

Rozwiązanie układów równań z wykorzystaniem metody BSOR

SPRAWOZDANIE

I. TEMAT PROJEKTU

Projekt polega na przedstawieniu metody BSOR (Backward Successive Over-Relaxation) do rozwiązywania układów równań liniowych $Ax = b$, gdzie $A \in R^{n \times n}$, $b \in R^n$.

II. OPIS METODY

Backward SOR (BSOR) to wariant metody Successive Over-Relaxation (SOR), który jest stosowany w rozwiązywaniu układów równań liniowych $Ax = b$, gdzie $A \in R^{n \times n}$, $b \in R^n$, gdzie iteracje są wykonywane od ostatniej do pierwszej zmiennej. Schemat iteracyjny w metodzie BSOR dla $i = n, n-1, \dots, 1$ w iteracji k przyjmuje postać:

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega \left(\frac{b_i - \sum_{j=i+1}^n a_{ij}x_j^{(k+1)} - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)}}{a_{ii}} \right), \text{ gdzie}$$

- $x_i^{(k)}$ to i – ta zmienna w iteracji k
- $\omega \in R$ to parametr relaksacji
- b_i to element wektora z prawej strony
- a_{ij} to element macierzy A

W metodzie BSOR, iteracje rozpoczynamy od przybliżenia początkowego x_0 , które ma kluczowe znaczenie dla efektywności algorytmu, szczególnie pod względem szybkości zbieżności. Ponadto, zbieżność metody BSOR jest ściśle uzależniona od wartości parametru ω . W przypadku gdy $\omega \notin (0,2)$ metoda BSOR (oraz klasyczna metoda SOR) nie jest zbieżna. Z kolei, gdy $\omega \in (0,2)$, wartość tego parametru ma wpływ na szybkość zbieżności algorytmu.

- Dla $\omega = 1$ metoda BSOR odpowiada standardowej metodzie Gaussa-Seidela.
- Dla $\omega > 1$ (tzw. nadrelaksacja) metoda może przyspieszyć zbieżność, pod warunkiem odpowiedniego doboru parametru.
- Dla $\omega < 1$ (tzw. podrelaksacja) zbieżność może zostać spowolniona, co prowadzi do dłuższego czasu potrzebnego do osiągnięcia rozwiązania.

III. OPIS PROGRAMU

Program został zaimplementowany w MATLAB-ie i zawiera następujące moduły:

1. Funkcja BSOR(A, b, omega, tol, max_iter, x):

Funkcja jako parametry przyjmuje:

- **A** - macierz współczynników z układu równań $Ax = b$, który chcemy rozwiązać
- **b** - wektor prawej strony układu równań
- **omega** - parametr relaksacji

- **tol** – parametr tolerancji zbieżności, który określa, kiedy iteracje powinny zakończyć się, jeśli zmiany w wektorze rozwiązania są wystarczająco małe.
- **max_iter** - maksymalna liczba iteracji, po której funkcja zakończy działanie, nawet jeśli nie osiągnięto zbieżności.
- **x** - przybliżenie początkowe wektora rozwiązania

Funkcja zaczyna od obliczenia rozmiaru macierzy A (liczba zmiennych n) oraz inicjalizacji zmiennej iter na 0 (licznik iteracji). Zmienna rel_error jest inicjalizowana na wartość ∞ , aby w pierwszej iteracji zawsze wykonać sprawdzenie błędu.

Następnie obliczany jest promień spektralny macierzy iteracji B. Aby to zrobić macierz A rozkładana jest na trzy składniki:

- L – macierz dolna (elementy poniżej diagonal),
- D – macierz diagonalna (zawiera elementy na głównej przekątnej),
- U – macierz górna (elementy powyżej diagonal).

Konstrukcja macierzy iteracji B opiera się na wzorze: $B = (D + \omega L)^{-1}((1 - \omega)D - \omega U)$, a promień spektralny macierzy B jest wartością bezwzględną z jej wartości własnych.

W głównej pętli funkcja przeprowadza iteracje, aby znaleźć rozwiązanie układu równań. Pętla przebiega od $i = n$ do $i = 1$ (wstecz). Nowa wartość x_i jest obliczana według wzoru metody BSOR: $x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega \left(\frac{b_i - \sum_{j=i+1}^n a_{ij}x_j^{(k+1)} - \sum_{j=1}^{i-1} a_{ij}x_j^{(k)}}{a_{ii}} \right)$, a wartość ta jest następnie przypisywana do wektora wynikowego x.

Po każdej iteracji obliczany jest błąd względny. Jeśli błąd względny jest mniejszy niż zadana tolerancja (tol), pętla iteracyjna zostaje zakończona, a funkcja zwraca wynik. W przeciwnym wypadku, pętla działa dopóki liczba iteracji osiągnie wartość maksymalną (max_iter). Funkcja zwraca:

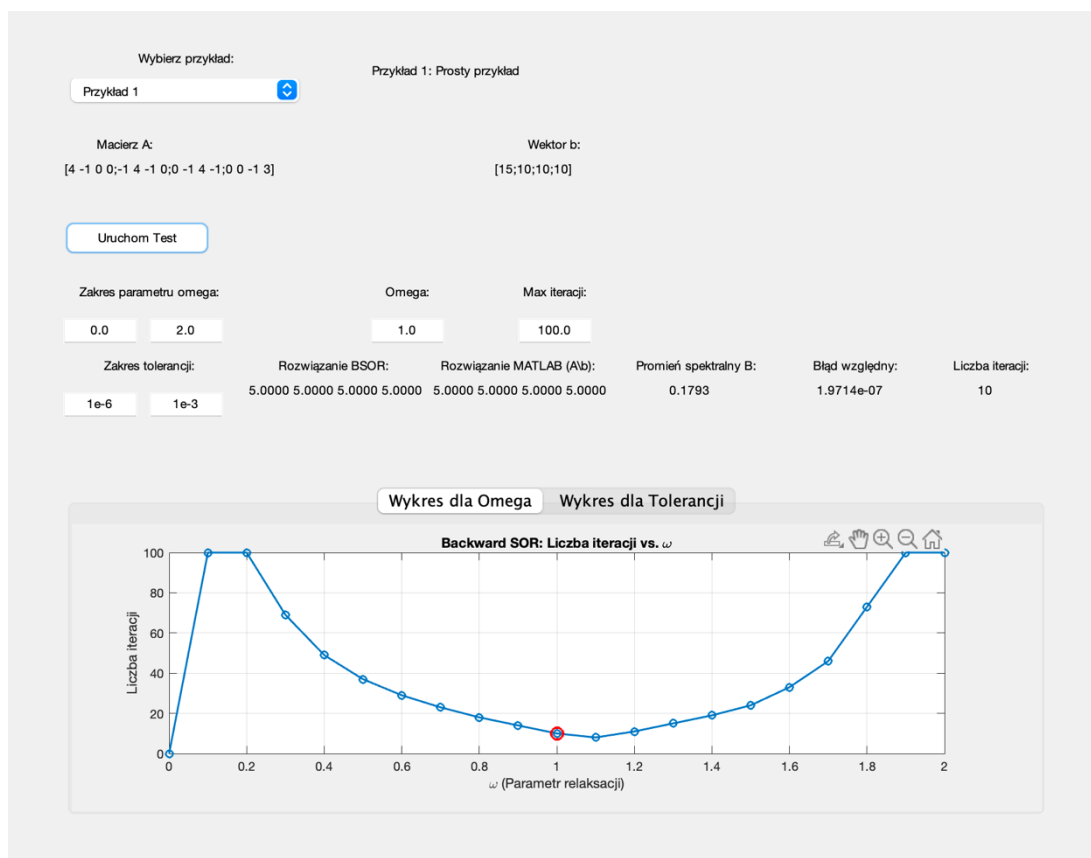
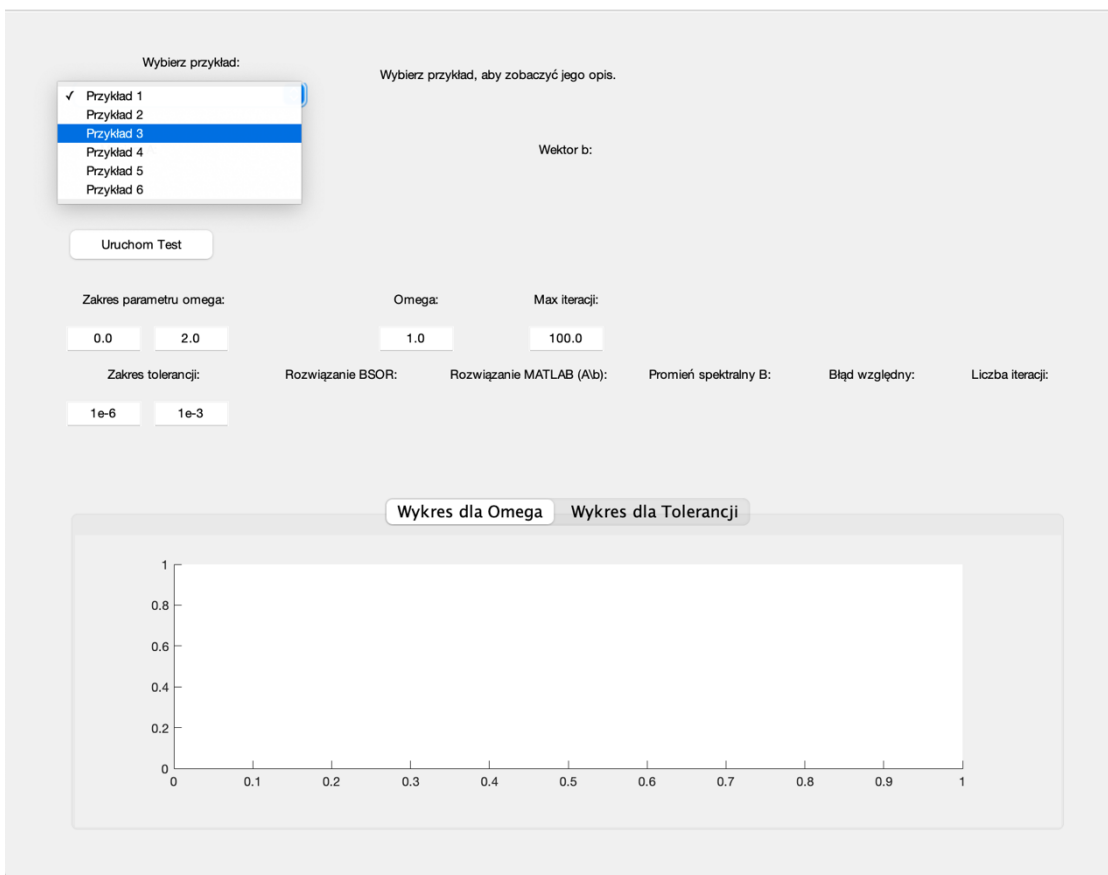
- **x** – wynikowy wektor
- **iter** – liczba wykonanych iteracji
- **rho_B** – promień spektralny B
- **rel_error** – błąd względny

2. Funkcja gui2():

Funkcja gui2() tworzy interfejs graficzny użytkownika, który umożliwia testowanie metody BSOR na kilku przykładach. Użytkownik może wybrać jeden z sześciu przykładów. Każdy z nich ma przypisaną inną macierz A i wektor b. Użytkownik może także wprowadzić wartości dla parametru relaksacji (omega) oraz maksymalnej liczby iteracji (max_iter). Parametr tolerancji ustawiony jest odgórnie na $1e-6$. Po naciśnięciu przycisku "Uruchom Test" GUI wyświetla następujące wyniki: wynikowy wektor obliczony przy użyciu funkcji BSOR2, oraz dokładne rozwiązanie obliczone przy użyciu operatora odwrotności macierzy A\b w MATLABie. Dzięki temu użytkownik może zobaczyć wyraźne różnice pomiędzy rozwiązaniami (lub ich brak). Wyświetlana jest także wartość promienia spektralnego macierzy iteracyjnej, błąd względny rozwiązania BSOR w porównaniu do rozwiązania dokładnego oraz liczbe iteracji, które były wymagane do osiągnięcia zbieżności.

Dodatkowo GUI wyświetla dwa wykresy, które można przełączając między sobą za pomocą zakładek. Pierwszy wykres przedstawia liczbę iteracji dla różnych wartości parametru relaksacji. Zakres tego parametru jest wcześniej ustawiany przez użytkownika. Drugi wykres

wyświetla liczbę iteracji dla różnych wartości parametru tolerancji, której zakres jest również ustalany przez użytkownika.



IV. PRZYKŁADY DZIAŁANIA

W okienku GUI do przetestowania jest 6 następujących przykładów:

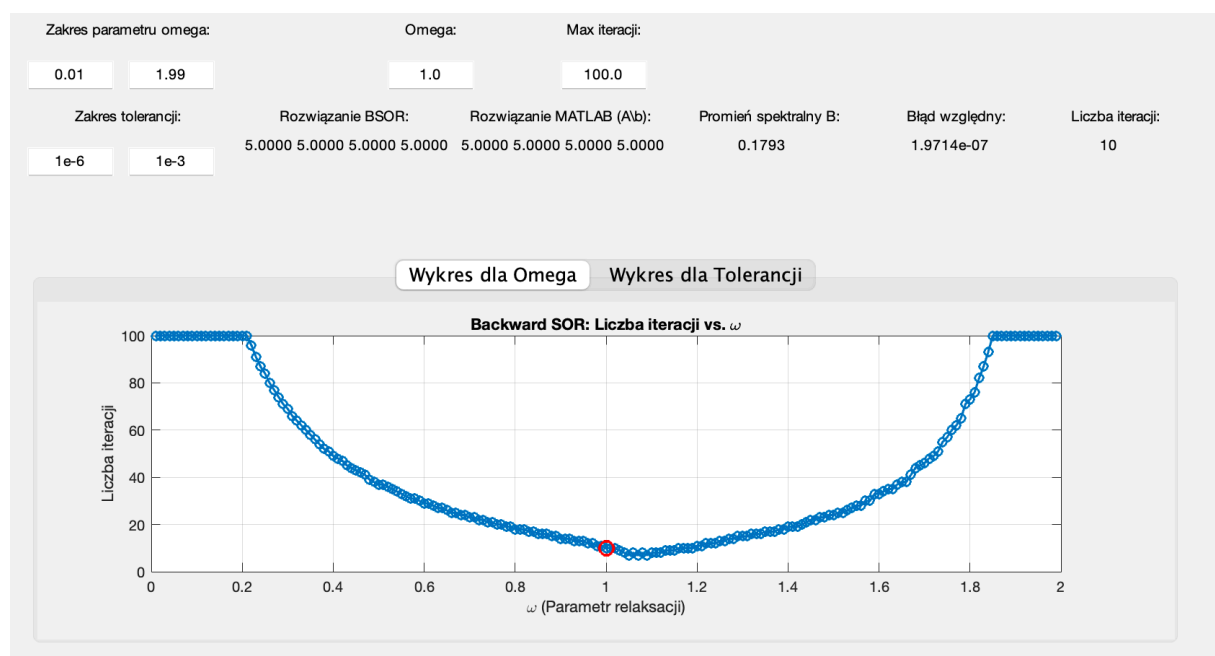
Przykład 1 - Prosty przykład:

$$A = \begin{bmatrix} 4 & -1 & 0 & 0 \\ -1 & 4 & -1 & 0 \\ 0 & -1 & 4 & -1 \\ 0 & 0 & -1 & 3 \end{bmatrix}$$

$$b = \begin{bmatrix} 15 \\ 10 \\ 10 \\ 10 \end{bmatrix}$$

Jest to klasyczny przykład układu równań, który ma jedno rozwiązanie i jest dobrze uwarunkowany. Metoda BSOR osiąga zbieżność dla wszystkich wartości parametrów tolerancji oraz relaksacji. Na wykresie zależności liczby iteracji od parametru relaksacji idealnie widać wpływ omegi na szybkość zbieżności, o której mowa była wcześniej (patrz. "OPIS METODY").

- **Zbieżność:** Szybka, ponieważ układ jest dobrze uwarunkowany.
- **Wyniki:** Liczba iteracji stosunkowo mała, a błąd względny niski.
- **Promień spektralny B:** Wartość ta stosunkowo mała, wskazując na dobrą zbieżność.



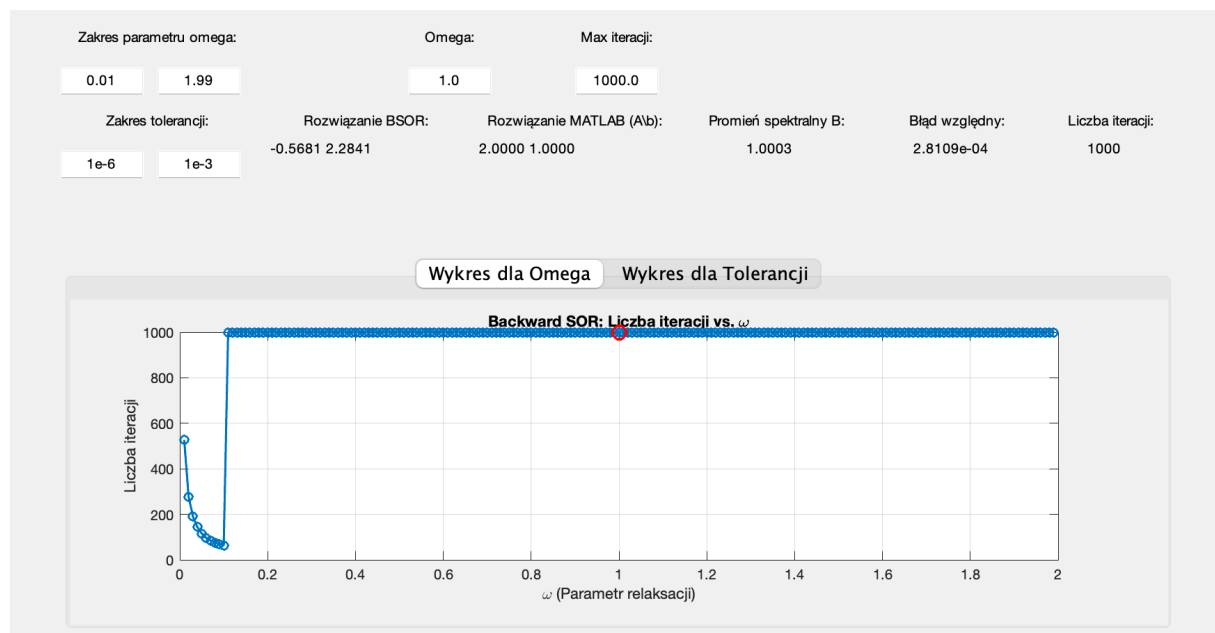
Przykład 2 - Macierz źle uwarunkowana:

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 3.999 \end{bmatrix}$$

$$b = \begin{bmatrix} 4 \\ 7.999 \end{bmatrix}$$

W tym przykładzie macierz A nie jest osobliwa, ale jest bardzo słabo uwarunkowana, co oznacza, że może wystąpić duża wrażliwość na zmiany w danych wejściowych lub zaokrąglenia. Taki układ prowadzi do długich czasów zbieżności w metodach iteracyjnych. Metoda BSOR wymaga ostrożnego doboru parametru ω , aby uzyskać zbieżność. Faktycznie test wykazał, że dla małych wartości ω (co dziwne) metoda zbiega znacznie szybciej niż dla większych. Przy parametrze ω większym od 0.2, metoda BSOR nie zbiega nawet dla bardzo wielu iteracji.

- **Zbieżność:** wolna, szczególnie jeśli wartość ω nie jest odpowiednio dobrana.
- **Wyniki:** większa liczba iteracji, a błąd względny dużo większy niż w przykładzie 1.
- **Promień spektralny B :** większy, wskazując na słabszą zbieżność metody.



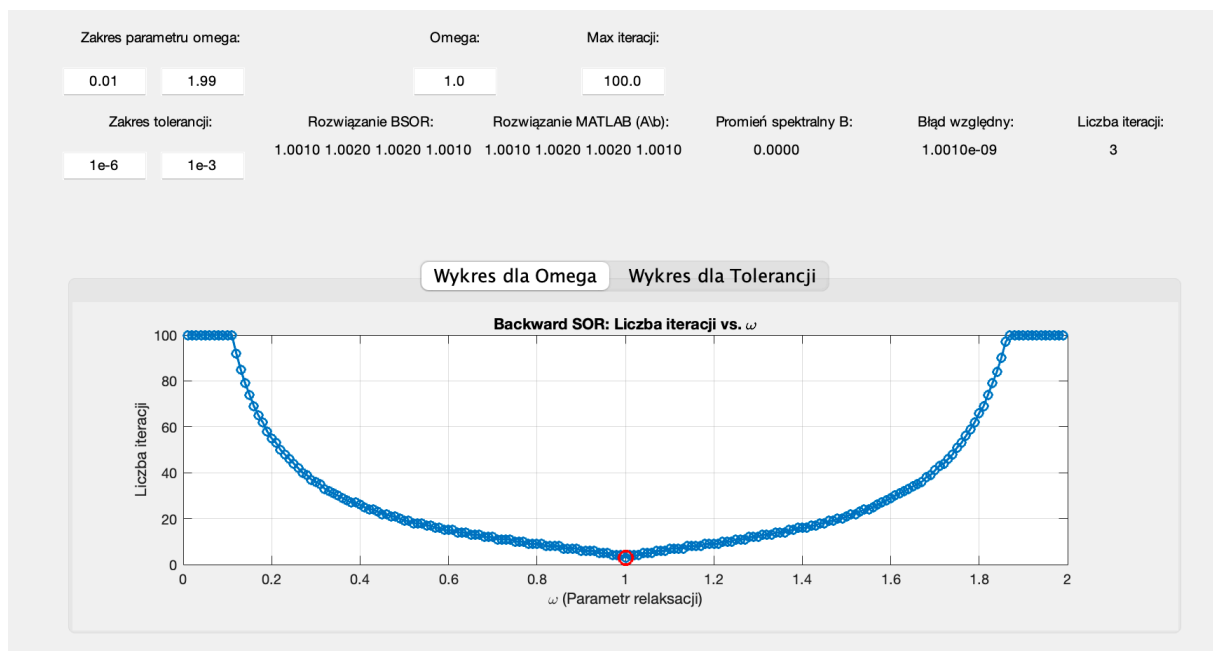
Przykład 3 - Układ z dominującymi wartościami własnymi:

$$A = \begin{bmatrix} 1000 & -1 & 0 & 0 \\ -1 & 1000 & -1 & 0 \\ 0 & -1 & 1000 & -1 \\ 0 & 0 & -1 & 1000 \end{bmatrix}$$

$$b = \begin{bmatrix} 1000 \\ 1000 \\ 1000 \\ 1000 \end{bmatrix}$$

W tym przypadku, wartości własne macierzy A są bardzo duże, co oznacza, że układ jest dobrze uwarunkowany. Metoda, podobnie jak w przykładzie 1, zbiega bardzo szybko, niezależnie od parametrów wejściowych

- **Zbieżność:** Szybka, metoda BSOR skuteczna
- **Wyniki:** Liczba iteracji niewielka, a błąd względny bardzo niski.
- **Promień spektralny B:** Zwykle mniejszy, ponieważ układ jest dobrze uwarunkowany.



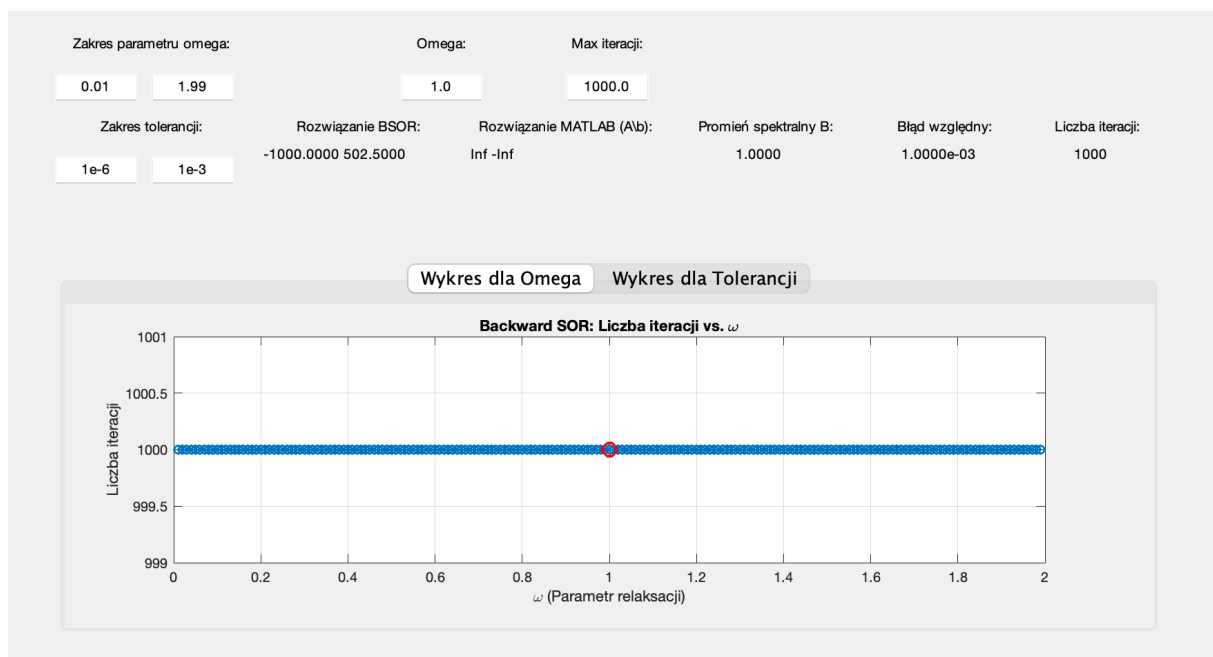
Przykład 4 - Układ bez rozwiązania (sprzeczny):

$$A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$$

$$b = \begin{bmatrix} 5 \\ 12 \end{bmatrix}$$

Układ ten jest sprzeczny, nie ma rozwiązań. Dodałam taki przykład, aby upewnić się, że funkcja BSOR działa. Faktycznie niezależnie od wejściowych parametrów, metoda nigdy nie jest zbieżna. To dobrze, bo nie ma do czego być zbieżna:)

- **Zbieżność:** Metoda BSOR nie osiąga zbieżności, ponieważ układ nie ma rozwiązania.
- **Wyniki:** liczba iteracji nie ma sensu.
- **Promień spektralny B:** trudno ocenić, ale metoda nie da rezultatu.

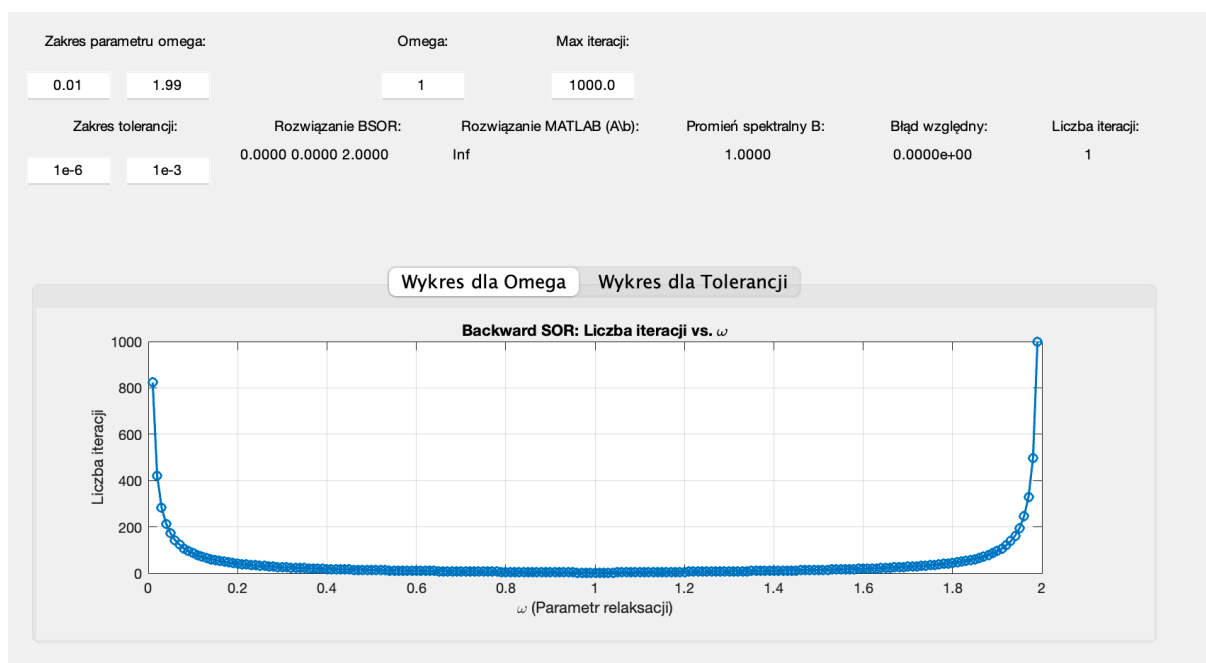


Przykład 5 - Układ z nieskończoną liczbą rozwiązań (niedookreślony):

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & 4 & 6 \\ 1 & 2 & 3 \end{bmatrix}, \quad b = \begin{bmatrix} 6 \\ 12 \\ 6 \end{bmatrix}$$

Jest to układ niedookreślony, ponieważ macierz A jest osobliwa, istnieje nieskończona liczba rozwiązań. Podczas testu, widać, że nasza metoda niezależnie od wejściowych parametrów zbiega do jakiegoś rozwiązania. Co zaskakujące, metoda z jakiegoś powodu potrzebuje jednak kilku iteracji, aby znaleźć właściwe rozwiązanie. Zaskakujące jest to, dlatego bowiem iż każde rozwiązanie jest poprawne.

- **Zbieżność:** Metoda BSOR zawsze (przy odpowiednim parametrze relaksacji) zbiega do wyniku
- **Wyniki:** Błąd względny zerowy
- **Promień spektralny B :** zawsze równy jeden



Przykład 6 - Duże wartości własne:

$$A = \begin{bmatrix} 2 & 3 \\ 5 & 4 \end{bmatrix}$$

$$b = \begin{bmatrix} 5 \\ 6 \end{bmatrix}$$

W tym przykładzie wartości własne macierzy są duże, co często powoduje, że metody iteracyjne nie są zbieżne. Dodatkowo, macierz A nie spełnia warunku dominacji diagonalnej (gdzie każdy element diagonalny powinien być większy od sumy wartości bezwzględnych pozostałych elementów w danym wierszu). Brak tej dominacji często wskazuje, że metoda Gaussa-Seidla lub BSOR i SOR nie będzie zbieżna. Faktycznie nasz test to potwierdza. Nasza metoda niezależnie od wejściowych parametrów nie zbiega do wyniku, który istnieje.

- Zakres parametru omega:

Omega:

Max iteracji:

0.01

1.99

1

1000.0

Zakres tolerancji:

Rozwiązanie BSOR:

Rozwiązanie MATLAB (A\b):

Promień spektralny B:

Błąd względny:

Liczba iteracji:

1e-6

1e-3

2865523796611145328970283-0.2857 1.8571

1.8750

4.6667e-01

1000

Wykres dla Omega

Wykres dla Tolerancji

Backward SOR: Liczba iteracji vs. ω

ω (Parametr relaksacji)	Liczba iteracji
0.0	1000
0.2	1000
0.4	1000
0.6	1000
0.8	1000
1.0	1000
1.2	1000
1.4	1000
1.6	1000
1.8	1000
2.0	1000

Podsumowując wyniki analizy, metoda BSOR najlepiej sprawdza się w przypadku układów równań, które są **dobrze uwarunkowane**. Radzi sobie w przypadku nieskończonej liczby rozwiązań. Dodatkowo **brak dominacji diagonalnej** często powoduje brak zbieżności metody BSOR. Potwierdziliśmy też wpływ i zależność między **szybkością zbieżności** metody BSOR, **a parametrem relaksacji**. Analiza drugiego wykresu pozwoliła także wysunąć wnioski, iż **parametr tolerancji (tol)** oczywiście **ma wpływ na ilość iteracji** - im mniejszy tym więcej iteracji potrzeba do zbieżności.