

Sprawozdanie projekt 2

Temat projektu:

Rozwiązywanie układu równań liniowych $Ax=b$ metodą SOR

Matematyczny opis metody

6.8 Metody iteracyjne rozwiązywania układów równań liniowych

Niech, jak poprzednio, $Ax = b$ będzie układem równań liniowych, który chcemy rozwiązać, przy czym $A \in \mathbb{R}^{n \times n}$ jest nieosobliwa a $x, b \in \mathbb{R}^n$.

Idea metod iteracyjnych jest następująca: startując z danego przybliżenia początkowego

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

tworzymy ciąg kolejnych przybliżeń $\{x^{(k)}\}$, taki, że $x^{(k)} \rightarrow x$ przy $k \rightarrow \infty$.

Każdy element tego ciągu jest wektorem (tj. $x^{(k)} \in \mathbb{R}^n$ dla $k = 0, 1, \dots$). Zbieżność jest rozumiana w sensie normy, tj. $\|x^{(k)} - x\| \rightarrow 0$.

6.8.2 Możliwe warunki zakończenia obliczeń

Niech d będzie parametrem definiującym dokładność (niewielką liczbą rzeczywistą dodatnią). Jako kryterium zakończenia obliczeń można użyć jednego z następujących warunków:

- 1) $\|Ax^{(k)} - b\| < d$ (warunek bardziej kosztowny niż następne – wymaga obliczenia iloczynu $Ax^{(k)}$),
- 2) $\|x^{(k+1)} - x^{(k)}\| < d$ („błąd” bezwzględny),
- 3) $\|x^{(k+1)} - x^{(k)}\| < d\|x^{(k)}\|$ („błąd” względny),
- 4) (warunek Gilla) $\|x^{(k+1)} - x^{(k)}\| < d_1\|x^{(k)}\| + d_2$,

gdzie d_1 i d_2 to parametry określające dokładność (liczby dodatnie, bliskie zeru, przy czym $d_1 > d_2$), przykładowo $d_1 = 10^{-10}$ i $d_2 = 10^{-20}$ lub $d_1 = 10^{-13}$ i $d_2 = 10^{-25}$ a $\|\cdot\|$ jest dowolną normą wektorową.

6.8.5 Metoda SOR

Nazwa SOR (ang. **S**uccessive **O**ver**R**elaxation), nazywana też metodą nadrelaksacji, jest uogólnieniem metody Gaussa-Seidla.

W metodzie tej występuje parametr $\omega \in \mathbb{R}$, zwany parametrem relaksacji.

Algorytm (metoda SOR)

$x^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})^T$ – przybliżenie początkowe

for $k = 0, 1, \dots$, (dopóki nie będzie spełniony wybrany warunek stopu)

for $i = 1, 2, \dots, n$

$$x_i^{(k+1)} = (1 - \omega)x_i^{(k)} + \omega(b_i - \sum_{j=1, j < i}^n a_{ij}x_j^{(k+1)} - \sum_{j=1, j > i}^n a_{ij}x_j^{(k)})/a_{ii}$$

end

end

Wzór na i -ty element wektora $X^{(k+1)}$ - $(k+1)$ -tego przybliżenia wektora X :

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

Jeżeli stworzymy wektor $X_{k,k+1}$ taki, że w momencie obliczenia $x_i^{(k+1)}$ to:

$$X_{k,k+1} = [x_1^{(k+1)}, x_2^{(k+1)}, \dots, x_{i-1}^{(k+1)}, x_i^{(k)}, x_{i+1}^{(k)}, \dots, x_n^{(k)}]$$

Oznaczenie: $X_{k,k+1}[i]$ to i -ty element wektora $X_{k,k+1}$, to:

$$\begin{aligned} \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k+1)} + \sum_{j=1, j < i}^n a_{ij}x_j^{(k)} &= \sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \sum_{j=i+1}^n a_{ij}x_j^{(k)} = \\ &= (a_{i1}x_1^{(k+1)} + a_{i2}x_2^{(k+1)} + \dots + a_{i(i-1)}x_{i-1}^{(k+1)}) + (a_{i(i+1)}x_{i+1}^{(k)} + a_{i(i+2)}x_{i+2}^{(k)} + \dots + a_{in}x_n^{(k)}) = \\ &= a_{i1} \cdot X_{k,k+1}[1] + a_{i2} \cdot X_{k,k+1}[2] + \dots + a_{i(i-1)} \cdot X_{k,k+1}[i-1] + a_{i(i+1)} \cdot X_{k,k+1}[i+1] + \dots + a_{in} \cdot X_{k,k+1}[n] = \\ &= \sum_{j=1}^{i-1} a_{ij} \cdot X_{k,k+1}[j] + \sum_{j=i+1}^n a_{ij} \cdot X_{k,k+1}[j] + a_{ii} \cdot X_{k,k+1}[i] - a_{ii} \cdot X_{k,k+1}[i] = \\ &= \sum_{j=1}^n a_{ij} \cdot X_{k,k+1}[j] - a_{ii} \cdot X_{k,k+1}[i] \end{aligned}$$

Zatem wzór na $x_i^{(k+1)}$ to:

$$x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \omega \left(b_i - \left(\sum_{j=1}^n a_{ij} \cdot X_{k,k+1}[j] - a_{ii} \cdot X_{k,k+1}[i] \right) \right) / a_{ii}$$

Rysunek 2: Wyprowadzenie wzoru na obliczanie i -tego elementu k -tego przybliżenia wektora X , użytego w implementacji funkcji

Opis programu

Podczas działania programu czasami wartości wektorów np. x_k przyjmują tak duże wartości, że MATLAB zamienia je na wartość Inf , co powoduje błędy w działaniu programu (pojawienie się NaN) tam, gdzie występuje mnożenie wartości $Inf * 0$. Dlatego w takiej sytuacji ustalam wynik na wartość 0 (Bo to działanie to w rzeczywistości bardzo duża liczba, ale skończona razy 0).

Program zawiera funkcję `iteracyjneSor`, która przyjmuje sześć argumentów:

- A – nieosobliwa macierz $\mathbb{R}^{n \times n}$
- b – wektor \mathbb{R}^n
- ω – parametr relaksacji
- d – dokładność
- `warunek` – numer warunku stopu dla metody SOR (domyślna wartość `warunek = 1`)

Numery warunków:

- 1 -> $\|x_k - x_{k-1}\| < d$
 - 2 -> $\|Ax_k - b\| < d$
 - 3 -> $\|x_k - x_{k-1}\| / \|x_{k-1}\| < d$
 - 4 -> $\|x_k - x_{k-1}\| < d \|x_{k-1}\| + d2$
- $d2$ – druga dokładność, gdy stosujemy warunek Gilla (`warunek == 4`)
(domyślna wartość $d2 = d - 0.1d$)

i zwraca wektor x – przybliżoną wartość wektora rozwiązań układu równań $Ax=b$ z dokładnością d .

Funkcja *iteracyjneSor* wykorzystuje funkcję pomocniczą *czySpełniaWarunekStopu*, która przyjmuje pięć argumentów:

- x_k – przybliżony wektor rozwiązań X w k -tej iteracji metody
- x_{k-1} – przybliżony wektor rozwiązań X w $(k-1)$ -tej iteracji metody
- A – macierz $\mathbb{R}^{n \times n}$
- b – wektor \mathbb{R}^n
- *typ* – numer warunku stopu dla metody SOR (domyślna wartość *typ* = 1)

Typy warunków:

- 1 -> $c = \|x_k - x_{k-1}\|$
- 2 -> $c = \|Ax_k - b\|$
- 3 -> $c = \|x_k - x_{k-1}\| / \|x_{k-1}\|$
- 4 -> $c = \|x_k - x_{k-1}\|$

a zwraca c , czyli wartość błędu dla określonego typu warunku.

Działanie funkcji *czySpełniaWarunekStopu*:

1. Sprawdza wartość argumentu *typ*
2. Wyznacza wartość c , ze wzoru (podany powyżej) dla podanego typu
3. Zwraca wartość c

Funkcja *iteracyjneSor* wykorzystuje też funkcję pomocniczą *WartoscXk*, która przyjmuje cztery argumenty:

- x_k – przybliżony wektor rozwiązań x wyznaczony w $(k-1)$ -tej iteracji metody
- A – macierz $\mathbb{R}^{n \times n}$
- b – wektor \mathbb{R}^n
- w – parametr relaksacji

a zwraca nowy wektor x_k , czyli przybliżony wektor rozwiązań x wyznaczony w k -tej iteracji metody

Funkcja *WartoscXk* wykorzystuje funkcję pomocniczą *policzSume*, która przyjmuje trzy argumenty:

- i – indeks elementu wektora x , który przybliżamy
- x_k – przybliżony wektor rozwiązań x , gdzie elementy o indeksach do i zostały wyznaczone w k -tej iteracji metody, a elementy o indeksach większych od i zostały wyznaczone w $(k-1)$ -tej iteracji.
- A – macierz $\mathbb{R}^{n \times n}$

a zwraca wartość zmiennej *suma*, czyli wartość sumy $\sum_{j=1, j \neq i}^n a_{ij} x_{kj}$.

Działanie funkcji *policzSume*:

1. Ustawia wartość zmiennej *suma* = 0
2. Przechodzi po wartościach $j = 1:n$
3. Pomija iterację, gdzie $j == i$
4. W pozostałych iteracjach dodaje do sumy wartość wyrażenia $a_{ij} x_{kj}$
5. Po skończeniu pętli zwraca wartość zmiennej *suma*

Działanie funkcji *WartoscXk*:

1. Przechodzi po wartościach $i=1:n$
2. Oblicza wartość zmiennej suma korzystając z funkcji *policzSume*(i, X_k, A)
3. Wylicza i -ty element k -tego przybliżenia wektora x ze wzoru wyprowadzonego na Rysunku 2.
4. Zwraca nowy X_k .

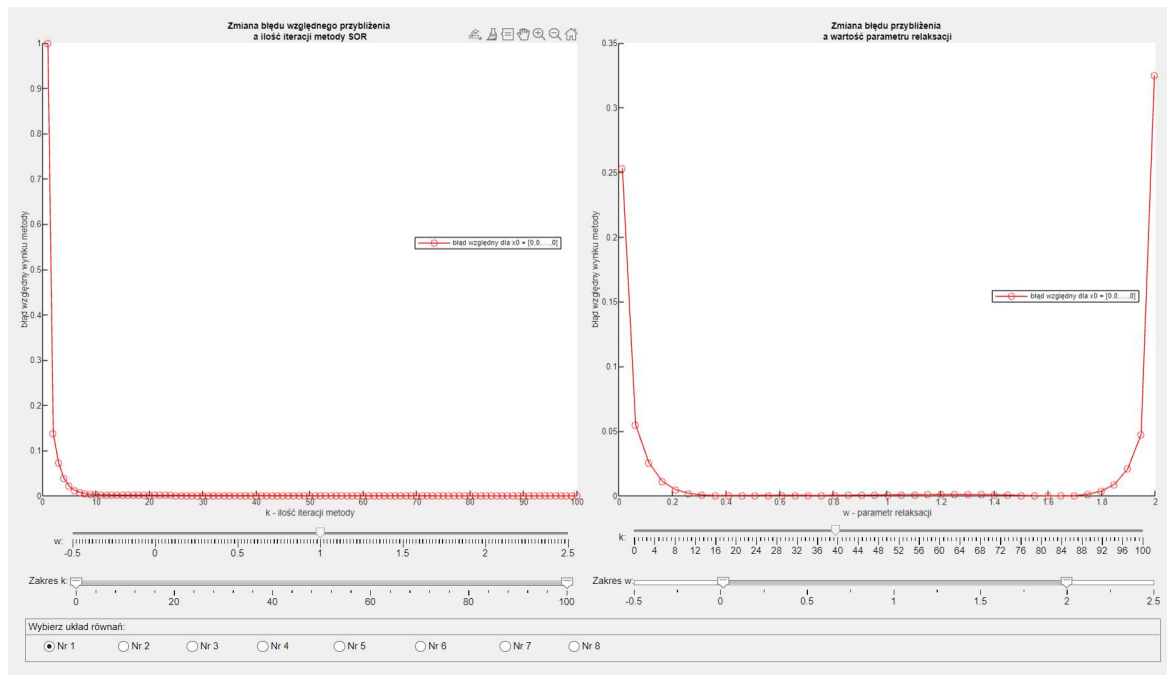
Działanie funkcji *iteracyjneSor*:

1. Ustala $k = 0$
2. Ustala X_k i X_{k-1} jako wektory zerowe o długości n
3. Ustala *wartoscWarunku* = *Inf*
4. W zależności od wybranego warunku, przechodzi po pętli *while* dopóki nie spełni tego warunku:
 - a. Gdy *warunek* ma wartość 1,2 lub 3 to dopóki spełniona jest nierówność *wartoscWarunku* > d :
 - i. Do zmiennej X_{k-1} przypisujemy X_k
 - ii. Obliczamy wartość X_k w k -tej iteracji korzystając z funkcji *WartoscXk*(X_k, A, b, w)
 - iii. Zwiększamy wartość k o 1
 - iv. Wyliczamy nową *wartoscWarunku* korzystając z funkcji *czySpeLniaWarunekStopu*($X_k, X_{k-1}, A, b, warunek$)
 - b. Gdy *warunek* ma wartość 4 to dopóki spełniona jest nierówność: *wartoscWarunku* > $d\|x_{k-1}\| + d_2$
 - i. Do zmiennej X_{k-1} przypisujemy X_k
 - ii. Obliczmy wartość X_k w k -tej iteracji korzystając z funkcji *WartoscXk*(X_k, A, b, w)
 - iii. Zwiększamy wartość k o 1
 - iv. Wyliczamy nową *wartoscWarunku* korzystając z funkcji *czySpeLniaWarunekStopu*($X_k, X_{k-1}, A, b, warunek$)
2. Przypisujemy zmiennej x wartość wektora X_k i zwracamy x .

Przykłady

1. $Ax=b$, gdzie $A = \begin{pmatrix} 1 & -3 & 2 \\ -3 & 10 & -5 \\ 2 & -5 & 6 \end{pmatrix}$, $b = \begin{pmatrix} 3 \\ -8 \\ 1 \end{pmatrix}$, $x = 0$

Macierz A jest symetryczna i dodatnio określona.



Rysunek 3: Widok GUI z wybranym układem równań nr 1

Na prawym wykresie (Rysunek 3) widać, że metoda daje niewielki błąd dla parametru $w \in (0,2)$.

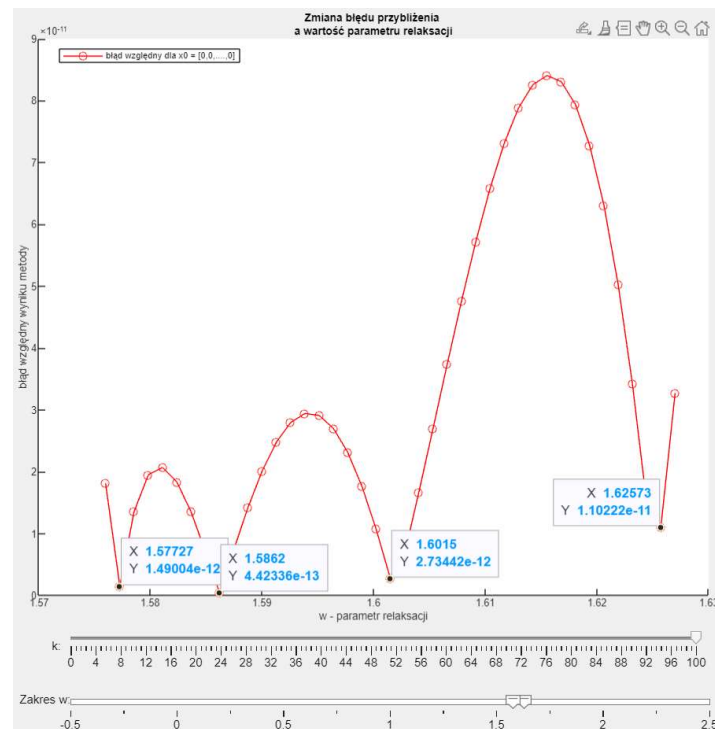
Z lewego wykresu (Rysunek 3) widać, że dla parametru $w = 1$ metoda jest zbieżna.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	5.2546e-05	488
0.2	5.2566e-05	151
0.3	5.2361e-05	293
0.4	5.2439e-05	266
0.5	5.211e-05	227
0.6	5.2026e-05	190
0.7	5.1976e-05	157
0.8	5.2383e-05	136
0.9	5.1384e-05	128
1	5.2043e-05	117
1.1	5.1138e-05	108
1.2	5.0706e-05	97
1.3	5e-05	85
1.4	4.9257e-05	72
1.5	5.0588e-05	57
1.6	4.9188e-05	34
1.7	4.0088e-05	50
1.8	2.2845e-05	71
1.9	3.4047e-05	129
2	NaN	Inf

Tabela 1: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

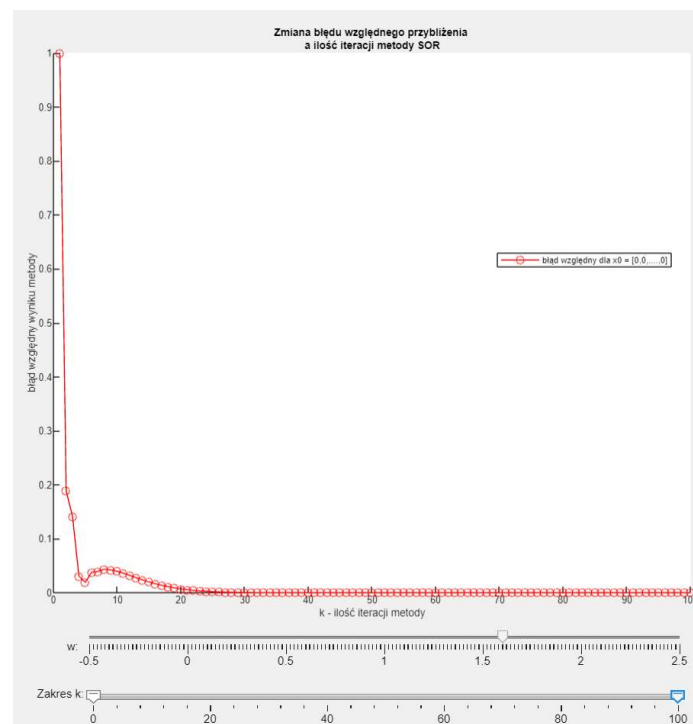
Tabela 1 potwierdza, że dla $w \in (0,2)$ metoda SOR działa i dobrze przybliża wartości wektora x .

Z tabeli wynika też, że najlepszym parametrem relaksacji z wybranych jest $w = 1.6$.



Rysunek 4: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

Na wykresie (Rysunek 4) zakres parametru w został zawężony do zakresu, gdzie przybliżenie przyjmuje najmniejsze błędy. Wykres potwierdza obserwację z analizy tabeli, że metoda działa najszybciej dla parametru relaksacji $w \approx 1.6$.



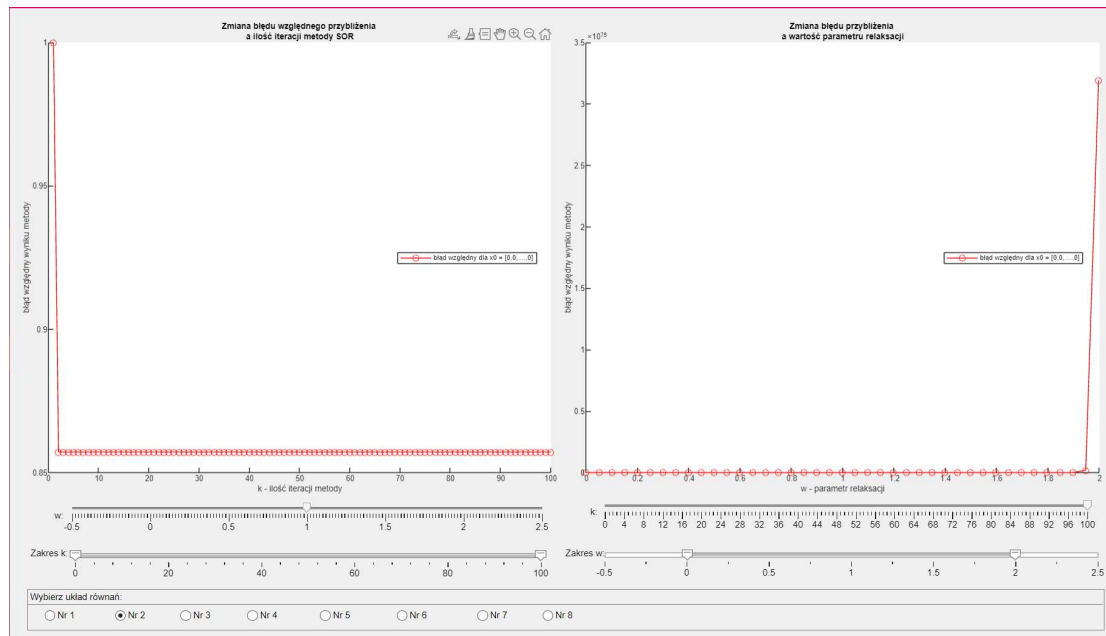
Rysunek 5: Wykres błędów przybliżenia dla kolejnych iteracji metody SOR, dla $w \approx 1.6$

2. $Ax=b$, gdzie $A|b =$

$$\left(\begin{array}{cccc|c} 1 & 0 & 0 & -3 & 3 \\ 1 & -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 1 & -2 \\ 0 & 0 & -2 & 2 & 1 \end{array} \right)$$

$x =$

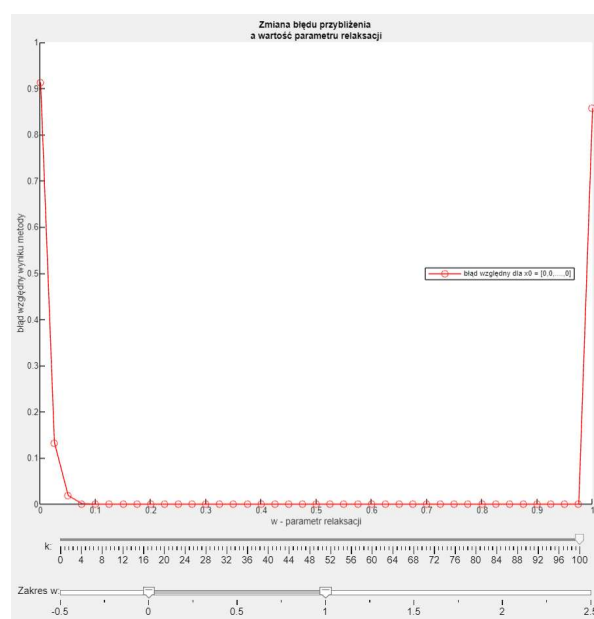
$$\begin{pmatrix} 3 \\ 4 \\ 7 \\ 4 \\ -5 \\ 4 \\ -3 \\ 4 \end{pmatrix}$$



Rysunek 6: Widok GUI z wybranym układem równań nr 2

Na prawym wykresie (Rysunek 6) widać, że metoda SOR zwraca przybliżenie z bardzo dużym błędem dla wartości parametru w bliskich 2.

Z lewego wykresu (Rysunek 6) widać, że metoda nie jest zbieżna dla parametru $w = 1$.



Rysunek 7: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

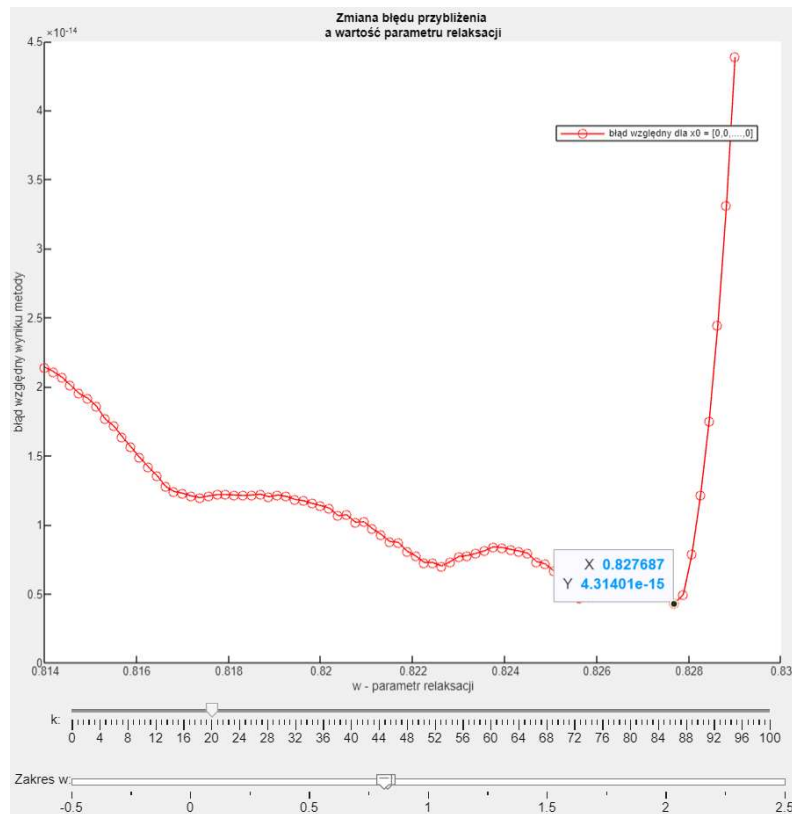
Na powyższym wykresie (Rysunek 7), gdzie zmniejszony został zakres parametru w do $[0, 1]$, widać, że dla $w \in (0,1)$ metoda daje niewielkie błędy przybliżeń.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	0.00013261	102
0.2	0.00012867	48
0.3	0.0001203	30
0.4	0.0001057	21
0.5	0.00013246	15
0.6	6.7935e-05	12
0.7	6.8815e-05	9
0.8	0.00012421	6
0.9	8.9103e-05	18
1	NaN	Inf
1.1	NaN	Inf
1.2	NaN	Inf
1.3	NaN	Inf
1.4	NaN	Inf
1.5	NaN	Inf
1.6	NaN	Inf
1.7	NaN	Inf
1.8	NaN	Inf
1.9	NaN	Inf
2	NaN	Inf

Tabela 2: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

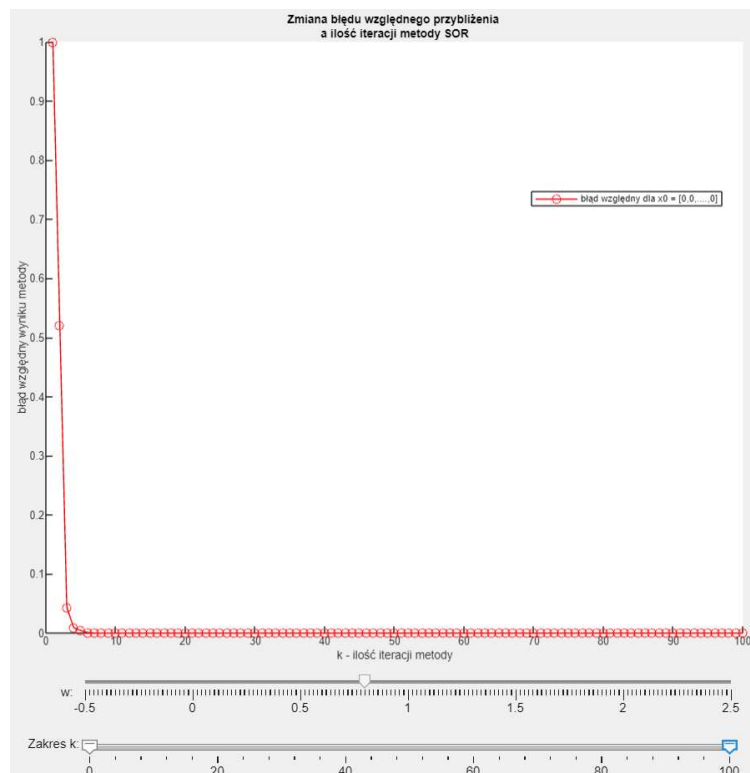
Tabela 2 potwierdza wnioski z wykresu (Rysunek 7), czyli że metoda SOR daje dobre wyniki tylko dla $w \in (0,1)$. Dla takich parametrów metoda działa szybko i dokładnie.

Z tabeli wynika też, że najlepszym parametrem relaksacji z wybranych jest $w = 0.8$.



Rysunek 8: Wykres błędu przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

Na wykresie (Rysunek 8) zakres parametru w został zawężony do zakresu, gdzie przybliżenie przyjmuje najmniejsze błędy. Wykres potwierdza obserwację z tabeli (Tabela 2), że metoda działa najszybciej dla parametru relaksacji $w \approx 0.8$.



Rysunek 9: Wykres błędu przybliżenia dla kolejnych iteracji metody SOR, dla $w \approx 0.82$.

3. $Ax = b$, gdzie $A =$

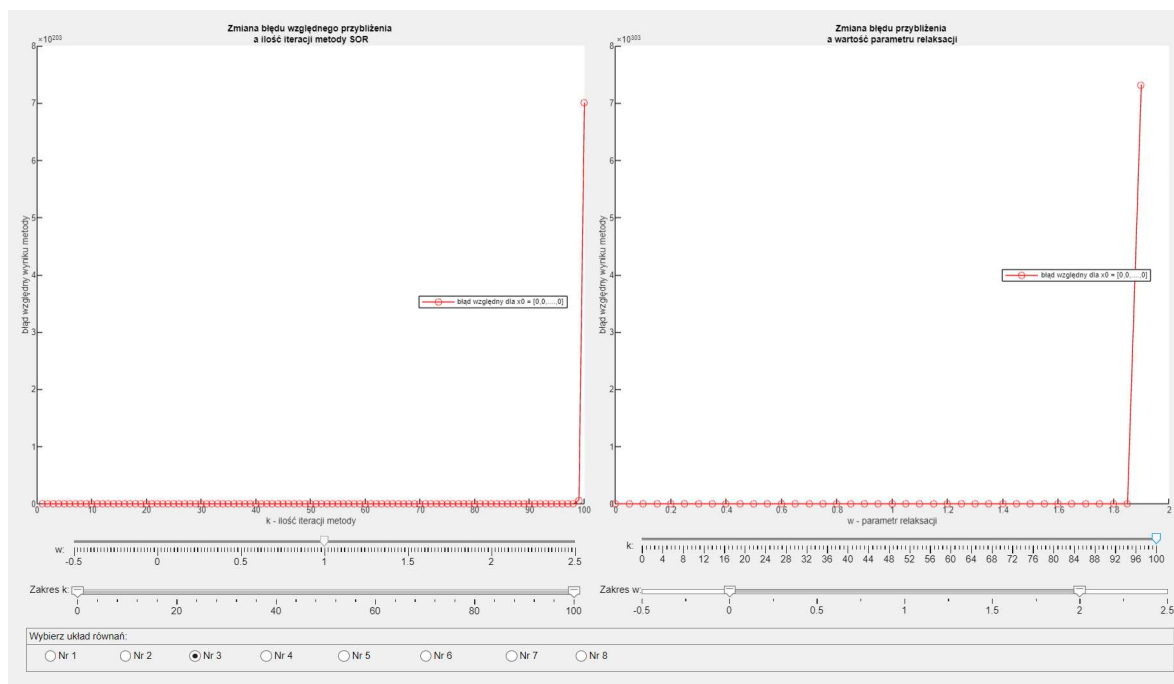
$$\begin{pmatrix} 1 & 4 & 7 & -8 & -9 \\ 0 & 8 & 5 & 3 & 8 \\ 9 & 0 & 6 & 5 & -2 \\ 7 & -8 & 0 & -2 & 5 \\ -9 & 0 & 0 & 6 & 1 \end{pmatrix}$$

$b =$

$$\begin{pmatrix} -35 \\ 5 \\ 13 \\ -22 \\ 14 \end{pmatrix}$$

$x =$

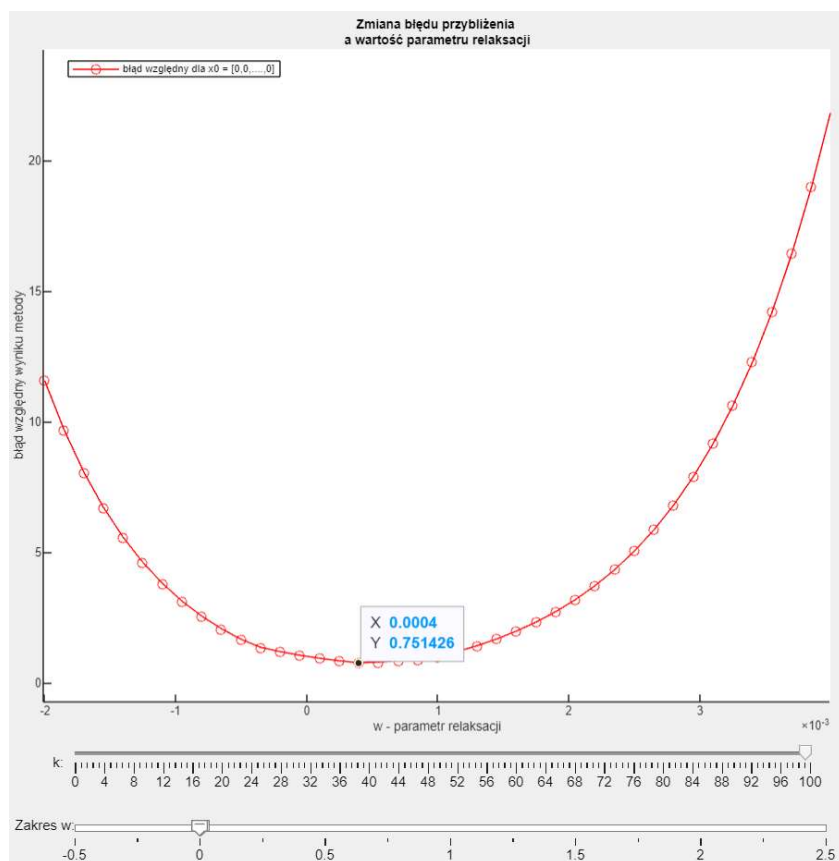
$$\begin{pmatrix} 1 \\ 2 \\ -3 \\ 4 \\ -1 \end{pmatrix}$$



Rysunek 10: Widok GUI z wybranym układem równań nr 3

Na prawym wykresie (Rysunek 10) widać, że metoda SOR zwraca przybliżenie z bardzo dużym błędem dla wartości parametru w bliskich 2.

Z lewego wykresu (Rysunek 10) widać, że metoda nie jest zbieżna dla parametru $w = 1$.



Rysunek 11: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

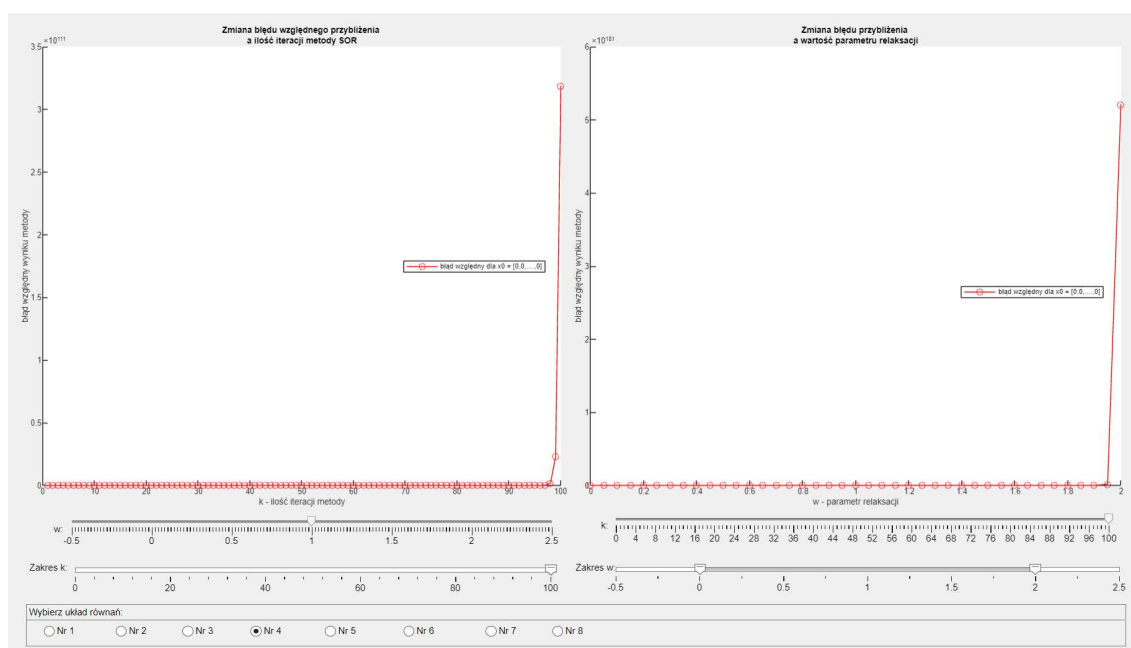
Na wykresie (Rysunek 11) zmniejszając zakres parametru w do zakresu $[-0.002, 0.004]$, gdzie parametr przyjmuje najmniejsze wartości błędów, nadal błąd przybliżenia dla 100 iteracji metody jest dość duży.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	NaN	Inf
0.2	NaN	Inf
0.3	NaN	Inf
0.4	NaN	Inf
0.5	NaN	Inf
0.6	NaN	Inf
0.7	NaN	Inf
0.8	NaN	Inf
0.9	NaN	Inf
1	NaN	Inf
1.1	NaN	Inf
1.2	NaN	Inf
1.3	NaN	Inf
1.4	NaN	Inf
1.5	NaN	Inf
1.6	NaN	Inf
1.7	NaN	Inf
1.8	NaN	Inf
1.9	NaN	Inf
2	NaN	Inf

Tabela 3: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

Tabela 3 pokazuje, że metoda nie jest zbieżna dla żadnego z wybranych parametrów $w \in (0,2)$.

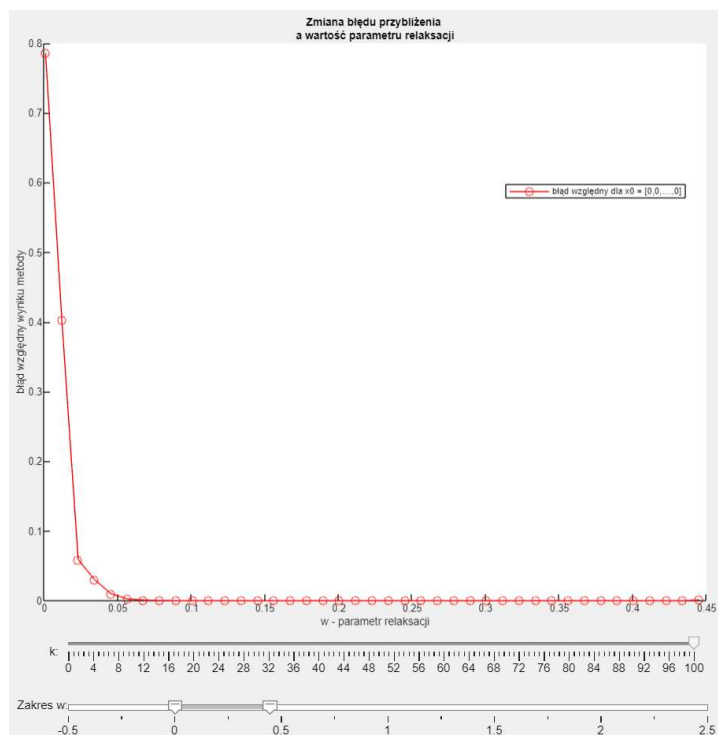
4. $Ax=b$, gdzie $A = \begin{pmatrix} 5 & 5 & 2 \\ 2 & 3 & -4 \\ 0 & 9 & 1 \end{pmatrix}$, $b = \begin{pmatrix} 60 \\ 33 \\ 8 \end{pmatrix}$, $x = \begin{pmatrix} 11.8823 \\ 1.0724 \\ -1.6516 \end{pmatrix}$



Rysunek 12: Widok GUI z wybranym układem równań nr 4

Na prawym wykresie (Rysunek 12) widać, że metoda SOR zwraca przybliżenie z bardzo dużym błędem dla wartości parametru w bliskich 2.

Z lewego wykresu (Rysunek 12) wynika, że metoda nie jest zbieżna dla parametru $w = 1$.



Rysunek 13: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

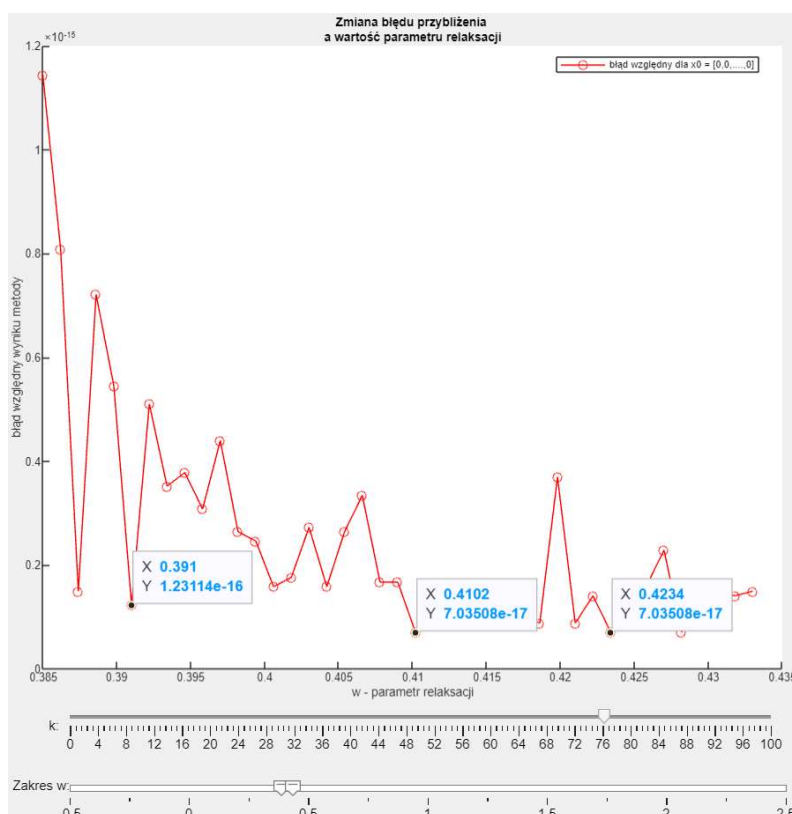
Na wykresie (Rysunek 13) po zmniejszeniu zakresu parametru w do $[0, 0.45]$ widać, że dla większości wartości w z tego zakresu przybliżenie rozwiązania przyjmuje stosunkowo małe błędy.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	9.5889e-06	114
0.2	7.0576e-06	54
0.3	8.8959e-06	33
0.4	6.1518e-06	23
0.5	NaN	Inf
0.6	NaN	Inf
0.7	NaN	Inf
0.8	NaN	Inf
0.9	NaN	Inf
1	NaN	Inf
1.1	NaN	Inf
1.2	NaN	Inf
1.3	NaN	Inf
1.4	NaN	Inf
1.5	NaN	Inf
1.6	NaN	Inf
1.7	NaN	Inf
1.8	NaN	Inf
1.9	NaN	Inf
2	NaN	Inf

Tabela 4: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

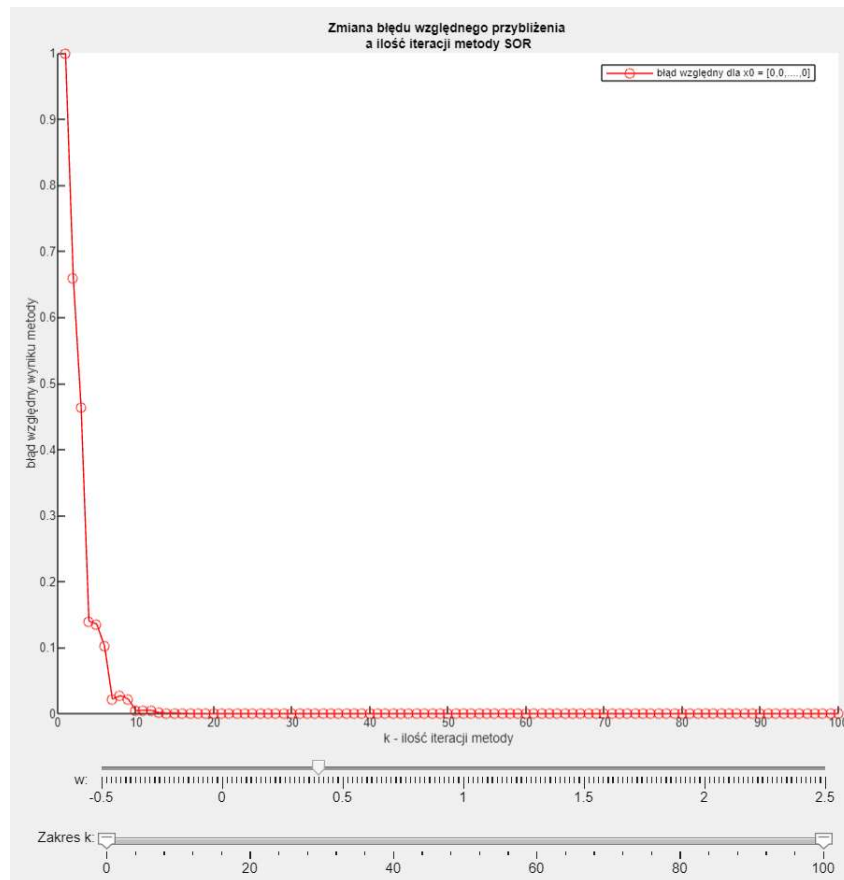
Tabela 4 potwierdza wnioski z wykresu (Rysunek 13), czyli że metoda SOR może być zbieżna tylko dla $w \in (0, 0.5)$. Dla takich parametrów działa ona szybko i dokładnie.

Z tabeli wynika też, że najlepszym parametrem relaksacji z wybranych jest $w = 0.4$.



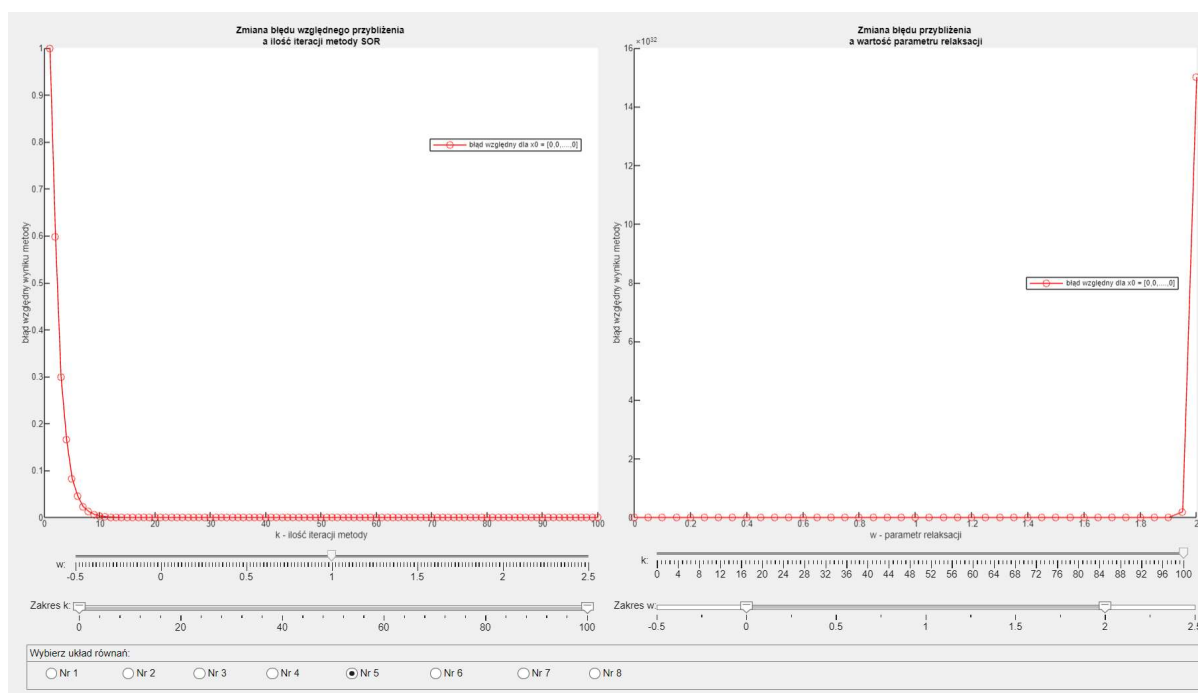
Rysunek 14: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

Na wykresie (Rysunek 14) zakres parametru w został zawężony do zakresu, gdzie przybliżenie przyjmuje najmniejsze błędy. Wykres potwierdza obserwację z tabeli, że metoda działa najszybciej dla parametru relaksacji $w \approx 0.4$.



Rysunek 15: Wykres błędów przybliżenia dla kolejnych iteracji metody SOR, dla $w \approx 0.4$

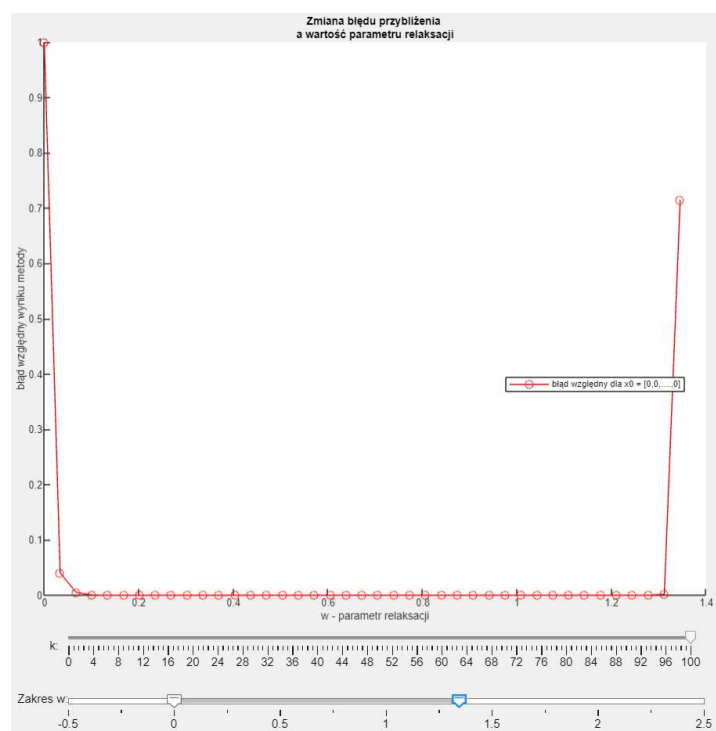
5. $Ax = b$, gdzie $A = \begin{pmatrix} 2 & 5 & 0 \\ 0 & 2 & 2 \\ 1 & 0 & 9 \end{pmatrix}$, $b = \begin{pmatrix} 36 \\ 30 \\ 84 \end{pmatrix}$, $x = \begin{pmatrix} 3 \\ 6 \\ 9 \end{pmatrix}$



Rysunek 16: Widok GUI z wybranym układem równań nr 5

Na prawym wykresie (Rysunek 16) widać, że metoda SOR zwraca przybliżenie z bardzo dużym błędem dla wartości parametru w bliskich 2.

Z lewego wykresu (Rysunek 16) wynika, że metoda jest zbieżna dla parametru $w = 1$.



Rysunek 17: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu.

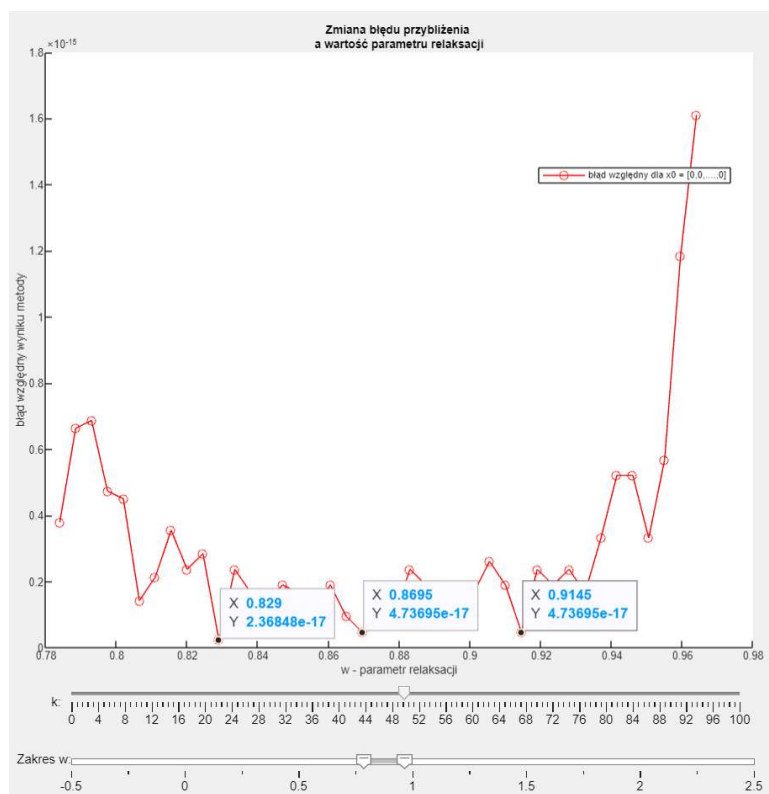
Na wykresie (Rysunek 17) po zmniejszeniu zakresu parametru w do $[0, 1.4)$ widać, że dla większości wartości w z tego zakresu przybliżenie rozwiązania przyjmuje stosunkowo małe błędy.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	6.2694e-06	163
0.2	5.891e-06	79
0.3	5.0649e-06	51
0.4	5.7439e-06	36
0.5	5.443e-06	27
0.6	5.791e-06	23
0.7	6.0352e-06	19
0.8	6.599e-06	17
0.9	3.387e-06	17
1	5.886e-06	19
1.1	5.2584e-06	26
1.2	5.5397e-06	46
1.3	6.3091e-06	148
1.4	NaN	Inf
1.5	NaN	Inf
1.6	NaN	Inf
1.7	NaN	Inf
1.8	NaN	Inf
1.9	NaN	Inf
2	NaN	Inf

Tabela 5: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

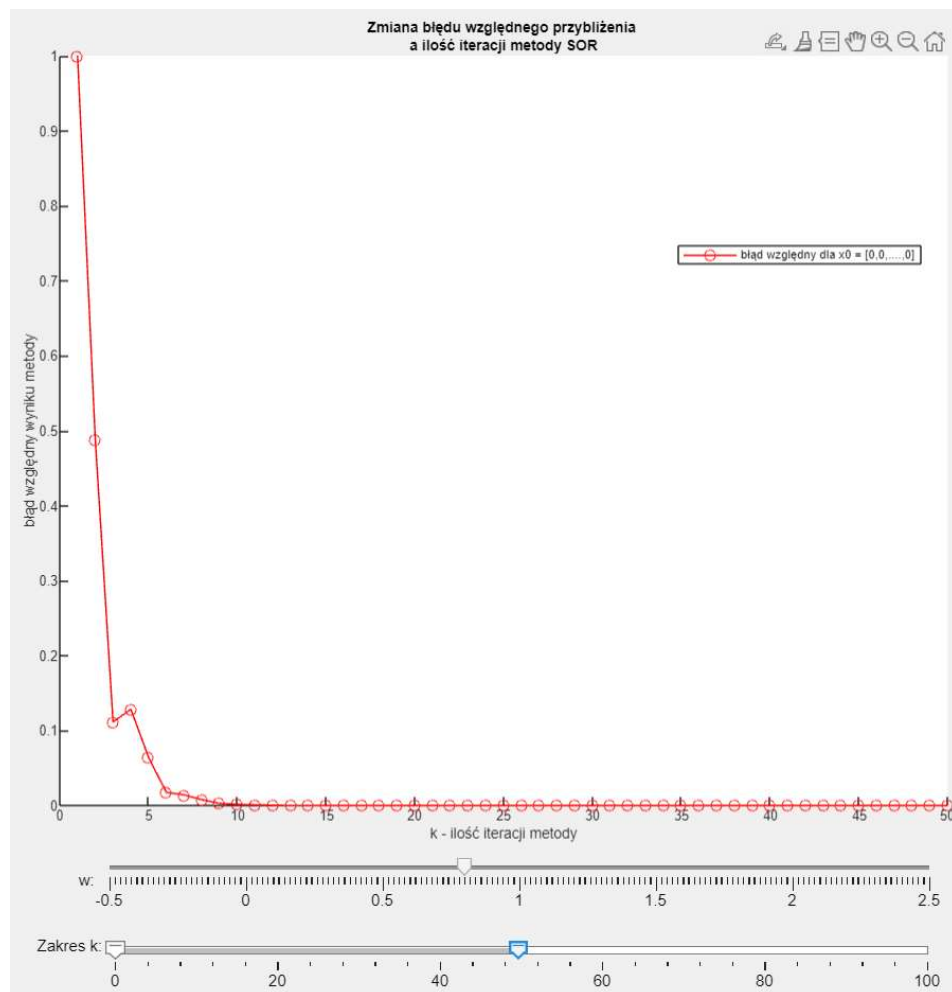
Tabela 5 potwierdza wniosek z wykresu (Rysunek 17), czyli że metoda może być zbieżna tylko dla $w \in (0, 1.4)$. Dla takich parametrów działa ona szybko i dokładnie.

Z tabeli wynika też, że najlepszym parametrem relaksacji z wybranych jest $w = 0.8$ i $w = 0.9$.



Rysunek 18: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

Na wykresie (Rysunek 18) zakres parametru w został zawężony do zakresu, gdzie przybliżenie przyjmuje najmniejsze błędy. Wykres potwierdza obserwację z tabeli, że metoda działa najszybciej dla parametru relaksacji $w \in (0.8, 0.9)$.



Rysunek 19: Wykres błędów przybliżenia dla kolejnych iteracji metody SOR, dla $w \approx 0.8$

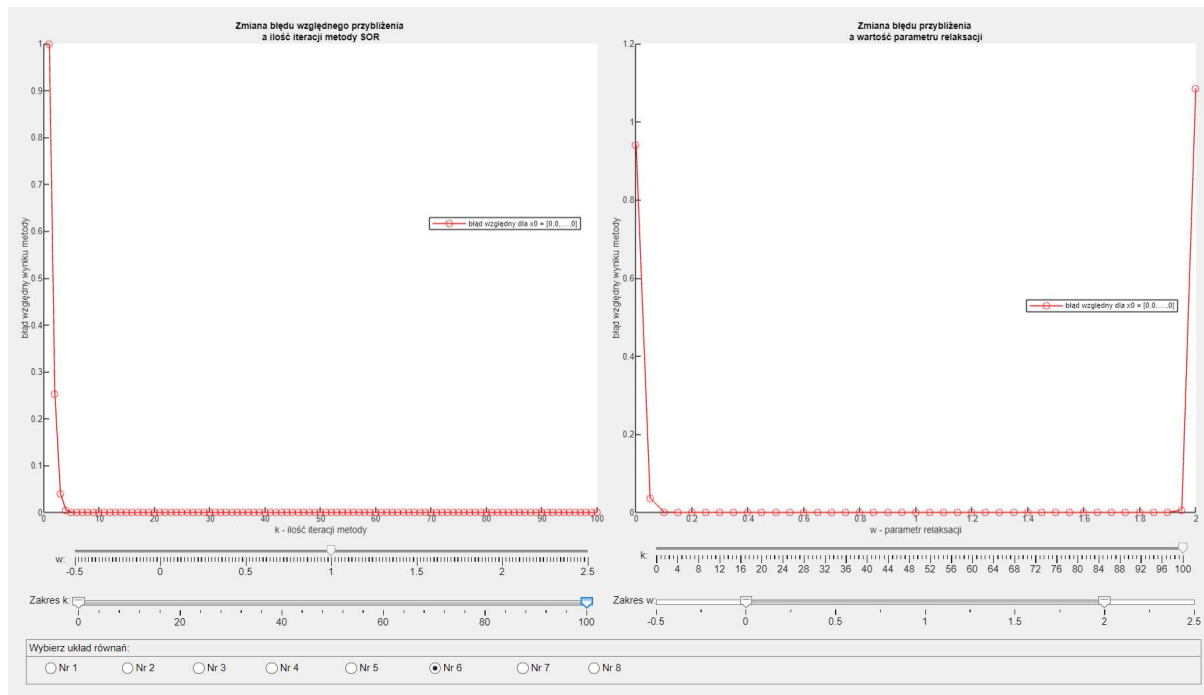
6. $Ax = b$, gdzie $A|b =$

$$\left(\begin{array}{ccc|c} 4 & -1 & 0 & 2 \\ -1 & 4 & -1 & 6 \\ 0 & -1 & 4 & 2 \end{array} \right)$$

$x =$

$$\begin{pmatrix} 1 \\ 2 \\ 1 \end{pmatrix}$$

Macierz A jest symetryczna i dodatnio określona.



Rysunek 20: Widok GUI z wybranym układem równań nr 6

Na prawym wykresie (Rysunek 20) widać, że metoda daje niewielki błąd dla parametru $w \in (0, 2)$.

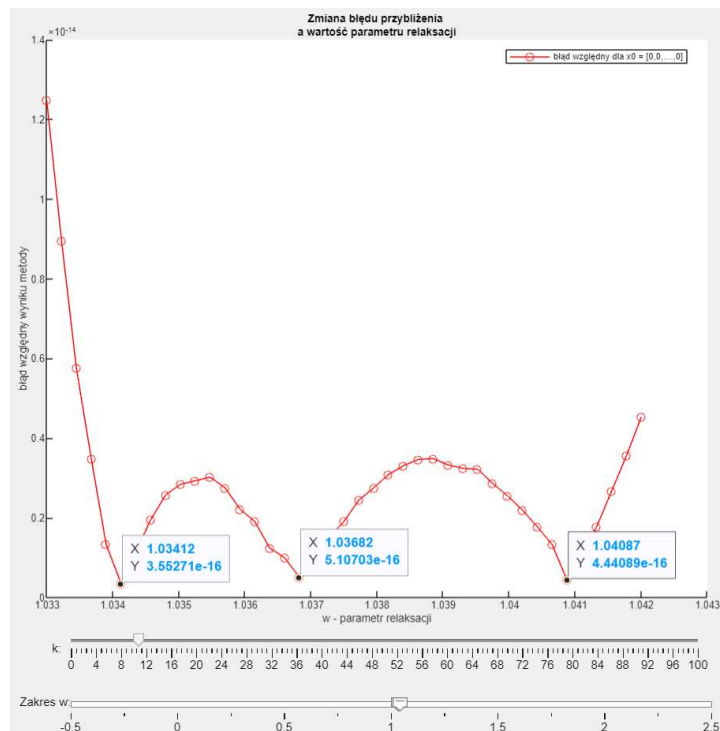
Z lewego wykresu (Rysunek 20) widać, że dla $w = 1$ metoda jest zbieżna.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	9.4729e-05	137
0.2	9.1483e-05	65
0.3	8.6476e-05	41
0.4	7.9305e-05	29
0.5	9.923e-05	21
0.6	5.7019e-05	17
0.7	6.4297e-05	13
0.8	6.6588e-05	10
0.9	3.4972e-05	8
1	8.0109e-05	5
1.1	1.324e-05	5
1.2	9.1881e-05	6
1.3	7.3147e-05	8
1.4	4.5378e-05	11
1.5	9.1117e-05	14
1.6	7.3988e-05	19
1.7	8.6125e-05	26
1.8	6.8985e-05	43
1.9	9.8435e-05	89
2	NaN	Inf

Tabela 6: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

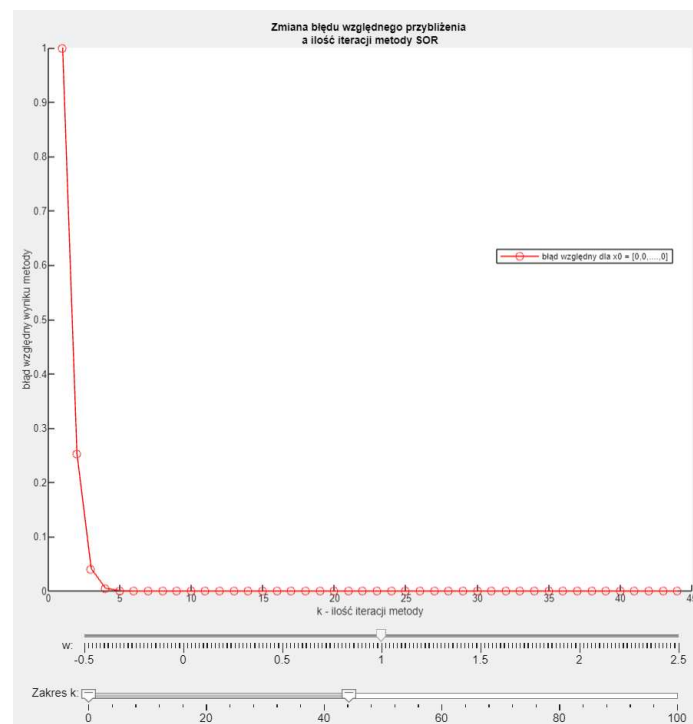
Tabela 6 potwierdza, że dla $w \in (0,2)$ metoda SOR działa i dobrze przybliża wartości wektora x .

Z tabeli wynika też, że najlepszym parametrem relaksacji z wybranych jest $w = 1$ i $w = 1.1$.



Rysunek 21: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

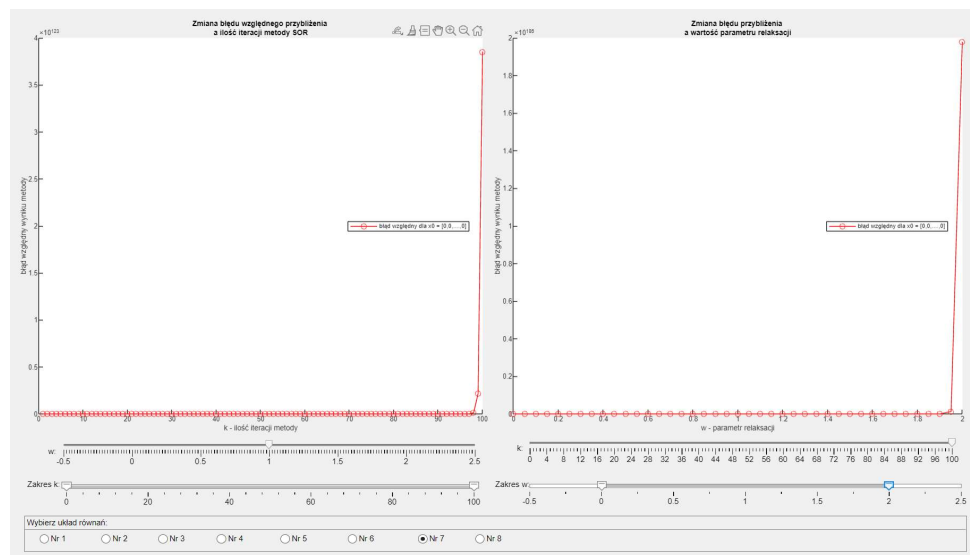
Na wykresie (Rysunek 21) zakres parametru w został zawężony do zakresu, gdzie przybliżenie przyjmuje najmniejsze błędy. Wykres potwierdza obserwację z tabeli 6, że metoda działa najszybciej dla parametru relaksacji $w \approx 1$.



Rysunek 22: Wykres błędów przybliżenia dla kolejnych iteracji metody SOR, dla $w \approx 1$.

7. $Ax = b$, gdzie:

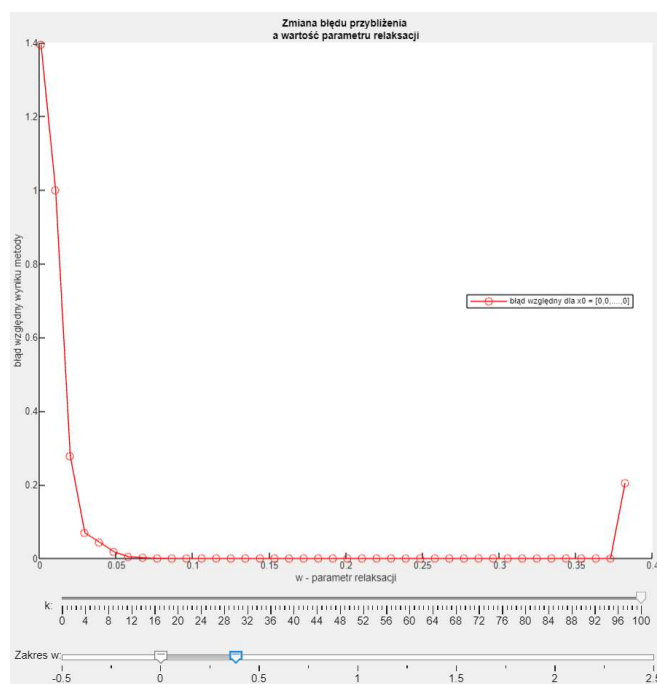
$$\begin{pmatrix} 2 & -9 & 0 & 0 & 0 \\ 2 & 1 & 8 & 6 & 0 \\ 0 & 11 & -10 & 0 & 0 \\ 0 & 0 & 0 & 6 & 2 \\ 0 & 0 & 0 & -3 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} -16 \\ 52 \\ -8 \\ 34 \\ 3 \end{pmatrix}$$



Rysunek 23: Widok GUI z wybranym układem równań nr 7

Na prawym wykresie (Rysunek 23) widać, że metoda SOR zwraca przybliżenie z bardzo dużym błędem dla wartości parametru w bliskich 2.

Z lewego wykresu (Rysunek 23) wynika, że metoda nie jest zbieżna dla parametru $w = 1$.



Rysunek 24: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu.

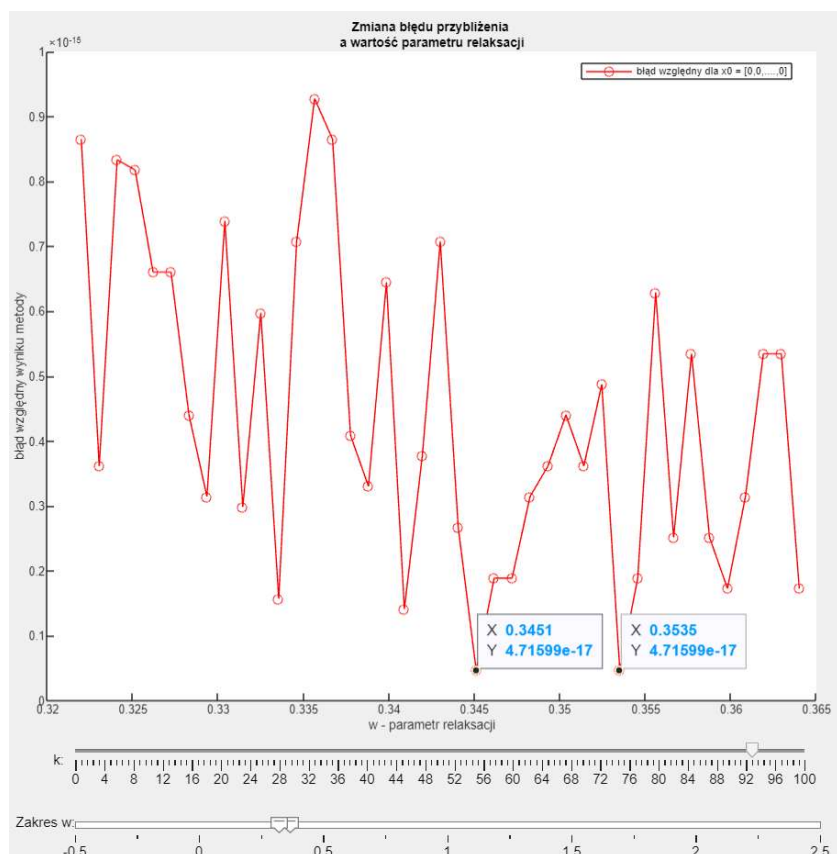
Na wykresie (Rysunek 24) po zmniejszeniu zakresu parametru w do $[0, 0.4]$ widać, że dla większości wartości w z tego zakresu przybliżenie rozwiązania przyjmuje stosunkowo małe błędy.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	8.0162e-06	112
0.2	7.5867e-06	54
0.3	8.4618e-06	33
0.4	NaN	Inf
0.5	NaN	Inf
0.6	NaN	Inf
0.7	NaN	Inf
0.8	NaN	Inf
0.9	NaN	Inf
1	NaN	Inf
1.1	NaN	Inf
1.2	NaN	Inf
1.3	NaN	Inf
1.4	NaN	Inf
1.5	NaN	Inf
1.6	NaN	Inf
1.7	NaN	Inf
1.8	NaN	Inf
1.9	NaN	Inf
2	NaN	Inf

Tabela 7: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

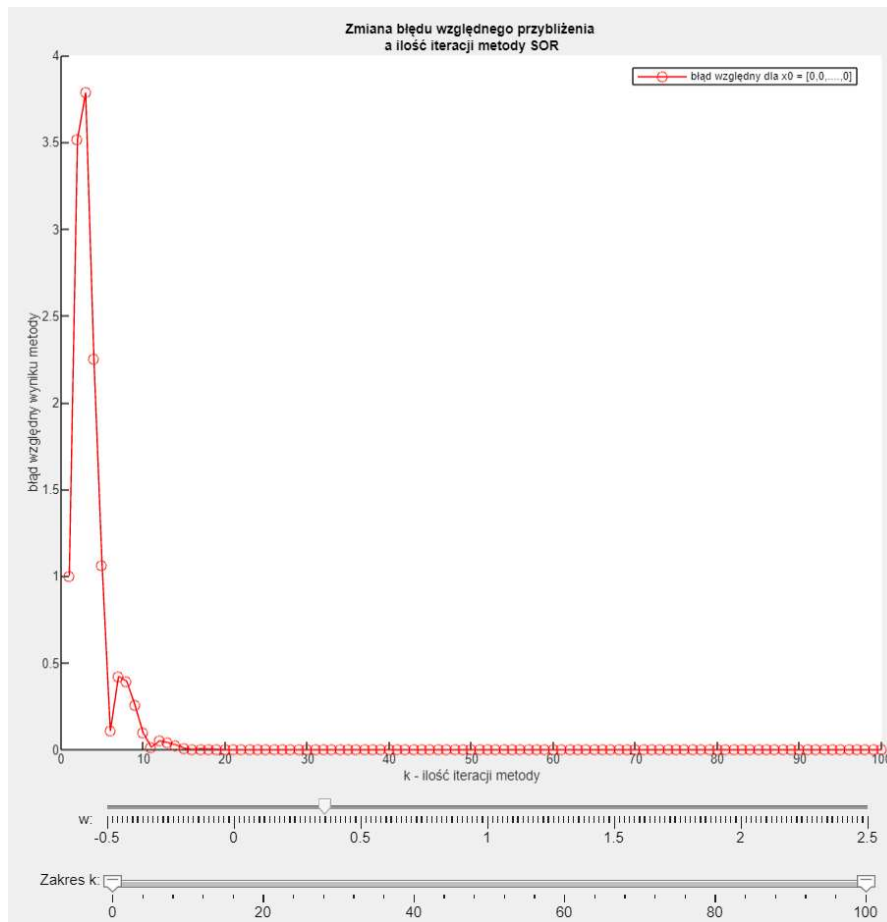
Tabela 7 potwierdza wniosek z wykresu (Rysunek 24), czyli że metoda może być zbieżna tylko dla $w \in (0, 0.4)$. Dla takich parametrów działa ona szybko i dokładnie.

Z tabeli wynika też, że najlepszym parametrem relaksacji z wybranych jest $w = 0.3$.



Rysunek 25: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

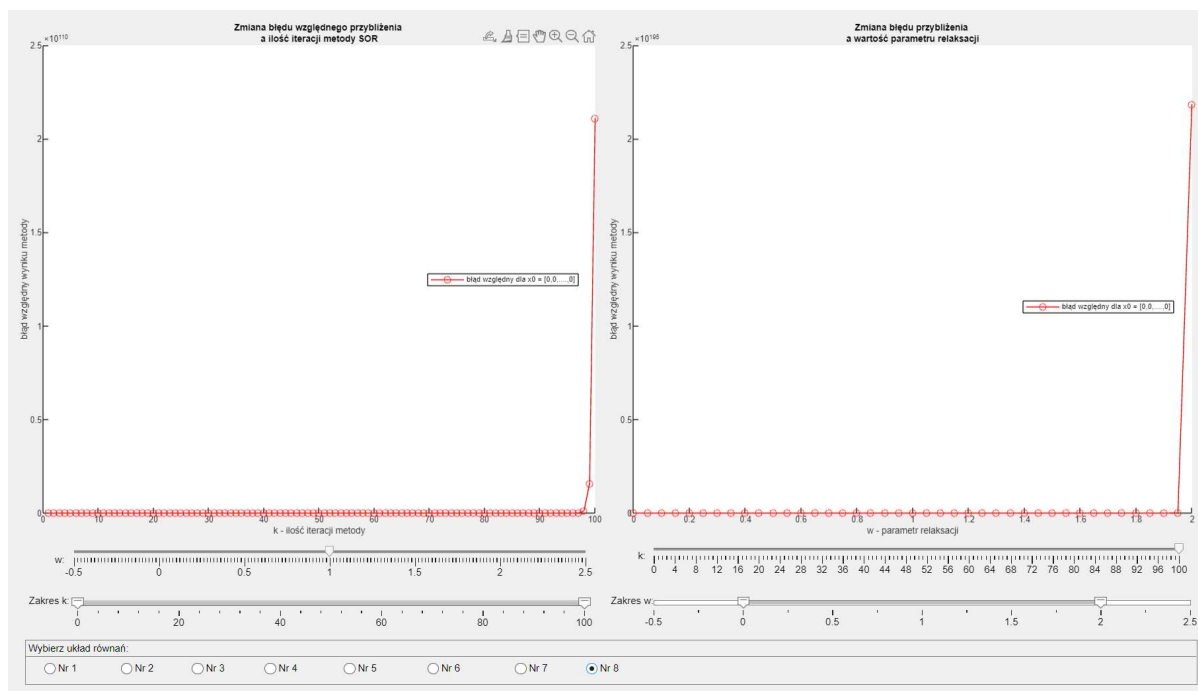
Na wykresie (Rysunek 25) zakres parametru w został zawężony do zakresu, gdzie przybliżenie przyjmuje najmniejsze błędy. Wykres pozwala na znalezienie jeszcze lepszego parametru relaksacji niż Tabela 7 - metoda działa najszybciej dla parametru relaksacji $w \approx 0.35$.



Rysunek 26: Wykres błędów przybliżenia dla kolejnych iteracji metody SOR, dla $w \approx 0.35$

8. $Ax = b$, gdzie

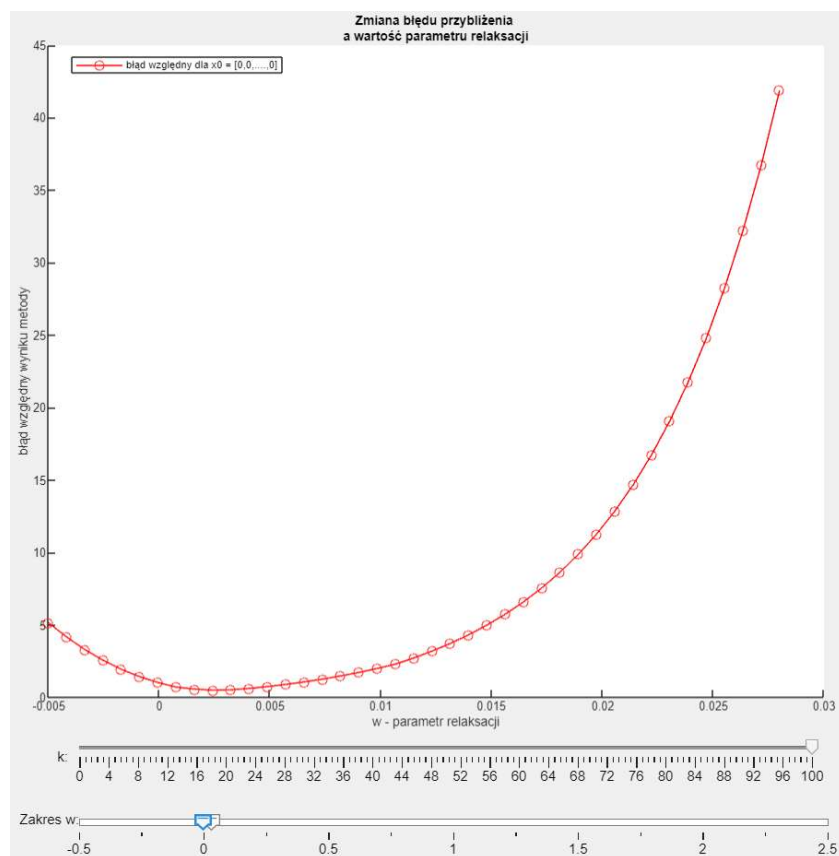
$$\begin{pmatrix} 1 & 0 & 8 & 0 & 0 \\ 0 & 3 & 0 & -5 & 0 \\ 0 & 0 & 2 & 0 & 8 \\ 0 & 0 & 3 & 1 & 0 \\ 0 & 7 & 0 & 11 & 4 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} = \begin{pmatrix} 25 \\ -14 \\ 46 \\ 13 \\ 78 \end{pmatrix}$$



Rysunek 27: Widok GUI z wybranym układem równań nr 8

Na prawym wykresie (Rysunek 27) widać, że metoda SOR zwraca przybliżenie z bardzo dużym błędem dla wartości parametru w bliskich 2.

Z lewego wykresu (Rysunek 27) widać, że metoda nie jest zbieżna dla parametru $w = 1$.



Rysunek 28: Wykres błędów przybliżenia dla różnych wartości parametru relaksacji dla zawężonego zakresu

Na wykresie (Rysunek 28) zmniejszając zakres parametru w do zakresu $[-0.005, 0.03]$, gdzie przybliżenie ma najmniejsze wartości błędów, nadal błąd przybliżenia dla 100 iteracji metody jest dość duży.

parametr relaksacji	Błąd względny przybliżenia	numer potrzebnych iteracji dla dokładności d
0	NaN	Inf
0.1	NaN	Inf
0.2	NaN	Inf
0.3	NaN	Inf
0.4	NaN	Inf
0.5	NaN	Inf
0.6	NaN	Inf
0.7	NaN	Inf
0.8	NaN	Inf
0.9	NaN	Inf
1	NaN	Inf
1.1	NaN	Inf
1.2	NaN	Inf
1.3	NaN	Inf
1.4	NaN	Inf
1.5	NaN	Inf
1.6	NaN	Inf
1.7	NaN	Inf
1.8	NaN	Inf
1.9	NaN	Inf
2	NaN	Inf

Tabela 8: Tabela pokazująca potrzebną ilość iteracji metody SOR dla dokładności $d=0.01$ dla różnych wartości parametru relaksacji

Tabela 8 pokazuje, że metoda nie jest zbieżna dla żadnego z wybranych parametrów $w \in (0,2)$.

Podsumowanie wyników:

Przykłady 1. i 6. potwierdzają wniosek z poniższego twierdzenia z wykładu.

Twierdzenie 6.7. (Kahan) Promień spektralny $\rho(B_{SOR})$ macierzy iteracji spełnia zależność: $\rho(B_{SOR}) \geq |\omega - 1|$.

Wniosek 6.1. Jeśli $\omega \notin (0, 2)$, to metoda SOR nie jest zbieżna.

Rysunek 29: Twierdzenie z notatek z wykładu

W tych przykładach były macierze symetryczne i dodatnio określone oraz metoda SOR dla nich była zbieżna dla $\omega \in (0, 2)$.

Z przykładów 3. i 8. widać, że niestety metoda SOR nie zawsze ma taki parametr relaksacji, aby była zbieżna. Przez to nie pozwala ona na rozwiązanie każdego układu równań.

Przykłady 2., 4. i 7. pokazują, że metoda SOR jest lepsza od metody Gaussa-Seidla. Metoda SOR z parametrem relaksacji $\omega=1$ jest metodą Gaussa-Seidla. Dla wymienionych przykładów metoda SOR dla $\omega = 1$ nie jest zbieżna, czyli nie działa dla nich metoda Gaussa-Seidla, ale istnieją takie parametry $\omega \neq 1$, takie, że dla nich metoda SOR jest zbieżna.

Przykład praktycznego zastosowania

Metoda SOR jest stosowana do rozwiązywania problemów odtwarzania i odszumiania obrazów. W tych problemach trzeba znaleźć rozkład intensywności pikseli, który minimalizuje pewną funkcję kosztu i spełnia warunek nieujemności. Metoda SOR jest wykorzystywana przy tych problemach, bo jest szybka i skuteczna oraz pozwala znaleźć ich przybliżone rozwiązanie.