję nawet dla większych rozmiarów macierzy.

3.1.2 Dokładność dla macierzy trójdzielnej

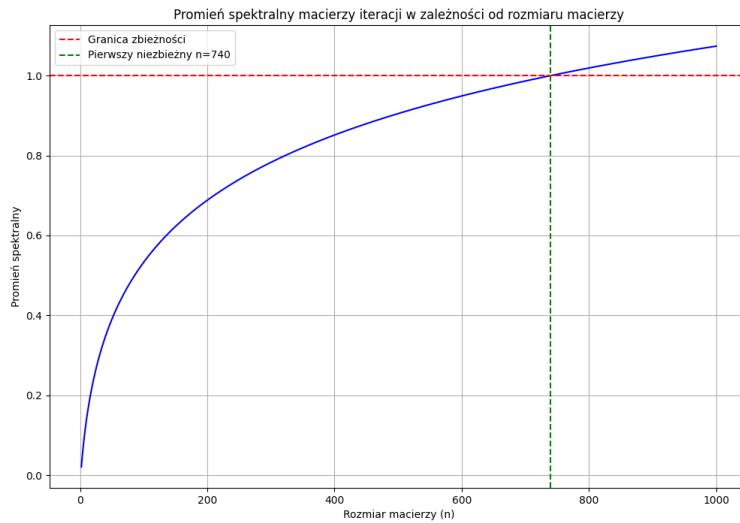
Tabela 3: Błędy dla układu 2.4.3 w zależności od wielkości macierzy A_3 , precyzja float

| 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 |
| 8 | 4.26×10^0 | 1.19×10^{-7} | 1.19×10^{-7} |
| 10 | 5.24×10^0 | 1.19×10^{-7} | 1.19×10^{-7} |
| 12 | 6.22×10^0 | 1.19×10^{-7} | 1.19×10^{-7} |
| 18 | 9.18×10^0 | 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} |

Dla macierzy A_3 , obie metody – Thomasa i Gaussa – wykazują niezwykle **wysoką i stabilną dokładność**. Względny błąd $\sigma(x, y)$ utrzymuje się na bardzo niskim poziomie, rzędu 10^{-7} , niezależnie od rozmiaru macierzy n (aż do $n = 50$). To świadczy o bardzo dobrym uwarunkowaniu macierzy A_3 oraz o efektywności obu algorytmów dla tego typu struktury.

3.1.3 Dokładność układu przy użyciu metody Jacobiego

Rysunek 1: Wykres wartości promienia spektralnego dla układu A_4



W metodzie Jacobiego dokładność jest związana ściśle z tolerancją. Tak więc możemy nią dowolnie manipulować, w zależności od pożądanej liczby iteracji. Kluczowym w stabilności numerycznej i zbieżności macierzy jest promień spektralny, którego wartość nie może przekraczać 1. Ta wartość, dla układu $A_4(2.4.4)$, została przekroczona przy rozmiarze $n = 740$.

Dokonano również dla danych kryterium stopu i wektora początkowego sprawdzenia zaburzeń rozwiązań układów z użyciem normy maksimum $\sigma(x, y)$. Jednak wyniki poniżej $n = 740$ były w dokładności na poziomie precyzji maszynowej, co sprawiło, że nie zostały przedstawione w sprawozdaniu.

3.2 Czas wykonania

3.2.1 Porównanie czasów wykonania dla metod Thomasa i Gaussa

Rysunek 2: Wykresy czasów wykonania w zależności od n dla precyzji float w macierzy A_3

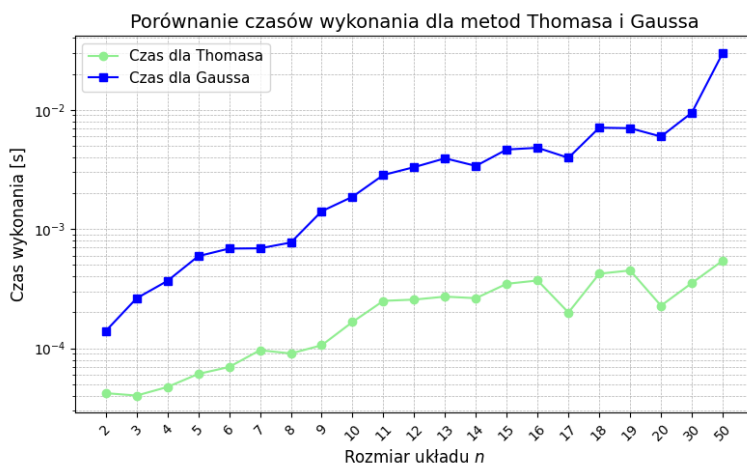


Tabela 4: Czas wykonania w zależności od wielkości macierzy A_3 , precyzja float

| 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} |
| 12 | 1.33×10^{-4} | 1.76×10^{-3} |
| 14 | 1.56×10^{-4} | 2.18×10^{-3} |
| 16 | 1.86×10^{-4} | 3.05×10^{-3} |
| 18 | 2.08×10^{-4} | 3.80×10^{-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 5: Porównanie czasów wykonania dla wybranych n

| n | Thomas [s] | Gauss [s] | Stosunek (Gauss/Thomas) |
|-----|-----------------------|-----------------------|-------------------------|
| 10 | 1.10×10^{-4} | 1.35×10^{-3} | ≈ 12 |
| 20 | 2.30×10^{-4} | 4.71×10^{-3} | ≈ 21 |
| 50 | 5.54×10^{-4} | 2.46×10^{-2} | ≈ 44 |

Zaobserwowano w tabelach, że metoda Thomasa jest znacznie szybsza niż metoda Gaussa. Czas wykonania dla **metody Thomasa** rośnie proporcjonalnie do n (złożoność rzędu $\mathcal{O}(n)$), co jest widoczne jako relatywnie liniowy wzrost na wykresie w skali logarytmicznej. Natomiast czas dla **metody Gaussa** rośnie znacząco szybciej, co odpowiada złożoności rzędu $\mathcal{O}(n^3)$ dla ogólnych macierzy. Ta dysproporcja staje się coraz widoczniejsza wraz ze wzrostem n .

3.2.2 Czas wykonania dla metody Jacobiego w zależności od kryterium stopu i wektora początkowego z wykorzystaniem układu A_4

• Kryterium Residualne

Tabela 6: Liczba iteracji dla kryterium residualnego w zależności od n dla zerowego wektora początkowego

| n | ϵ | | |
|-----|------------|-----------|------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 1 | 1 | 1 |
| 50 | 1 | 2 | 4 |
| 100 | 1 | 4 | 10 |
| 200 | 5 | 18 | 45 |
| 300 | 21 | 59 | 100000 |
| 350 | 19 | 66 | 100000 |
| 400 | 32 | 89 | 100000 |
| 600 | 65 | 241 | 100000 |
| 700 | 205 | 888 | 100000 |
| 739 | 18800 | 69500 | 100000 |

Tabela 7: Czas średni pojedynczej iteracji w sekundach [s] dla kryterium residualnego w zależności od n dla zerowego wektora początkowego

| n | ϵ | | |
|-----|-----------------------|-----------------------|-----------------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 0 | 0 | 0 |
| 50 | 0 | 8.52×10^{-5} | 0 |
| 100 | 0 | 0 | 0 |
| 200 | 2.00×10^{-4} | 1.69×10^{-5} | 1.78×10^{-5} |
| 300 | 0 | 4.32×10^{-5} | 2.96×10^{-5} |
| 350 | 4.76×10^{-5} | 3.22×10^{-5} | 3.71×10^{-5} |
| 400 | 4.60×10^{-5} | 4.60×10^{-5} | 4.27×10^{-5} |
| 600 | 7.35×10^{-5} | 7.56×10^{-5} | 8.50×10^{-5} |
| 700 | 2.44×10^{-4} | 2.49×10^{-4} | 2.23×10^{-4} |
| 739 | 1.77×10^{-4} | 1.84×10^{-4} | 1.86×10^{-4} |

Tabela 8: Liczba iteracji dla kryterium residualnego w zależności od n dla wektora początkowego z losowymi permutacjami $\{-100, 100\}$

| n | Tolerancja ϵ | | |
|-----|-----------------------|-----------|------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 2 | 6 | 16 |
| 50 | 10 | 20 | 39 |
| 100 | 15 | 29 | 60 |
| 200 | 25 | 50 | 103 |
| 300 | 44 | 81 | 100000 |
| 350 | 60 | 106 | 100000 |
| 400 | 70 | 127 | 100000 |
| 600 | 165 | 342 | 100000 |
| 700 | 886 | 1569 | 100000 |
| 739 | 61362 | 100000 | 100000 |

Tabela 9: Czas średni pojedynczej iteracji w sekundach [s] dla kryterium residualnego w zależności od n dla wektora początkowego z losowymi permutacjami $\{-100, 100\}$

| n | Tolerancja ϵ | | |
|-----|-----------------------|-----------------------|-----------------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 0 | 0 | 0 |
| 50 | 0 | 0 | 0 |
| 100 | 0 | 0 | 1.96×10^{-5} |
| 200 | 4.05×10^{-5} | 2.00×10^{-5} | 1.96×10^{-5} |
| 300 | 2.29×10^{-5} | 2.73×10^{-5} | 3.14×10^{-5} |
| 350 | 5.01×10^{-5} | 3.77×10^{-5} | 3.97×10^{-5} |
| 400 | 4.29×10^{-5} | 4.33×10^{-5} | 4.51×10^{-5} |
| 600 | 7.88×10^{-5} | 7.61×10^{-5} | 8.36×10^{-5} |
| 700 | 1.72×10^{-4} | 1.82×10^{-4} | 1.73×10^{-4} |
| 739 | 1.82×10^{-4} | 1.91×10^{-4} | 1.93×10^{-4} |

Powyższe tabel dla kryterium residualnego (metoda Jacobiego, macierz A_4) ujawniają, że liczba iteracji gwałtownie rośnie wraz z zaostrzeniem tolerancji ϵ oraz ze wzrostem rozmiaru macierzy n , zwłaszcza w pobliżu $n = 740$, gdzie zbieżność gwałtownie spada. Zauważalny jest także znaczący wpływ wektora początkowego: losowy wektor wymagał istotnie więcej iteracji niż zerowy. Średni czas pojedynczej iteracji rośnie z n , lecz jest niezależny od tolerancji, utrzymując się na niskim poziomie (rzędu $10^{-5} - 10^{-4}$ s).

Podsumowując, tolerancja i warunki początkowe determinują liczbę iteracji, natomiast rozmiar macierzy wpływa na czas pojedynczego kroku.

• Kryterium Przyrostowe

Tabela 10: Liczba iteracji dla kryterium residualnego w zależności od n dla zerowego wektora początkowego

| n | Tolerancja ϵ | | |
|-----|-----------------------|-----------|------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 1 | 1 | 1 |
| 50 | 1 | 1 | 3 |
| 100 | 1 | 3 | 9 |
| 200 | 3 | 15 | 40 |
| 300 | 12 | 50 | 125 |
| 350 | 9 | 55 | 148 |
| 400 | 19 | 77 | 191 |
| 600 | 25 | 201 | 554 |
| 700 | 51 | 734 | 2100 |
| 739 | 7330 | 58000 | 100000 |

Tabela 11: Czas średni pojedynczej iteracji w sekundach [s] dla kryterium residualnego w zależności od n dla zerowego wektora początkowego

| n | Tolerancja ϵ | | |
|-----|-----------------------|-----------------------|-----------------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 0 | 0 | 0 |
| 50 | 0 | 0 | 0 |
| 100 | 0 | 0 | 2.22×10^{-5} |
| 200 | 0 | 0 | 1.36×10^{-5} |
| 300 | 1.12×10^{-4} | 2.14×10^{-5} | 2.84×10^{-5} |
| 350 | 0 | 2.03×10^{-5} | 2.13×10^{-5} |
| 400 | 5.89×10^{-5} | 1.35×10^{-5} | 2.65×10^{-5} |
| 600 | 3.57×10^{-5} | 4.34×10^{-5} | 4.18×10^{-5} |
| 700 | 2.50×10^{-4} | 1.30×10^{-4} | 1.29×10^{-4} |
| 739 | 1.56×10^{-4} | 9.55×10^{-5} | 1.03×10^{-4} |

Tabela 12: Liczba iteracji dla kryterium residualnego w zależności od n dla wektora początkowego z losowymi permutacjami $\{-100, 100\}$

| n | Tolerancja ϵ | | |
|-----|-----------------------|-----------|------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 1 | 5 | 15 |
| 50 | 8 | 17 | 37 |
| 100 | 11 | 26 | 55 |
| 200 | 20 | 44 | 94 |
| 300 | 35 | 73 | 148 |
| 350 | 50 | 96 | 189 |
| 400 | 57 | 114 | 228 |
| 600 | 125 | 302 | 655 |
| 700 | 732 | 1415 | 2781 |
| 739 | 49914 | 100000 | 100000 |

Tabela 13: Czas średni pojedynczej iteracji w sekundach [s] dla kryterium residualnego w zależności od n dla wektora początkowego z losowymi permutacjami $\{-100, 100\}$

| n | Tolerancja ϵ | | |
|-----|-----------------------|-----------------------|-----------------------|
| | 10^{-3} | 10^{-7} | 10^{-15} |
| 10 | 0 | 0 | 0 |
| 50 | 1.34×10^{-4} | 0 | 0 |
| 100 | 0 | 3.88×10^{-5} | 0 |
| 200 | 0 | 0 | 1.08×10^{-5} |
| 300 | 3.53×10^{-5} | 6.92×10^{-6} | 2.03×10^{-5} |
| 350 | 3.96×10^{-5} | 2.29×10^{-5} | 2.65×10^{-5} |
| 400 | 3.52×10^{-5} | 1.75×10^{-5} | 2.63×10^{-5} |
| 600 | 4.80×10^{-5} | 4.36×10^{-5} | 5.48×10^{-5} |
| 700 | 9.00×10^{-5} | 9.67×10^{-5} | 9.15×10^{-5} |
| 739 | 9.57×10^{-5} | 9.66×10^{-5} | 1.01×10^{-4} |

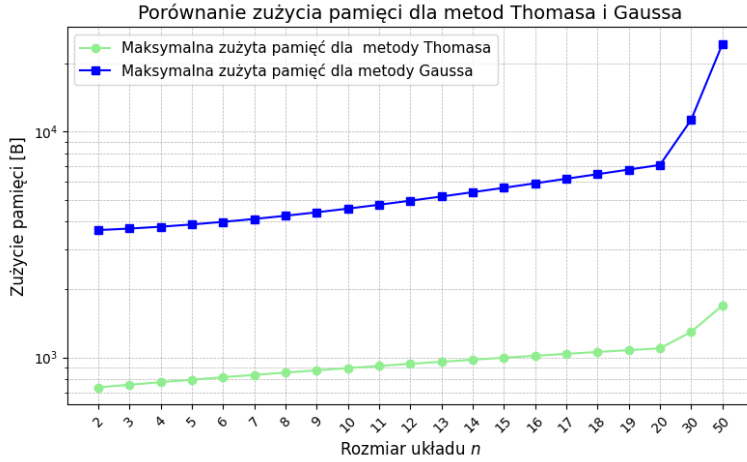
Analiza tabel dla kryterium przyrostowego, zastosowanego w metodzie Jacobiego dla macierzy A_4 , przedstawia podobne tendencje do kryterium residualnego. Obserwuje się, że liczba iteracji wymaganych do osiągnięcia zbieżności **rośnie wraz z zaostrzaniem tolerancji ϵ** oraz **wzrostem rozmiaru macierzy n** . Blisko $n = 740$, gdzie promień spektralny przekracza 1, liczba iteracji gwałtownie wzrasta, osiągając limit iteracji. Podobnie jak dla kryterium residualnego, **wybór wektora początkowego ma istotny wpływ na liczbę iteracji**; wektor początkowy z losowymi permutacjami wymagał większej liczby kroków niż zerowy.

Średni czas pojedynczej iteracji dla kryterium przyrostowego również **rośnie wraz z n** , co jest zgodne ze złożonością obliczeniową metody, ale pozostaje na niskim poziomie (rzędu $10^{-5} - 10^{-4}$ s). Wartości te są zbliżone do tych uzyskanych dla kryterium residualnego, co sugeruje, że koszt obliczeniowy pojedynczej iteracji jest głównie zależny od rozmiaru macierzy, a nie od wybranego kryterium stopu.

3.3 Zużycie pamięci

3.3.1 Porównanie metody Gaussa i Thomasa

Rysunek 3: Wykresy maksymalnej zużytej pamięci w zależności od n dla precyzji float w macierzy A_3



Rysunek 4: Zużycie pamięci w zależności od wielkości macierzy A_3 , 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 |
| 8 | 8.56×10^2 | 4.24×10^3 |
| 10 | 8.96×10^2 | 4.56×10^3 |
| 12 | 9.36×10^2 | 4.94×10^3 |
| 14 | 9.76×10^2 | 5.39×10^3 |
| 16 | 1.02×10^3 | 5.90×10^3 |
| 18 | 1.06×10^3 | 6.48×10^3 |
| 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 14: Porównanie zużycia pamięci

| n | Thomas [B] | Gauss [B] | Stosunek (Gauss/Thomas) |
|-----|--------------------|--------------------|-------------------------|
| 10 | 8.96×10^2 | 4.56×10^3 | ≈ 5 |
| 20 | 1.10×10^3 | 7.12×10^3 | ≈ 7 |
| 50 | 2.12×10^3 | 3.08×10^4 | ≈ 14.5 |

Metoda Thomasa jest znacznie mniej kosztowna pod względem ze względu na jej $\mathcal{O}(n)$ złożoność pamięciową (przechowywanie tylko przekątnych). Metoda Gaussa, operująca na pełnej macierzy, wykazuje złożoność $\mathcal{O}(n^2)$, co skutkuje znacznie szybszym wzrostem zużycia pamięci wraz z n .

Przykładowo, dla $n = 50$, metoda Gaussa zużywa niemal **14.5 razy więcej pamięci** niż metoda Thomasa (2.44×10^4 B vs 1.70×10^3 B). Podkreśla to przewagę metody Thomasa dla dużych macierzy trójdzielnych.

3.3.2 Metoda Jacobiego

Metoda Jacobiego wymaga przechowywania kilku wektorów o rozmiarze n , w tym wektora rozwiązania, wektora prawej strony oraz elementów macierzy, co dla pełnej macierzy współczynników A . Skutkuje to złożonością pamięciową rzędu $\mathcal{O}(n^2)$. W przypadku rzadkich macierzy możliwe jest znaczne ograniczenie zużycia pamięci poprzez zastosowanie odpowiedniej reprezentacji rzadkiej. Dodatkowe zużycie zależy również od liczby iteracji oraz ewentualnego buforowania pomocniczych obliczeń.

4 Wnioski

Przeprowadzone eksperymenty porównujące eliminację Gaussa, metodę Thomasa i metodę Jacobiego dostarczają kluczowych wniosków dotyczących ich dokładności, czasu wykonania i zużycia pamięci.

4.1 Dokładność rozwiązań

- **Uwarunkowanie macierzy:** Złe uwarunkowanie macierzy (A_1) drastycznie obniża dokładność metody Gaussa (tabela 1). Dla macierzy dobrze uwarunkowanych (A_2) dokładność metody Gaussa jest zadowalająca (tabela 1).
- **Macierze trójdzielne:** Metody Gaussa i Thomasa dla macierzy trójdzielnych (A_3) wykazują bardzo wysoką i stabilną dokładność (tabela 3).
- **Metoda Jacobiego:** Dokładność metody Jacobiego zależy od tolerancji ϵ i promienia spektralnego. Promień spektralny powyżej 1 (dla $n \geq 740$) powoduje brak zbieżności (rysunek 1). Wektor początkowy wpływa na liczbę iteracji (tabele 6, 8, 10, 12).

4.2 Czas wykonania

- **Metody bezpośrednie:** Metoda Thomasa jest znacznie szybsza ($\mathcal{O}(n)$) od metody Gaussa ($\mathcal{O}(n^2)$) dla trójdzielnych macierzy A_3 , co wyraźnie widać na wykresie 2 oraz w tabelach 4 i 5. Różnica wydajności rośnie z n .
- **Metoda Jacobiego:** Całkowity czas wykonania metody Jacobiego jest silnie zależny od liczby iteracji, która z kolei zależy od tolerancji ϵ , rozmiaru n oraz wektora początkowego (tabele 6 - 12). Średni czas pojedynczej iteracji rośnie wraz z n (tabele 7 - 13), ale jest niezależny od tolerancji.

4.3 Zużycie pamięci

- **Metody bezpośrednie:** Metoda Thomasa jest znacznie bardziej pamięciowooszczędna ($\mathcal{O}(n)$) niż metoda Gaussa ($\mathcal{O}(n^2)$) dla macierzy A_3 . Ta różnica jest krytyczna dla dużych macierzy, jak pokazano na wykresie 3 oraz w tabelach 4 i 14.
- **Metoda Jacobiego:** Wymaga $\mathcal{O}(n^2)$ pamięci dla pełnych macierzy. Dla macierzy rzadkich zużycie pamięci może być znacznie ograniczone poprzez zastosowanie odpowiednich reprezentacji.

4.4 Podsumowanie ogólne

- **Metoda Thomasa** jest idealna dla macierzy trójdzielnych ze względu na szybkość, dokładność i efektywność pamięciową.
- **Metoda Gaussa** jest uniwersalna, ale jej wydajność i dokładność są silnie zależne od uwarunkowania macierzy i rosną wraz z n .
- **Metoda Jacobiego** jest odpowiednia dla dużych, rzadkich macierzy z dominującą przekątną, wymagając weryfikacji zbieżności poprzez promień spektralny.