

NUM5

Kacper Tatrocki (*kacper.tatrocki@student.uj.edu.pl*)

21 listopada 2022

1 Teoria

Rozwiązywanie układów równań za pomocą wcześniej stosowanych technik prowadziło do otrzymywania dokładnych rozwiązań równań. Metody te nazywaliśmy metodami dokładnymi.

Natomiast w metodach iteracyjnych rozwiązanie dokładne teoretycznie otrzymujemy po wykonaniu nieskończenie wielu kroków. Oczywiście nie wykonujemy naszych programów w nieskończoność, lecz oczekujemy że po skończonej ilości kroków zbliżymy się do dokładnego wyniku który będzie się mieścił w przyjętych granicach błędu zaokrąglenia.

Aby metoda Jacobiego była zbieżna nasza macierz musi być diagonalnie dominująca. Znaczy to nic innego, że element na diagonalu musi być większy niż suma pozostałych elementów w tym samym wierszu. Suma właśnie tych pozostałych elementów jest równa 2,4 co jest oczywiście mniejsze niż nasz element na diagonalu w danym wierszu.

Metoda Gaussa-Seidela jest zbieżna gdy mamy macierz symetryczną i dodatnio określoną co też zachodzi dla tej macierzy.

Porównywane dalej metody to metoda *Jacobiego* i metoda *Gaussa-Seidela* należą do ogólnej kategorii metod iteracyjnych:

$$Mx^{(k+1)} = Nx^{(k)} + b \quad (1)$$

Gdzie indeks k oznacza numer iteracji. Dla równania $Ax = b$, $A = M - N$ jest podziałem wybranym w różny sposób w zależności od metody iteracyjnej. Podział dla metody *Jacobiego* (2) i *Gaussa-Seidela* (3)

$$A = D + (L + U) \quad M = D \quad N = -(L + U) \quad (2)$$

$$A = (D + L) + U \quad M = D + L \quad N = -U \quad (3)$$

Z powyższych wzorów można wyprowadzić wyrażenia na i -ty element wektora w $k + 1$ iteracji odpowiednio dla obu metod:

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)} - \sum_{j=i+1}^N a_{ij}x_j^{(k)} \right) \quad (4)$$

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^N a_{ij}x_j^{(k)} \right) \quad (5)$$

2 Ćwiczenie

W ćwiczeniu należy rozwiązać układ równań $Ax = b$ za pomocą omawianych metod iteracyjnych i porównać ich tempo zbieżności do rozwiązania.

$$B = \begin{bmatrix} 3 & 1 & 0.2 & & & \\ 1 & 3 & 1 & 0.2 & & \\ 0.2 & 1 & 3 & 1 & 0.2 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & 0.2 & 1 & 3 & 1 & 0.2 \\ & & & 0.2 & 1 & 3 & 1 \\ & & & & 0.2 & 1 & 3 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ \vdots \\ N-1 \\ N \end{bmatrix} \quad (6)$$

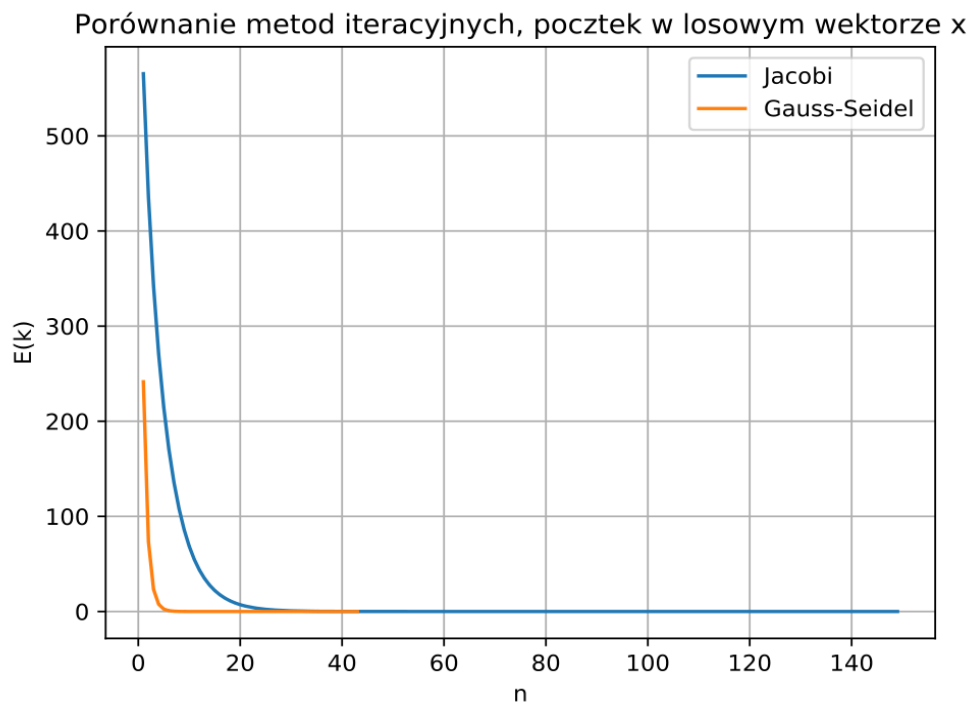
3 Wyniki

$$x = \begin{bmatrix} 0.17126009249156168 \\ 0.37523973745159345 \\ 0.5548999253689408 \\ 0.7406038489242017 \\ 0.9260230950962519 \\ 1.111087426360602 \\ 1.2962972717647263 \\ 1.481482921363617 \\ 1.666666089813768 \\ \vdots \\ 17.73605398322124 \\ 18.107440202728714 \\ 18.03115406572111 \\ 16.956038061792984 \\ 26.47924370835428 \end{bmatrix} \quad (7)$$

Rysunek 1: Rozwiązanie równania używając metody Jacobiego

$$x = \begin{bmatrix} 0.1712600924915493 \\ 0.3752397374515706 \\ 0.5548999253689074 \\ 0.7406038489241576 \\ 0.9260230950961977 \\ 1.1110874263605375 \\ 1.2962972717646517 \\ 1.481482921363533 \\ \vdots \\ 18.10744020272868 \\ 18.031154065721086 \\ 16.956038061792967 \\ 26.47924370835427 \end{bmatrix} \quad (8)$$

Rysunek 2: Rozwiązanie równania używając metody Gaussa-Seidela



Rysunek 3: Błąd E od ilości iteracji k dla metod *Jacobiego* i *Gaussa-Seidela*

Wykres przedstawia wartość błędu bezwzględnego $E(k) = |x^{(k)} - x|$ dla obu metod. Dla zadanej macierzy A metoda *Gaussa-Seidela* zbiega się szybciej, ponieważ w $k + 1$ kroku iteracji do obliczenia danej składowej wektora $x^{(k+1)}$ oprócz $x^{(k)}$, wykorzystuje ona obliczone już w poprzednich krokach danej iteracji składowe tego $x^{(k+1)}$, natomiast metoda *Jacobiego* używa z wartości tylko z $x^{(k)}$.

4 Podsumowanie

Przewaga metod iteracyjnych jest widoczna gdy, rozwiązanie równania za pomocą faktoryzacji macierzy staje się zbyt kosztowne.

Dla gęstej macierzy A złożoność rozkładu to $O(n^3)$. Złożoność pojedynczej iteracji dla takiej macierzy to $O(n^2)$, stąd złożoność metody iteracyjnej dla k iteracji $O(k * n^2)$.

W porównywanych metodach iteracyjnych istotne jest, aby wykorzystać strukturę macierzy i nie iterować po znanych elementach zerowych, co w równaniu z zadania pozwala to na osiągnięcie liniowej złożoności pojedynczej iteracji.