

# Metody numeryczne w elektrotechnice

## Ćwiczenie 2 — Macierzowy opis problemów elektrotechniki, podstawy obliczeń macierzowych

Dariusz Borkowski

### Wstęp

W tym ćwiczeniu rozwiążecie pewne typowe problemy techniczne, które zwykle opisuje się za pomocą równań macierzowych.

## 1 Rozwiązanie obwodu stałoprądowego metodą prądów oczkowych — 2 pkt

### 1.1 Obliczenie prądów oczkowych metodą Cramera

Metodą prądów oczkowych<sup>1</sup> wyprowadź (na papierze) układ równań opisujący obwód pokazany na rysunku 1, a następnie przekształć te równania do postaci macierzowej

$$\mathbf{R}\mathbf{I} = \mathbf{E} \quad (1)$$

gdzie:  $\mathbf{R}$  – macierz rezystancji,  $\mathbf{I} = [I_1 \ I_2 \ I_3]^T$  – wektor prądów oczkowych (zaznaczonych na rysunku),  $\mathbf{E}$  – wektor napięć źródłowych.

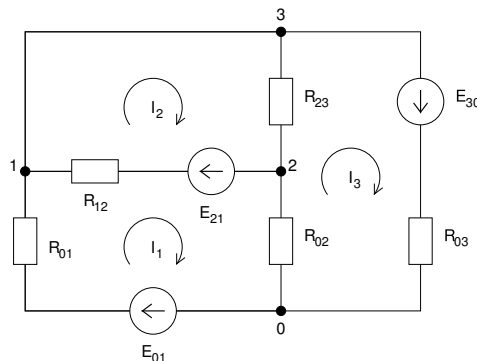
Następnie napisz w Matlabie program rozwiązujący układ równań (1) metodą Cramera. W metodzie Cramera poszukiwane niewiadome oblicza się według zależności

$$I_i = \frac{\det \mathbf{R}_i}{\det \mathbf{R}} \quad (2)$$

gdzie  $\mathbf{R}_i$  jest macierzą, której  $i$ -ta kolumna została zastąpiona wektorem  $\mathbf{E}$ .

Wartości elementów obwodu przepisuj z tabeli 1. Tabela ta prezentuje typowy sposób opisu obwodów elektrycznych stosowany przy wprowadzaniu danych tego typu w celu rozwiązania obwodu (np. program Spice). UWAGA! Napięcia w tabeli mają zwrot od węzła A (o niższym numerze) do węzła B (o wyższym numerze), i w dwóch przypadkach są niezgodne ze zwrotami napięć na rysunku 1. Aby rozwiązać wątpliwości dopisałem po prawej stronie tabeli dwie kolumny z oznaczeniami elementów.

Na podstawie wyliczonych wartości prądów oczkowych  $\mathbf{I}$  oblicz napięcia na rezystorach  $R_{02}$  i  $R_{12}$ .



Rysunek 1: Obwód stałoprądowy do zadania 1.1.

<sup>1</sup>Przypominam, że tworzenie równań metodą napięć oczkowych, polega na wykorzystaniu II prawa Kirchhoffa, mówiącego, że suma napięć w każdym oczku wynosi zero.

| Nr gałęzi | Nr węzła A | Nr węzła B | Rezystancja [ $\Omega$ ] | Napięcie [V] | Rezystor | Źródło napięcia    |
|-----------|------------|------------|--------------------------|--------------|----------|--------------------|
| 1         | 0          | 1          | 2                        | 10           | $R_{01}$ | $E_{01}$           |
| 2         | 0          | 2          | 0,5                      |              | $R_{02}$ |                    |
| 3         | 0          | 3          | 2                        | -10          | $R_{03}$ | $-E_{03} = E_{30}$ |
| 4         | 1          | 2          | 0,25                     | -5           | $R_{12}$ | $-E_{12} = E_{21}$ |
| 5         | 1          | 3          | 0                        |              | $R_{13}$ |                    |
| 6         | 2          | 3          | 0,5                      |              | $R_{23}$ |                    |

Tabela 1: Wartości elementów obwodu do zadania 1.1.

## 2 Sprawdzenie rozwiązania poprzedniego zadania w języku C — 2 pkt

Wykorzystując macierz  $R$  oraz wektor  $I$  wyznaczone w poprzednim zadaniu oblicz (z pomocą biblioteki GSL wektor  $E$  według (1) i sprawdź czy otrzymany wynik jest równy wektorowi  $E$  wyprowadzonemu w poprzednim zadaniu.

W środowisku CodeBlocks utwórz nowy projekt konsolowy. Następnie do pliku głównego main.c wklej szkielet programu z pliku lab02z02.c i uzupełnij brakujące części zgodnie z poleceniami podanymi w komentarzach. Przedstawiam skrócony<sup>2</sup> opis potrzebnych funkcji.

`gsl_vector * gsl_vector_alloc (size_t n)` tworzy wektor o długości  $n$  i zwraca wskaźnik do niego. Typ danych `gsl_vector` jest strukturą opisaną m.in. w pliku nagłówkowym `include/gsl/gsl_vector_double.h`.

`gsl_matrix * gsl_matrix_alloc (size_t n1, size_t n2)` tworzy macierz o  $n1$  wierszach i  $n2$  kolumnach i zwraca wskaźnik do niej. Przykład: `gsl_matrix *A = gsl_matrix_alloc (3, 3);` .

`void gsl_vector_set (gsl_vector * v, size_t i, double x)` wpisuje wartość  $x$  do  $i$ -tego elementu wektora  $v$ . Przykład: `gsl_vector_set(b, 3, 0.75);` .

`void gsl_matrix_set (gsl_matrix * m, size_t i, size_t j, double x)` wpisuje wartość  $x$  do  $i, j$ -tego elementu macierzy  $m$ . Przykład: `gsl_matrix_set(A, 3, 2, -1.53);` .

`int gsl_blas_dgemv (CBLAS_TRANSPOSE_t TransA, double alpha, const gsl_matrix * A, const gsl_vector * x, double beta, gsl_vector * y)` wykonuje operację<sup>3</sup>  
 $y = \alpha \text{op}(A)x + \beta y$ , czyli skalar  $\alpha$  razy macierz  $A$  (na której wykonano operację `op()`) razy wektor  $x$  plus skalar  $\beta$  razy wektor  $y$ . Wynik zostaje umieszczony w wektorze  $y$ . Przed wykonaniem działań można wykonać operację `op()` na macierzy  $A$ . Możliwe operacje to: brak operacji, transpozycja macierzy, transpozycja ze sprzężeniem liczb zespolonych, a odpowiadające tym operacjom wartości parametru `TransA` to: `CblasNoTrans`, `CblasTrans`, `CblasConjTrans`.

`double gsl_vector_get (const gsl_vector * v, size_t i)` zwraca wartość  $i$ -tego elementu wektora  $v$ . Przykład `liczba = gsl_vector_get(b, 4)` .

`double gsl_matrix_get (const gsl_matrix * m, size_t i, size_t j)` zwraca wartość  $i, j$ -tego elementu macierzy  $m$ . Przykład `liczba = gsl_matrix_get(A, 3, 2)` .

`void gsl_vector_free (gsl_vector * v)` zwalnia pamięć po wektorze  $v$ .

`void gsl_matrix_free (gsl_matrix * m)` zwalnia pamięć po macierzy  $m$ .

Uwaga! Istnieją także wersje funkcji do alokacji wektorów i macierzy, które od razu inicjalizują wektor lub macierz zerami. Od opisanych wyżej różnią się tylko nazwą np.: `gsl_vector_calloc` oraz `gsl_matrix_calloc`.

Do wszystkich opisanych funkcji wektory i macierze przekazywane są nie przez wartość, ale przez wskaźnik. Wskaźnik do tablicy to adres pierwszego elementu tej tablicy w pamięci. To rozwiązanie stosuje się ze względu na szybkość operacji, gdyż nie wymaga ono kopiowania całych tablic w pamięci.

<sup>2</sup>Dokładny opis (po angielsku) można znaleźć pod adresem [http://www.gnu.org/software/gsl/manual/html\\_node/Vectors-and-Matrices.html](http://www.gnu.org/software/gsl/manual/html_node/Vectors-and-Matrices.html)

<sup>3</sup>Funkcja ta należy podsystemu BLAS (*ang. Basic Linear Algebra Subprograms*) złożonego z efektywnych niskopoziomowych<sup>4</sup> funkcji wykonujących podstawowe operacje z dziedziny algebry liniowej (np. mnożenie macierzy, wektorów, itp.). Funkcje macierzowe i wektorowe wyższego poziomu t.j. GSL pozwalają tylko na mnożenie elementów macierzy przez odpowiadające im elementy drugiej macierzy.

W zadaniu interesuje nas jedynie iloczyn macierzy i wektora  $RI$ , bez transpozycji macierzy  $R$  przed mnożeniem. Wobec tego wartości obu skalarów mają być równe 1, a pierwszy argument funkcji `int gsl_blas_dgemv` ma mieć wartość `CblasNoTrans`.

Skoro wynik mnożenia ma być umieszczony w wektorze  $E$ , a funkcja `int gsl_blas_dgemv` zawsze realizuje podany wyżej wzór, to należy mieć pewność, że przed operacją wektor  $E$  jest wypełniony zerami! Można w tym celu alokować wektor  $E$  za pomocą w.w. funkcji `gsl_vector_calloc`.

Aby poprawnie skompilować i zlinkować program należy dodać do projektu biblioteki `libgsl.so` i `libgslcblas.so` (jak pokazałem w instrukcji 0) lub wywołać kompilator `gcc` z opcjami `-lgsl -lgslcblas`.

### 3 Obliczanie obwodu trójfazowego w stanie ustalonym — 1 pkt

Na rysunku 2 pokazano uproszczony schemat systemu energetycznego średniego napięcia. Źródło (gwiazda z uziemionym punktem neutralnym) odpowiada stronie wtórnej transformatora mocy w rozdzielni WN/SN. Odbiornik typu trójkąt odpowiada uzwojeniom pierwotnym transformatora rozdzielczego SN/nn. Linię przedstawiono w postaci modelu typu T o parametrach skupionych. Impedancje  $Z_g$  to suma impedancji faz źródła oraz linii od strony źródła, a  $Z_l$  to impedancje linii od strony obciążenia.  $Z_d$  to impedancje doziemne linii (głównie pojemnościowe).  $Z_{A...C}$  to impedancje górnych uzwojeń transformatora rozdzielczego. Przyjęto założenie, że system pracuje w stanie ustalonym, a przebiegi napięć i prądów są sinusoidalne. W takim przypadku obwód ten można opisać za fazorów napięć i prądów. Fazor to po prostu liczba zespolona, której moduł odpowiada wartości skutecznej, a argument odpowiada fazie.

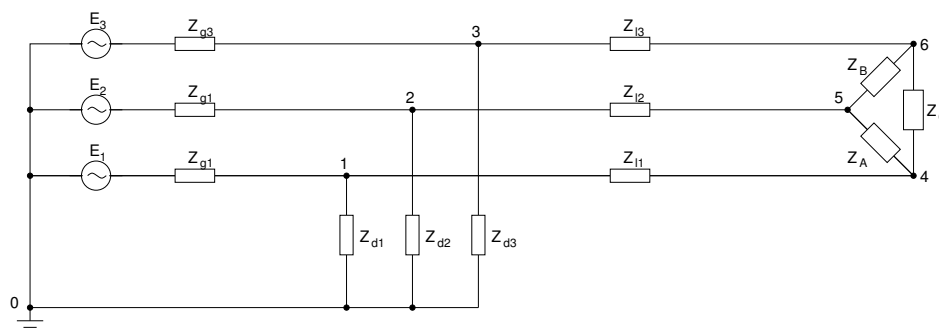
Korzystając z metody napięć węzłowych<sup>5</sup> wyprowadź równania na kartce (będą sprawdzone!) opisujące trójfazowy system energetyczny (rys. 2) w stanie sinusoidalnym ustalonym, a następnie zapisz te równania w formie macierzowej:

$$YV = J \quad (3)$$

i rozwiąż je w Matlabie metodą zwykłego odwracania macierzy admitancyjnej  $Y$ . Plik `lab02z03.m` zawiera wartości napięć źródeł i admitancji elementów systemu energetycznego, które będą potrzebne do zbudowania macierzy admitancyjnej  $Y$  i wektora prądów źródłowych  $J$ . Na rysunku 2 zaznaczono numery węzłów. Poszukiwane napięcia węzłowe  $V = [V_1 \ V_2 \ V_3 \ V_4 \ V_5 \ V_6]^T$  to napięcia skierowane od węzła 0 (ziemia) do odpowiednich węzłów 1...6.

Sprawdź jak zmienia się uwarunkowanie macierzy  $Y$  (zmienna `rc`) w stanie zwarcia doziemnego jednej fazy. Efekt ten można uzyskać ustawiając bardzo małą wartość jednej z impedancji doziemnych  $Z_d$ . Nie można podać zerowej impedancji  $Z_d$ , gdyż odpowiadająca jej admitancja  $Y_d$  byłaby nieskończona, dlatego najlepiej ustawić dużą wartość  $Y_d$  np.  $\approx 1e18$ .

W tym zadaniu macie do czynienia z macierzą i wektorami zawierającymi liczby zespolone. Wiele funkcji Matlabu działa zarówno na liczbach rzeczywistych jak i zespolonych. Należy pamiętać, że transpozycja macierzy zespolonej  $A$ , oznaczanej w Matlabie przez  $A'$  (apostrof), daje macierz transponowaną o elementach sprzężonych czyli  $(a_{i,j})' = (a_{j,i}^*)$ . Dla macierzy o elementach rzeczywistych będzie to zwykła transpozycja. W literaturze ta transpozycja sprzężona, w odróżnieniu od zwykłej transpozycji oznaczanej  $A^T$ , jest oznaczana przez  $A^H$ .



Rysunek 2: Schemat systemu energetycznego do zadania 3.

<sup>5</sup>Konstruowanie równań w tej metodzie opiera się na założeniu, że suma prądów w każdym węźle jest równa zero. Prądy  $I$  (w postaci fazorów) wyrażamy za pomocą iloczynów admitancji  $Y$  gałęzi i napięć węzłowych  $V$  (ew. źródłowych  $E$ )

#### 4 Wyznaczanie sygnału wyjściowego na podstawie znajomości odpowiedzi impulsowej linii z wykorzystaniem macierzy splotowej — 1 pkt

Rozważmy system komunikacji składający się z modemu nadawczego, linii i modemu odbiorczego. Znana jest odpowiedź impulsowa  $h(t)$  linii, która opisuje zniekształcający wpływ linii na transmitowany sygnał. Mając do dyspozycji zdyskretyzowane w ograniczonym czasie odpowiedź impulsową  $h_k$  oraz sygnał wejściowy  $u_n$ , próbki sygnału wyjściowego  $y_n$  można wyznaczyć jako splot:

$$y_n = u_n * h_n = \sum_{k=0}^L u_{n-k} h_k \quad (4)$$

Wyznaczenie całego wektora  $\mathbf{y} = [y_1 \ y_2 \ \dots \ y_{L+N}]^T$  na podstawie znajomości  $N$  próbek wektora  $\mathbf{u} = [u_1 \ u_2 \ \dots \ u_N]^T$  w „jednym kroku” można uzyskać poprzez utworzenie na podstawie równania (4) układu równań i zapisaniu go w postaci macierzowej:

$$\mathbf{y}_{(N+L-1 \times 1)} = \mathbf{H}_{(N+L-1 \times N)} \mathbf{u}_{(N \times 1)} \quad (5)$$

gdzie  $\mathbf{H}$  jest macierzą splotową o strukturze Toeplitza:

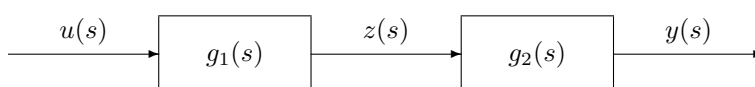
$$\mathbf{H} = \begin{bmatrix} h_0 & 0 & \dots & 0 \\ h_1 & h_0 & \ddots & \vdots \\ \vdots & h_1 & \ddots & 0 \\ h_{L-1} & \ddots & \ddots & h_0 \\ 0 & h_{L-1} & \ddots & h_1 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & h_{L-1} \end{bmatrix} \quad (6)$$

W Matlabie załaduj poleceniem `load` plik `lab02z04.mat` zawierający niezbędne dane czyli próbki sygnału wejściowego  $\mathbf{u}$  i spróbkowaną odpowiedź impulsową  $\mathbf{h}$ . Korzystając z funkcji `toeplitz` zbuduj macierz  $\mathbf{H}$ <sup>6</sup> oraz oblicz próbki sygnału wyjściowego  $\mathbf{y}$  według zależności (5). Wyznacz próbki wyjścia  $\mathbf{y}_a$  alternatywną metodą korzystając z funkcji `conv`. Wyrysuj na jednym wykresie sygnały wejściowy  $\mathbf{u}$  i wyjściowe  $\mathbf{y}$  (linią ciągłą) i  $\mathbf{y}_a$  (linią przerywaną).

#### 5 Opis obiektu dynamicznego w postaci macierzowego równania stanu — 1 pkt

Rozważmy układ pomiarowy złożony z szeregowo połączonych termopary i wzmacniacza operacyjnego (rys. 3). Obydwa elementy zamodelujemy jako obiekty dynamiczne inercyjne pierwszego rzędu o transmitancji operatorowej:

$$g_i(s) = \frac{k_i}{1 + sT_i} \quad (7)$$



Rysunek 3: Szeregowo połączenie termopary i wzmacniacza z zadania 5.

Wzmacniacz ma dużą impedancję wejściową, dlatego można przyjąć, że elementy się wzajemnie nie obciążają (wypadkowa transmitancja układu jest iloczynem transmitancji elementów). Wiedząc, że układy nie obciążają się, zbuduj układ dwóch równań operatorowych:

$$\begin{cases} z(s) = g_1(s)u(s) \\ y(s) = g_2(s)z(s) \end{cases} \quad (8)$$

<sup>6</sup>Przeczytaj help do funkcji `toeplitz` i pamiętaj o właściwym rozmiarze macierzy  $\mathbf{H}$  podanym w (5)

a następnie po podstawieniu (7) do (8) przejdź do układu równań różniczkowych w dziedzinie czasu, wykorzystując odwrotną transformację Laplace'a<sup>7</sup>. Ostatecznie otrzymane równania przekształć do postaci macierzowych równań stanu:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx + Du \end{cases} \quad (9)$$

lub po rozpisaniu dla obiektu o jednym wejściu, jednym wyjściu (SISO) i dwóch zmiennych stanu

$$\begin{bmatrix} \dot{x}_2 \\ \dot{x}_1 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot \\ \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} + \begin{bmatrix} \cdot \\ \cdot \end{bmatrix} u \quad (10)$$

$$y = \begin{bmatrix} \cdot & \cdot \end{bmatrix} \begin{bmatrix} x_2 \\ x_1 \end{bmatrix} + \begin{bmatrix} \cdot \end{bmatrix} u \quad (11)$$

Mając dane parametry obiektów  $k_1 = 0,001$ ,  $T_1 = 10$ ,  $k_2 = 1000$ ,  $T_2 = 0,001$  wyprowadź równania stanu opisujące ten układ. Sprawdź poprawność rozwiązania wykorzystując funkcję bode Matlaba. Załamania charakterystyki amplitudowo-częstotliwościowej powinny występować przy częstotliwościach odpowiadających stałym czasowym czujnika i wzmacniacza.

Pomimo, że do opisu dynamiki układu, wystarcza pierwsze z równań (9), to aby wykorzystać funkcję bode dla systemu opisanego równaniami stanu niezbędne jest podanie jako parametrów wszystkich macierzy  $A$ ,  $B$ ,  $C$ ,  $D$  o właściwych rozmiarach. Do wyznaczenia drugiego z równań (9) wystarczy informacja, że wyjście układu  $y = x_2$ .

Na koniec sprawdź jak zmieni się uwarunkowanie macierzy  $A$  po przyjęciu stałych czasowych  $T_1 = T_2 = 10$  lub na odwrót  $T_1 = T_2 = 0,001$ .

## Na następne zajęcia

Proszę się zapoznać z metodami rozwiązywania układów równań liniowych przez eliminację Gaussa-Jordana oraz eliminację Gaussa i podstawienie wstecz oraz z metodami iteracyjnymi. Jak ma wpływ wybór metody na błędy uzyskanego rozwiązania. Proszę przeczytać pomoc do funkcji realizujących dekompozycje LU, QR, Cholesky'ego (lu, qr, tttchol) aby wiedzieć jak ich używać. Zapoznać się z dokumentacją do biblioteki GSL dotyczącą rozwiązywania układów równań liniowych metodą faktoryzacji LU (i innymi funkcjami algebry liniowej), a przede wszystkim z przykładem na końcu dokumentacji [http://www.gnu.org/software/gsl/manual/html\\_node/Linear-Algebra.html](http://www.gnu.org/software/gsl/manual/html_node/Linear-Algebra.html). Dowiedzieć się jak rozwiązywane są układy równań w przypadku macierzy rzadkich.

---

<sup>7</sup>Przypominam, że  $\mathcal{L}^{-1}\{sx(s)\} = \frac{dx}{dt} = \dot{x}$