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:

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

Poszczególne metody charakteryzują się inną metodyką określania macierzy Q.

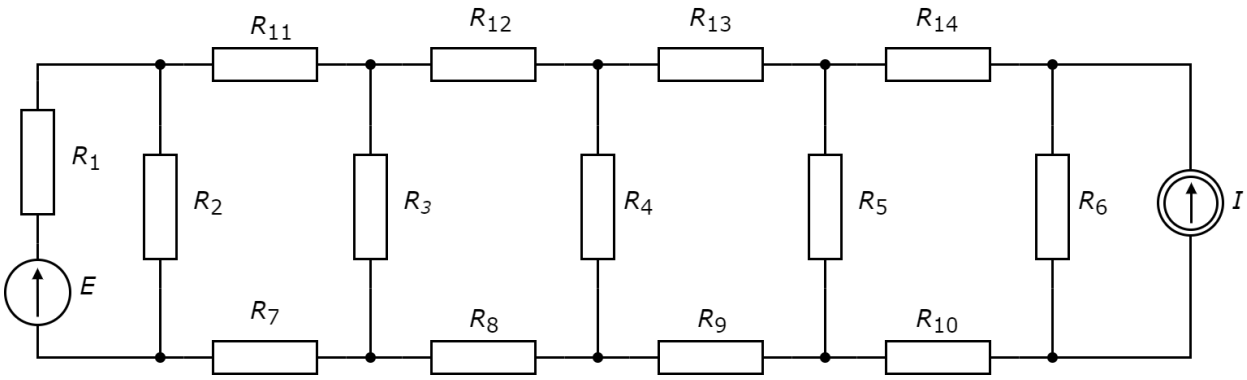
1) Ogólna metoda iteracyjna

Kolejne przybliżenia rozwiązania w metodzie ogólnej wyznaczane są na podstawie równań powstałych poprzez wyłączenie wybranych niewiadomych z kolejnych równań

układu. Optymalnie z pierwszego wyciąga się pierwszą niewiadomą, z drugiego drugą itd.

$$A \cdot x = b \quad \longrightarrow \quad \hat{x} = Bx + c \tag{2}$$

Implementacja numeryczna metody wymaga określenia macierzy B i c na podstawie algorytmu (2) na podstawie macierzy A i B z zadanego równania macierzowego. Dla przykładowego obwodu z Rys.1 układ równań zapisano poniżej. Warunkiem koniecznym dla przeprowadzenia obliczeń metodą ogólną jest brak elementów zerowych na przekątnej diagonalnej. W przypadku ich występowania konieczne jest przetworzenie macierzy głównej równania aby spełniało zadany warunek za pomocą przekształceń elementarnych.



Rys. 1: Schemat drabinkowego obwodu rozgałęzionego prądu stałego

$$\begin{aligned}
&(R_1 + R_2)I_I - R_2 I_{II} = E \\
&-R_2 I_I + (R_2 + R_3 + R_7 + R_{11})I_{II} - R_3 I_{III} = 0 \\
&-R_3 I_{II} + (R_3 + R_4 + R_8 + R_{12})I_{III} - R_4 I_{IV} = 0 \\
&-R_4 I_{III} + (R_4 + R_5 + R_9 + R_{13})I_{IV} - R_5 I_V = 0 \\
&-R_5 I_{IV} + (R_5 + R_6 + R_{10} + R_{14})I_V = -R_6 I_z
\end{aligned} \tag{3}$$

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 wykonywanie niepotrzebnych obliczeń gdy model jest rozbieżny.

Liczba iteracji modelu obliczeniowego może być z góry określona, co jest rozwiązaniem nieefektywnym i może się wiązać z niewystarczająca 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

2) Metoda Richardsona

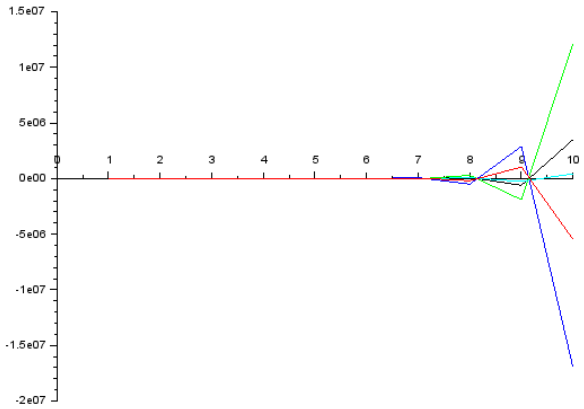
Metoda Richardsona przyjmuje macierz jednostkową I za macierz Q, algorytm ogólny (1) jest więc sprowadzany do postaci:

$$x^k = (I - A)x^{k-1} + b \tag{4}$$

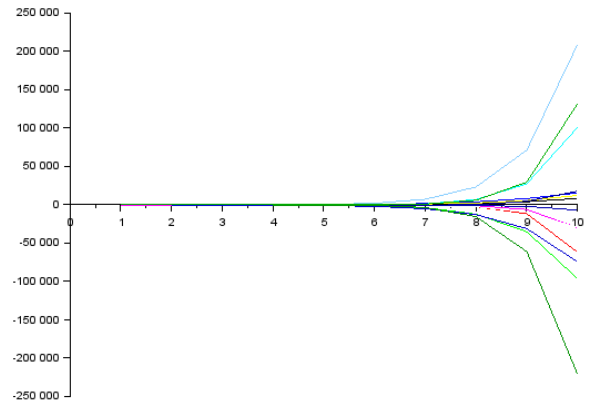
Model numeryczny w środowisku obliczeniowym ***Scilab***'a może być oparty bezpośrednio na zapisanym powyżej modelu obliczeniowym lub zapisanym poniżej algorytmie iteracyjnym. Ze względu na odwoływanie się bezpośrednio do komórek macierzy i wektorów może być zaimplementowany w każdym języku programowania.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

$$\begin{aligned} &for\ i = 1 : n \\ &\quad r(i) = b(i) - \sum_{j=1}^{n} a(i,j) \cdot x_k(i) \\ &\quad x_{k+1}(i) = x_k(i) + r(i) \\ &gdzie\ n\ jest\ liczbą\ równań \end{aligned} \tag{5}$$

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) `norm(I-A)=3.77`, a dla układu uproszczonego (oczkowego) `norm(I-A)=6.65`. Metoda w obu przypadkach nie powinna być zbieżna, co potwierdziły symulacje pokazane na Rys. 2 i 3.



Rys. 2: Wykres zbieżności modelu oczkowego



Rys. 3: Wykres zbieżności modelu Kirchhoffa

3) Metoda Jacoby'ego

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

$$A = L + D + U \tag{6}$$

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

$$(L + D + U) \cdot x = b \tag{7}$$

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

$$D\hat{x} = -(L + U) \cdot x + b \tag{8}$$

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

$$\hat{x} = -D^{-1} \cdot (L + U) \cdot x + D^{-1} \cdot b \quad \longrightarrow \quad \hat{x} = B \cdot x + c \tag{9}$$

Analogicznie jak w przypadku metody Richardson'a można model (9) 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.

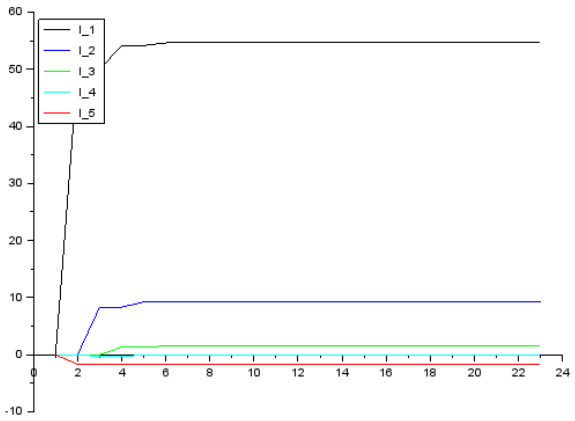
$$\begin{aligned} &for\ i = 1 : n \\ &\quad suma(i) = \sum_{j=1(j\neq i)}^{n} a(i,j) \cdot x_k(j) \\ &\quad x_{k+1}(i) = \frac{b(i) - suma(i)}{a(i,i)} \\ &gdzie\ n\ jest\ liczbą\ równań \end{aligned} \tag{10}$$

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 $i = 1, 2, ..., n$ zachodzi nierówność:

$$|a_{(i,i)}| \geq \sum_{j=1,(j\neq i)}^{n} |a_{(i,j)}| \tag{11}$$

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. 4. Wyniki dla obydwu algorytmów obliczeniowych uzyskano identyczne.

```
"Wyniki uzyskano po 23 krokach."
   I_1=54.7
   I_2=9.37
   I_3=1.56
   I_4=-0.02
   I_5=-1.67
```



Rys. 4: Wykres zbieżności modelu oczkowego

4) Metoda Gaussa-Seidela

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

$$(L + D) \cdot \hat{x} = -U \cdot x + b \tag{12}$$

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

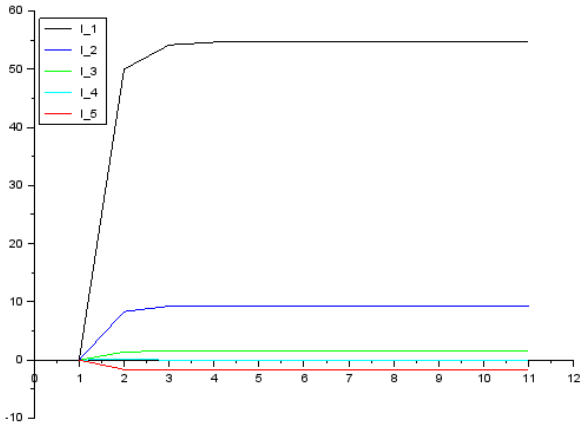
$$\hat{x} = -(L + D)^{-1} U \cdot x + (L + D)^{-1} \cdot b \tag{13}$$

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

$$\begin{aligned} &for\ i = 1 : n \\ &\quad suma_I = \sum_{j=1,(j<i)}^{n} a(i,j) * x_{k+1}(j) \\ &\quad suma_{II} = \sum_{j=1,(j>i)}^{n} a(i,j) * x_k(j) \\ &\quad x_{k+1}(i) = \frac{b(i) - suma_I - suma_{II}}{a(i,i)} \\ &gdzie\ n\ jest\ liczbą\ równań \end{aligned} \tag{14}$$

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. W odróżnieniu od wcześniej opisanych metod algorytm Gaussa-Seidela jest ponad dwukrotnie szybszy, zwracając wyniki dla żądanej dokładności po 11 krokach.

```
"Wyniki uzyskano po 11 krokach."
   I_1=54.7
   I_2=9.37
   I_3=1.56
   I_4=-0.02
   I_5=-1.67
```



Rys. 5: Wykres zbieżności modelu oczkowego

5) Metoda Nadrelaksacji

Jest z kolei modyfikacją metody Gaussa-Seidela, której celem jest przyśpieszenie 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.

$$\omega \cdot (L + D + U) \cdot x = \omega \cdot b \tag{15}$$

Po przekształceniach algorytm macierzowy przyjmuje ostateczną formę pokazano poniżej.

$$\hat{x} = (D + \omega L)^{-1} \cdot ((1 - \omega) \cdot D - \omega U) \cdot x + \omega \cdot (D + \omega L)^{-1} \cdot b \tag{16}$$

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

przez model macierzowy metody (18).

$$
\begin{aligned}
&for\ i = 1 : n \\
&\quad suma_I = \sum_{j=1,(j<i)}^{n} a(i,j) * x_{k+1}(j) \\
&\quad suma_{II} = \sum_{j=1,(j>i)}^{n} a(i,j) * x_k(j) \\
&\quad x_{k+1}(i) = \frac{(1-\omega)\cdot a(i,i)\cdot x_k(i) + \omega\cdot(b(i) - suma_I - suma_{II})}{a(i,i)}
\end{aligned}
\tag{17}
$$

gdzie n jest liczbą równań

Poszukiwanie rozwiązania układu równań metodą nadrelaksacji dla żądanej dokładności ε należny przeprowadzić dla zakresu wartości ω z zakresu (0,2). W zależności od zadanego układu równań powyżej pewnej granicznej wartości ω poniżej wartości granicznej 2 model może utrącić zbieżność.

Dla przykładowych układów równań przeprowadzono symulację z wykorzystaniem algorytmu nadrelaksacji z skokiem wartości współczynnika ω co 0.01 do 1.9 uzyskując najszybszą zbieżność o wartości 9 kroków dla ω=1.04. Metoda nadrelaksacji pozwoliła także na uzyskanie zbieżności dla układu równań na podstawie praw Kirchhoffa dla ω w zakresie do 0.31, uzyskując najszybszą zbieżność po 1776 krokach dla $\omega = 0.18$. Powyżej górnych granic model dla obydwu układów równań tracił zbieżność. Powyższe wyniki w sposób identyczny uzyskano dla algorytmu macierzowego i iteracyjnego.

6) Zadanie

Na podstawie informacji zawartych w niniejszej instrukcji wyznaczyć rozpływ prądu w rozgałęzionym obwodzie elektrycznym prądu stałego w układzie drabinkowym rozpatrywanym w poprzednik ćwiczeniu. 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 do 2 w poszukiwaniu optymalnej wartości. Dodać wykres zbieżności dla wybranego prądu dla zakresu wartości ω spełniających kryterium zbieżności.