NUM5

Mikołaj Kowalski

November 27, 2024

1 Wstep

Celem sprawozdania jest rozwiazanie układu równań liniowych o macierzy diagonalnie dominujacej za pomoca metod iteracyjnych: **Jacobi** oraz **Gaussa-Seidela**. Układ równań ma postać:

$$A \cdot x = b,$$

gdzie A jest macierza wypełniona według określonego wzoru, x to wektor niewiadomych, a b to wektor wyników.

Dla $N = 200$ macierz A przyjmuje strukture:

$$A = \begin{bmatrix} d & 0.5 & 0.1 & & & \cdots & \\ 0.5 & d & 0.5 & 0.1 & & \cdots & \\ 0.1 & 0.5 & d & 0.5 & 0.1 & \cdots & \\ & \ddots & \ddots & \ddots & \ddots & \ddots & \\ & & 0.1 & 0.5 & d & 0.5 & 0.1 \\ & & & 0.1 & 0.5 & d & 0.5 \\ & & & & 0.1 & 0.5 & d \end{bmatrix},$$

natomiast wektor b przyjmuje postać:

$$b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \\ N \end{bmatrix}.$$

Wartość elementu diagonalnego d oraz punkty startowe beda zmieniane w celu zbadania wpływu tych parametrów na zbieżność metod.

2 Metody Rozwiazania

Do rozwiazania zadania zastosowano dwie metody iteracyjne:

- **Metoda Jacobiego** - w każdym kroku iteracji korzysta z wartości z poprzedniej iteracji.

- **Metoda Gaussa-Seidela** - w trakcie każdej iteracji wykorzystuje aktualizowane wartości zmiennych.

Obie metody porównano pod wzgledem liczby iteracji potrzebnej do osiagniecia zadanej tolerancji błedu 10^{-10} oraz dokładności wzgledem rozwiazania dokładnego, wyznaczonego za pomoca funkcji `numpy.linalg.solve`.

3 Opis Implementacji

Poniżej przedstawiono główne elementy algorytmu:

- Funkcja `matrix_builder(d)` generuje macierz A i wektor b zgodnie z opisanym schematem.
- Funkcje `Jacobi(x_for_check)` oraz `Gauss_Seidel(x_for_check)` implementuja odpowiednie metody iteracyjne.
- Funkcja `plot_differences(d)` wizualizuje różnice miedzy rozwiazaniem dokładnym a przybliżonym w kolejnych iteracjach.

4 Wyniki

Dla różnych wartości d (*np.* $d = 2, 3, 4$) przeprowadzono symulacje. Na wykresach przedstawiono różnice miedzy dokładnym rozwiazaniem a przybliżeniem w kolejnych iteracjach w skali logarytmicznej.

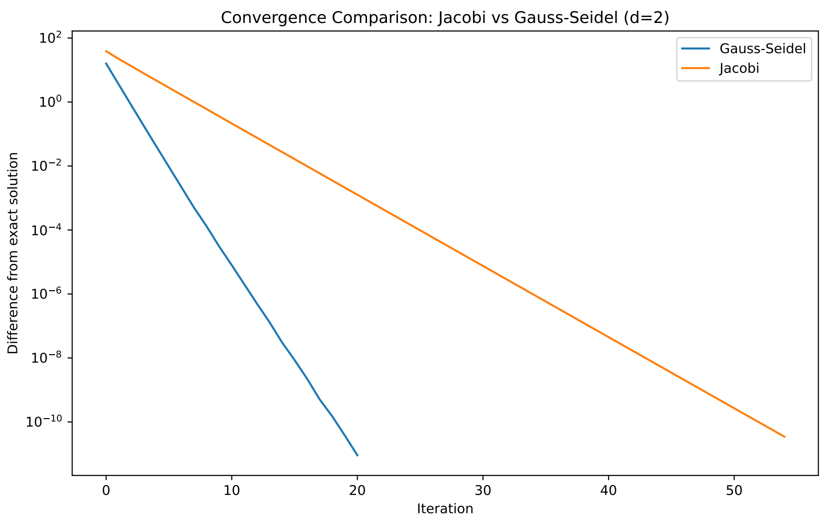
4.1 Wyniki dla $d = 2$ (Jacobi)

[0.29582854 0.62922177 0.9373203 1.24982028 1.5625551 1.87499564 2.18749817 2.5000007 2.81249992 3.12499998 3.43750001 3.75 4.0625 4.375 4.6875 5. 5.3125 5.625 5.9375 6.25 6.5625 6.875 7.1875 7.5 7.8125 8.125 8.4375 8.75 9.0625 9.375 9.6875 10. 10.3125 10.625 10.9375 11.25 11.5625 11.875 12.1875 12.5 12.8125 13.125 13.4375 13.75 14.0625 14.375 14.6875 15. 15.3125 15.625 15.9375 16.25 16.5625 16.875 17.1875 17.5 17.8125 18.125 18.4375 18.75 19.0625 19.375 19.6875 20. 20.3125 20.625 20.9375 21.25 21.5625 21.875 22.1875 22.5 22.8125 23.125 23.4375 23.75 24.0625 24.375 24.6875 25. 25.3125 25.625 25.9375 26.25 26.5625 26.875 27.1875 27.5 27.8125 28.125 28.4375 28.75 29.0625 29.375 29.6875 30. 30.3125 30.625 30.9375 31.25 31.5625 31.875 32.1875 32.5 32.8125 33.125 33.4375 33.75 34.0625 34.375 34.6875 35. 35.3125 35.625 35.9375 36.25 36.5625 36.875 37.1875 37.5 37.8125 38.125 38.4375 38.75 39.0625 39.375 39.6875 40. 40.3125 40.625 40.9375 41.25 41.5625 41.875 42.1875 42.5 42.8125 43.125 43.4375 43.75 44.0625 44.375 44.6875 45. 45.3125 45.625 45.9375 46.25 46.5625 46.875 47.1875 47.5 47.8125 48.125 48.4375 48.75 49.0625 49.375 49.6875 50. 50.3125 50.625 50.9375 51.25 51.5625 51.875 52.1875 52.5 52.8125 53.125 53.4375 53.75 54.0625 54.375 54.6875 55. 55.3125 55.625 55.9375 56.25 56.5625 56.875 57.1875 57.5 57.8125 58.125 58.43749991 58.75000041 59.06249972 59.37499356 59.68753578 59.99995087 60.31206249 60.62799772 60.93147142 61.22242521 61.80639439 61.23416907 60.65766947 81.77387418]

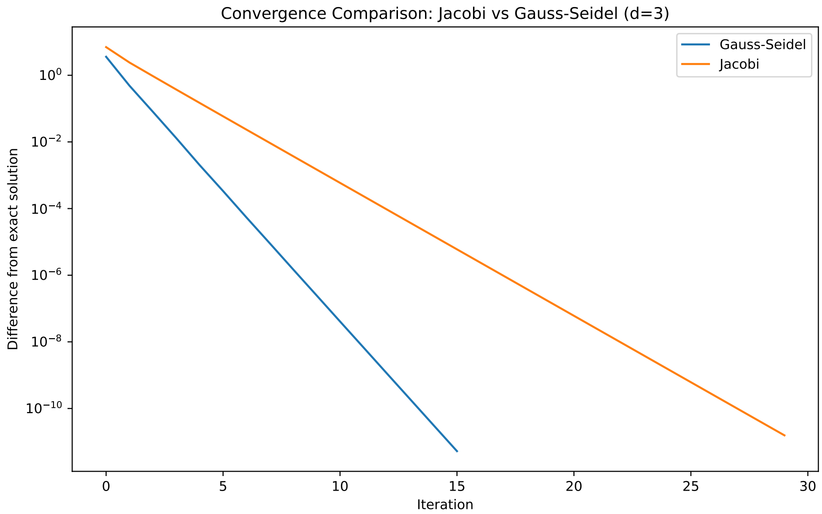
4.2 Wyniki dla $d = 2$ (Gauss-Seidel)

[0.29582854 0.62922177 0.9373203 1.24982028 1.5625551 1.87499564 2.18749817 2.5000007 2.81249992 3.12499998 3.43750001 3.75 4.0625 4.375 4.6875 5. 5.3125 5.625 5.9375 6.25 6.5625 6.875 7.1875 7.5 7.8125 8.125 8.4375 8.75 9.0625 9.375 9.6875 10. 10.3125 10.625 10.9375 11.25 11.5625 11.875 12.1875 12.5 12.8125 13.125 13.4375 13.75 14.0625 14.375 14.6875 15. 15.3125 15.625 15.9375 16.25 16.5625 16.875 17.1875 17.5 17.8125 18.125 18.4375 18.75 19.0625 19.375 19.6875 20. 20.3125 20.625 20.9375 21.25 21.5625 21.875 22.1875 22.5 22.8125 23.125 23.4375 23.75 24.0625 24.375 24.6875 25. 25.3125 25.625 25.9375 26.25 26.5625 26.875 27.1875 27.5 27.8125 28.125 28.4375 28.75 29.0625 29.375 29.6875 30. 30.3125 30.625 30.9375 31.25 31.5625 31.875 32.1875 32.5 32.8125 33.125 33.4375 33.75 34.0625 34.375 34.6875 35. 35.3125 35.625 35.9375 36.25 36.5625 36.875 37.1875 37.5 37.8125 38.125 38.4375 38.75 39.0625 39.375 39.6875 40. 40.3125 40.625 40.9375 41.25 41.5625 41.875 42.1875 42.5 42.8125 43.125 43.4375 43.75 44.0625 44.375 44.6875 45. 45.3125 45.625 45.9375 46.25 46.5625 46.875 47.1875 47.5 47.8125 48.125 48.4375 48.75 49.0625 49.375 49.6875 50. 50.3125 50.625 50.9375 51.25 51.5625 51.875 52.1875 52.5 52.8125 53.125 53.4375 53.75 54.0625 54.375 54.6875 55. 55.3125 55.625 55.9375 56.25 56.5625 56.875 57.1875 57.5 57.8125 58.125 58.43749991 58.75000041 59.06249972 59.37499356 59.68753578 59.99995087 60.31206249 60.62799772 60.93147142 61.22242521 61.80639439 61.23416907 60.65766947 81.77387418]

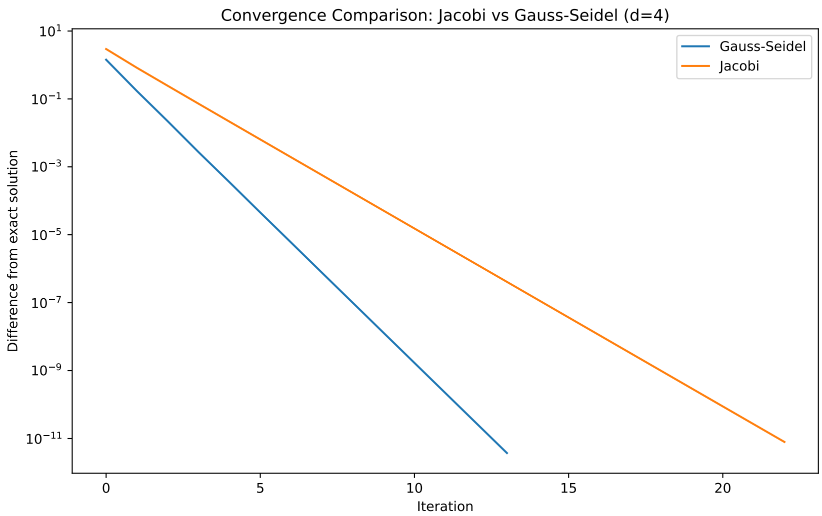
4.3 Wykres dla $d = 2$



4.4 Wykres dla $d = 3$



4.5 Wykres dla $d = 4$



4.6 Porównanie metod dla różnych wartości d

Dla każdej wartości d zaobserwowano, że:

- Metoda Gaussa-Seidela wykazuje szybsza zbieżność w porównaniu z metoda Jacobiego.
- Obie metody osiagaja zbieżność w mniej niż 10 000 iteracji dla $d > 1$.
- Procedura iteracyjna nie jest zbieżna dla małych wartości d, gdy macierz A nie spełnia warunku diagonalnej dominacji.

5 Wnioski

Z przeprowadzonych symulacji wynika, że:

- Zbieżność metody Jacobi i Gaussa-Seidela zależy od wartości d. Dla $d > 1$ obie metody sa zbieżne, a Gauss-Seidel osiaga zbieżność szybciej.
- Przy małych wartościach d metody iteracyjne moga nie być zbieżne ze wzgledu na brak diagonalnej dominacji macierzy A.
- Dobór poczatkowych wartości wektora x nie ma istotnego wpływu na liczbe iteracji, jeśli d jest dostatecznie duże.