



Politechnika Białostocka
Wydział Elektryczny
Katedra Elektrotechniki Teoretycznej i Metrologii

Instrukcja
do pracowni specjalistycznej z przedmiotu

Informatyka 1

Kod przedmiotu: **ES1E2009**
(studia stacjonarne)

MATLAB - SKRYPTY I FUNKCJE

Numer ćwiczenia

INF08

Autor:
dr inż. Jarosław Forenc

Białystok 2017

Spis treści

1. Opis stanowiska	3
1.1. Stosowana aparatura	3
1.2. Oprogramowanie.....	3
2. Wiadomości teoretyczne.....	3
2.1. Skrypty	3
2.2. Komentarze i system pomocy.....	5
2.3. Wczytywanie danych do skryptu.....	6
2.4. Funkcje.....	8
2.5. Funkcje inline i funkcje anonimowe	10
3. Przebieg ćwiczenia.....	11
4. Literatura.....	12
5. Pytania kontrolne	13
6. Wymagania BHP.....	13

Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.

© Wydział Elektryczny, Politechnika Białostocka, 2017 (wersja 3.1)

Wszelkie prawa zastrzeżone. Żadna część tej publikacji nie może być kopiowana i odtwarzana w jakiegokolwiek formie i przy użyciu jakichkolwiek środków bez zgody posiadacza praw autorskich.

1. Opis stanowiska

1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows (XP/7/10).

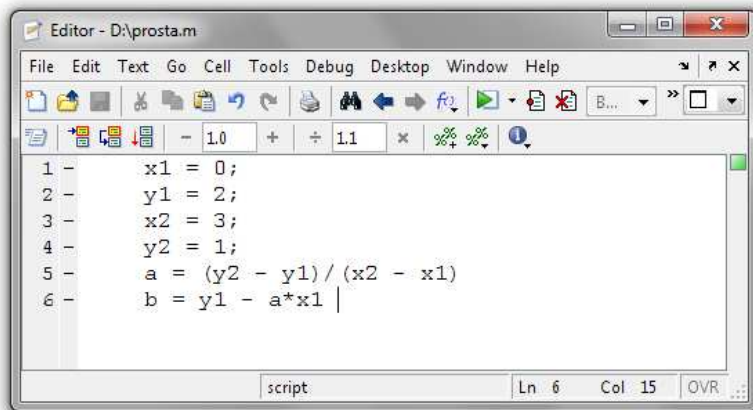
1.2. Oprogramowanie

Na komputerach zainstalowane jest środowisko Matlab R2007b (Version 7.5.0.342), classroom license.

2. Wiadomości teoretyczne

2.1. Skrypty

Skrypt jest to plik tekstowy o rozszerzeniu `.m` zawierający polecenia i instrukcje Matlaba. Skrypt można utworzyć w dowolnym edytorze zapisującym niesformatowane pliki tekstowe. Środowisko Matlab zawiera własny edytor m-plików (Rys. 1) wywołwany z menu **Desktop** → **Editor**.



Rys. 1. Edytor m-plików

Poniżej znajduje się przykładowa zawartość skryptu obliczającego i wyświetlającego współczynniki **a** i **b** równania prostej $y = ax + b$ przechodzącej przez dwa punkty o współrzędnych **(0, 2)** i **(3, 1)**.

```
x1 = 0;
y1 = 2;
x2 = 3;
y2 = 1;
a = (y2 - y1) / (x2 - x1)
b = y1 - a*x1
```

Skrypt uruchamia się podając jego nazwę (bez rozszerzenia) w wierszu poleceń Matlaba. Jeśli powyższy skrypt będzie zapisany pod nazwą **prosta.m**, to jego wykonanie wymaga wpisania polecenia:

```
>> prosta
a =
    -0.3333
b =
     2
```

Wykonanie skryptu jest równoważne z wpisywaniem i zatwierdzaniem kolejnych jego poleceń w wierszu poleceń (*Command Window*).

Matlab wykona skrypt jeśli będzie on znajdował się w bieżącym katalogu lub w katalogu udostępnionym poleceniem **path**.

path	wyświetla aktualną listę ścieżek (katalogów)
path (path, kat1)	dodaje do listy ścieżek katalog o podanej nazwie (kat1)

Bieżący katalog (*Current Directory*) jest to katalog, w którym zapisywane są pliki tworzone podczas pracy. Nazwa bieżącego katalogu wyświetlana jest w górnej części głównego okna programu Matlab (Rys. 2).



Rys. 2. Okno bieżącego katalogu

Informację o bieżącym katalogu można wyświetlić także poleceniem **pwd**. Katalog można zmienić wywołując polecenie **cd**.

Przy nazywaniu skryptów obowiązują takie same zasady jak w przypadku innych zmiennych w Matlabie. Nazwa skryptu nie powinna być taka sama jak nazwa zmiennej w przestrzeni roboczej lub zmiennej występującej w skrypcie. Jeśli powyższa sytuacja wystąpi, to wprowadzenie w wierszu poleceń takiej nazwy spowoduje wyświetlenie wartości zmiennej, a nie wykonanie skryptu.

2.2. Komentarze i system pomocy

W skryptach można wstawiać komentarze. Komentarz rozpoczyna się znakiem procentu (%). Matlab zignoruje wszystko co znajdzie się między znakiem % a końcem wiersza. Jeśli pierwsze linie skryptu zaczynają się od znaków %, to stanowią pomoc wyświetlaną na ekranie po wywołaniu polecenia:

```
>> help skrypt
```

gdzie słowo **skrypt** jest nazwą skryptu (bez rozszerzenia **.m**).

Przykładowa zawartość skryptu **prosta.m** zawierającego pomoc i komentarze:

```
% PROSTA - równanie prostej y = ax + b
% Skrypt obliczający współczynniki a i b równania
% prostej y = ax + b przechodzącej przez dwa punkty
% o współrzędnych (x1,y1) i (x2,y2)
x1 = 0;
y1 = 2;
x2 = 3;
y2 = 1;
% Obliczenie wartości a i b
a = (y2 - y1)/(x2 - x1);
b = y1 - a*x1;
% Wyświetlenie wartości a i b
txt = sprintf('a = %g', a);
disp(txt);
txt = sprintf('b = %g', b);
disp(txt);
```

Polecenie:

```
>> help prosta
```

spowoduje wyświetlenie informacji o skrypcie:

```
PROSTA - równanie prostej y = ax + b
Skrypt obliczający współczynniki a i b równania
prostej y = ax + b przechodzącej przez dwa punkty
o współrzędnych (x1,y1) i (x2,y2)
```

Natomiast wpisanie:

```
>> prosta
```

spowoduje wykonanie skryptu.

```
a = -0.333333
b = 2
```

W powyższym skrypcie do sformatowania wyświetlanych wyników zastosowano funkcję **sprintf**, która ma identyczną składnię jak funkcja **printf** w języku C. Funkcja **sprintf** zwraca łańcuch znaków. Łańcuch ten wyświetlany jest poleceniem **disp**. Wywołanie funkcji **sprintf** można wstawić bezpośrednio do polecenia **disp**:

```
% Wyświetlenie wartości a i b
disp(sprintf('a = %g', a));
disp(sprintf('b = %g', b));
```

Skrypty nie pobierają żadnych argumentów wejściowych, ani nie zwracają żadnych argumentów wyjściowych. Operują tylko na zmiennych dostępnych w przestrzeni roboczej Matlab. Wyniki wykonania skryptu (np. utworzone zmienne) pozostają w przestrzeni roboczej.

2.3. Wczytywanie danych do skryptu

Podczas wykonywania skryptu można wczytywać do niego dane z klawiatury wykorzystując funkcję **input**:

<code>x=input(napis)</code>	wyświetla napis , a następnie czeka na wprowadzenie liczby, która przypisywana jest zmiennej x
<code>x=input(napis, 's')</code>	działa j.w., ale służy do wczytania łańcucha znaków

Przykładowy skrypt wykorzystujący wczytywanie danych:

```
% PROSTA - równanie prostej y = ax + b
% Skrypt obliczający współczynniki a i b równania
% prostej y = ax + b przechodzącej przez dwa punkty
% o współrzędnych (x1,y1) i (x2,y2)
disp('Współrzędne punktu nr 1:');
x1 = input('x1: ');
y1 = input('y1: ');
disp('Współrzędne punktu nr 2:');
x2 = input('x2: ');
y2 = input('y2: ');
% Obliczenie wartości a i b
a = (y2 - y1)/(x2 - x1);
b = y1 - a*x1;
% Wyświetlenie wartości a i b
disp(sprintf('a = %g', a));
disp(sprintf('b = %g', b));
```

Wykonanie powyższego skryptu:

```
>> prosta
Współrzędne punktu nr 1:
x1: 0
y1: 2
Współrzędne punktu nr 2:
x2: 3
y2: 1
a = -0.333333
b = 2
```

Wykonywanie skryptu może być zatrzymane przy użyciu polecenia **pause**.

<code>pause</code>	zatrzymuje wykonywanie skryptu do naciśnięcia przez użytkownika dowolnego klawisza
<code>pause(x)</code>	zatrzymuje wykonywanie skryptu na x sekund

2.4. Funkcje

Funkcja różni się tym od skryptu, że można do niej przekazywać argumenty podczas wywołania oraz może ona zwracać wartości.

Funkcje definiowane przez użytkownika przechowywane są w m-plikach (plik tekstowy o rozszerzeniu **.m**). Pierwszy wiersz funkcji musi zawierać **definicję funkcji** składającą się z:

- słowa kluczowego **function**;
- listy wartości zwracanych przez funkcję;
- nazwy funkcji (musi być taka sama jak nazwa m-pliku, w którym się znajduje);
- listy parametrów funkcji.

Definicja funkcji ma następującą postać:

```
function [wart1, wart2, ...]=nazwa(par1,par2,...)
% opis funkcji - jako komentarz
instrukcje
```

Wśród instrukcji funkcji muszą znajdować się przypisania o postaci:

```
wart1 = ...;
wart2 = ...;
```

Przykładowa funkcja obliczająca i zwracająca wartość współczynników **a** i **b** równania prostej **y = ax + b** przechodzącej przez dwa punkty o współrzędnych **(x1,y1)** i **(x2,y2)**:

```
function [a,b] = prosta(x1,y1,x2,y2)
% PROSTA - równanie prostej y = ax + b
% Funkcja obliczająca współczynniki a i b równania
% prostej y = ax + b przechodzącej przez dwa punkty
% o współrzędnych (x1,y1) i (x2,y2)
a = (y2 - y1)/(x2 - x1);
b = y1 - a*x1;
```

W wywołaniu funkcji należy podać wartości jej argumentów oraz nazwy zmiennych pod które zostaną podstawione zwrócone wartości:

```
>> [wa,wb] = prosta(0,2,3,1)
wa =
    -0.3333
wb =
     2
```

Jeśli nie zostaną podane nazwy zmiennych wynikowych, to wyświetlona zostanie tylko jedna zwracana wartość:

```
>> prosta(0,2,3,1)
ans =
    -0.3333
```

W przypadku, gdy funkcja zwraca tylko jedną wartość, to nie trzeba stosować nawiasów kwadratowych w pierwszym wierszu jej definicji. Poniższa funkcja oblicza pole koła o promieniu r.

```
function p = pole(r)
% POLE - pole koła o promieniu r
p = pi*r.^2;
```

Przykładowe wywołanie funkcji **pole**:

```
>> pole(2.5)
ans =
    19.6350
```

Argumentami funkcji mogą być także zmienne, wyrażenia arytmetyczne lub wywołania innych funkcji.

Podczas wykonywania funkcji nie są wyświetlane wyniki działania poleceń, które kończą się średnikiem. Zmienne oraz argumenty wejściowe występujące w funkcjach są lokalne w ciele funkcji i nie wchodzą w skład przestrzeni roboczej Matlaba. Z poziomu funkcji nie ma dostępu do zmiennych występujących w przestrzeni roboczej Matlaba.

Pojawienie się w funkcji instrukcji **return** powoduje natychmiastowe przerwanie jej wykonywania i powrót do miejsca wywołania.

2.5. Funkcje inline i funkcje anonimowe

Funkcja **inline** jest to funkcja, którą można zdefiniować bezpośrednio w wierszu polecenia. Składnia takiej funkcji jest następująca:

```
nazwa = inline(wyrażenie)
```

Nazwa określa nazwę funkcji, zaś **wyrażenie** jest łańcuchem znaków opisującym funkcję. Tak stworzona funkcja może być wywoływana dla różnych argumentów. Poniżej przedstawiono definicję i wywołanie jednoargumentowej funkcji **fun1()**:

```
>> fun1 = inline('sin(2*x-pi/2)')
fun1 =
    Inline function:
    fun1(x) = sin(2*x-pi/2)
>> fun1(4)
ans =
    0.1455
```

oraz definicję i wywołanie dwuargumentowej funkcji **fun2()**:

```
>> fun2 = inline('x.^2+y.^2')
fun2 =
    Inline function:
    fun2(x,y) = x.^2+y.^2
>> fun2(5,6)
ans =
    61
```

Funkcja **anonimowa** jest to funkcja jednej lub kilku zmiennych, tworzona w wierszu polecenia, najczęściej w celu wyznaczenia wartości funkcji dla różnych argumentów. Funkcja taka nie ma nazwy, a dostęp do niej odbywa się poprzez zmienną tworzoną przez użytkownika i przechowującą uchwyt funkcji. Składnia funkcji anonimowej jest następująca:

```
nazwa = @(zmienne)wyrażenie
```

Nazwa określa nazwę uchwytu funkcji, **zmienne** jest to lista zmiennych wejściowych, zaś **wyrażenie** określa wzór funkcji. Poniżej przedstawiono definicję i wywołanie funkcji jednoargumentowej:

```
>> fun3 = @(x) sin(2*x-pi/2)
fun3 =
    @(x) sin(2*x-pi/2)
>> fun3(4)
ans =
    0.1455
```

oraz definicję i wywołanie funkcji dwuargumentowej:

```
>> fun4 = @(x,y) x.^2+y.^2
fun4 =
    @(x,y) x.^2+y.^2
>> fun4(5,6)
ans =
    61
```

3. Przebieg ćwiczenia

Na pracowni specjalistycznej należy wykonać wybrane zadania wskazane przez prowadzącego zajęcia. W różnych grupach mogą być wykonywane różne zadania.

- Przez opornik o rezystancji **R** płynie stały prąd **I**. Napisz skrypt, który obliczy i wyświetli napięcie na oporniku **U** oraz wydzielającą się w nim moc **P**. Wartości rezystancji i prądu wczytaj z klawiatury w skrypcie. Dodaj pomoc do skryptu. Wywołanie skryptu i wyświetlenie wyników powinno mieć następującą postać:

```
Podaj R [Om]:      470
Podaj I [A]:       0.25
-----
Napiecie U [V]:   117.5
Moc P [W]:        29.375
```

- Przez opornik o rezystancji **R** płynie stały prąd **I**. Napisz funkcję, która obliczy i zwróci napięcie na oporniku **U** oraz wydzielającą się w nim moc **P**. Wartości rezystancji i prądu powinny być argumentami funkcji. Dodaj pomoc do funkcji. Wywołaj napisaną funkcję.

- Napisz funkcję obliczającą częstotliwość rezonansową **f_r** układu o rezystancji **R**, indukcyjności **L** i pojemności **C**. Dodaj pomoc do funkcji.

$$f_r = \frac{1}{2\pi\sqrt{LC}} \sqrt{1 - \frac{L}{R^2C}} \quad (1)$$

Sprawdź poprawność funkcji dla danych:

$$R = 100 \, \Omega, \quad L = 0,05 \, H, \quad C = 5 \, mF \quad \rightarrow \quad f_r = 10,0608 \, Hz$$

- Admitancja zastępcza obwodu elektrycznego określona jest wzorem (2). Napisz skrypt, który wyświetli wykres konduktancji **G** i susceptancji **B** w funkcji pulsacji ω . Pulsacja powinna zmieniać się w zakresie od **100** do **500 rad/s**. Podziel okno graficzne na dwie części (umieść wykresy obok siebie). Opisz osie, dodaj tytuły wykresów, włącz wyświetlanie pomocniczej siatki.

$$\underline{Y} = G + jB = \frac{R}{R^2 + X_C^2} + j \left(\frac{X_C}{R^2 + X_C^2} - \frac{1}{X_L} \right) \quad (2)$$

Przyjmij następujące parametry obwodu:

$$R = 15 \, \Omega, \quad L = 0,125 \, H, \quad C = 0,254 \, mF$$

- Napisz skrypt wyświetlający w jednym, podzielonym, oknie graficznym cztery wykresy funkcji (dwa w wierszu i dwa w kolumnie) $z = e^{\cos(2x)} \cdot e^{\cos(2y)}$ dla $x, y \in \langle -3, 3 \rangle$ z krokiem **0,2**. Do narysowania wykresów zastosuj funkcje: **mesh(z)**, **mesh(x,y,z)**, **meshc(x,y,z)**, **meshz(x,y,z)**. Dodaj opisy do wykresów.

4. Literatura

- Mrozek B., Mrozek Z.: MATLAB i Simulink. Poradnik użytkownika. Wydanie IV. Helion, Gliwice, 2018.
- Pratap R.: MATLAB dla naukowców i inżynierów. Wydanie 2. Wydawnictwo Naukowe PWN, Warszawa, 2015.

- [3] Banasiak K.: Algorytmizacja i programowanie w Matlabie. Wydawnictwo BTC, Legionowo, 2017.
- [4] Stachurski M. Treichel W.: Matlab dla studentów. Ćwiczenia, zadania, rozwiązania. Witkom, Warszawa, 2009.
- [5] Brzóska J., Dorobczyński L.: Matlab: środowisko obliczeń naukowo-technicznych. „Mikom”, Wydawnictwo Naukowe PWN, Warszawa, 2008.
- [6] Kamińska A., Pańczyk B.: Ćwiczenia z Matlab. Przykłady i zadania. Wydawnictwo MIKOM, Warszawa, 2002.
- [7] Sobierajski M., Łabuzek M.: Programowanie w Matlabie dla elektryków. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 2005.
- [8] Dyka E., Markiewicz P., Sikora R.: Modelowanie w elektrotechnice z wykorzystaniem środowiska MATLAB. Wydawnictwa Politechniki Łódzkiej, Łódź, 2006.
- [9] Sradomski W.: Matlab. Praktyczny podręcznik modelowania. Helion, Gliwice, 2015.
- [10] Czajka M.: MATLAB. Ćwiczenia. Helion, Gliwice, 2005.

5. Pytania kontrolne

1. W jaki sposób definiuje się pomoc do skryptów w Matlabie?
2. Jaka jest struktura definicji funkcji w Matlabie?
3. Jakie są różnice pomiędzy skryptami a funkcjami w Matlabie?
4. W jaki sposób definiuje się funkcje inline i funkcje anonimowe?

6. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciwpożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.
- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.