

23 grudnia 2021

Laura Hoang  
Grupa 2

## Metoda SOR rozwiązywania układu równań liniowych

Projekt nr 2

### 1 Opis metod

W tej sekcji opisane zostaną metody wykorzystane do rozwiązywania zadania odnajdywania rozwiązań układu równań  $Ax = b$ , gdzie  $A \in \mathbb{R}^{n \times n}$ ,  $b \in \mathbb{R}^n$ .

#### 1.1 Metody iteracyjne rozwiązywania układów równań liniowych

Niech  $Ax = b$  będzie układem równań liniowych, przy czym  $A \in \mathbb{R}^{n \times n}$  jest nieosobliwa,  $b \in \mathbb{R}^n$ . Należy odnaleźć taką wartość  $x \in \mathbb{R}^n$ , która spełnia powyższy układ równań.

Idea metod iteracyjnych: poczynając z danego przybliżenia początkowego

$$x^{(0)} = \begin{pmatrix} x_1^{(0)} \\ \dots \\ x_n^{(0)} \end{pmatrix} \in \mathbb{R}^n$$

tworzony jest ciąg kolejnych przybliżeń  $\{x^{(k)}\}$ , taki, że  $x^{(k)} \in \mathbb{R}^n$  dla  $k = 0, 1, \dots$  oraz  $x^{(k)} \rightarrow x$  przy  $k \rightarrow \infty$ .

Jedną z grup metod iteracyjnych są tzw. *metody iteracji prostej*, których zasada

konstrukcji jest taka, że układ wyjściowy  $Ax = b$  zapisuje się w postaci równoważnej  $x = Bx + c$ , a następnie kolejne przybliżenia oblicza się według wzoru

$$x^{(k+1)} = Bx^{(k)} + c,$$

dla  $k = 0, 1, \dots$ . Macierz  $B$  to *macierz iteracji*. Jako początkowe przybliżenie  $x^{(0)}$  można взять wektor o elementach losowych lub wektor zerowy.

## 1.2 Warunek zakończenia obliczeń

Jednym z warunków zakończenia obliczeń jest *warunek Gilla*, który jest połączeniem warunku z uwzględnieniem „ błędów”: bezwzględnego i względnego.

Niech  $d_1$  i  $d_2$  to parametry określające dokładność (liczby dodatnie, bliskie zeru, przy czym  $d_1 > d_2$ ), a  $\|\cdot\|$  jest dowolną normą wektorową. Warunek Gilla wygląda następująco:  $\|x^{(k+1)} - x^{(k)}\| < d_1 \|x^{(k)}\| + d_2$ .

## 1.3 Zbieżność metod iteracyjnych

Cieźko jest stwierdzić czy przybliżenie początkowe  $x^{(0)}$  gwarantuje zbieżność metody. Z tego powodu, należy sprawdzać warunki zapewniające globalną zbieżność (czyli dla dowolnego przybliżenia początkowego  $x^{(0)} \in \mathbb{R}^n$ ). Poniższe twierdzenie jest tego warunkiem koniecznym:

**Twierdzenie.** *Metoda iteracji prostej (tzn. postaci  $x^{(k+1)} = Bx^{(k)} + c$ ) jest zbieżna globalnie, wtedy i tylko wtedy, gdy  $\rho(B) < 1$ , gdzie  $\rho(B)$  - promień spektralny macierzy  $B$ .*

## 1.4 Metoda SOR

Jedną ze wspomnianych powyżej metod iteracyjnych (iteracji prostej) jest metoda SOR<sup>1</sup> (ang. Successive OverRelaxation, in. metoda nadrelaksacji). W tej metodzie występuje parametr  $\omega \in \mathbb{R}$ , zwany parametrem relaksacji<sup>2</sup>.

Metodę SOR można wyprowadzić zapisując macierz  $A$  jako

$$A = L + D + U,$$

gdzie  $L$  - macierz trójkątna dolna,  $D$  - macierz diagonalna,  $U$  - macierz trójkątna górska.

Mnożąc powyższe równanie stronami przez  $\omega \neq 0$ , a następnie dodając stronami macierz  $D$  i odpowiednio grupując wyrazy, równanie przybierze postać:

$$\omega A = (D + \omega L) - ((1 - \omega)D - \omega U).$$

Podstawiając powyższą zależność do równania  $\omega Ax = wb$  (równoważnemu  $Ax = b$ ), otrzymane zostanie:

$$(D + \omega L)x - ((1 - \omega)D - \omega U)x = wb,$$

---

<sup>1</sup>Jest to uogólnienie metody Gaussa-Seidla, również z grupy metod iteracji prostej.

<sup>2</sup>Gdy  $\omega = 1$ , metoda ta staje się metodą Gaussa-Seidla.

a dalej

$$x = (D + \omega L)^{-1}((1 - \omega)D - \omega U)x + \omega(D + \omega L)^{-1}b.$$

Zatem, **wzór iteracyjny** ma postać:

$$x^{(k+1)} = B_{SOR}x^{(k)} + c_{SOR},$$

gdzie

$$B_{SOR} = (D + \omega L)^{-1}((1 - \omega)D - \omega U)$$

oraz

$$c_{SOR} = \omega(D + \omega L)^{-1}b.$$

### Algorytm metody SOR:

```

 $x^{(0)} \leftarrow (x_1^{(0)}, \dots, x_n^{(0)})^T$  – przybliżenie początkowe
for  $k = 0, 1, \dots$ , (dopóki nie będzie spełniony wybrany warunek wyjścia) do
    for  $i = 1, \dots, n$  do
         $x_i^{(k+1)} \leftarrow (1 - \omega)x_i^{(k)} + \omega(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum_{j=i+1}^n a_{ij}x_j^{(k)})/a_{ii}$ 
    end for
end for
```

#### 1.4.1 Parametr relaksacji

Istnieje zależność między promieniem spektralnym  $\rho(B_{SOR})$  a parametrem  $\omega$ , o której mówi poniższe twierdzenie:

**Twierdzenie.** (Kahan) Promień spektralny  $\rho(B_{SOR})$  macierzy iteracji spełnia zależność:  $\rho(B_{SOR}) \geq |\omega - 1|$ .

Z twierdzenia przytoczonego w sekcji 1.3 wiadomo, że aby metoda była zbieżna globalnie, musi zachodzić  $\rho(B_{SOR}) < 1$ . Zatem, na mocy tych dwóch twierdzeń:  $|\omega - 1| \leq \rho(B_{SOR}) < 1$ . Stąd można wyciągnąć następujący wniosek:

**Wniosek.** Jeżeli  $\omega \notin (0, 2)$ , to metoda SOR nie jest zbieżna.

Dlatego, w programie obliczeniowym, rozpatrywane będą jedynie wartości  $\omega \in (0, 2)$ .

## 2 Opis programu obliczeniowego

### 2.1 Omówienie funkcji

Program składa się z 5 funkcji:

- **czyZbiezna** - sprawdza czy dla danej macierzy  $A$  i parametru relaksacji  $\omega$  metoda SOR jest zbieżna globalnie, czyli promień spektralny macierzy iteracji  $B_{SOR} < 1$

Dane wejściowe:

**A** - macierz współczynników układu równań  $Ax = b$ ,

**om** - parametr relaksacji  $\omega$

Dane wyjściowe:

**zbiezna** - czy metoda SOR jest zbieżna globalnie (1-tak, 0-nie)

- **SOR** - funkcja rozwiązuje układ równań  $Ax = b$ , dla parametru relaksacji  $\omega$ , przy pomocy Metody Sukcesywnej Nadrelaksacji (SOR).

Dane wejściowe:

**A** - macierz współczynników układu równań  $Ax = b$ ,

**b** - wektor wyrazów wolnych

**om** - parametr relaksacji  $\omega$

Dane wyjściowe:

**steps** - liczba wykonanych iteracji

**x** - końcowe rozwiązanie układu równań

**xMatr** - macierz z kolejnymi przybliżeniami rozwiązania (kolumny to kolejne wartości  $x^{(k)}$ )

- **wizualizacja** - tworzy wizualizacje kolejnych przybliżeń  $x^{(k)}$ . Na wykresie oś OX reprezentuje numer iteracji, a os OY - przybliżone wartości  $x^{(k)}$  dla k-tej iteracji.

Dane wejściowe:

**A** - macierz współczynników

**b** - wektor wyrazów wolnych

**om** - omega, parametr relaksacji

Dane wyjściowe: (brak)

- **wypisz** - funkcja wypisuje tabelę z porównaniem dokładnego i szacowanego wyniku, w kolejnych kolumnach: real - dokładny wynik, approx. - szacowany wynik, blad - błąd bezwzględny

Dane wejściowe:

**xDokl** - dokładny wynik

**x** - szacowany wynik

Dane wyjściowe:

**Out** - tabela z porównaniem dokładnego i szacowanego wyniku

- **wizIter** - tworzy wizualizacje liczby wykonanych iteracji potrzebnych do rozwiązania układu równań  $Ax=b$ , względem parametru relaksacji  $\omega$ . Wykres: OX - omega na przedziale (0,2), OY - liczba wykonanych iteracji

Dane wejściowe:

**A** - macierz współczynników

**b** - wektor wyrazów wolnych

Dane wyjściowe: (brak)

Oprócz tego, program zawiera skrypt **test.m** z przykładami.

## 2.2 Prezentacja działania programu na przykładzie

Dany jest układ równań  $Ax = b$ , gdzie

$$A = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \quad b = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}.$$

Dokładne rozwiązań tego układu jest równe:

$$x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}.$$

Przy rozwiązywaniu go za pomocą metody SOR, zdefiniowanej wzorem:

$$x^{(k+1)} = (D + \omega L)^{-1}((1 - \omega)D - \omega U)x^{(k)} + \omega(D + \omega L)^{-1}b,$$

dla  $\omega = 1, 2$ , kolejne przybliżenia będą równe:

$$x^{(2)} = \begin{bmatrix} 0,6000 \\ 1,9800 \\ 1,1940 \end{bmatrix}, \quad x^{(3)} = \begin{bmatrix} 1,0740 \\ 2,0844 \\ 0,9865 \end{bmatrix}, \quad x^{(4)} = \begin{bmatrix} 1,0105 \\ 1,9822 \\ 0,9974 \end{bmatrix}, \quad x^{(5)} = \begin{bmatrix} 0,9926 \\ 2,0005 \\ 1,0007 \end{bmatrix}, \dots$$

Natomiast dla  $\omega = 1, 1$ :

$$x^{(2)} = \begin{bmatrix} 0,5500 \\ 1,8013 \\ 1,0453 \end{bmatrix}, \quad x^{(3)} = \begin{bmatrix} 0,9903 \\ 2,0297 \\ 1,0036 \end{bmatrix}, \quad x^{(4)} = \begin{bmatrix} 1,0091 \\ 2,0005 \\ 0,9998 \end{bmatrix}, \quad x^{(5)} = \begin{bmatrix} 0,9998 \\ 1,9997 \\ 0,9999 \end{bmatrix}, \dots$$

Z powyższych zestawień wynika więc, że metoda SOR szybko zbiega oraz w przypadku  $\omega = 1, 1$  zbiega szybciej niż dla  $\omega = 1, 2$ .

Zatem, przy odpowiednim doborze parametru  $\omega$ , metoda ta może być bardzo efektywna.

### 3 Przykłady obliczeniowe

W niniejszej sekcji przeanalizowane zostanie działanie programu na poniższych przykładach:

$$1. \ A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}, \ b = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

$$2. \ A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \ b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix},$$

$$3. \ A = \begin{bmatrix} 3 & -2 & 0 \\ 4 & 3 & 0 \\ -1 & 1 & -1 \end{bmatrix}, \ b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},$$

$$4. \ A = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 2 \end{bmatrix}, \ b = \begin{bmatrix} 3 \\ -1 \\ -2 \\ 1 \end{bmatrix},$$

$$5. \ A = \begin{bmatrix} 5 & 5 & 2 \\ 2 & 3 & -4 \\ 0 & 9 & 1 \end{bmatrix}, \ b = \begin{bmatrix} 60 \\ 33 \\ 8 \end{bmatrix},$$

$$6. \ A = \begin{bmatrix} 2 & 5 & 0 \\ 0 & 2 & 2 \\ 1 & 0 & 9 \end{bmatrix}, \ b = \begin{bmatrix} 36 \\ 30 \\ 84 \end{bmatrix},$$

$$7. \ A = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, \ b = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix},$$

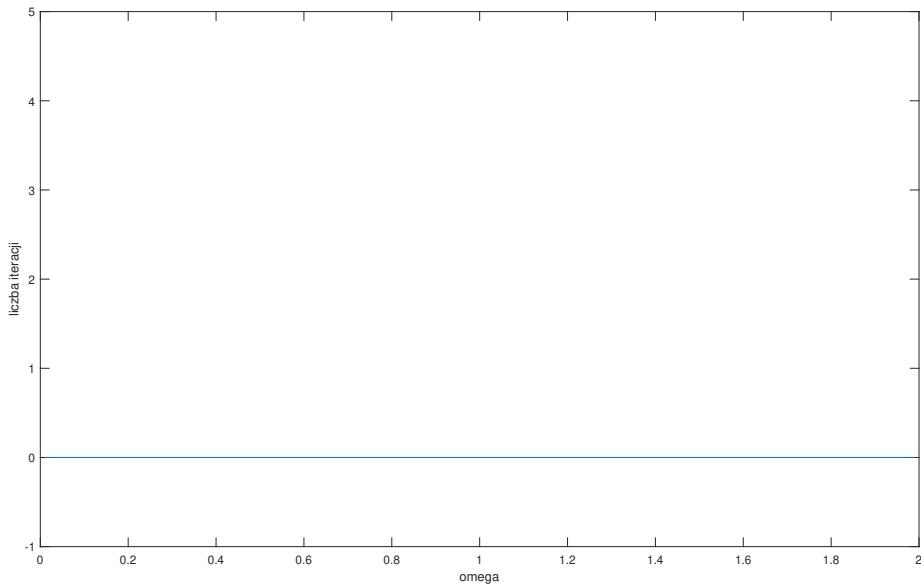
$$8. \ A = \begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^{10} \\ 1 & 2 & 2^2 & \dots & 2^{10} \\ 1 & 3 & 3^2 & \dots & 3^{10} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 10 & 10^2 & \dots & 10^{10} \end{bmatrix}, \ b = \begin{bmatrix} 10 \\ \dots \\ 10 \end{bmatrix},$$

9.  $A$  - symetryczna i dodatnio określona macierz  $\mathbb{R}^{50 \times 50}$ ,  $b$  - wektor  $\mathbb{R}^{50}$ .

### 3.1 Przykład 1

$$A = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

W tym przykładzie, metoda SOR nie jest zbieżna<sup>3</sup> dla żadnego  $\omega$  (z przedziału  $(0, 2)$ ).



Rysunek 1: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Zatem, funkcja zwraca pustą macierz i wykonanych zostaje 0 iteracji dla każdej  $\omega$ .

---

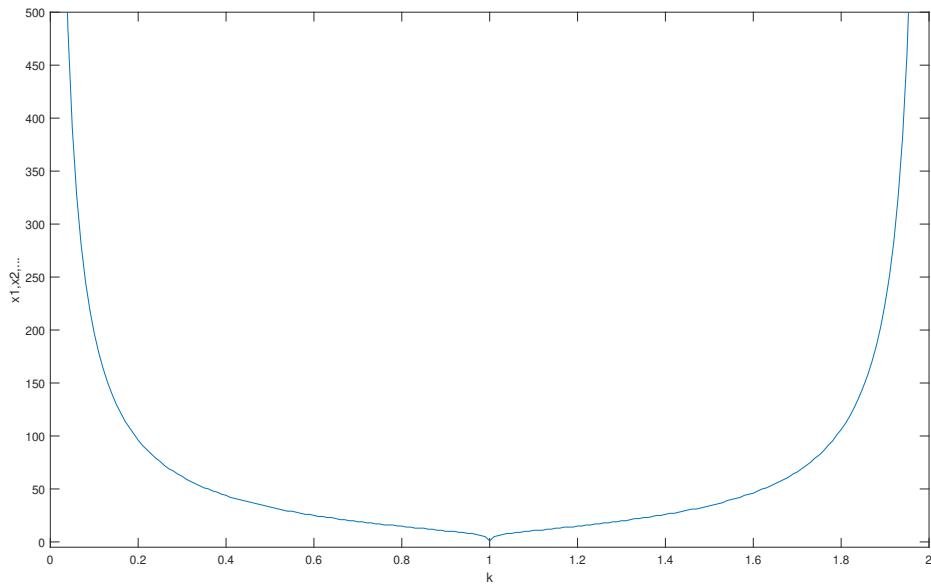
<sup>3</sup>Nie jest ona zbieżna, gdyż promień spektralny macierzy iteracyjnej  $B_{SOR}$  jest większy niż 1.

### 3.2 Przykład 2

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Ten przykład jest szczególnym przypadkiem, gdyż wynik można otrzymać bez żadnych obliczeń ( $x = b = [1, 1, 1, 1]^T$ ). W tym wypadku, nieprawidłowo dobrana wartość  $\omega$  w metodzie SOR może kosztować dużą liczbą operacji, a w efekcie być tylko utrudnieniem w rozwiązywaniu równania.

Poniższy wykres jest symetryczny<sup>4</sup> względem prostej  $\omega = 1$ , a dla  $\omega = 1$  potrzebna jest 1 iteracja.



Rysunek 2: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

---

<sup>4</sup>Nie jest dokładnie symetryczny, ale na wykresie tak się może wydawać.

Jak widać w poniższej tabeli, dla  $\omega = 1$  szacowany wynik jest równy poprawnemu.

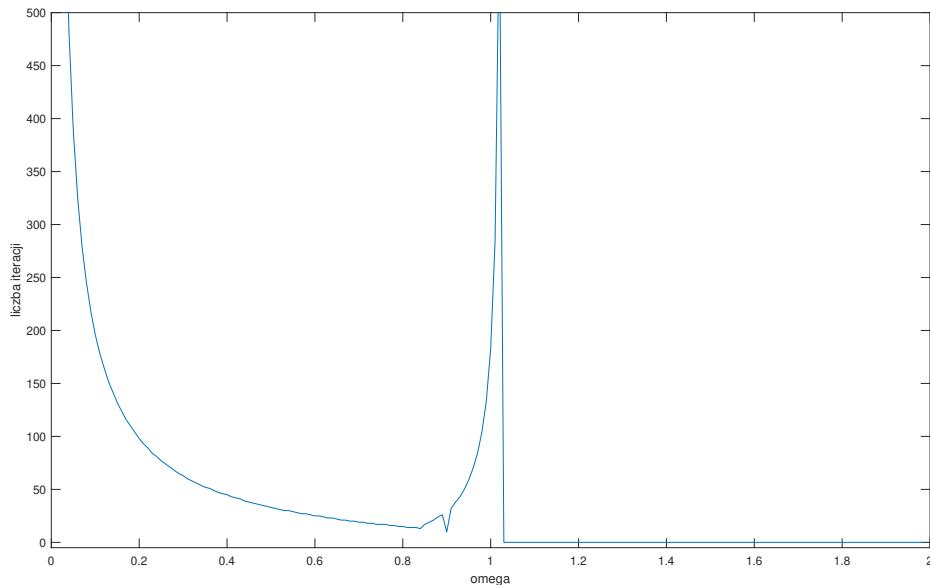
Tabela 1: Porównanie wartości  $x_i$  dla  $\omega = 1$ .

rozwiązanie dokładne	rozwiązanie przybliżone	błąd bezwzględny
1.0000	1.0000	$0.0000e + 00$
1.0000	1.0000	$0.0000e + 00$
1.0000	1.0000	$0.0000e + 00$
1.0000	1.0000	$0.0000e + 00$

### 3.3 Przykład 3

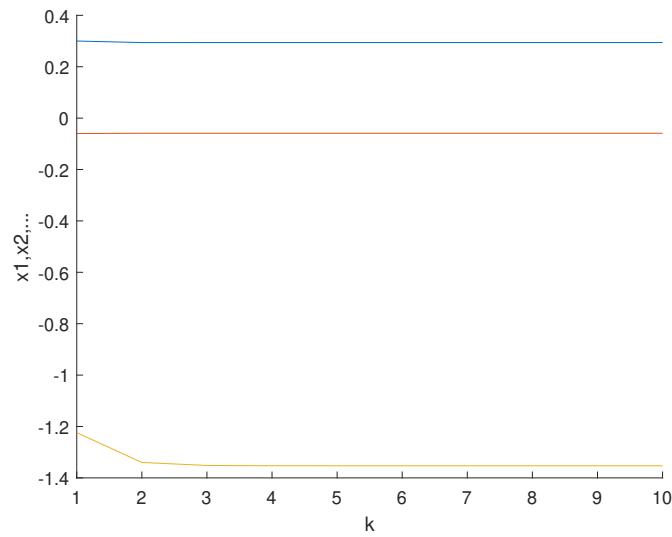
$$A = \begin{bmatrix} 3 & -2 & 0 \\ 4 & 3 & 0 \\ -1 & 1 & -1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

W tym przykładzie, w  $\omega \approx 1,01$  występuje duży uskok - dla tej wartości potrzebna jest duża liczba iteracji, a najmniejsza jest dla  $\omega \approx 0,9$ . Dla  $\omega > 1,01$  metoda nie jest zbieżna.



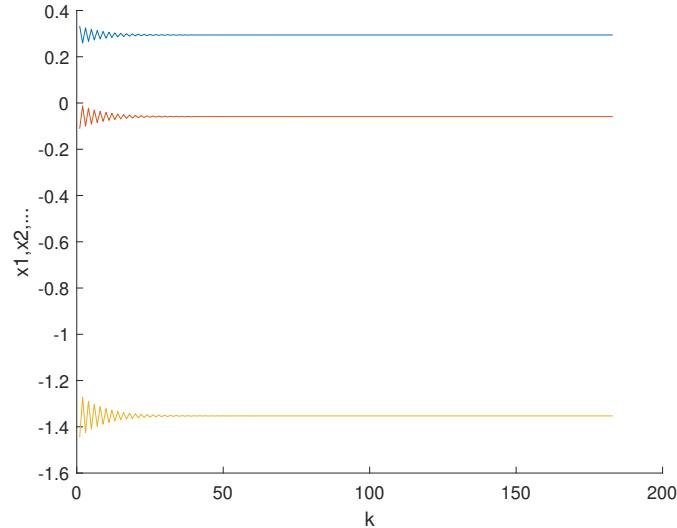
Rysunek 3: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Rozwiązaniem tego układu równań jest  $x = \begin{bmatrix} 0,2941 \\ -0,0588 \\ -1,3529 \end{bmatrix}$ .



Rysunek 4: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 0, 9$ .

Z powyższego wykresu widać, że dla  $\omega = 0, 9$  metoda ta niemalże natychmiast<sup>5</sup> odnajduje właściwe rozwiązanie.



Rysunek 5: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 1$ .

Natomiast z Rysunku 5, dla  $\omega = 1$  początkowe szacowania są nieprecyzyjne - jest na zmianę za duże i za małe, przez co potrzeba więcej iteracji<sup>6</sup> do „stabilizacji” wyniku.

---

<sup>5</sup>Program wykonał 10 iteracji.

<sup>6</sup>Program wykonał 183 iteracji.

Tabela 2: Porównanie wartości  $x_i$  dla  $\omega = 0, 9$ .

rozwiązań dokładne	rozwiązań przybliżone	błąd bezwzględny
0.2941	0.2941	$0.0000e + 00$
-0.0588	-0.0588	$6.9389e - 18$
-1.3529	-1.3529	$1.3000e - 11$

Tabela 3: Porównanie wartości  $x_i$  dla  $\omega = 1$ .

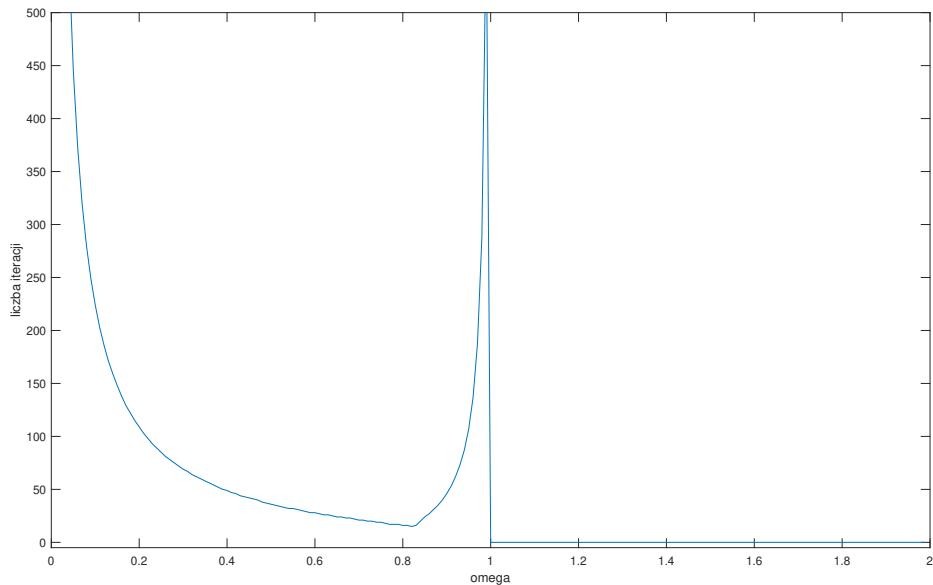
rozwiązań dokładne	rozwiązań przybliżone	błąd bezwzględny
0.2941	0.2941	$1.7082e - 11$
-0.0588	-0.0588	$2.2776e - 11$
-1.3529	-1.3529	$3.9859e - 11$

Z powyższych tabel widać, że dla  $\omega = 0, 9$  wynik jest bardziej precyzyjny niż dla  $\omega = 1$ . Dzieje się tak, gdyż w drugim wypadku metoda zbiega dużo wolniej.

### 3.4 Przykład 4

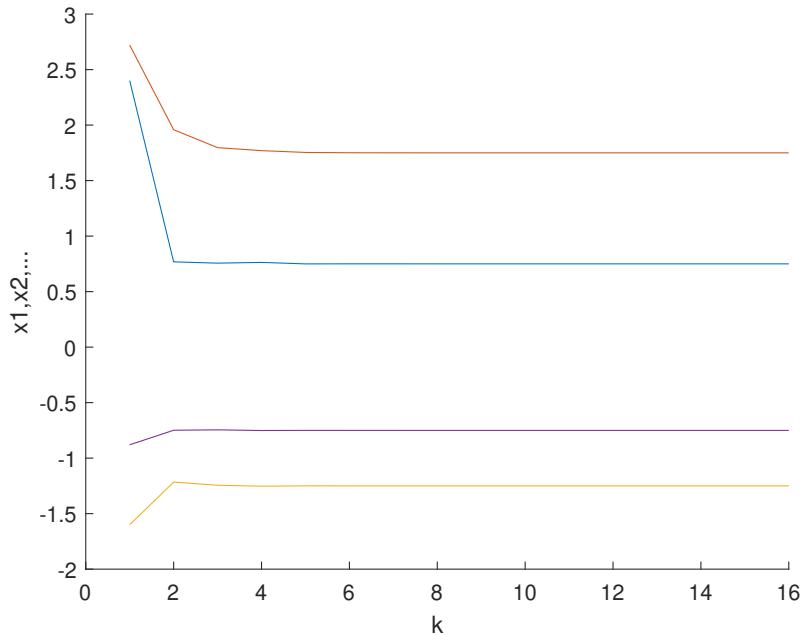
$$A = \begin{bmatrix} 1 & 0 & 0 & -3 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 2 \end{bmatrix}, \quad b = \begin{bmatrix} 3 \\ -1 \\ -2 \\ 1 \end{bmatrix}$$

W tym przykładzie, liczba potrzebnych iteracji maleje dla  $0 < \omega \leq 0,8$ , a dla  $\omega$  bliskiego od lewej strony 1 liczba ta znacznie wzrasta, aż dla  $\omega \geq 1$  metoda już nie jest zbieżna.



Rysunek 6: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Rozwiązaniem tego układu równań jest  $x = \begin{bmatrix} 0,7500 \\ 1,7500 \\ -1,2500 \\ -0,7500 \end{bmatrix}$ .



Rysunek 7: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 0,8$ .

Tabela 4: Porównanie wartości  $x_i$  dla  $\omega = 0,8$ .

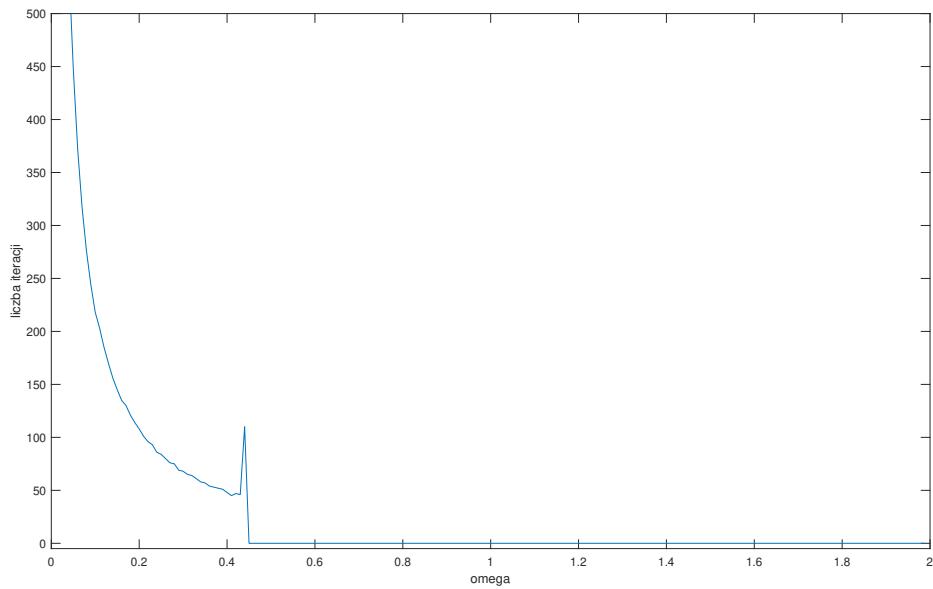
rozwiązanie dokładne	rozwiązanie przybliżone	błąd bezwzględny
0.7500	0.7500	$1.2504e - 12$
1.7500	1.7500	$5.1133e - 11$
-1.2500	-1.2500	$1.7275e - 12$
-0.7500	-0.7500	$9.9987e - 13$

Program wykonał tylko 16 iteracji i błąd bezwzględny między szacowanymi a prawdziwymi wartościami nie przekracza  $10^{-10}$ .

### 3.5 Przykład 5

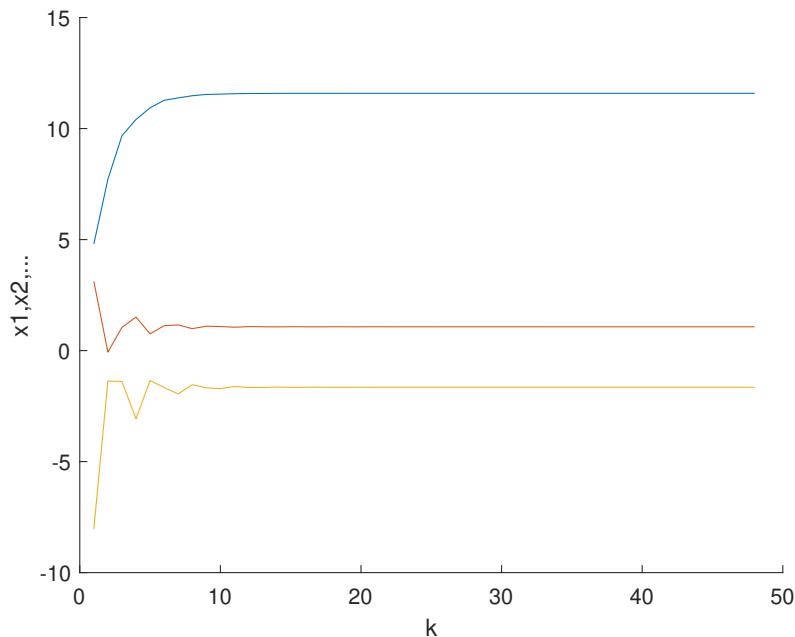
$$A = \begin{bmatrix} 5 & 5 & 2 \\ 2 & 3 & -4 \\ 0 & 9 & 1 \end{bmatrix}, \quad b = \begin{bmatrix} 60 \\ 33 \\ 8 \end{bmatrix}$$

W tym przypadku, liczba potrzebnych iteracji maleje dla  $0 < \omega \leq 0,4$ , następnie następuje mały uskok i dla  $\omega \geq 0,45$  metoda nie jest zbieżna.



Rysunek 8: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Rozwiązaniem tego układu równań jest  $x = \begin{bmatrix} 11,5882 \\ 1,0724 \\ -1,6516 \end{bmatrix}$ .



Rysunek 9: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 0, 4$ .

Tabela 5: Porównanie wartości  $x_i$  dla  $\omega = 0, 4$ .

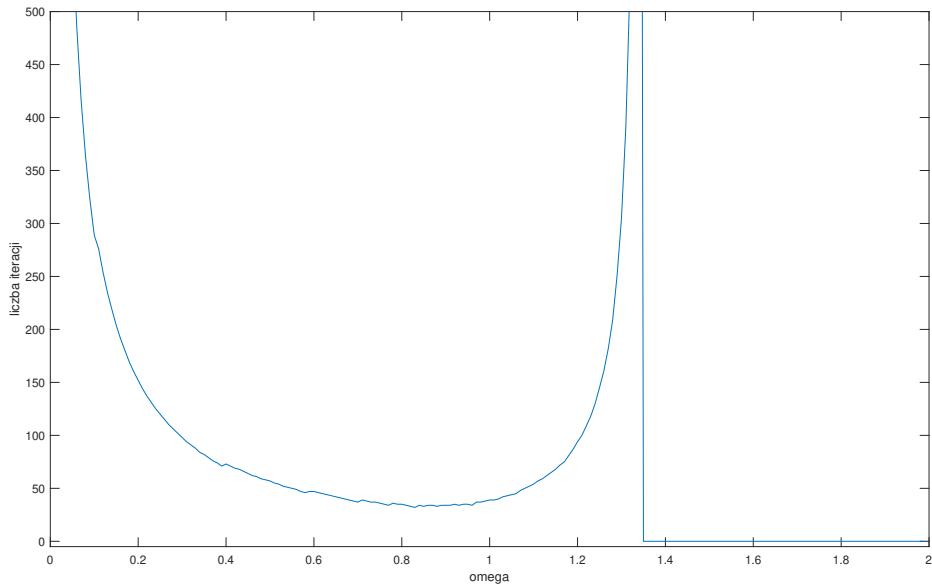
rozwiązanie dokładne	rozwiązanie przybliżone	błąd bezwzględny
11.5882	11.5882	$5.4495e - 11$
1.0724	1.0724	$1.2340e - 11$
-1.6516	-1.6516	$3.4612e - 10$

Program wykonał 48 iteracji i błąd bezwzględny między szacowanymi a prawdziwymi wartościami nie przekracza  $10^{-9}$ .

### 3.6 Przykład 6

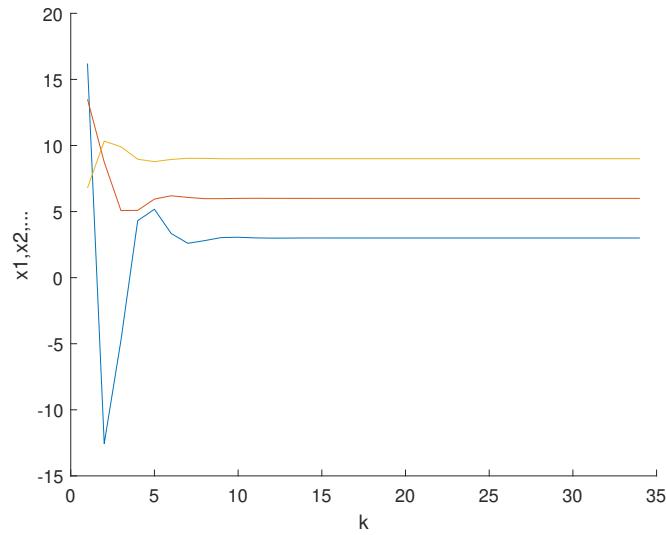
$$A = \begin{bmatrix} 2 & 5 & 0 \\ 0 & 2 & 2 \\ 1 & 0 & 9 \end{bmatrix}, \quad b = \begin{bmatrix} 36 \\ 30 \\ 84 \end{bmatrix}$$

Najmniejsza liczba potrzebnych iteracji jest dla  $\omega \approx 0,9$ , a dla  $\omega \geq 1,35$  metoda nie jest zbieżna.

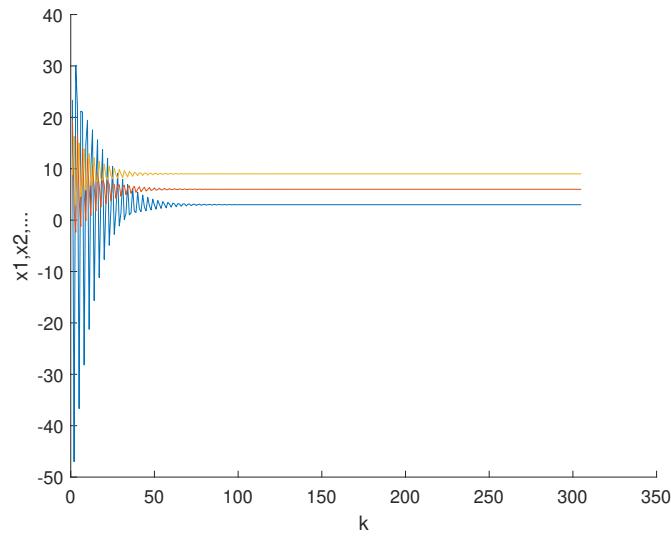


Rysunek 10: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Rozwiązaniem tego układu równań jest  $x = \begin{bmatrix} 3 \\ 6 \\ 9 \end{bmatrix}$ . Poniżej zestawione jest porównanie metody SOR dla  $\omega = 0,9$  oraz dla  $\omega = 1,3$ :



Rysunek 11: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 0, 9$ .



Rysunek 12: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 1, 3$ .

Dla  $\omega = 0, 9$  program wykonał 34 iteracje. Jednak, dla  $\omega = 1, 3$  (podobnie jak w przykładzie 3) początkowe szacowania są nieprecyzyjne, przez co program musiał wykonać więcej iteracji, bo aż 305 - jest to niemalże 10 razy tyle co dla  $\omega = 0, 9$ .

Tabela 6: Porównanie wartości  $x_i$  dla  $\omega = 0, 9$ .

rozwiązańe dokładne	rozwiązańe przybliżone	błąd bezwzględny
3.0000	3.0000	$6.2801e - 10$
6.0000	6.0000	$9.4893e - 12$
9.0000	9.0000	$6.4629e - 11$

Tabela 7: Porównanie wartości  $x_i$  dla  $\omega = 1, 3$ .

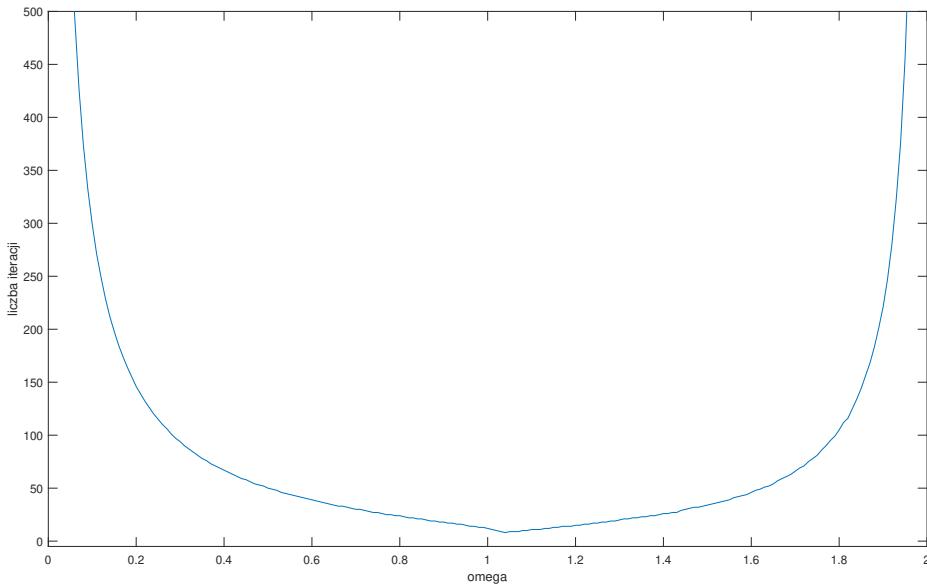
rozwiązańe dokładne	rozwiązańe przybliżone	błąd bezwzględny
3.0000	3.0000	$4.5162e - 10$
6.0000	6.0000	$3.9021e - 10$
9.0000	9.0000	$7.6117e - 12$

Choć metoda zbiega wolniej dla  $\omega = 1, 3$ , to oba wyniki mają podobne precyzje.

### 3.7 Przykład 7

$$A = \begin{bmatrix} 4 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}, b = \begin{bmatrix} 2 \\ 6 \\ 2 \end{bmatrix}$$

Ten przykład został przytoczony w sekcji 2.2<sup>7</sup>. Dobra tam wartość  $\omega$  nie była przypadkowa - dla  $\omega = 1,1$  metoda najszybciej zbiega, co można zauważyć na poniższym wykresie:



Rysunek 13: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Dla tej wartości  $\omega$ , program wykonuje jedynie 11 iteracji, a przykładowo dla  $\omega = 0,2$  aż 146 iteracji.

Rozwiązaniem tego układu równań jest  $x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$ .

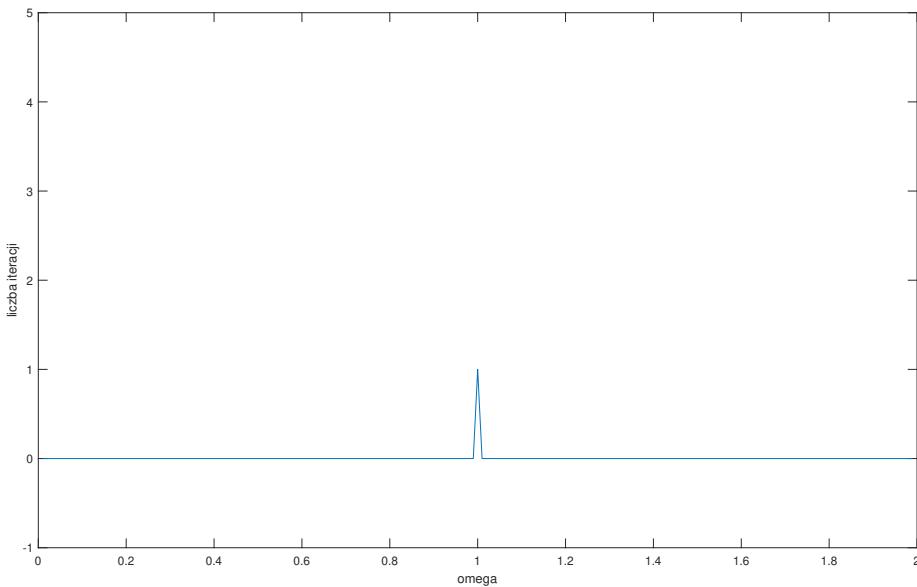
---

<sup>7</sup> „Prezentacja działania programu na przykładzie”

### 3.8 Przykład 8

$$A = \begin{bmatrix} 1 & 1 & 1^2 & \dots & 1^{10} \\ 1 & 2 & 2^2 & \dots & 2^{10} \\ 1 & 3 & 3^2 & \dots & 3^{10} \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 10 & 10^2 & \dots & 10^{10} \end{bmatrix}, \quad b = \begin{bmatrix} 10 \\ \dots \\ 10 \end{bmatrix}.$$

Podobnie jak w przykładzie 2, dla  $\omega = 1$  potrzebna jest tylko jedna iteracja. Dla pozostałych wartości metoda nie jest nawet zbieżna.



Rysunek 14: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Rozwiązaniem tego układu równań jest  $x = \begin{bmatrix} 10 \\ 0 \\ \dots \\ 0 \end{bmatrix}$ .

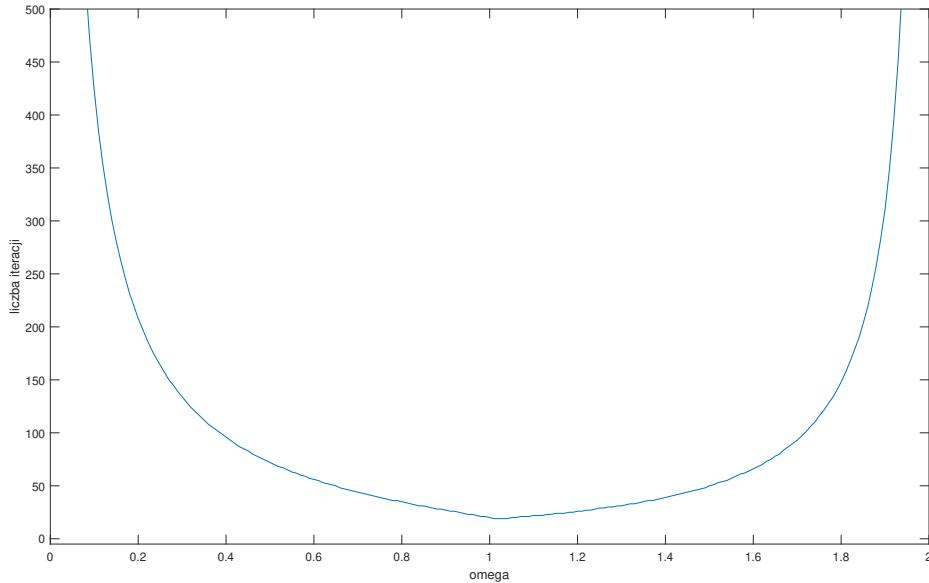
Tabela 8: Porównanie wartości  $x_i$  dla  $\omega = 1$ .

rozwiązańe dokładne	rozwiązańe przybliżone	błąd bezwzględny
10.0000	10.0000	$0.0000e + 00$
0.0000	0.0000	$0.0000e + 00$
-0.0000	0.0000	$0.0000e + 00$
0.0000	0.0000	$0.0000e + 00$
-0.0000	0.0000	$0.0000e + 00$
0.0000	0.0000	$0.0000e + 00$
0.0000	0.0000	$0.0000e + 00$
-0.0000	0.0000	$0.0000e + 00$
-0.0000	0.0000	$0.0000e + 00$

Szacowny wynik nie różni się od dokładniejszego.

### 3.9 Przykład 9

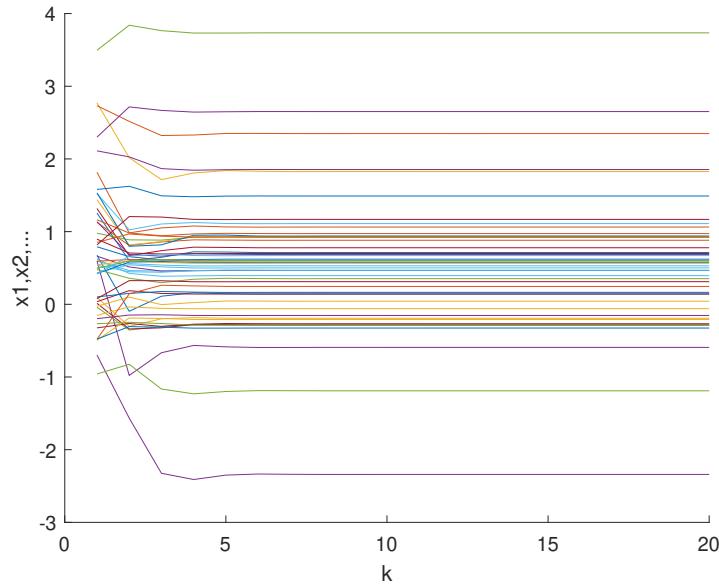
$A$  - symetryczna i dodatnio określona macierz  $\mathbb{R}^{50 \times 50}$ ,  $b$  - wektor  $\mathbb{R}^{50}$ .



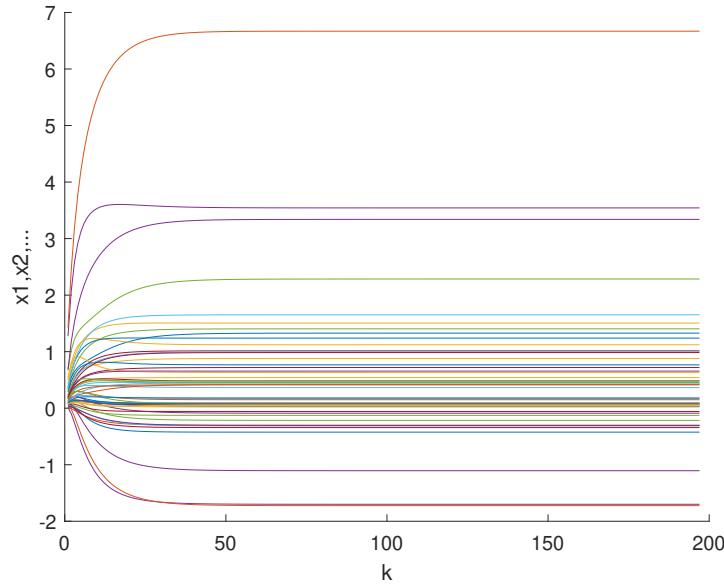
Rysunek 15: Przedstawia liczbę wykonanych iteracji, względem parametru relaksacji  $\omega$ .

Dla wielu przykładów tego typu, wykres liczby iteracji względem parametru relaksacji  $\omega$  będzie wyglądał podobnie (jak w przykładzie 7). Jeżeli macierz jest symetryczna i dodatnio określona, to metoda SOR jest zbieżna globalnie wtedy i tylko wtedy, gdy  $0 < \omega < 2$ . Stąd również widać na wykresie, że metoda jest zbieżna na całym przedziale  $(0, 2)$ .

Wykres przybiera kształt paraboli z końcami uniesionymi do góry, a dla pewnej  $\omega$  o drobinkę większej niż 1 liczba iteracji jest najmniejsza.



Rysunek 16: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 1,05$ .



Rysunek 17: Przedstawia szacowania  $x_i^{(k)}$  dla kolejnych iteracji  $k$ ,  $\omega = 0,2$ .

Dla  $\omega = 1,05$  program wykonał 20 iteracji. Jednak, dla  $\omega = 0,2$  wykonał ich 197, ale jak widać z rysunku 17, szacowania te są dobre (zbiegają do prawdziwych wyników z małym „krokiem”, przez co wykres wygląda na „gładki” a nie „kanciasty”, w przeciwieństwie do przykładu 6: na rysunku 12 widać jak wartości te na początku na zmianę rosną i maleją).

Tabela 9: Porównanie wartości  $x_i$  dla  $\omega = 1, 05$ .

rozwiązańe dokładne	rozwiązańe przybliżone	błąd bezwzględny
0.3420	0.3420	$6.2600e - 11$
-0.2414	-0.2414	$5.0800e - 11$
0.3396	0.3396	$2.6605e - 11$
0.8201	0.8201	$1.1031e - 10$
0.8331	0.8331	$1.6482e - 11$
0.2520	0.2520	$4.4226e - 11$
0.4227	0.4227	$5.1159e - 11$
-0.0121	-0.0121	$8.1671e - 11$
0.4726	0.4726	$3.0884e - 11$
2.3064	2.3064	$3.3882e - 11$
0.4109	0.4109	$5.8131e - 11$
1.0083	1.0083	$3.3759e - 11$
0.1291	0.1291	$2.7896e - 12$
-0.2039	-0.2039	$4.1645e - 11$
0.6137	0.6137	$1.2747e - 10$
-0.7044	-0.7044	$1.8848e - 11$
-2.2761	-2.2761	$1.3648e - 10$
0.9986	0.9986	$4.9394e - 12$
0.8781	0.8781	$2.2039e - 11$
0.3246	0.3246	$7.8212e - 12$
-0.4395	-0.4395	$5.5343e - 12$
0.4072	0.4072	$5.6863e - 12$
-0.4994	-0.4994	$3.0154e - 11$
0.9755	0.9755	$4.1936e - 11$
2.8883	2.8883	$1.6616e - 10$
8.5168	8.5168	$1.8021e - 10$
...	...	...

Błąd bezwzględny szacowań nie przekracza  $10^{-9}$ .

## 4 Analiza wyników

Najistotniejszy w tej metodzie jest odpowiedni dobór wartości  $\omega$  - zapewniając to, można otrzymać ogromną przewagę czasową nad innymi metodami iteracyjnymi rozwiązywania układów równań, m.in metodą Gaussa-Seidla, czyli gdy  $\omega = 1$ .

Jednak dla zle dobranych wartości, korzystanie z metody SOR może mieć wręcz odwrotny skutek (tak jak to było w przykładzie 3 dla  $\omega = 1$ , czy przykładzie 6 dla  $\omega = 1, 3$ ).

Wybór współczynnika relaksacji  $\omega$  niekoniecznie jest łatwy i zależy od właściwości macierzy współczynników. Przykładowo, możliwe jest wyprowadzenie jego optymalnej wartości, gdy spełnione są następujące warunki:

- parametr relaksacji jest właściwy:  $\omega \in (0, 2)$
- Macierz iteracji Jacobiego  $C_{\text{Jac}} := I - D^{-1}A$  ma tylko rzeczywiste wartości własne
- Metoda Jacobiego jest zbieżna:  $\mu := \rho(C_{\text{Jac}}) < 1$
- rozkład macierzy  $A = D + L + U$  spełnia właściwość:  
$$\det(\lambda D + zL + \frac{1}{z}U) = \det(\lambda D + L + U) \text{ dla dowolnych } z \in \mathbb{C} \setminus \{0\} \text{ i } \lambda \in \mathbb{C}.$$

Wówczas optymalna wartość parametru relaksacji  $\omega$  jest równa <sup>8</sup>:

$$\omega_{\text{opt}} := 1 + \left( \frac{\mu}{1 + \sqrt{1 - \mu^2}} \right)^2.$$

---

<sup>8</sup>[https://en.wikipedia.org/wiki/Successive\\_over-relaxation#Convergence\\_Rate](https://en.wikipedia.org/wiki/Successive_over-relaxation#Convergence_Rate)

## 5 Podsumowanie

Na podstawie przeprowadzonej analizy działania programu, można wysnuć następujące wnioski:

- Metoda Sukcesywnej Nadrelaksacji (SOR) może być dla danego układu równań szybciej lub wolniej zbieżna do rozwiązań niż metoda Gaussa-Seidla (czyli gdy  $\omega = 1$ ). Wszystko zależy od doboru współczynnika relaksacji w stosunku do układu równań.
- Nie wszystkie układy równań są zbieżne na całym przedziale dopuszczalnego współczynnika relaksacji (przedział  $(0, 2)$  ).
- Odpowiedni dobór współczynnika relaksacji pozwala nam uzyskać zbieżność dla układów, które nie spełniają warunku zbieżności w podstawowej metodzie Gaussa-Seidla.
- Dla macierzy symetrycznych i dodatnio określonych (jak z przykładu 7 i 9) najczęściej najbardziej optymalne wartości  $\omega$  są niewiele większe od 1.