

11 stycznia 2022 r.

Marta Szuwarska  
Grupa 1 (środa 14:15)

## Rozwiązywanie układu równań liniowych z macierzą trójdagonalną za pomocą metody BSOR

Projekt nr 2

### 1 Opis metody

Celem zadania jest rozwiązywanie układu równań liniowych  $Ax = b$ , gdzie  $A \in \mathbb{R}^{n \times n}$  jest macierzą trójdagonalną,  $b \in \mathbb{R}^n$ , metoda BSOR (Backward SOR).

Niech  $A \in \mathbb{R}^{n \times n}$  będzie macierzą trójdagonalną,  $n \in \mathbb{R}$ .

$$A = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 & \cdots & 0 & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & \ddots & 0 & 0 \\ 0 & a_{32} & a_{33} & \ddots & \ddots & a_{n-2,n-1} & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & a_{n-1,n-1} & a_{n-1,n} \\ 0 & 0 & \cdots & \cdots & \cdots & a_{n,n-1} & a_{n,n} \end{pmatrix} \quad (1)$$

Aby oszczędzić pamięć i przyspieszyć obliczenia, naszą macierz będziemy przechowywać w postaci macierzy o rozmiarze  $3 \times n$ :

$$A^* = \begin{pmatrix} 0 & a_{21} & a_{32} & \cdots & a_{n-1,n-2} & a_{n,n-1} \\ a_{11} & a_{22} & a_{33} & \cdots & a_{n-1,n-1} & a_{n,n} \\ a_{12} & a_{23} & a_{34} & \cdots & a_{n-1,n} & 0 \end{pmatrix} \quad (2)$$

Niech  $b \in \mathbb{R}^n$  będzie wektorem.  $b = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$

Metoda BSOR jest analogiczna do metody SOR. Jedyna różnica jest taka, że zajmujemy się równaniami w odwrotnej kolejności.

Algorytm:

$\omega \in \mathbb{R}$  - parametr relaksacji.

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

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

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

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

end

end

Przeróbmy ten algorytm tak, aby skorzystać z macierzy  $A^*$ .

Wartości z  $i$ -tego wiersza macierzy  $A$  znajdują się w  $i$ -tej kolumnie  $A^*$ .

Dla  $i = 1$  w  $i$ -tym wierszu niezerowe wartości znajdują się w kolumnach 1, 2.  
Dla  $i = n$  w  $i$ -tym wierszu niezerowe wartości znajdują się w kolumnach  $n, n-1$ .  
W pozostałych przypadkach w  $i$ -tym wierszu niezerowe wartości znajdują się w kolumnach  $i, i+1, i+2$ .

Zmodyfikujmy zatem:

$x^{(0)} = (0, x_1^{(0)}, \dots, x_n^{(0)}, 0)^T$ .

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

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

$x_i^{k+1} = (1 - \omega)x_i^k + \omega(b_i - \sum_{j=1, j>i}^3 a_{ji}^*x_{i+j-1}^{k+1} - \sum_{j=1, j<i}^3 a_{ji}^*x_{i+j-1}^k)/a_{2j}^*$

end

Ostatecznym rozwiązaniem jest  $x$  z ostatniej iteracji bez zer na początku i końcu.

Dodatkowo będziemy chcieli obliczyć promień spektralny macierzy iteracji  $B$ .

$$L = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ a_{12}^* & 0 & \cdots & 0 & 0 \\ 0 & a_{13}^* & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & a_{1n}^* & 0 \end{pmatrix} \quad (3)$$

$$D = \begin{pmatrix} a_{21}^* & 0 & 0 & \cdots & 0 \\ 0 & a_{22}^* & 0 & \cdots & 0 \\ 0 & 0 & a_{23}^* & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{2n}^* \end{pmatrix} \quad (4)$$

$$U = \begin{pmatrix} 0 & a_{31}^* & 0 & \cdots & 0 \\ 0 & 0 & a_{32}^* & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & a_{3,n-1}^* \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \quad (5)$$

$$B = (D - \omega U)^{-1}((1 - \omega)D + \omega L) \quad (6)$$

$$\rho(B) = \max_{\lambda \in \sigma(B)} |\lambda| \quad (7)$$

Ponadto będziemy też obliczać wskaźnik uwarunkowania macierzy:

$$\text{cond}(A) = \|A^{-1}\| \cdot \|A\| \quad (8)$$

Będziemy również liczyć błąd względny, korzystając ze wzoru:

$$\delta = \frac{x - x_0}{x_0}, \quad (9)$$

gdzie  $x$  - rozwiązanie dokładne,  $x_0$  - rozwiązanie przybliżone, wyliczone za pomocą metody BSOR.

Z racji, że mamy do czynienia z macierzą trójdziagonalną, do policzenia rozwiązania dokładnego będziemy używać algorytmu Thomasa.

Algorytm:

$$e = (a_{11}^*, a_{12}^*, \dots, a_{1n}^*)$$

$$f = (a_{21}^*, a_{22}^*, \dots, a_{2n}^*)$$

$$g = (a_{31}^*, a_{32}^*, \dots, a_{3n}^*)$$

for  $i = 2, 3, \dots, n$

$$w = \frac{e_i}{f_{i-1}}$$

$$f_i = f_i - w g_{i-1}$$

$$b_i = b_i - w b_{i-1}$$

end

$$x_n = \frac{b_n}{f_n}$$

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

$$x_i = \frac{b_i - g_i x_{i+1}}{f_i}$$

end

## 2 Opis programu obliczeniowego

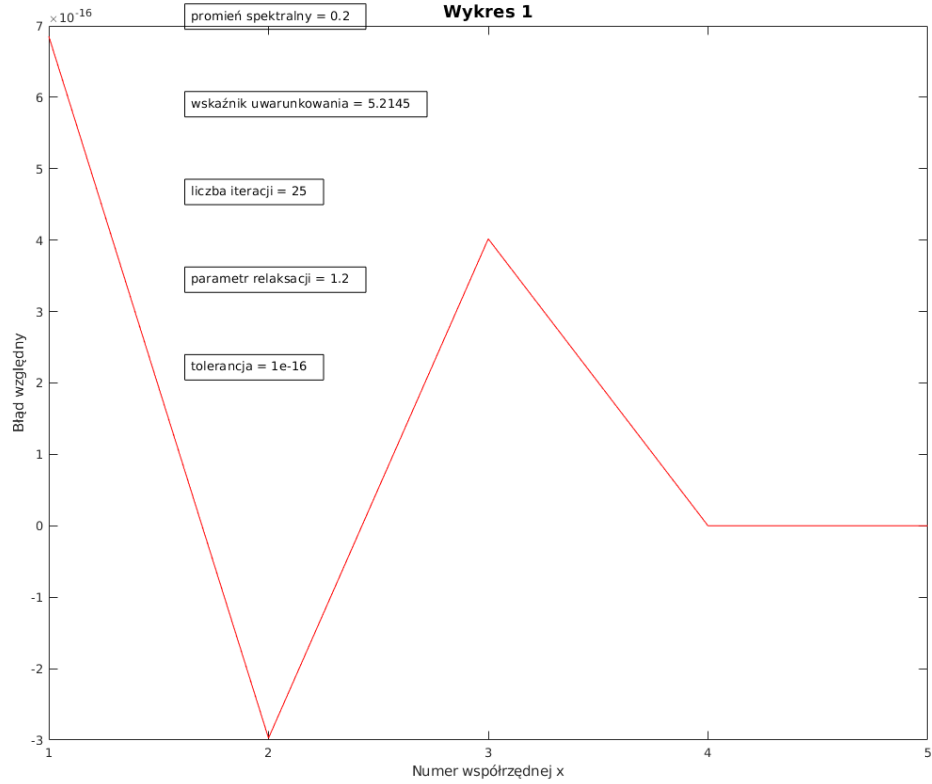
Program składa się z kilku plików:

1. **Main.m** - główny skrypt, w którym inicjalizujemy zmienne, rozwiązujemy układ  $Ax = b$  za pomocą metody dokładnej i przybliżonej BSOR, liczymy błąd względny i rysujemy jego wykres.
2. **BSOR.m** - funkcja rozwiązująca układ  $Ax = b$  metodą BSOR z warunkiem stopu  $\|Ax^{(k)} - b\| < d$ . Przyjmuje macierz trójdziagonalną  $A$ , wektor  $b$ , parametr relaksacji  $\omega$ , początkowy wektor rozwiązania  $x_0$ , tolerancję dla warunku stopu  $d$  i maksymalną liczbę iteracji  $M$ . Zwraca rozwiązanie  $x$ , promień spektralny macierzy iteracji  $\rho_B$ , wskaźnik uwarunkowania macierzy  $A$   $\text{cond}_A$  i liczbę iteracji  $k$ .
3. **Solve.m** - funkcja rozwiązuje układ równań  $Ax = b$  dla macierzy trójdziagonalnej za pomocą algorytmu Thomasa. Przyjmuje macierz trójdziagonalną  $A$  i wektor  $b$ . Zwraca rozwiązanie  $x$ .

## 3 Przykłady obliczeniowe

### 3.1 Przykład 1

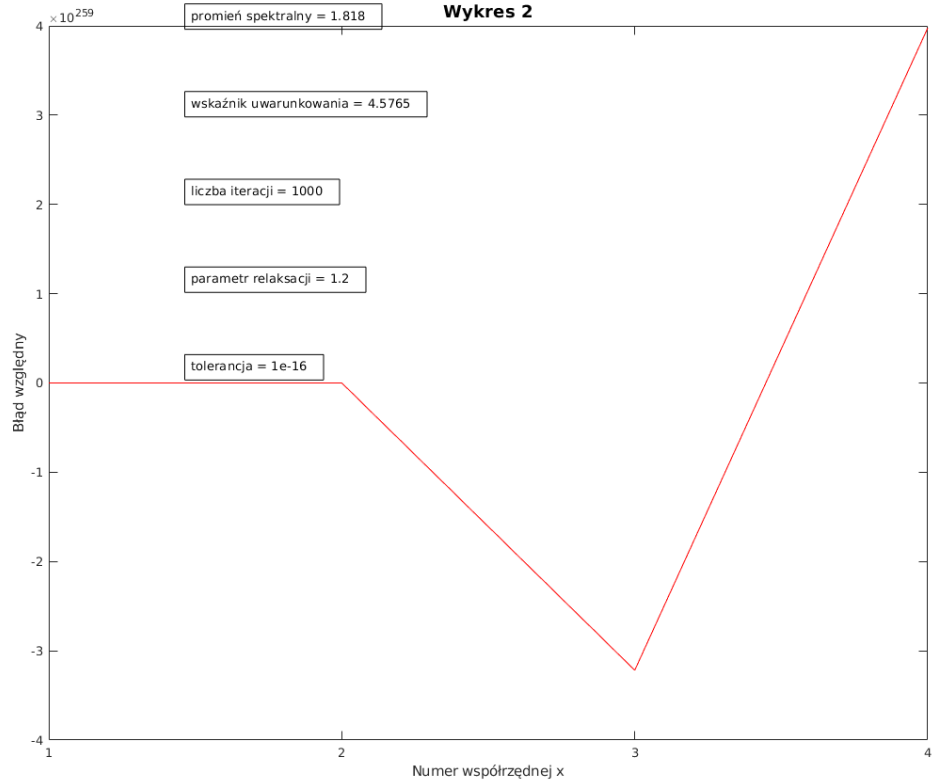
$$A = \begin{pmatrix} 5 & 4 & 0 & 0 & 0 \\ 1 & 6 & 3 & 0 & 0 \\ 0 & 2 & 7 & 2 & 0 \\ 0 & 0 & 3 & 8 & 1 \\ 0 & 0 & 0 & 4 & 9 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \quad (10)$$



Rysunek 1: Wykres przedstawia błąd względny dla kolejnych współrzędnych rozwiązania. Promień spektralny jest mały, więc metoda zbiega szybko. Widać to też po tym, że potrzebne było jedynie 25 iteracji, żeby poradzić sobie z tolerancją  $10^{-16}$ .

### 3.2 Przykład 2

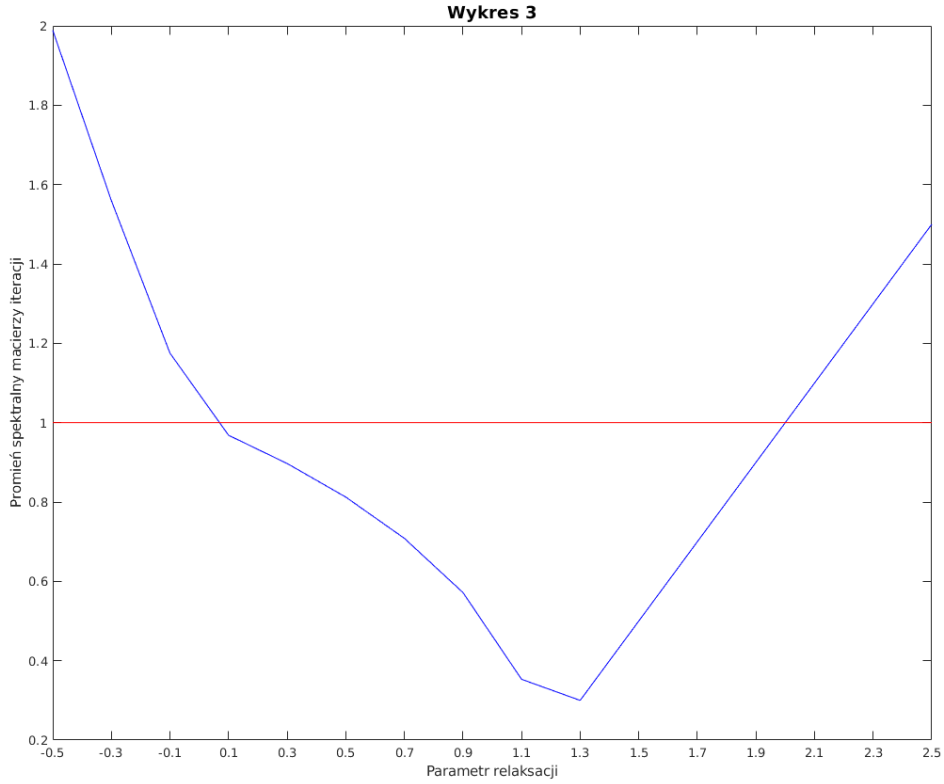
$$A = \begin{pmatrix} 1 & 0 & 0 & -3 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & -2 & 2 \end{pmatrix}, b = \begin{pmatrix} 3 \\ -1 \\ -2 \\ 1 \end{pmatrix} \quad (11)$$



Rysunek 2: Wykres przedstawia błąd względny dla kolejnych współrzędnych rozwiązania. Promień spektralny przekracza 1, więc metoda jest rozbieżna. Z tego powodu błąd względny wychodzi ogromny.

### 3.3 Przykład 3

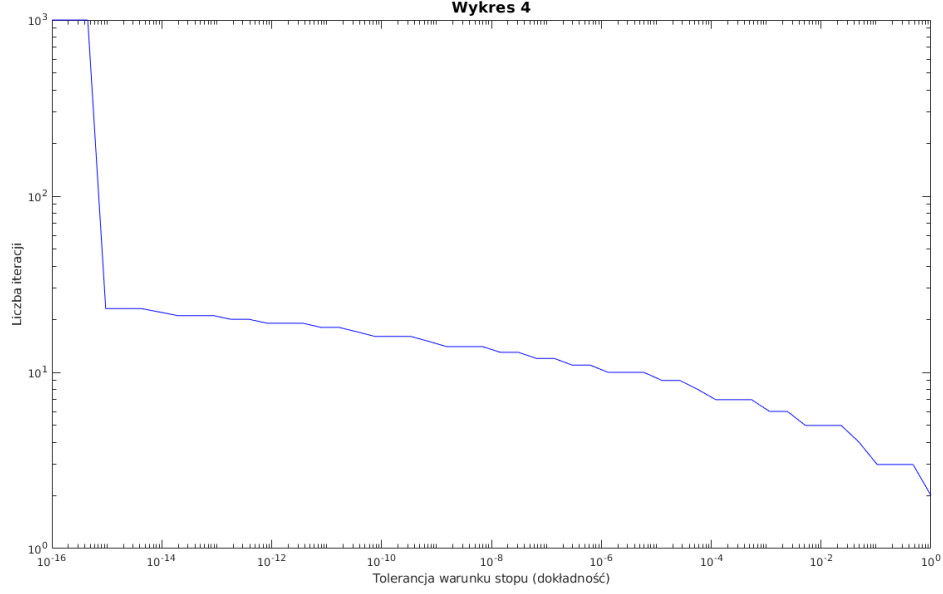
$$A = \begin{pmatrix} 2.04 & -1 & 0 \\ -1 & 2.04 & -1 \\ 0 & -1 & 2.04 \end{pmatrix}, b = \begin{pmatrix} 48.8 \\ 0.8 \\ 0.8 \end{pmatrix} \quad (12)$$



Rysunek 3: Wykres przedstawia zależność między parametrem relaksacji a promieniem spektralnym macierzy iteracji. Widać, że promień spektralny jest mniejszy od 1, a więc metoda jest zbieżna, dla  $\omega \in (0, 2)$ . Zbiega natomiast najszybciej dla parametru relaksacji w okolicach 1.3.

### 3.4 Przykład 4

$$A = \begin{pmatrix} 0 & -1 & -1 \\ 3 & 3 & 3 \\ -1 & -1 & 0 \end{pmatrix}, b = \begin{pmatrix} -1 \\ 7 \\ 7 \end{pmatrix} \quad (13)$$

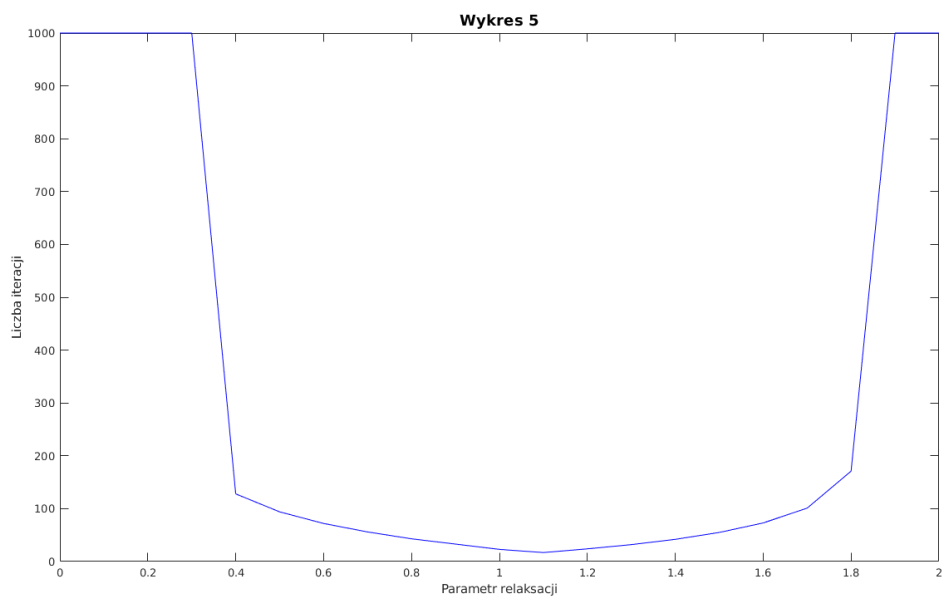


Rysunek 4: Wykres przedstawia zależność między tolerancją warunku stopu (dokładnością) a liczbą iteracji. Widać, że im mniejsza tolerancja jest wymagana, tym więcej iteracji potrzeba, aby uzyskać wynik z tą tolerancją. W pewnym momencie również (dla  $d \approx 10^{-15.5}$ ) liczba iteracji przekracza maksymalną liczbę iteracji równą 1000.

### 3.5 Przykład 5

$$A = \begin{pmatrix} 6 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 6 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 1 \\ 2 \\ -6 \\ 2 \\ 1 \\ 0 \end{pmatrix} \quad (14)$$

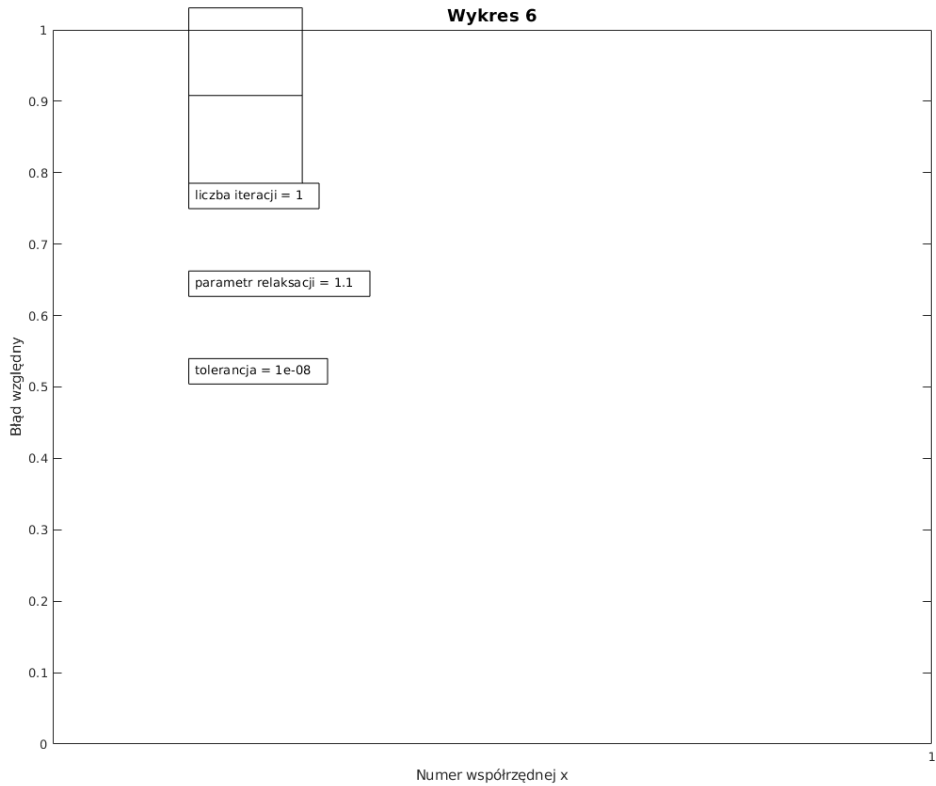




Rysunek 5: Wykres przedstawia zależność między parametrem relaksacji a potrzebną liczbą iteracji do uzyskania rozwiązania z daną stałą tolerancją ( $d = 10^{-15}$ ). Wykres przypomina parabolę. Liczba iteracji robi się bardzo duża dla  $\omega \in (-\infty, 0.4) \cup (1.8, \infty)$ . Najmniej iteracji potrzeba dla  $\omega \approx 1.1$ .

### 3.6 Przykład 6

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}, b = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{pmatrix} \quad (15)$$



Rysunek 6: Co się dzieje w przypadku układu sprzecznego? Program kończy pracę po 1 iteracji. Zwracany jest wektor  $(NaN, NaN, NaN, NaN, Inf)^T$

## 4 Analiza wyników

Tabela 1: Wyniki dla przykładu 1

L.p.	wartość dokładna	wartość przybliżona	błąd względny
1	0.0506	0.0506	$0.6856e-15$
2	0.1867	0.1867	$-0.2973e-15$
3	0.2763	0.2763	$0.4018e-15$
4	0.3462	0.3462	0
5	0.4017	0.4017	0

Tabela 2: Wyniki dla przykładu 2

L.p.	wartość dokładna	wartość przybliżona	błąd względny
1	3.0000	0	0
2	4.0000	0	0
3	-1.2500	$4.0221e+259$	$-3.2176e+259$
4	-0.7500	$-2.9830e+259$	$3.9773e+259$

Tabela 3: Wyniki dla przykładu 3

L.p.	parametr relaksacji	promień spektralny
1	-0.5000	1.9888
2	-0.3000	1.5597
3	-0.1000	1.1752
4	-0.1000	0.9682
5	-0.2000	0.8970
6	-0.5000	0.8124
7	-0.7000	0.7084
8	-0.9000	0.5718
9	1.1000	0.3532
10	1.3000	0.3000
11	1.5000	0.5000
12	1.7000	0.7000
13	1.9000	0.9000
14	2.1000	1.1000
15	2.3000	1.3000
16	2.5000	1.5000

Tabela 4: Wyniki dla przykładu 4

L.p.	tolerancja (dokładność)	liczba iteracji
1	1	2
2	$10^{-1}$	3
3	$10^{-2}$	5
4	$10^{-3}$	7
5	$10^{-4}$	8
6	$10^{-5}$	9
7	$10^{-6}$	11
8	$10^{-7}$	12
9	$10^{-8}$	14
10	$10^{-9}$	15
11	$10^{-10}$	16
12	$10^{-11}$	18
13	$10^{-12}$	19
14	$10^{-13}$	21
15	$10^{-14}$	22
16	$10^{-15}$	23
17	$10^{-16}$	1000

Tabela 5: Wyniki dla przykładu 5

L.p.	parametr relaksacji	liczba iteracji
1	0	1000
2	0.1000	1000
3	0.2000	1000
4	0.3000	1000
5	0.4000	128
6	0.5000	94
7	0.6000	72
8	0.7000	56
9	0.8000	43
10	0.9000	33
11	1.0000	23
12	1.1000	17
13	1.2000	24
14	1.3000	32
15	1.4000	42
16	1.5000	55
17	1.6000	73
18	1.7000	101
19	1.8000	171
20	1.9000	1000
21	2.0000	1000

Tabela 6: Wyniki dla przykładu 6

L.p.	wartość dokładna	wartość przybliżona	błąd względny
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	NaN	NaN	NaN
5	NaN	Inf	NaN

Na podstawie wyników można zauważyć kilka obserwacji:

- gdy metoda zbiega szybko, mało iteracji jest potrzebnych do uzyskania nawet bardzo dokładnych wyników,
- gdy metoda jest rozbieżna, błąd względny wychodzi bardzo duży,

- metoda jest rozbieżna, gdy  $\omega \notin (0, 2)$ ,
- metoda zbiega najszybciej dla  $\omega \approx 1.3$ ,
- przy zmniejszaniu tolerancji zwiększa się liczba potrzebnych iteracji,
- w przypadku  $d = 10^{-15}$  liczba iteracji przekracza 100 dla  $\omega \in (-\infty, 0.4) \cup (1.8, \infty)$ ,
- najmniej iteracji potrzeba dla  $\omega \approx 1.1$ .

## 5 Literatura

1. <https://www.degruyter.com/document/doi/10.1515/math-2016-0065/html>
2. [http://www.industrial-maths.com/ms6021\\_thomas.pdf](http://www.industrial-maths.com/ms6021_thomas.pdf)
3. <https://www.sciencedirect.com/science/article/pii/S0377042700007056>
4. [https://www-users.cse.umn.edu/saad/IterMethBook\\_2ndEd.pdf](https://www-users.cse.umn.edu/saad/IterMethBook_2ndEd.pdf)
5. <https://journalofinequalitiesandapplications.springeropen.com/counter/pdf/10.1186/s13660-016-1100-6.pdf>
6. [http://www-solar.mcs.st-andrews.ac.uk/clare/Lectures/num-analysis/Numan\\_chap2.pdf](http://www-solar.mcs.st-andrews.ac.uk/clare/Lectures/num-analysis/Numan_chap2.pdf)