

NUM5

Łukasz Kowalik

1 Polecenie zadania

Zadanie numeryczne NUM5

Rozwiązać układ równań:

$$\begin{pmatrix} d & 0.5 & 0.1 & & & \\ 0.5 & d & 0.5 & 0.1 & & \\ 0.1 & 0.5 & d & 0.5 & 0.1 & \\ & \ddots & \ddots & \ddots & \ddots & \\ & & 0.1 & 0.5 & d & 0.5 \\ & & & 0.1 & 0.5 & d \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{N-1} \\ x_N \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \\ N \end{pmatrix}.$$

Dla $N = 200$, za pomocą metod Jacobiego i Gaussa-Seidela, gdzie d jest elementem diagonalnym.

Dla różnych wartości d i punktów startowych przedstaw graficznie różnice pomiędzy dokładnym rozwiązaniem a jego przybliżeniami w kolejnych iteracjach. Odpowiednio dobierając zakres parametrów, porównaj dwie metody. Czy procedura iteracyjna zawsze jest zbieżna?

2 Opis metod

Metody Jacobiego oraz Gaussa-Seidela to iteracyjne algorytmy numeryczne służące do rozwiązywania układów równań liniowych. Oznacza to, że rozwiązanie dokładne jest osiągane w granicach błędu zaokrąglenia, przy odpowiednio dużej liczbie iteracji. W praktyce iteracyjne metody są skuteczne, gdy układ równań zbiega się w stosunkowo niewielkiej liczbie kroków.

Metody te są szczególnie przydatne dla macierzy rzadkich, takich jak macierz opisana w treści zadania, gdzie większość elementów jest zerowa, co pozwala na uniknięcie zbędnych obliczeń i znacząco przyspiesza działanie algorytmu.

2.1 Wzór Jacobiego

W metodzie Jacobiego obliczenia dla $x_i^{(k+1)}$ w k -tej iteracji opierają się wyłącznie na wartościach $x_j^{(k)}$ z poprzedniej iteracji:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^N a_{ij} x_j^{(k)}}{a_{ii}},$$

gdzie:

- $a_{ii} = d$ to element diagonalny macierzy,
- b_i to element wektora prawej strony,
- k to indeks iteracji.

2.2 Wzór Gaussa-Seidela

W metodzie Gaussa-Seidela wartości $x_i^{(k+1)}$ są obliczane sekwencyjnie, wykorzystując już zaktualizowane wartości z bieżącej iteracji. Wzór ma postać:

$$x_i^{(k+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^N a_{ij} x_j^{(k)}}{a_{ii}}.$$

2.3 Porównanie obu metod

Obie metody różnią się sposobem wykorzystania wyników:

- W metodzie Jacobiego wszystkie wartości $x_i^{(k+1)}$ są obliczane niezależnie od siebie i używają wyników $x_j^{(k)}$ z poprzedniej iteracji.
- W metodzie Gaussa-Seidela każda wartość $x_i^{(k+1)}$ natychmiast wykorzystuje obliczone już wartości $x_j^{(k+1)}$ z bieżącej iteracji, co zazwyczaj przyspiesza zbieżność.

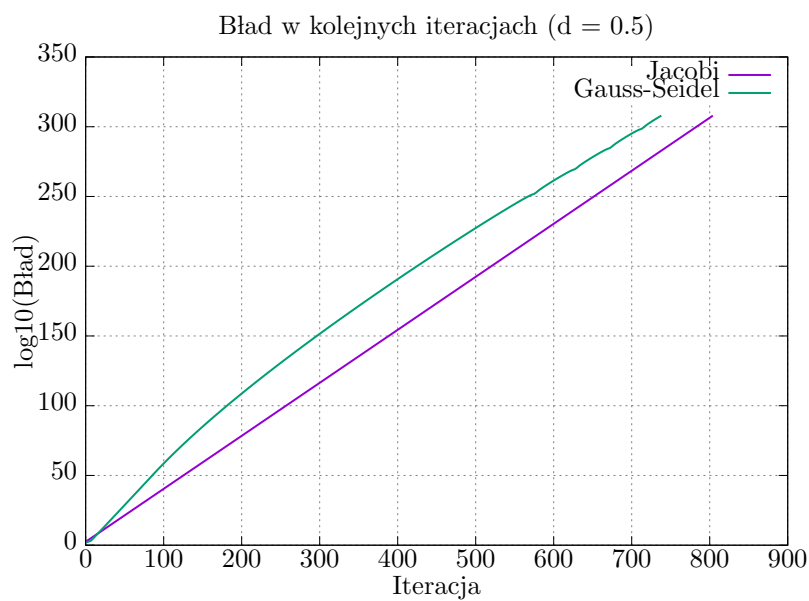
2.4 Specyfika macierzy w zadaniu

Macierz opisana w zadaniu jest rzadką macierzą z pięcioma diagonalami, co oznacza, że większość jej elementów wynosi zero. Dzięki temu podczas implementacji można uniknąć zbędnych obliczeń związanych z mnożeniem przez zera, co znacząco zwiększa efektywność obu metod.

3 Wyniki dla $d = 0.5$ i różnych punktów startowych

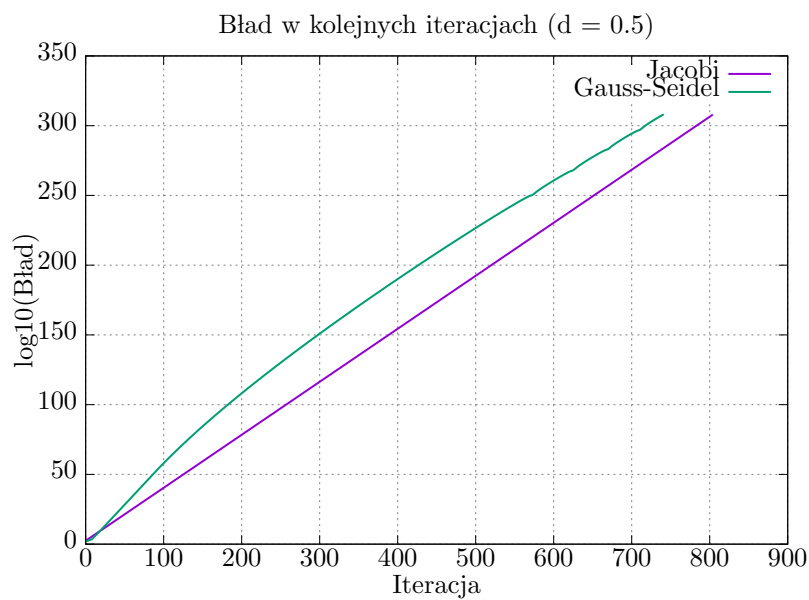
Dla wartości $d = 0.5$ przeprowadzono analizę zbieżności metod Jacobiego oraz Gaussa-Seidela, stosując różne punkty startowe. Wyniki przedstawiono w postaci wykresów ilustrujących logarytmiczny błąd w kolejnych iteracjach.

3.1 Punkt startowy: -1.0



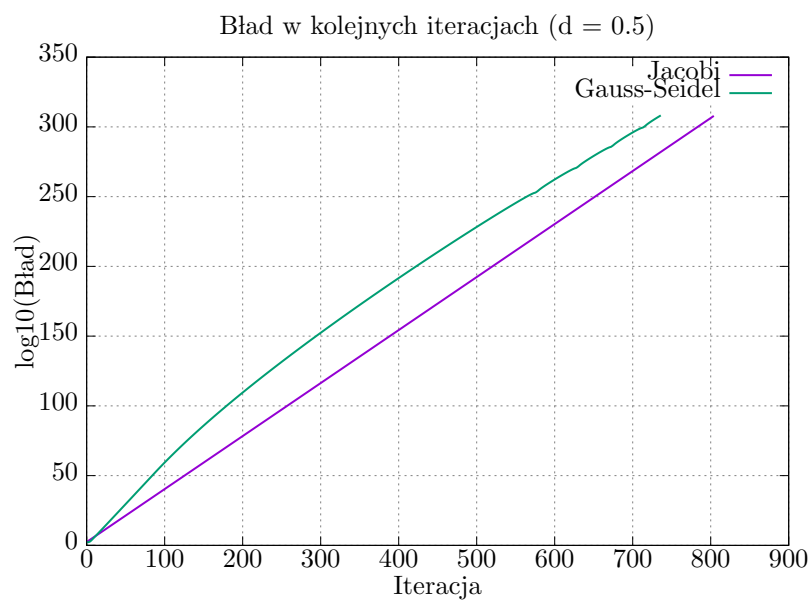
Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 0.5$, punkt startowy -1.0 .

3.2 Punkt startowy: 0.0



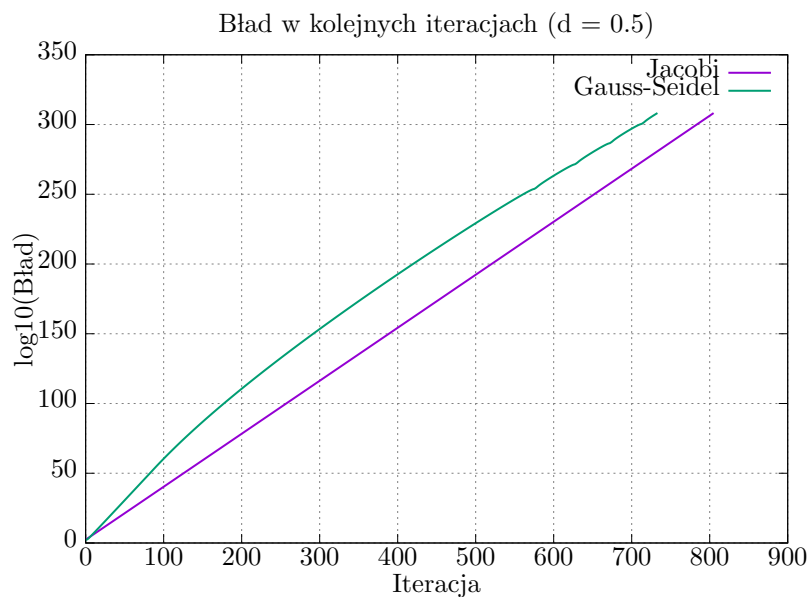
Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 0.5$, punkt startowy 0.0.

3.3 Punkt startowy: 10.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 0.5$, punkt startowy 10.0.

3.4 Punkt startowy: 100.0

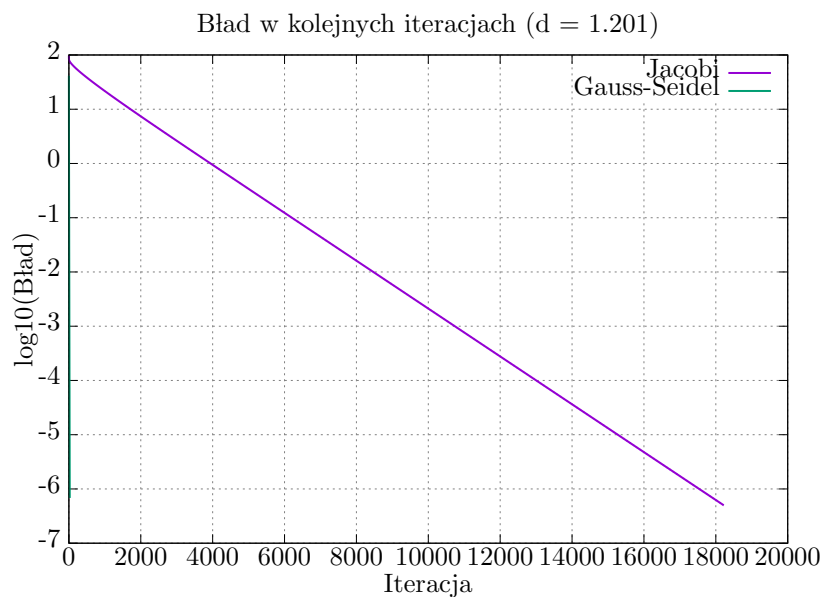


Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 0.5$, punkt startowy 100.0.

4 Wyniki dla $d = 1.201$ i różnych punktów startowych

Dla wartości $d = 1.201$ przeanalizowano zbieżność metod Jacobiego oraz Gaussa-Seidela, stosując różne punkty startowe. Wyniki przedstawiono w postaci wykresów logarytmicznego błędu w kolejnych iteracjach.

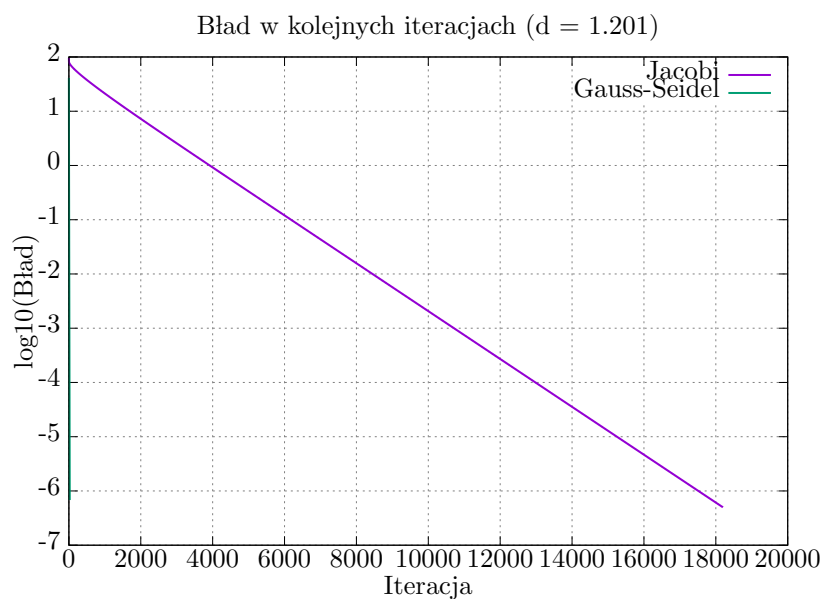
4.1 Punkt startowy: -1.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.201$, punkt startowy -1.0 .

Dla punktu startowego -1.0 metoda Gaussa-Seidela zbiega szybciej niż metoda Jacobiego. Wartość $d = 1.201$ sprawia, że układ pozostaje przekatniowo dominujący, co umożliwia zbieżność obu metod, jednak z różnicami w szybkości.

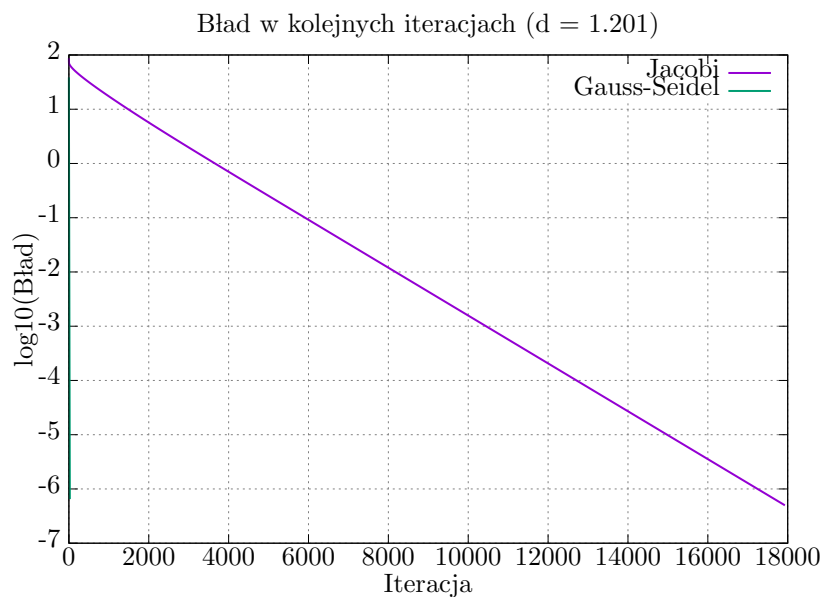
4.2 Punkt startowy: 0.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.201$, punkt startowy 0.0.

W przypadku punktu startowego 0.0 metoda Gaussa-Seidela osiąga mniejszy błąd w mniejszej liczbie iteracji w porównaniu do Jacobiego. Zbieżność obu metod jest gwarantowana dzięki dominacji przekątnej macierzy.

4.3 Punkt startowy: 10.0

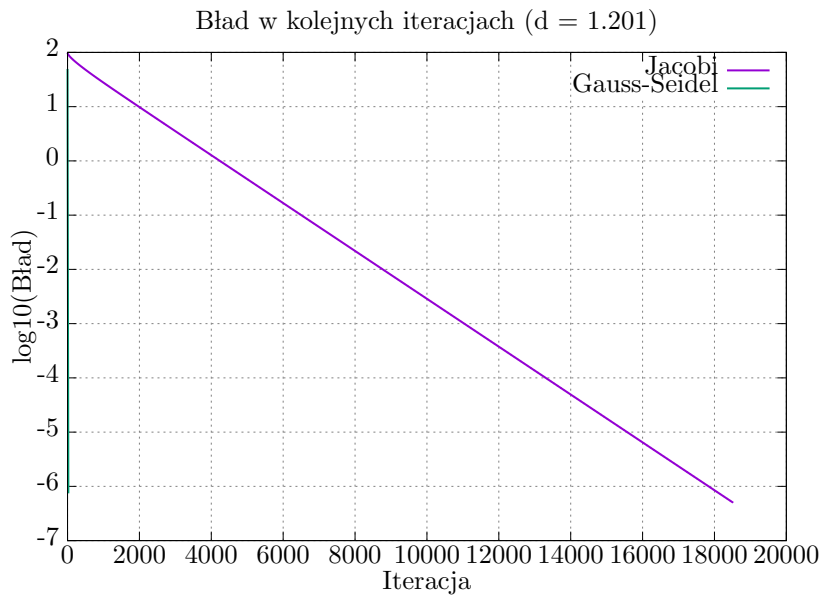


Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.201$, punkt startowy 10.0.

Przy punkcie startowym 10.0 metoda Gaussa-Seidela szybko redukuje błąd, natomiast metoda Jacobiego wymaga większej liczby iteracji, aby osiągnąć zadany poziom dokładności. Różnica w efektywności staje się bardziej widoczna w przypadku większych punktów startowych.

4.4 Punkt startowy: 100.0

Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.201$, punkt startowy 100.0.

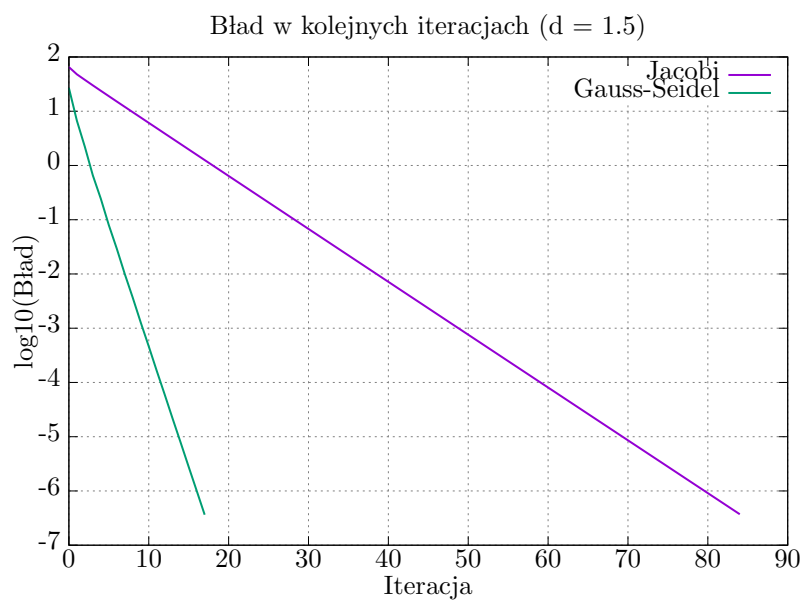


Przy bardzo dużym punkcie startowym, 100.0, metoda Gaussa-Seidela ponownie potwierdza swoją wyższość w szybkości zbieżności. Metoda Jacobiego, choć wolniejsza, osiąga podobny poziom dokładności po znacznie większej liczbie iteracji.

5 Wyniki dla $d = 1.5$ i różnych punktów startowych

Dla wartości $d = 1.5$ przeprowadzono analizę zbieżności metod Jacobiego oraz Gaussa-Seidela, stosując różne punkty startowe. Wyniki przedstawiono w postaci wykresów logarytmicznego błędów w kolejnych iteracjach.

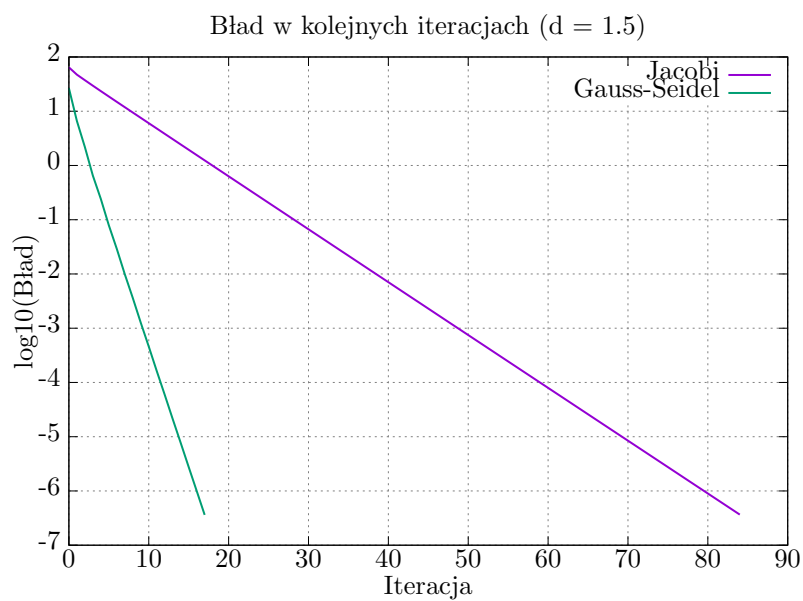
5.1 Punkt startowy: -1.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.5$, punkt startowy -1.0 .

Dla punktu startowego -1.0 metoda Gaussa-Seidela charakteryzuje się szybszą zbieżnością w porównaniu do metody Jacobiego. Wartość $d = 1.5$ wzmacnia dominację przekatniową macierzy, co sprzyja stabilności iteracji.

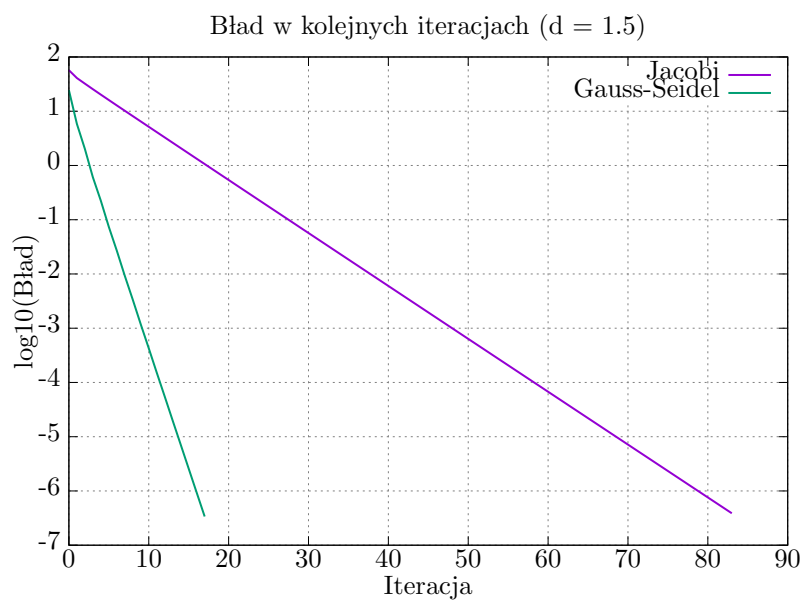
5.2 Punkt startowy: 0.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.5$, punkt startowy 0.0.

W przypadku punktu startowego 0.0 metoda Gaussa-Seidela szybko redukuje błąd do poziomu poniżej tolerancji, podczas gdy metoda Jacobiego wymaga większej liczby iteracji. Obie metody wykazują jednak stabilną zbieżność.

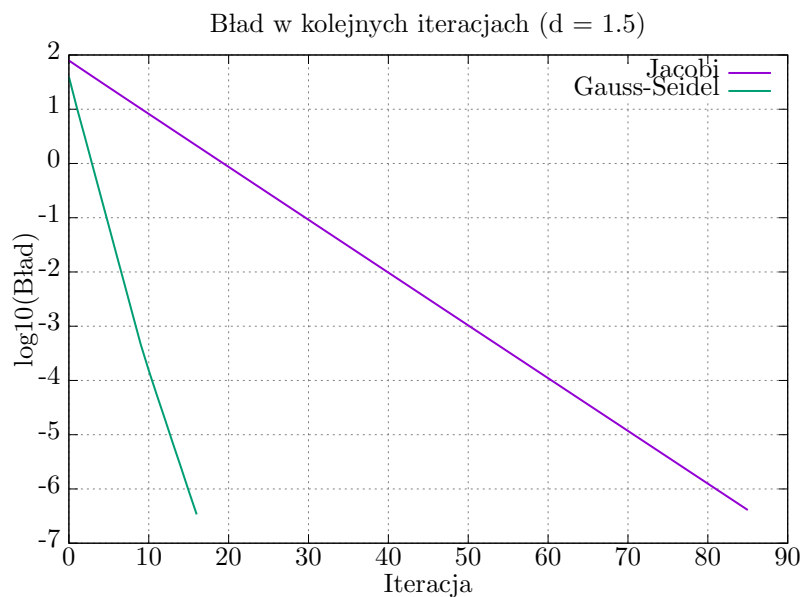
5.3 Punkt startowy: 10.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.5$, punkt startowy 10.0.

Dla większego punktu startowego, 10.0, metoda Gaussa-Seidela wykazuje wyraźną przewagę w szybkości zbieżności. Z kolei metoda Jacobiego potrzebuje znacznie większej liczby iteracji, aby osiągnąć podobny poziom dokładności.

5.4 Punkt startowy: 100.0



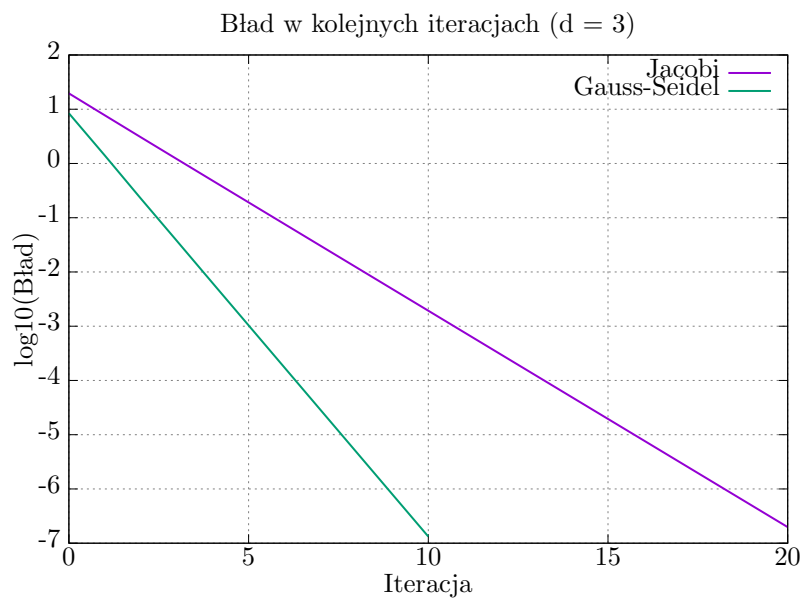
Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 1.5$, punkt startowy 100.0.

Przy bardzo dużym punkcie startowym, 100.0, metoda Gaussa-Seidela wykazuje znaczną przewagę nad metodą Jacobiego, zarówno pod względem szybkości, jak i stabilności iteracji. Metoda Jacobiego jest mniej efektywna przy takich warunkach początkowych.

6 Wyniki dla $d = 3.0$ i różnych punktów startowych

Dla wartości $d = 3.0$ przeprowadzono analizę zbieżności metod Jacobiego oraz Gaussa-Seidela, stosując różne punkty startowe. Wyniki przedstawiono w postaci wykresów logarytmicznego błędów w kolejnych iteracjach.

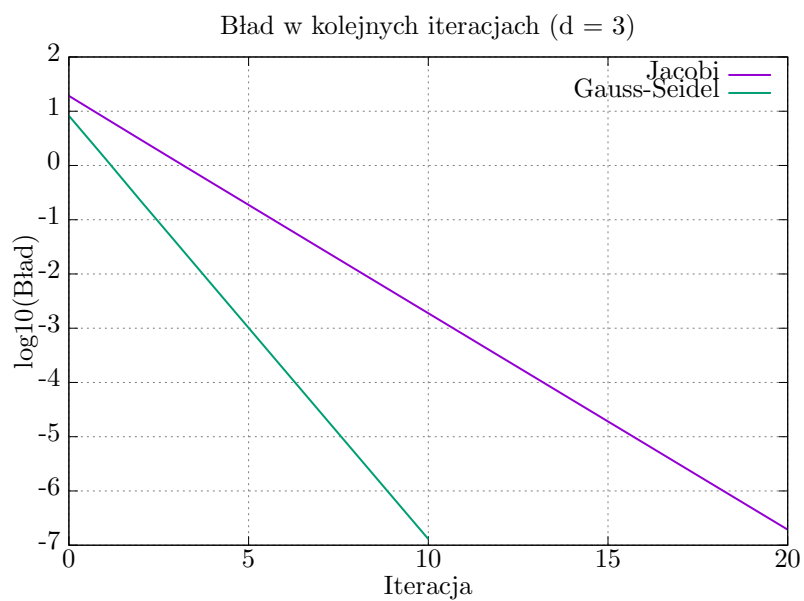
6.1 Punkt startowy: -1.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 3.0$, punkt startowy -1.0 .

Przy punkcie startowym -1.0 obie metody wykazują stabilną zbieżność, jednak metoda Gaussa-Seidela osiąga znacznie mniejszy błąd w krótszym czasie. Wyższa wartość d znacząco poprawia dominację przekatniową, co pozytywnie wpływa na szybkość iteracji.

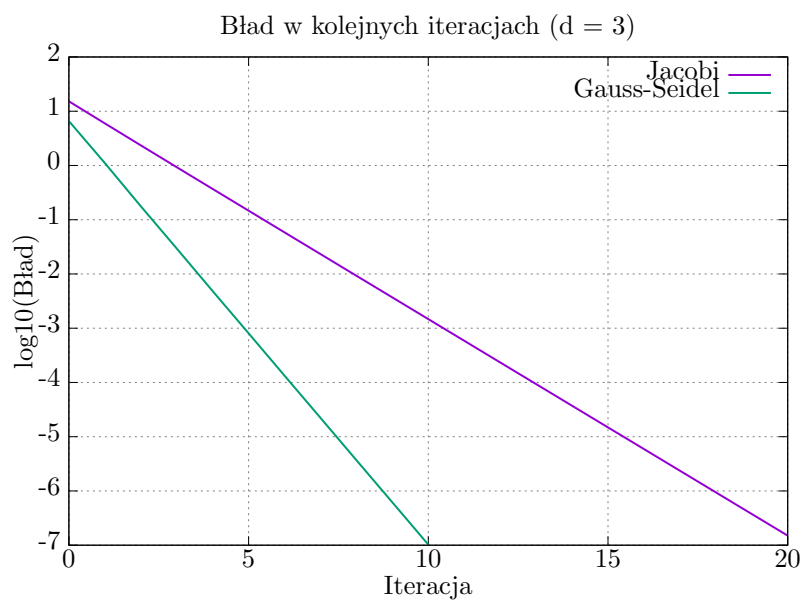
6.2 Punkt startowy: 0.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 3.0$, punkt startowy 0.0.

Dla punktu startowego 0.0 metoda Gaussa-Seidela ponownie charakteryzuje się znacznie szybszą zbieżnością niż metoda Jacobiego. Jest to widoczne szczególnie w początkowych iteracjach, gdzie redukcja błędów jest bardziej dynamiczna.

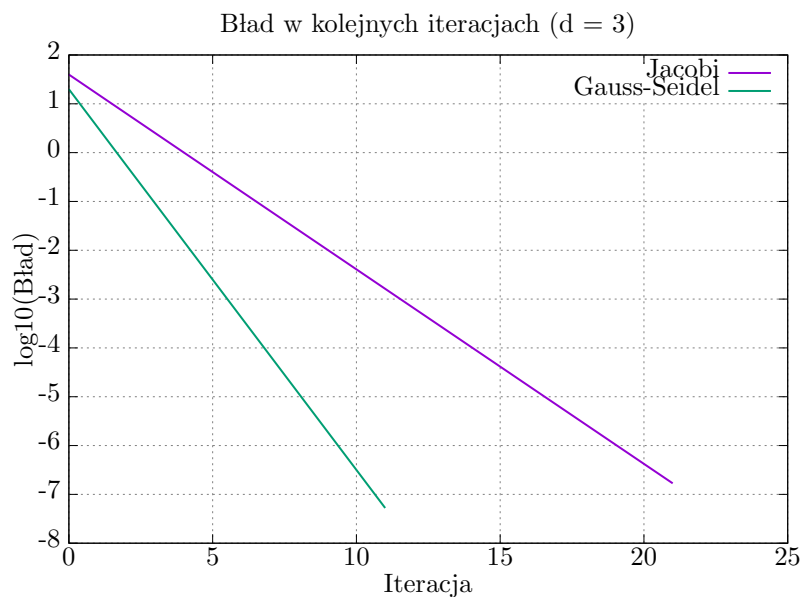
6.3 Punkt startowy: 10.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 3.0$, punkt startowy 10.0.

Dla punktu startowego 10.0 metoda Gaussa-Seidela potwierdza swoją przewagę w redukcji błędu. W metodzie Jacobiego zbieżność jest zauważalnie wolniejsza, co wskazuje na jej ograniczenia w przypadku większych wartości początkowych.

6.4 Punkt startowy: 100.0



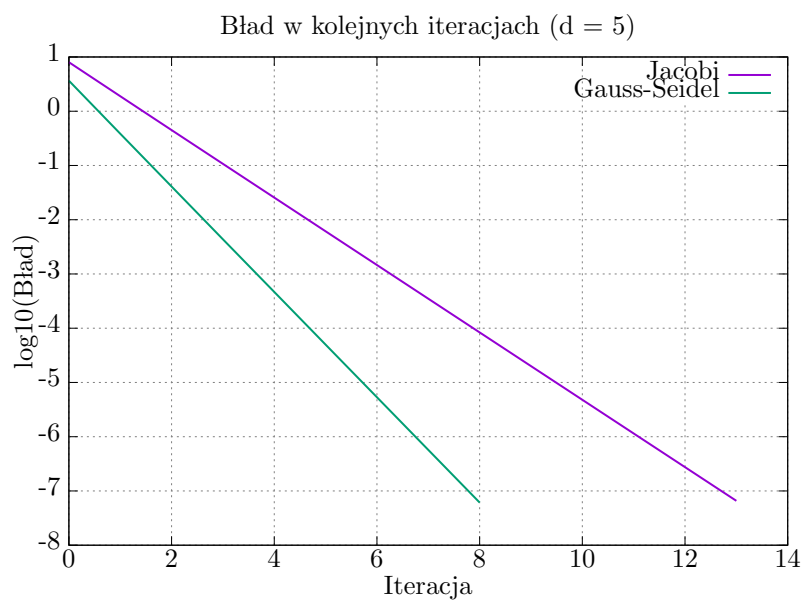
Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 3.0$, punkt startowy 100.0.

Przy bardzo dużym punkcie startowym, 100.0, metoda Gaussa-Seidela skutecznie redukuje błąd w stosunkowo krótkim czasie. Metoda Jacobiego, choć stabilna, wymaga znacznie większej liczby iteracji, aby osiągnąć porównywalny poziom dokładności.

7 Wyniki dla $d = 5.0$ i różnych punktów startowych

Dla wartości $d = 5.0$ przeprowadzono analizę zbieżności metod Jacobiego oraz Gaussa-Seidela, stosując różne punkty startowe. Wyniki przedstawiono w postaci wykresów logarytmicznego błędów w kolejnych iteracjach.

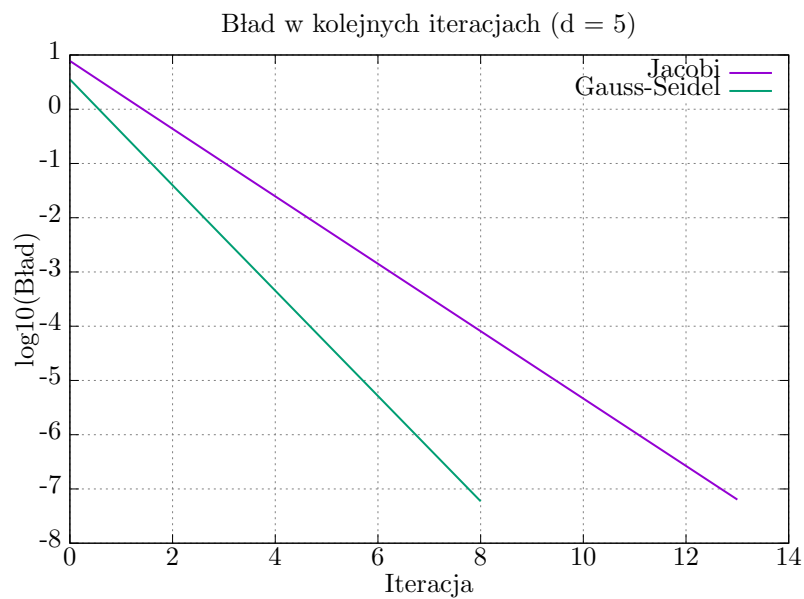
7.1 Punkt startowy: -1.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 5.0$, punkt startowy -1.0 .

Przy punkcie startowym -1.0 obie metody wykazują bardzo dobrą zbieżność, jednak metoda Gaussa-Seidela osiąga wymagany poziom błędu znacznie szybciej. Duża wartość $d = 5.0$ dodatkowo poprawia stabilność obliczeń i wzmacnia dominację przekatniową macierzy.

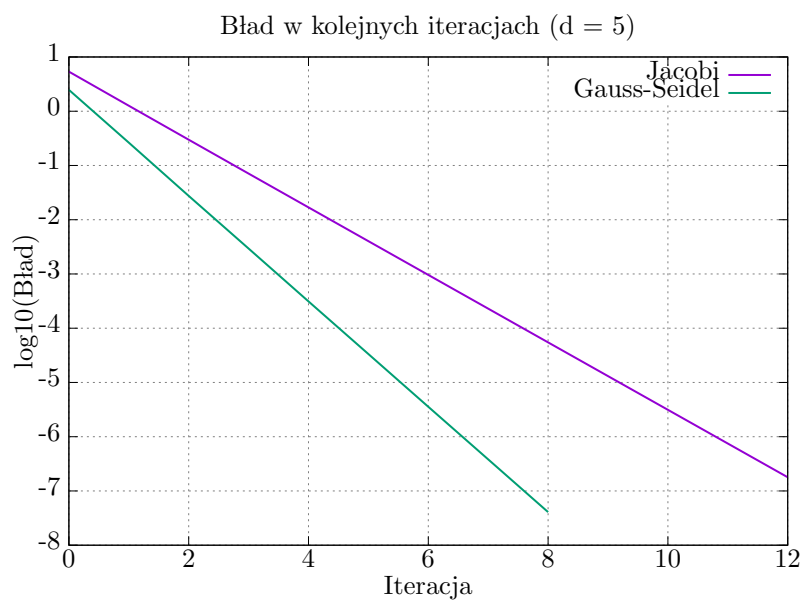
7.2 Punkt startowy: 0.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 5.0$, punkt startowy 0.0.

Dla punktu startowego 0.0 metoda Gaussa-Seidela zachowuje swoją przewagę w szybkości zbieżności. Metoda Jacobiego, mimo stabilności, wymaga większej liczby iteracji, aby osiągnąć porównywalny poziom błędu.

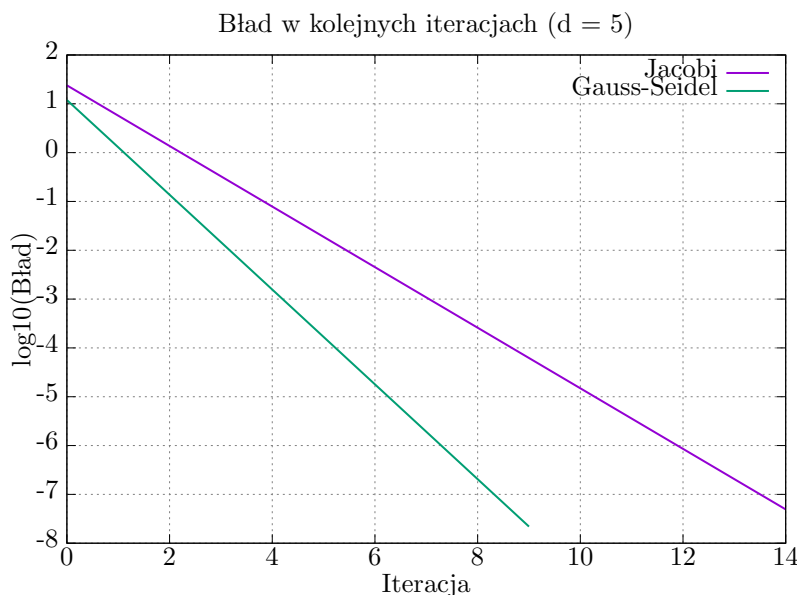
7.3 Punkt startowy: 10.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 5.0$, punkt startowy 10.0.

Dla punktu startowego 10.0 widoczna jest znaczna przewaga metody Gaussa-Seidela w szybkości redukcji błędu. Metoda Jacobiego wymaga więcej iteracji, co potwierdza jej ograniczenia przy większych wartościach początkowych.

7.4 Punkt startowy: 100.0



Zbieżność metod Jacobiego i Gaussa-Seidela dla $d = 5.0$, punkt startowy 100.0.

Przy bardzo dużym punkcie startowym, 100.0, metoda Gaussa-Seidela wyraźnie przewyższa metodę Jacobiego pod względem szybkości i efektywności. Metoda Jacobiego, choć stabilna, wymaga znacznie większej liczby iteracji, aby osiągnąć podobny poziom dokładności.

8 Analiza wyników dla $d = 0.5$

Dla wartości $d = 0.5$ przeprowadzono testy metod Jacobiego i Gaussa-Seidela, stosując różne punkty startowe. Jednakże, dla tej wartości d zarówno metoda Jacobiego, jak i Gaussa-Seidela nie zbiegały do poprawnego rozwiązania. Wyniki zawierają czas wykonania, błąd końcowy oraz liczbę iteracji, jednak warto zauważyć, że rozwiązania były niepoprawne ($-nan$).

8.1 Rozwiązanie dokładne

Eigen LU : $[141.248, -131.718, -37.6535, 160.614, -67.4623, \dots, 242.647, 161.137, -42.7842, 184.258, 224.298]$

8.2 Punkt startowy: -1.0

Metoda Jacobiego:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0281047 s
- Błąd końcowy: 0

- Liczba iteracji: 807

Metoda Gaussa-Seidela:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0222791 s
- Błąd końcowy: 0
- Liczba iteracji: 741

8.3 Punkt startowy: 0.0

Metoda Jacobiego:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0304998 s
- Błąd końcowy: 0
- Liczba iteracji: 807

Metoda Gaussa-Seidela:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.023265 s
- Błąd końcowy: 0
- Liczba iteracji: 744

8.4 Punkt startowy: 10.0

Metoda Jacobiego:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0306459 s
- Błąd końcowy: 0
- Liczba iteracji: 807

Metoda Gaussa-Seidela:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0224078 s
- Błąd końcowy: 0
- Liczba iteracji: 739

8.5 Punkt startowy: 100.0

Metoda Jacobiego:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0290605 s
- Błąd końcowy: 0
- Liczba iteracji: 808

Metoda Gaussa-Seidela:

- Rozwiązanie: $[-nan, -nan, -nan, -nan, -nan, \dots, -nan, -nan, -nan, -nan, -nan]$
- Czas wykonania: 0.0233394 s
- Błąd końcowy: 0
- Liczba iteracji: 736

8.6 Podsumowanie wyników

Dla $d = 0.5$ metody iteracyjne Jacobiego i Gaussa-Seidela nie były w stanie znaleźć poprawnego rozwiązania, a wszystkie wartości rozwiązania przyjęły wartość $-nan$. Wynika to z braku dominacji przekatniowej macierzy, co powoduje, że metody iteracyjne są niestabilne. W takich przypadkach należy zastosować inne metody numeryczne lub zmodyfikować macierz, aby osiągnąć dominację przekatniową.

9 Analiza wyników dla $d = 1.201$

9.1 Rozwiązanie dokładne

Eigen LU : $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$

9.2 Punkt startowy: -1.0

Metoda Jacobiego:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.620068 s
- Błąd końcowy: 4.99573×10^{-7}
- Liczba iteracji: 18222

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.000766206 s
- Błąd końcowy: 6.82956×10^{-7}
- Liczba iteracji: 25

9.3 Punkt startowy: 0.0

Metoda Jacobiego:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.600652 s
- Błąd końcowy: 4.9945×10^{-7}
- Liczba iteracji: 18199

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.000812242 s
- Błąd końcowy: 6.79915×10^{-7}
- Liczba iteracji: 25

9.4 Punkt startowy: 10.0

Metoda Jacobiego:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.580617 s
- Błąd końcowy: 4.99584×10^{-7}
- Liczba iteracji: 17930

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.000851245 s
- Błąd końcowy: 6.49501×10^{-7}
- Liczba iteracji: 25

9.5 Punkt startowy: 100.0

Metoda Jacobiego:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.616096 s
- Błąd końcowy: 4.9964×10^{-7}
- Liczba iteracji: 18522

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.374034, 0.852627, 1.24472, 1.66617, 2.08286, \dots, 81.1547, 82.7847, 83.8168, 69.5365, 130.6]$
- Czas wykonania: 0.000782075 s
- Błąd końcowy: 7.44474×10^{-7}
- Liczba iteracji: 24

10 Analiza wyników dla $d = 1.5$

10.1 Rozwiązanie dokładne

Eigen LU : $[0.34252, 0.750479, 1.1098, 1.48121, 1.85205, \dots, 72.4267, 73.5377, 72.8664, 68.2314, 105.732]$

10.2 Punkt startowy: -1.0

Metoda Jacobiego:

- Rozwiązanie: $[0.34252, 0.750479, 1.1098, 1.48121, 1.85205, \dots, 72.4267, 73.5377, 72.8664, 68.2314, 105.732]$
- Czas wykonania: 0.00340229 s
- Błąd końcowy: 3.72182×10^{-7}
- Liczba iteracji: 85

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.34252, 0.750479, 1.1098, 1.48121, 1.85205, \dots, 72.4267, 73.5377, 72.8664, 68.2314, 105.732]$
- Czas wykonania: 0.000685254 s
- Błąd końcowy: 3.6464×10^{-7}
- Liczba iteracji: 18

10.3 Punkt startowy: 0.0

Metoda Jacobiego:

- Rozwiązanie: $[0.34252, 0.750479, 1.1098, 1.48121, 1.85205, \dots, 72.4267, 73.5377, 72.8664, 68.2314, 105.732]$
- Czas wykonania: 0.0057583 s
- Błąd końcowy: 3.66512×10^{-7}
- Liczba iteracji: 85

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.34252, 0.750479, 1.1098, 1.48121, 1.85205, \dots, 72.4267, 73.5377, 72.8664, 68.2314, 105.732]$

- Czas wykonania: 0.000767619 s
- Błąd końcowy: 3.6223×10^{-7}
- Liczba iteracji: 18

10.4 Punkt startowy: 10.0

Metoda Jacobiego:

- Rozwiązanie: [0.34252, 0.750479, 1.1098, 1.48121, 1.85205, ..., 72.4267, 73.5377, 72.8664, 68.2314, 105.732]
- Czas wykonania: 0.00806504 s
- Błąd końcowy: 3.87853×10^{-7}
- Liczba iteracji: 84

Metoda Gaussa-Seidela:

- Rozwiązanie: [0.34252, 0.750479, 1.1098, 1.48121, 1.85205, ..., 72.4267, 73.5377, 72.8664, 68.2314, 105.732]
- Czas wykonania: 0.000902711 s
- Błąd końcowy: 3.38125×10^{-7}
- Liczba iteracji: 18

10.5 Punkt startowy: 100.0

Metoda Jacobiego:

- Rozwiązanie: [0.34252, 0.750479, 1.1098, 1.48121, 1.85205, ..., 72.4267, 73.5377, 72.8664, 68.2314, 105.732]
- Czas wykonania: 0.00342478 s
- Błąd końcowy: 4.09053×10^{-7}
- Liczba iteracji: 86

Metoda Gaussa-Seidela:

- Rozwiązanie: [0.34252, 0.750479, 1.1098, 1.48121, 1.85205, ..., 72.4267, 73.5377, 72.8664, 68.2314, 105.732]
- Czas wykonania: 0.000657943 s
- Błąd końcowy: 3.39822×10^{-7}
- Liczba iteracji: 17

11 Analiza wyników dla $d = 5$

11.1 Rozwiązanie dokładne

Eigen LU : [0.229931, 0.477544, 0.714341, 0.952325, 1.19048, ..., 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]

11.2 Punkt startowy: -1.0

Metoda Jacobiego:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.00109817 s
- Błąd końcowy: 1.97896×10^{-7}
- Liczba iteracji: 21

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.000498343 s
- Błąd końcowy: 1.32573×10^{-7}
- Liczba iteracji: 11

11.3 Punkt startowy: 0.0

Metoda Jacobiego:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.00178054 s
- Błąd końcowy: 1.93548×10^{-7}
- Liczba iteracji: 21

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.00117186 s
- Błąd końcowy: 1.29827×10^{-7}
- Liczba iteracji: 11

11.4 Punkt startowy: 10.0

Metoda Jacobiego:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.00112125 s
- Błąd końcowy: 1.50068×10^{-7}
- Liczba iteracji: 21

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.000531255 s
- Błąd końcowy: 1.02501×10^{-7}
- Liczba iteracji: 11

11.5 Punkt startowy: 100.0**Metoda Jacobiego:**

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.00113125 s
- Błąd końcowy: 1.68414×10^{-7}
- Liczba iteracji: 22

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.229931, 0.477544, 0.714341, 0.952325, 1.19048, \dots, 46.6693, 46.9589, 46.8056, 47.4336, 57.2009]$
- Czas wykonania: 0.000594584 s
- Błąd końcowy: 5.26597×10^{-8}
- Liczba iteracji: 12

12 Analiza wyników dla $d = 5.0$

Dla wartości $d = 5.0$ przeprowadzono testy metod Jacobiego i Gaussa-Seidela, stosując różne punkty startowe. Wyniki zawierają czas wykonania, błąd końcowy oraz liczbę iteracji potrzebnych do osiągnięcia zbieżności. Rozwiązanie dokładne obliczono przy użyciu dekompozycji LU z wykorzystaniem biblioteki Eigen.

12.1 Rozwiązanie dokładne

Eigen LU : $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$

12.2 Punkt startowy: -1.0 **Metoda Jacobiego:**

- Rozwiązanie: $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$
- Czas wykonania: 0.000590537 s
- Błąd końcowy: 6.57393×10^{-8}

- Liczba iteracji: 14

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$
- Czas wykonania: 0.000362429 s
- Błąd końcowy: 6.11239×10^{-8}
- Liczba iteracji: 9

12.3 Punkt startowy: 0.0

Metoda Jacobiego:

- Rozwiązanie: $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$
- Czas wykonania: 0.000737732 s
- Błąd końcowy: 6.36513×10^{-8}
- Liczba iteracji: 14

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$
- Czas wykonania: 0.000418604 s
- Błąd końcowy: 5.92687×10^{-8}
- Liczba iteracji: 9

12.4 Punkt startowy: 10.0

Metoda Jacobiego:

- Rozwiązanie: $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$
- Czas wykonania: 0.000532728 s
- Błąd końcowy: 1.78985×10^{-7}
- Liczba iteracji: 13

Metoda Gaussa-Seidela:

- Rozwiązanie: $[0.158032, 0.322903, 0.483905, 0.645151, 0.806452, \dots, 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]$
- Czas wykonania: 0.000406352 s
- Błąd końcowy: 4.07727×10^{-8}
- Liczba iteracji: 9

12.5 Punkt startowy: 100.0

Metoda Jacobiego:

- Rozwiązanie: [0.158032, 0.322903, 0.483905, 0.645151, 0.806452, ..., 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]
- Czas wykonania: 0.000767037 s
- Błąd końcowy: 4.92253×10^{-8}
- Liczba iteracji: 15

Metoda Gaussa-Seidela:

- Rozwiązanie: [0.158032, 0.322903, 0.483905, 0.645151, 0.806452, ..., 31.6145, 31.7791, 31.8304, 32.3687, 36.1265]
- Czas wykonania: 0.000498434 s
- Błąd końcowy: 2.19126×10^{-8}
- Liczba iteracji: 10

13 Dyskusja

Otrzymane wyniki pokazują różnice w efektywności i zbieżności metod Jacobiego i Gaussa-Seidela. W przypadku wartości $d \geq 1.2$, obie metody wykazały poprawną zbieżność do rozwiązania dokładnego, co świadczy o ich poprawnym działaniu w takich warunkach. Jednak analiza wyników ujawnia kilka istotnych wniosków:

13.1 Porównanie metod Jacobiego i Gaussa-Seidela

1. Szybkość zbieżności:

- Metoda Gaussa-Seidela była konsekwentnie szybsza od metody Jacobiego w liczbie iteracji potrzebnych do osiągnięcia zadanej dokładności. Wynika to z faktu, że Gauss-Seidel natychmiast aktualizuje wartości w trakcie iteracji, co prowadzi do szybszej redukcji błędów.
- Metoda Jacobiego, choć bardziej równoległa, wymagała większej liczby iteracji, co przekłada się na dłuższy czas wykonania.

2. Zbieżność w różnych warunkach początkowych:

- Metoda Gaussa-Seidela była bardziej stabilna i efektywna niezależnie od punktu startowego. Nawet dla dużych wartości początkowych (10 lub 100) wykazywała szybką zbieżność do dokładnego rozwiązania.
- W przypadku Jacobiego, dla dużych punktów startowych liczba iteracji wzrastała, co podkreśla jej ograniczenia w bardziej ekstremalnych warunkach początkowych.

3. Wpływ wartości d :

- Dla wartości $d \geq 1.2$ macierz zachowywała dominację przekatniową, co gwarantowało zbieżność obu metod.
- Dla $d = 0.5$ obie metody były niestabilne i nie zbiegały do poprawnego rozwiązania (*-nan*). Brak dominacji przekatniowej w macierzy spowodował niestabilność procedur iteracyjnych.

13.2 Czy procedura iteracyjna zawsze jest zbieżna?

Odpowiedź na to pytanie brzmi: **nie**. Procedura iteracyjna, taka jak metoda Jacobiego czy Gaussa-Seidela, nie zawsze jest zbieżna. Zbieżność zależy od właściwości macierzy układu równań liniowych. Kluczowe warunki zbieżności to: - **Dominacja przekatniowa macierzy** ($|a_{ii}| > \sum_{j \neq i} |a_{ij}|$ dla każdego i): Gwarantuje stabilność i zbieżność iteracji. - **Macierz dodatnio określona**: W takich przypadkach metody iteracyjne mają lepsze właściwości zbieżności.

W przypadku $d = 0.5$, brak dominacji przekatniowej spowodował, że metody iteracyjne nie były w stanie znaleźć poprawnego rozwiązania. W praktyce oznacza to konieczność:

- Zastosowania innych metod numerycznych (np. dekompozycji LU, metody QR).
- Lub przekształcenia układu (np. przez dodanie dominacji przekatniowej).

14 Podsumowanie

Przeprowadzone analizy i wyniki testów pozwalają wyciągnąć następujące wnioski:

1. Efektywność metod:

- Metoda Gaussa-Seidela wykazała znacznie większą efektywność w porównaniu z metodą Jacobiego. Wymagała mniejszej liczby iteracji i krótszego czasu wykonania, co czyni ją bardziej praktycznym wyborem w zastosowaniach numerycznych.
- Metoda Jacobiego, choć mniej efektywna, pozostaje przydatna w sytuacjach, gdzie równoległość obliczeń ma kluczowe znaczenie.

2. Warunki zbieżności:

- Obie metody były zbieżne dla wartości $d \geq 1.2$, gdzie macierz układu była przekatniowo dominująca, co zapewniało stabilność iteracji.
- Dla $d = 0.5$ brak dominacji przekatniowej uniemożliwił zbieżność obu metod, co podkreśla znaczenie warunków stabilności macierzy w metodach iteracyjnych.

3. Zachowanie w różnych punktach startowych:

- Metoda Gaussa-Seidela była bardziej odporna na różnice w punktach startowych i zachowywała szybkie tempo zbieżności niezależnie od wartości początkowej.
- W przypadku Jacobiego większe punkty startowe prowadziły do wolniejszej zbieżności, co może ograniczać jej użyteczność w bardziej ekstremalnych warunkach.

4. Zastosowania praktyczne:

- Metody iteracyjne takie jak Jacobi i Gauss-Seidel znajdują zastosowanie w przypadkach, gdy dekompozycje bezpośrednie (np. LU, QR) są zbyt kosztowne obliczeniowo, szczególnie dla dużych i rzadkich macierzy.
- Warunkiem ich zastosowania jest odpowiednia struktura macierzy układu równań – w przeciwnym razie konieczne jest użycie innych metod numerycznych lub modyfikacja macierzy.

14.1 Odpowiedź na pytanie

Procedury iteracyjne, takie jak metody Jacobiego i Gaussa-Seidela, nie zawsze są zbieżne. Ich zbieżność zależy od własności macierzy układu równań, takich jak dominacja przekatniowa czy dodatnia określoność. W przypadku braku spełnienia tych warunków metody mogą być niestabilne lub nieosiągać poprawnych wyników.

14.2 Podsumowanie końcowe

Wyniki pokazują, że metody iteracyjne są potężnym narzędziem numerycznym, ale ich skuteczność zależy od właściwego doboru problemu. Metoda Gaussa-Seidela przewyższa metodę Jacobiego w większości analizowanych przypadków, a jej szybka zbieżność czyni ją preferowaną w zastosowaniach praktycznych. Warto jednak pamiętać o ograniczeniach tych metod i konieczności stosowania alternatywnych technik w przypadku nieodpowiednich warunków numerycznych.