**Jakub Rybak 333156 grupa nr 3**

**Treść zadania i opis metody.**

Rozwiązywanie układu równań Ax = b, gdzie A to macierz trójdiagonalna. Metodą BSOR

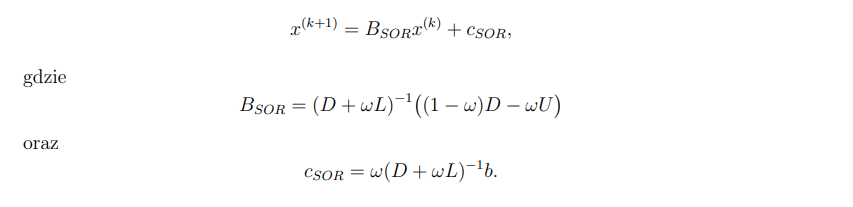
Metoda BSOR (Backword Successive Over-Relaxation) jest techniką iteracyjną wykorzystywaną do rozwiązywania układów równań liniowych. Jest to rozszerzenie metody Gaussa-Seidela, w której wprowadza się dodatkowy parametr zwany współczynnikiem relaksacji, aby przyspieszyć zbieżność rozwiązania.

Metoda BSOR polega na iteracyjnym przekształcaniu układu równań w następujący sposób:

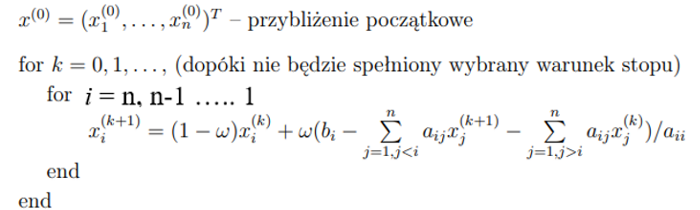
1. Rozbijamy macierz A na trzy składniki:
   * A=L+D+U , gdzie:
     + L to macierz dolna (zawiera elementy poniżej głównej przekątnej),
     + D to macierz diagonalna (zawiera tylko elementy na głównej przekątnej),
     + U to macierz górna (zawiera elementy powyżej głównej przekątnej).

Jako że macierze z tematu to macierze trójdiagonalne to postać L i U jest bardzo prosta.

1. Iteracyjne równanie BSOR jest zapisane w postaci:



Sposób bez macierzy iteracyjnej:



**Opis programu:**

**Sor\_classical.m**

Funkcja rozwiązuje układ równań liniowych za pomocą klasycznej metody BSOR (Backward Successive Over-Relaxation).

Parametry wejściowe:

A - macierz współczynników układu równań liniowych (rozmiar n x 3), gdzie A(:,1) to dolna przekątna, A(:,2) to główna przekątna, A(:,3) to górna przekątna (A jest macierzą trójdiagonalną).

b - wektor wyrazów wolnych (rozmiar n x 1).

omega - parametr relaksacji (0 < omega < 2), który wpływa na szybkość zbieżności metody.

tol - tolerancja błędu, określająca warunek zatrzymania iteracji (jeśli różnica między kolejnymi przybliżeniami jest mniejsza od tol, algorytm kończy obliczenia).

max\_iter - maksymalna liczba iteracji, po której algorytm zakończy się, nawet jeśli nie osiągnięto tolerancji.

Parametry wyjściowe:

x - wektor rozwiązania układu równań (rozmiar n x 1), który zawiera przybliżone wartości zmiennych układu.

Opis działania:

1. Funkcja wydobywa dolną, główną i górną przekątną macierzy A.

- lower\_diag to dolna przekątna,

- main\_diag to główna przekątna,

- upper\_diag to górna przekątna.

2. Tworzony jest wektor x, który początkowo zawiera same zera (rozpoczynamy od przybliżenia zerowego).

3. W pętli for wykonywane są iteracje metody BSOR. Dla każdej zmiennej i (od n do 1) obliczany jest nowy przybliżony wynik x\_i^(k+1) z uwzględnieniem poprzednich i następnych elementów układu równań.

Formuła aktualizacji to:

x(i) = (1 - omega) \* x(i) + (omega / main\_diag(i)) \* (b(i) - sum\_lower - sum\_upper),

gdzie sum\_lower i sum\_upper to składniki wynikające z sąsiednich elementów macierzy A.

4. Po każdej iteracji sprawdzany jest warunek zatrzymania. Błąd względny jest mniejszy od zadanej tolerancji, algorytm zatrzymuje się.

5. Jeśli po wykonaniu maksymalnej liczby iteracji rozwiązanie nie spełnia warunku tolerancji, proces zostaje zakończony.

Zwracany jest wektor x, który zawiera przybliżone rozwiązanie układu równań.

Przykład użycia:

A = [a1 b1 c1; a2 b2 c2; ...]; % Przykładowa macierz trójdiagonalna

b = [b1; b2; ...]; % Wektor wyrazów wolnych

omega = 1.25; % Parametr relaksacji

tol = 1e-6; % Tolerancja błędu

max\_iter = 1000; % Maksymalna liczba iteracji

x = sor\_classical(A, b, omega, tol, max\_iter);

**sor\_iteration\_matrix.m**

Funkcja rozwiązuje układ równań liniowych za pomocą metody BSOR (Backward Successive Over-Relaxation), (Używając macierzy iteracyjnej).

Parametry wejściowe:

A - macierz współczynników układu równań liniowych (rozmiar n x 3), gdzie A(:,1) to dolna przekątna, A(:,2) to główna przekątna, A(:,3) to górna przekątna (A jest macierzą trójdiagonalną).

b - wektor wyrazów wolnych (rozmiar n x 1).

omega - parametr relaksacji (0 < omega < 2), który wpływa na szybkość zbieżności metody.

tol - tolerancja błędu, określająca warunek zatrzymania iteracji (jeśli różnica między kolejnymi przybliżeniami jest mniejsza od tol, algorytm kończy obliczenia).

max\_iter - maksymalna liczba iteracji, po której algorytm zakończy się, nawet jeśli nie osiągnięto tolerancji.

Parametry wyjściowe:

x - wektor rozwiązania układu równań (rozmiar n x 1), który zawiera przybliżone wartości zmiennych układu.

liczba\_iteracji

macierz\_iteracyjna

Opis działania:

1. Funkcja najpierw wydobywa dolną, główną i górną przekątną macierzy A.

2. Następnie tworzy macierze D, L, U, które odpowiadają za rozkład macierzy A:

- D to macierz diagonalna (zawiera elementy z głównej przekątnej),

- L to macierz dolna (zawiera elementy z dolnej przekątnej),

- U to macierz górna (zawiera elementy z górnej przekątnej).

3. Wyznacza macierz iteracyjną B\_SOR oraz wektor c\_SOR, które są używane do aktualizacji przybliżenia rozwiązania.

4. Funkcja iteracyjnie aktualizuje wektor rozwiązania x, wykorzystując wzór metody BSOR:

x\_new = B\_SOR \* x + c\_SOR

5. Warunek zatrzymania sprawdza, błąd względny jest mniejszy od zadanej tolerancji.

6. Jeśli po wykonaniu maksymalnej liczby iteracji rozwiązanie nie spełnia warunku tolerancji, proces zostaje zakończony.

Zwracany jest wektor x, który zawiera przybliżone rozwiązanie układu równań.

Przykład użycia:

A = [a1 b1 c1; a2 b2 c2; ...]; % Przykładowa macierz trójdiagonalna

b = [b1; b2; ...]; % Wektor wyrazów wolnych

omega = 1.25; % Parametr relaksacji

tol = 1e-6; % Tolerancja błędu

max\_iter = 1000; % Maksymalna liczba iteracji

x = sor\_iteration\_matrix(A, b, omega, tol, max\_iter);

**skrypt\_testujacy.m**

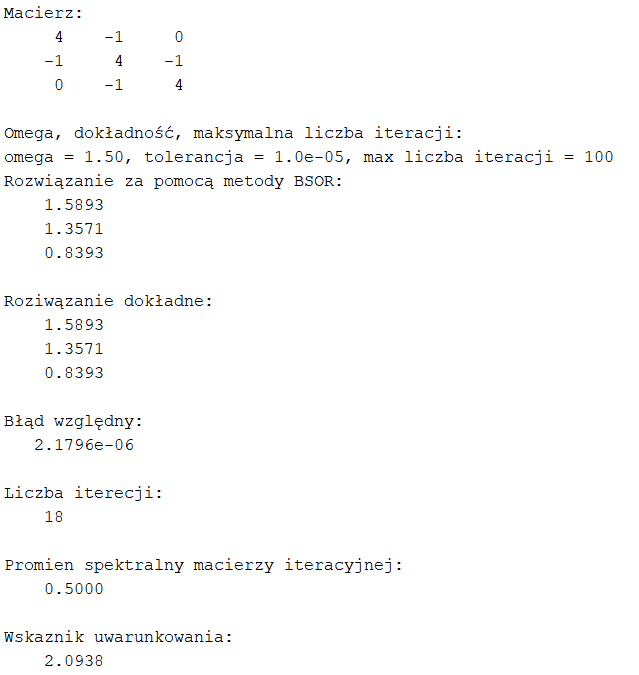
Skypt testujący.

**wykres.m**

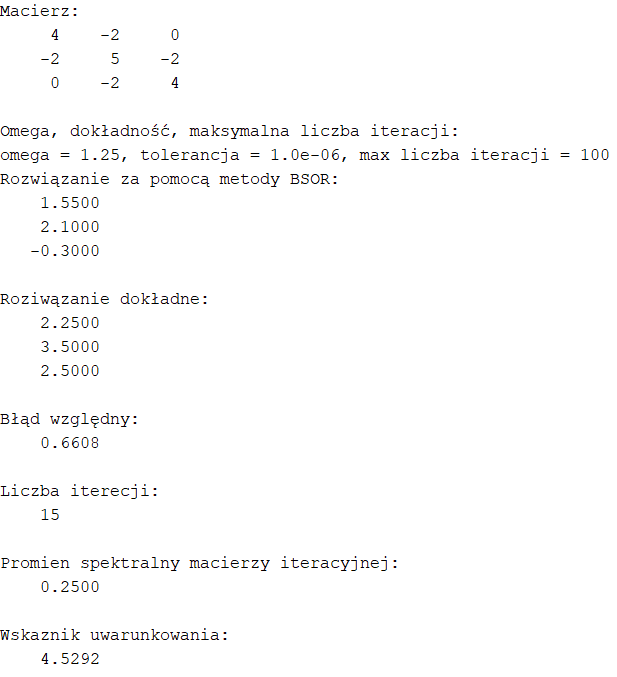
Wykres

**Ciekawe przypadki:**

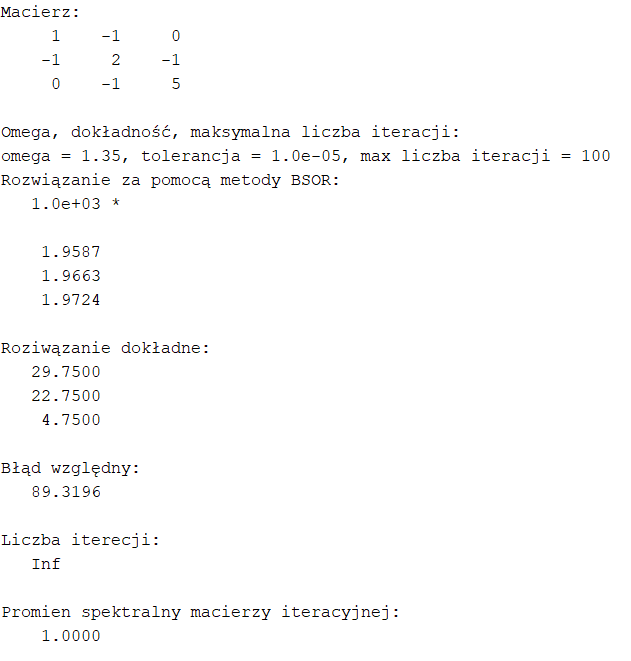
Pierwszy przypadek:



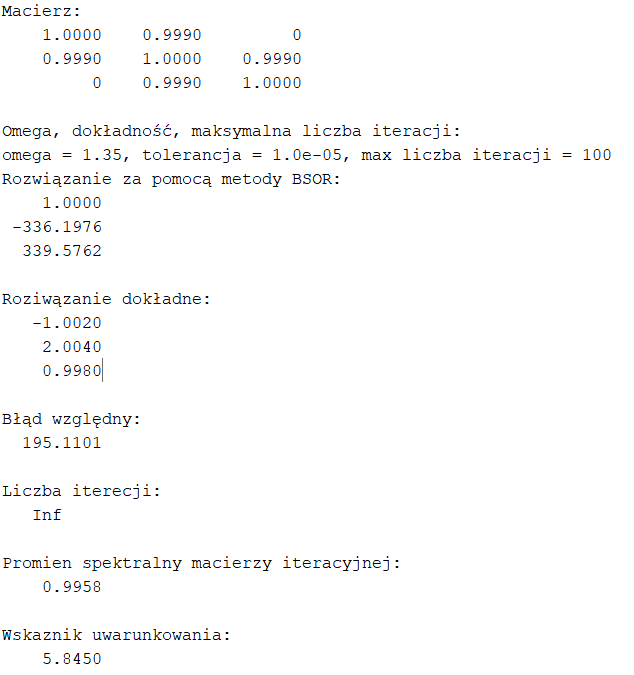
Drugi przypadek:



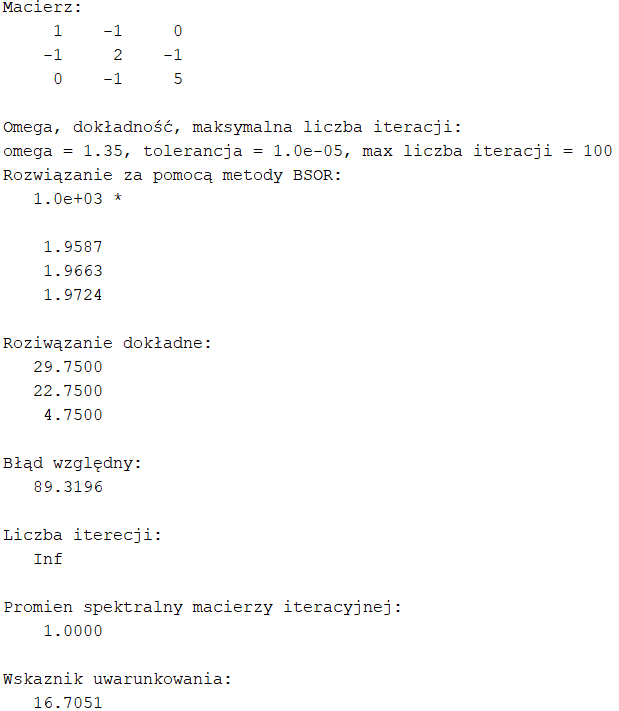
Trzeci przypadek:



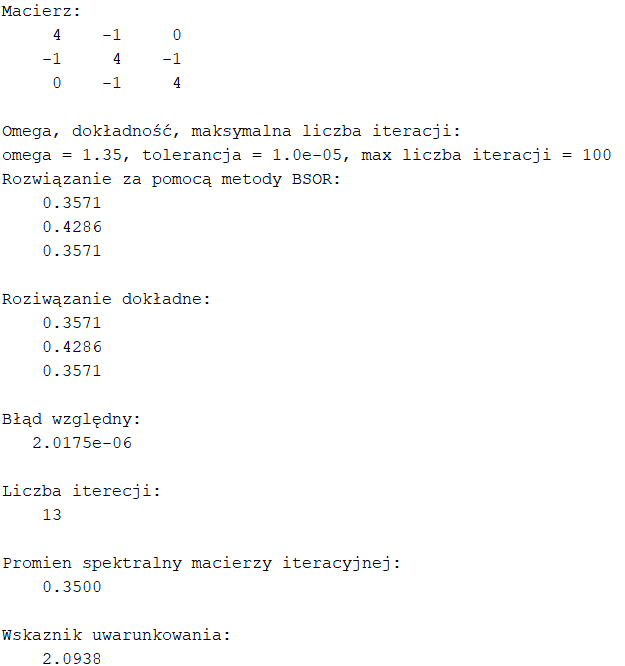
Czwarty przypadek:



Piąty przypadek:

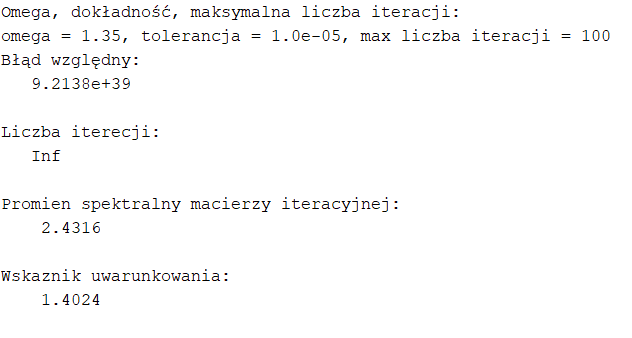


Szósty przypadek:



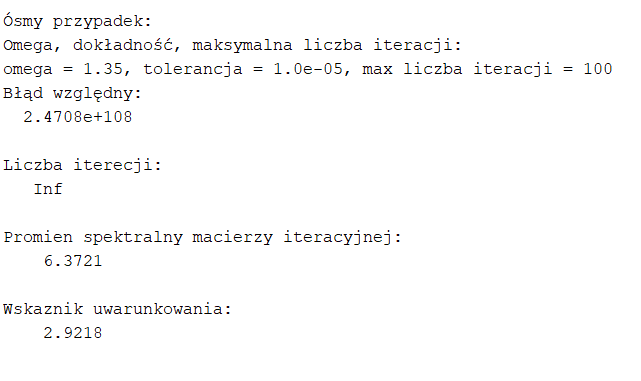
Siódmy przypadek

Macierz 20 x 20 (na głównej przekątnej 2, na górnej przekątnej 1, na dolnej przekątnej -1)

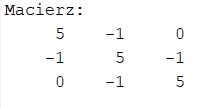


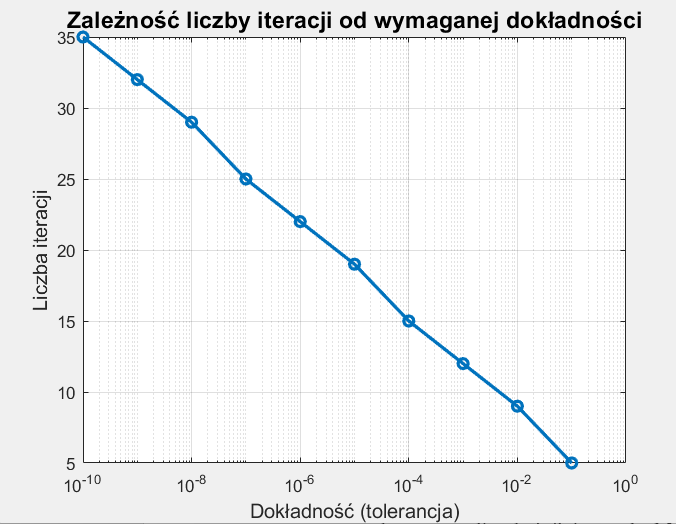
Ósmy przypadek

Macierz 100 x 100 (na głównej przekątnej 3, na górnej przekątnej 2, na dolnej przekątnej -3)



Wykres:





**Wnioski:**

* Wskaźnik uwarunkowania macierzy określa, jak bardzo rozwiązanie układu jest wrażliwe na błędy. Metoda BSOR działa efektywnie, gdy macierz A jest dobrze uwarunkowana czyli wskaźnik jest mały. Metoda działa nieefektywnie gdy wskaźnik jest duży.
* Promień spektralny macierzy iteracyjnej musi być mniejszy od 1, aby metoda była zbieżna. Jeśli promień spektralny wynosi co najmniej 1 to metoda nie jest zbieżna.
* Im mniejsza tolerancja błędu (dokładność) tym większa liczba iteracji jest potrzebna.