

Laboratorium Metod Numerycznych

Układy równań liniowych - metody iteracyjne

Laboratorium 5

Omówione i przećwiczone w poprzednim ćwiczeniu metody skończone do rozwiązywania układów równań liniowych zapewniają uzyskanie poprawnych wyników. Jednakże ich zastosowanie dla dużych układów równań (rzędu tysięcy lub większej liczby niewiadomych) może być nieefektywne i znacznym stopniu zasobożerne (w sensie sprzętowym) oraz czasochłonne pod względem liczby wykonywanych operacji.

Alternatywą może być zastosowanie metod iteracyjnych. Metody te pozwalają na uzyskanie rozwiązania zadanego układu równań z określoną dokładnością ε . Należy jednak pamiętać że w odróżnieniu od metod skończonych, metody iteracyjne nie dla każdego układu równań będą zbieżne do rozwiązania.

Celem ćwiczenia jest opracowanie skryptów obliczeniowych na podstawie omówionych algorytmów wybranych metod numerycznych i wyznaczenie na ich podstawie wartości prądów gałęziowych w układzie analizowanym w poprzednim ćwiczeniu.

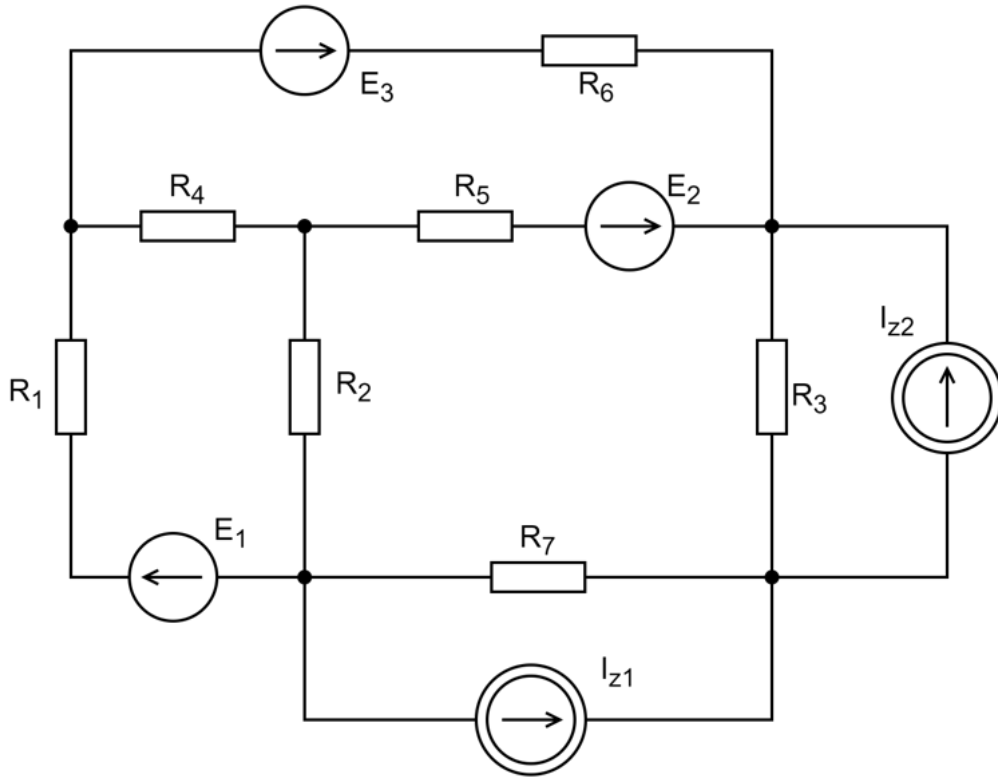
W ramach ćwiczenia analizowane będzie zastosowanie metod:

1. metoda Richardsona,
2. metoda Jacobiego,
3. metoda Gaussa-Seidela,
4. metoda Nadrelaksacji

Metody iteracyjne wyznaczają rozwiązanie układu równań liniowych poprzez ciąg kolejnych przybliżeń wyznaczanych na podstawie wcześniej wyznaczonych przybliżeń. Wszystkie metody iteracyjne wymagają więc podania wartości pierwszego przybliżenia rozwiązania. Zazwyczaj przyjmuje się warunki początkowe zerowe. Ogólną postać algorytmu obliczeniowego można zapisać:

$$x^k = (I - Q^{-1}A)x^{k-1} + Q^{-1}b, \quad \text{dla } k \geq 1 \quad (1)$$

Poszczególne metody charakteryzują się inną metodyką określania macierzy Q . Poszczególne metody omówione zostaną na przykładzie układu równań opisujących przykładowy obwód elektryczny analizowany w poprzednim ćwiczeniu, jak pokazano na Rys.1.



Rys. 1: Schemat obwodu rozgałęzionego prądu stałego

Dane dla zadania przykładowego.

$$R_x = 50\Omega, \quad E_x = 100V, \quad I_{zx} = 5A$$

Dla analizowanego układu można wyprowadzić układ równań liniowych $A \cdot I = b$. W ramach przykładowych obliczeń określono macierz A i wektor b z metody równań Kirchhoffa i metody oczkowej, jak pokazano poniżej:

- metoda równań Kirchhoffa

$$\begin{bmatrix} R_1 & -R_2 & 0 & R_4 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & -1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & -R_4 & -R_5 & R_6 & 0 \\ 0 & R_2 & -R_3 & 0 & R_5 & 0 & R_7 \end{bmatrix} \cdot \begin{bmatrix} I_1 \\ I_2 \\ I_3 \\ I_4 \\ I_5 \\ I_6 \\ I_7 \end{bmatrix} = \begin{bmatrix} E_1 \\ I_{z1} \\ -I_{z2} \\ 0 \\ 0 \\ E_3 - E_2 \\ E_2 \end{bmatrix} \quad (2)$$

- metod prądów oczkowych

$$\begin{bmatrix} R_1 + R_2 + R_4 & -R_2 & -R_4 \\ -R_2 & R_2 + R_3 + R_5 + R_7 & -R_5 \\ -R_4 & -R_5 & R_4 + R_5 + R_6 \end{bmatrix} \cdot \begin{bmatrix} I_I \\ I_{II} \\ I_{III} \end{bmatrix} = \begin{bmatrix} E_1 \\ E_2 - R_3 I_{z2} - R_7 I_{z1} \\ E_3 - E_2 \end{bmatrix} \quad (3)$$

,gdzie

$$\begin{aligned}I_1 &= I_I \\I_2 &= -I_4 + I_5 \\I_3 &= -I_{z2} - I_{II} \\I_4 &= I_1 - I_6 \\I_5 &= -I_3 - I_6 - I_{z2} \\I_6 &= I_{III} \\I_7 &= I_{z1} - I_{z2} - I_3\end{aligned}\tag{4}$$

Projektując algorytm obliczeniowy dla metod iteracyjnych dobrze jest zaimplementować w nim moduł detekcji rozbieżności modelu. Powinien on uwzględniać różne drogi dochodzenia do zbieżności (oscylacyjny, wykładniczy), a detekcję opierać na serii obliczeń a nie pojedynczej iteracji. Implementacja takiego algorytmu zabezpiecza funkcję obliczeniową przed wykonywaniem niepotrzebnych obliczeń gdy model jest rozbieżny.

Liczba iteracji modelu obliczeniowego może być z góry określona, co jest rozwiązaniem nieefektywnym i może się wiązać z niewystarczającą lub nadmiarową liczbą iteracji. Zdecydowanie korzystniejsze jest oparcie modelu obliczeniowego na pętli warunkowej testującej warunek jeden z dwóch warunków zbieżności do ε .

- $\|A \cdot x - b\| < \varepsilon$ - sprawdzenie czy norma wektora wyrażenia $|Ax-b|$ jest mniejsza od ε
- $\|x_n - x_{n-1}\| < \varepsilon$ - sprawdza czy norma różnica między ostatnimi dwoma przybliżeniami rozwiązania jest mniejsza od ε

W obliczeniach numerycznych realizowanych dla poszczególnych metod zaimplementowano oba mechanizmy kontroli zbieżności z dokładnością $\varepsilon = 10^{-10}$.

1) Metoda Richardsona

Metoda Richardsona przyjmuje macierz jednostkową I za macierz Q , algorytm ogólny (1) jest więc sprowadzany do postaci:

$$x^k = (I - A)x^{k-1} + b\tag{5}$$

Model numeryczny w środowisku obliczeniowym **Scilab**'a może być oparty bezpośrednio na zapisanym powyżej modelu obliczeniowym lub zapisanym poniżej algorytmie iteracyjnym. Ze względu na odwoływanie się bezpośrednio do komórek macierzy i wektorów może być zaimplementowany w każdym języku programowania. jeżeli dane środowisko programistyczne pozwala na wykonywanie działań na macierzach, można

zastosować oba modele obliczeniowe.

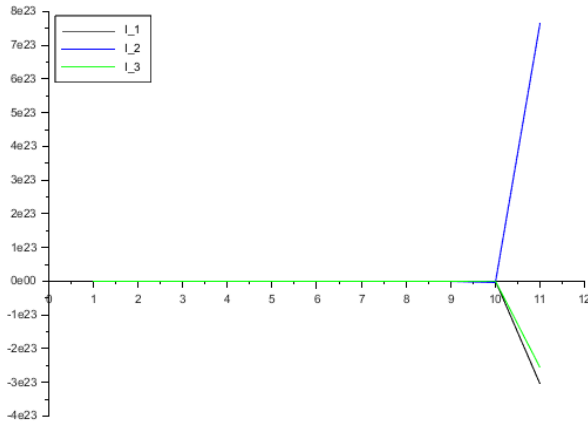
for $i = 1 : n$

$$r(i) = b(i) - \sum_{j=1}^n a(i, j) \cdot x_k(i) \quad (6)$$

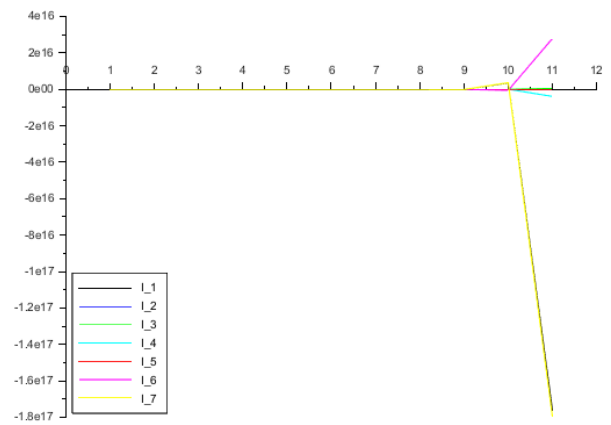
$$x_{k+1}(i) = x_k(i) + r(i)$$

gdzie n jest liczbą równań

Warunkiem podstawowym zbieżności metody jest aby norma $I - A$ była mniejsza od 1. Dla analizowanego obwodu i obu układów równań norma przyjmowała wartości większe od jedynki. Dla układu pełnego (równania Kirchhoffa) $\text{norm}(I-A)=108.3$, a dla układu uproszczonego (oczkowego) $\text{norm}(I-A)=235$. Metoda w obu przypadkach nie powinna być zbieżna, co potwierdziły symulacje pokazane na Rys. 7 i 8.



Rys. 2: Wykres zbieżności modelu oczkowego



Rys. 3: Wykres zbieżności modelu Kirchhoffa

Obliczenia dla obu zestawu danych wejściowych okazały się rozbieżne. Symulację przerwano w obu przypadkach po 11 iteracjach.

W metodzie Richardsona wprowadzenie współczynnika relaksacji może pozwolić uzyskanie zbieżności. Współczynnik ten koryguje wartość współczynnika przyrostowego kolejnego rozwiązania zwracanego przez metodę, jak pokazano poniżej.

$$x^k = (I - \omega \cdot A)x^{k-1} + \omega \cdot b \quad (7)$$

$$x^k = x^{k-1} + \omega (b - A \cdot x^k) \quad (8)$$

Dobierając wartość współczynnika relaksacji można sterować zbieżnością:

- $\omega = 1$ korekta nie wpływa na algorytm metody,
- $\omega < 1$ korekta jest "łagodniejsza", metoda jest wolniejsza, ale stabilna zbieżność,
- $\omega > 1$ korekta jest "agresywna", możliwa jest szybsza zbieżność metody, ale rośnie ryzyko rozbieżności.

Do określenia wartości współczynnika korygującego wykorzystuje się wartości własne macierzy $\mathbf{A}$. W środowisku obliczeniowym Scilab można je wyznaczyć za pomocą funkcji $\text{spec}(\mathbf{A})$. Dla analizowanego układu równań przyjmują one wartości:

$$\lambda = \text{spec}(A) = \begin{bmatrix} 79.3 \\ 221 \\ 250 \end{bmatrix} \quad (9)$$

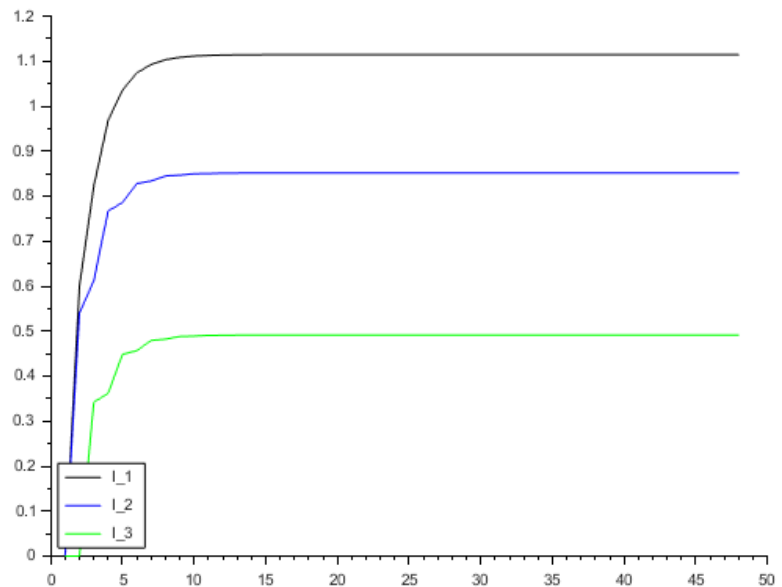
Wartość $\lambda_{\max} = 250$ a $\lambda_{\min} = 79.3$, na ich podstawie można określić przedział dopuszczalnych wartości współczynnika relaksacji ω oraz jego optymalną wartość:

$$0 < \omega < \frac{2}{\lambda_{\max}} = 0.008 \quad (10)$$

$$\omega = \frac{2}{\lambda_{\max} + \lambda_{\min}} = 0.00607 \simeq 0.006 \quad (11)$$

W analizowanym przypadku wprowadzenie współczynnika relaksacji nie poprawia zbieżności rozwiązania układu równań opartych na prawach Kirchhoffa, ale po ustawieniu wartości $\omega < 0.006$ model obliczeniowy staje się zbieżny i zwraca poprawne rozwiązanie przy przyjętej dokładności po 461 krokach, jak pokazano na Rys 4.

Wyniki po 48 krokach.
 $\text{Imetody}[1] = -0.417$
 $\text{Imetody}[2] = -2.333$
 $\text{Imetody}[3] = -0.917$



Rys. 4: Wykres zbieżności rozwiązania układu równań oczkowych dla metody Richardsona z współczynnikiem relaksacji 0.006

2) Metoda Jacoby'ego

Punktem wyjścia dla określenia algorytmu dla metody Jacoby'ego jest zastąpienie macierzy A układem równoważnym pokazanym poniżej.

$$A = L + D + U \quad (12)$$

Określenie elementów macierzy L , D i U polega na rozdzielaniu elementów macierzy A względem przekątnej diagonalnej, której elementy należy zapisać w macierzy D , występujące poniżej w macierzy L , a powyżej w macierzy U .

$$(L + D + U) \cdot x = b \quad (13)$$

W efekcie równanie (7) można w sposób analogiczny do metody ogólnej (2) sprowadzić do poniższej postaci.

$$D\hat{x} = -(L + U) \cdot x + b \quad (14)$$

Po przeniesieniu macierzy diagonalnej D na prawą stronę uzyskuje się równanie macierzowe zwracające kolejne przybliżenie równania macierzowego $Ax = b$.

$$\hat{x} = -D^{-1} \cdot (L + U) \cdot x + D^{-1} \cdot b \quad \longrightarrow \quad \hat{x} = B \cdot x + c \quad (15)$$

Analogicznie jak w przypadku metody Richardson'a można model (15) zaimplementować bezpośrednio w skrypcie obliczeniowym **Scilab**'a lub zastosować uniwersalny algorytm iteracyjny, zapisany poniżej. Warto zwrócić uwagę że zastosowanie algorytmu iteracyjnego nie wymaga rozkładu macierzy głównej na macierze L , D i U , gdyż rozkład ten jest zaimplementowany bezpośrednio w algorytmie.

$$\begin{aligned} & \text{for } i = 1 : n \\ & \quad \text{suma}(i) = \sum_{j=1, (j \neq i)}^n a(i, j) \cdot x_k(j) \\ & \quad x_{k+1}(i) = \frac{b(i) - \text{suma}(i)}{a(i, i)} \\ & \text{gdzie } n \text{ jest liczbą równań} \end{aligned} \quad (16)$$

Rozwiązanie danego równania $A \cdot x = b$ metodą Jacoby'ego jest zbieżne jeżeli macierz A jest nieredukowalna i diagonalnie dominująca. Macierz A jest nieredukowalna, jeżeli poprzez przestawienie wierszy i kolumn nie można jej sprowadzić do postaci blokowej górnej trójkątnej. Macierz kwadratowa A określa się jako diagonalnie dominującą, jeśli dla $i = 1, 2, \dots, n$ zachodzi nierówność:

$$|a_{(i,i)}| \geq \sum_{j=1, (j \neq i)}^n |a_{(i,j)}| \quad (17)$$

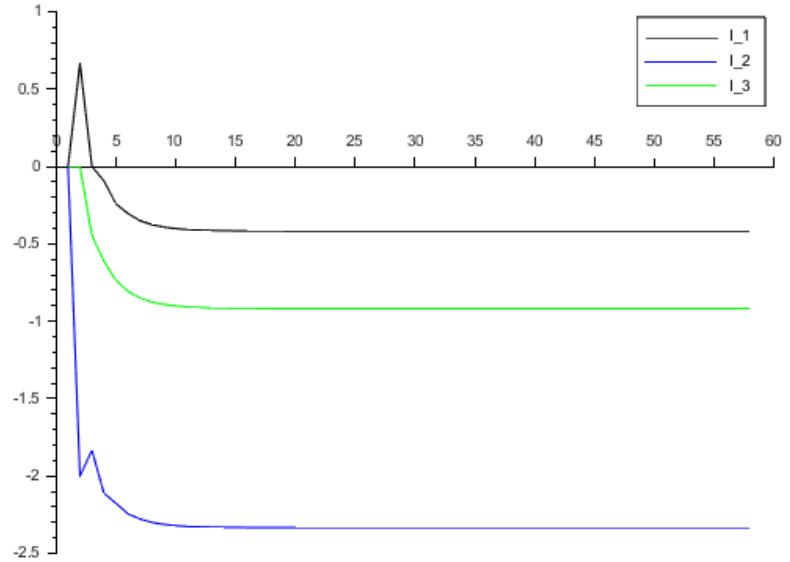
Za pomocą algorytmu macierzowego i iteracyjnego przeprowadzono symulację dla obu przykładowych układów równań. Wyniki zbieżne otrzymano dla oczkowego układu równań i zestawiono poniżej, w postaci wartości wyliczonych prądów i wykresu zbieżności Rys. 5. Wyniki dla obydwu algorytmów obliczeniowych uzyskano identyczne.

Wyniki po 58 krokach.

Imetody[1]=-0.417

Imetody[2]=-2.333

Imetody[3]=-0.917



Rys. 5: Wykres zbieżności modelu oczkowego

3) Metoda Gaussa-Seidela

Metoda Gaussa-Seidela jest modyfikacją metody Jacoby'ego, w której z wyjściowego równania $(L + D + U) \cdot x = b$ wyłącza się $L + D$.

$$(L + D) \cdot \hat{x} = -U \cdot x + b \quad (18)$$

Przekształcając powyższe równanie do postaci opisującej kolejne przybliżenie rozwiązania układu równań uzyskuje się postać zapisaną poniżej.

$$\hat{x} = -(L + D)^{-1}U \cdot x + (L + D)^{-1} \cdot b \quad (19)$$

Powyższe równanie macierzowe (13) można zapisać w postaci algorytmu iteracyjnego możliwego do zaimplementowania w dowolnym języku programowania.

for $i = 1 : n$

$$suma_I = \sum_{j=1, (j < i)}^n a(i, j) * x_{k+1}(j)$$

$$suma_{II} = \sum_{j=1, (j > i)}^n a(i, j) * x_k(j) \quad (20)$$

$$x_{k+1}(i) = \frac{b(i) - suma_I - suma_{II}}{a(i, i)}$$

gdzie n jest liczbą równań

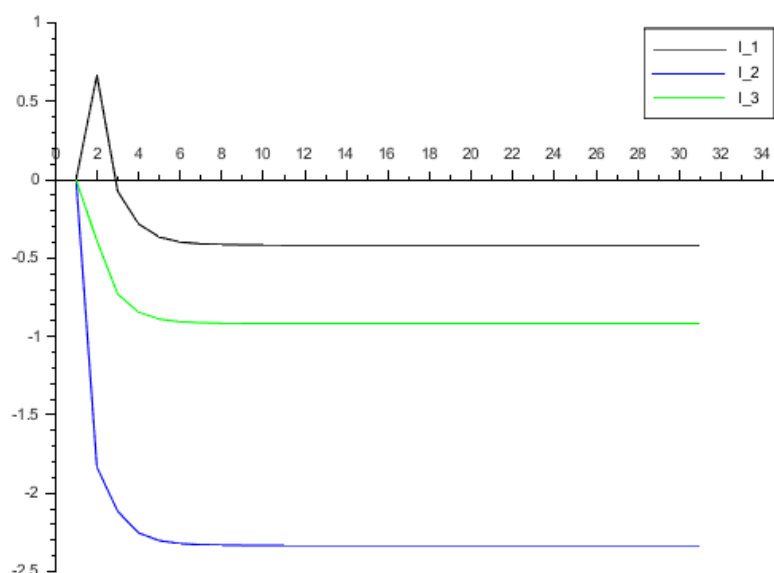
Za pomocą algorytmu macierzowego i iteracyjnego przeprowadzono symulację dla obu przykładowych układów równań. Wyniki zbieżne otrzymano dla oczkowego układu równań i zestawiono poniżej, w postaci wartości wyliczonych prądów i wykresu zbieżności Rys. 6. Wyniki dla obydwu algorytmów obliczeniowych uzyskano identyczne. W odróżnieniu od wcześniej opisanych metod algorytm Gaussa-Seidela jest prawie dwukrotnie szybszy, zwracając wyniki dla żądanej dokładności po 31 krokach.

Wyniki po 31 krokach.

Imetody[1]=-0.417

Imetody[2]=-2.333

Imetody[3]=-0.917



Rys. 6: Wykres zbieżności modelu oczkowego

4) Metoda Nadrelaksacji

Jest z kolei modyfikacją metody Gaussa-Seidela, której celem jest przyspieszenie obliczeń poprzez wprowadzenie współczynnika ω do przemnożenia poprawki obliczeniowej. Współczynnik ω dobiera się w dopuszczalnym zakresie $(0,2)$, ale zazwyczaj najlepsze efekty uzyskuje się dla ω w okolicach wartości 1.

Punktem wyjściowym wyprowadzenia równania macierzowego dla metody jest równanie macierzowe z rozłożeniem macierzy głównej A na sumę macierzy L , D i U obustronnie przemnożone przez współczynnik ω , jak pokazano poniżej.

$$\omega \cdot (L + D + U) \cdot x = \omega \cdot b \quad (21)$$

Po przekształceniach algorytm macierzowy przyjmuje ostateczną formę pokazano poniżej.

$$\hat{x} = (D + \omega L)^{-1} \cdot ((1 - \omega) \cdot D - \omega U) \cdot x + \omega \cdot (D + \omega L)^{-1} \cdot b \quad (22)$$

Analogicznie jak przy wcześniej opisanych metodach, dla metody nadrelaksacji także można zapisać algorytm iteracyjny realizujący rachunek macierzowy realizowany

przez model macierzowy metody (18).

for $i = 1 : n$

$$suma_I = \sum_{j=1, (j < i)}^n a(i, j) * x_{k+1}(j)$$

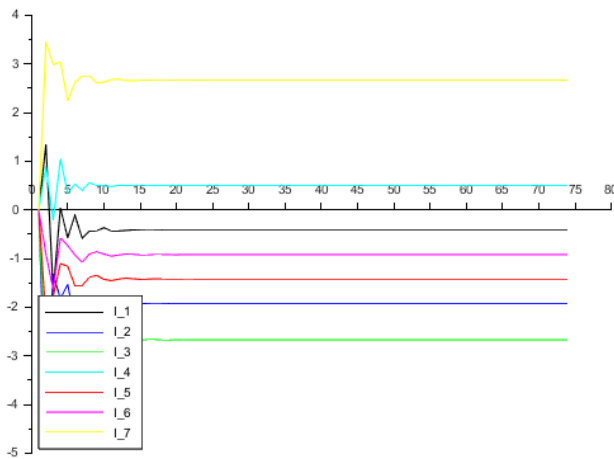
$$suma_{II} = \sum_{j=1, (j > i)}^n a(i, j) * x_k(j) \quad (23)$$

$$x_{k+1}(i) = \frac{(1 - \omega) \cdot a(i, i) \cdot x_k(i) + \omega \cdot (b(i) - suma_I - suma_{II})}{a(i, i)}$$

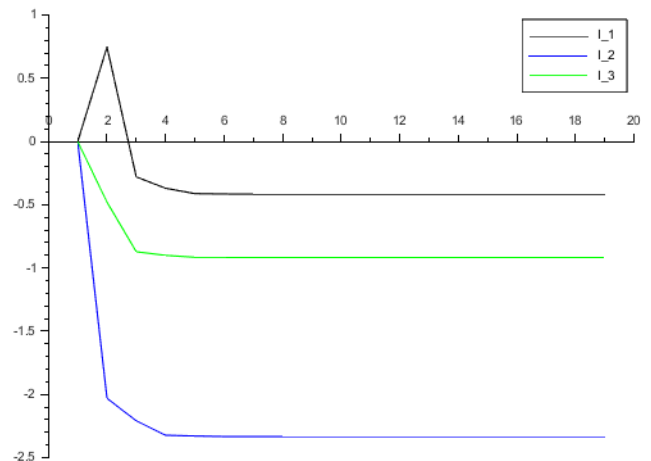
gdzie n jest liczbą równań

Poszukiwanie rozwiązania układu równań metodą nadrelaksacji dla żądanej dokładności ε należy przeprowadzić dla zakresu wartości ω z zakresu (0,2). W zależności od zadanego układu równań powyżej pewnej granicznej wartości ω (<2) model numeryczny traci zbieżność.

Dla przykładowych układów równań przeprowadzono symulację z wykorzystaniem algorytmu nadrelaksacji z skokiem wartości współczynnika ω co 0.01 do 2 uzyskując najszybszą zbieżność dla parametrów wejściowych z równań Kirchhoffa i $\omega=0.67$ zwracając wynik o żądanej dokładności po 74 krokach. Model obliczeniowy utracił zbieżność dla $\omega=0.71$. Znacznie lepsze efekty uzyskano dla układu równań z metody prądów oczkowych. Wykonano symulację obliczeniową dla ω w zakresie od 0.1 do 2 uzyskując najszybszą zbieżność po 19 krokach dla $\omega=1.12$. Model w tym przypadku tracił zbieżność dla $\omega=1.86$. Powyższe wyniki w sposób identyczny uzyskano dla algorytmu macierzowego i iteracyjnego, jak pokazano na Rys. i Rys. .



Rys. 7: Wykres zbieżności modelu Kirchhoffa



Rys. 8: Wykres zbieżności modelu oczkowego

5) Zadanie

Na podstawie informacji zawartych w niniejszej instrukcji wyznaczyć rozplływ prądu w rozgałęzionym obwodzie elektrycznym prądu stałego analizowanym w poprzednich ćwiczeniach. Wartości prądów wyznaczyć rozwiązując układ równań liniowych opisujący zaprojektowany układ metodami:

1. metodą Richardson'a,
2. metodą Jacoby'ego,
3. metodą Gaussa-Seidela,
4. metodą Nadrelaksacji.

Otrzymane wyniki odnieść do wartości uzyskanych w poprzednim ćwiczeniu.

Opracowany skrypt:

- wykonuje obliczenia zaleconymi metodami,
- wyświetla informacje o danych wejściowych zadania,
- wyświetla rozwiązywane układy równań w postaci macierzowej,
- wyświetla wyniki w postaci listy indeksowanej prądów gałęziowych,
- wykreślić wykres zbieżności rozwiązań,
- opracować i zaimplementować w modelach obliczeniowych algorytmy:
 - wykrywania spełnienia warunku dokładności ε ,
 - wykrywania utraty zbieżności (przerwanie obliczeń).

Poszczególne zadania obliczeniowe (metody obliczeniowe) i wybrane czynności prezentacji danych realizują funkcje. Zdefiniowane funkcje zapisać w zewnętrznym pliku *.sci i załadować w głównym skrypcie programu.

W przypadku metody Nadrelaksacji przeanalizować zakres wartości ω od 0.01 do 2 w poszukiwaniu optymalnej i granicznej wartości ω . W sprawozdaniu umieścić wykres zbieżności dla wybranego prądu dla zakresu wartości ω spełniających kryterium zbieżności.