

Spis treści

PRZEDMOWA	5
1. PODSTAWOWE WIADOMOŚCI O ŚRODOWISKU MATLAB	6
1.1. Wiadomości wstępne	6
1.2. Pomocnicze polecenia okna Command Window	7
1.3. Skrypty i funkcje użytkownika	11
1.4. Operacje na macierzach i tablicach wartości	14
1.5. Funkcje wspomagające generowanie macierzy	27
1.6. Liczby zespolone	28
1.7. Funkcje matematyczne	29
1.8. Instrukcje języka skryptowego	32
1.9. Instrukcje graficzne	34
2. METODY NUMERYCZNE	38
2.1. Interpolacja	38
2.2. Aproksymacja	42
2.3. Miejsca zerowe wielomianów	44
2.4. Miejsca zerowe funkcji	46
2.5. Algebra liniowa	47
2.5.1. Układy równań liniowych	47
2.5.2. Wartości własne i wektory własne macierzy	49
2.6. Całkowanie	51
2.7. Różniczkowanie	53
3. PRZYKŁADOWE OBLICZENIA ELEKTROENERGETYCZNE	58
3.1. Wyznaczanie przebiegu napięcia odkształconego	58
3.2. Obliczanie rozptywu prądów w sieci zamkniętej	59
3.3. Wyznaczenie macierzy impedancyjnej zwarciowej	63
3.4. Obliczanie prądów zwarciowych	66
3.5. Obliczanie rozkładu potencjału oraz natężenia pola elektrycznego dla uziomu półkulistego	71
4. WYKORZYSTANIE ŚRODOWISKA MATLAB DO MODELOWANIA WYBRANYCH ZAGADNIEŃ Z DZIEDZINY ELEKTROTECHNIKI	74
4.1. Modele matematyczne podstawowych elementów elektrycznych	74
4.2. Modelowanie obiektów złożonych	75
4.3. Szeregowy obwód RL zasilany napięciem stałym lub sinusoidalnie zmiennym	76
4.4. Szeregowy obwód RC zasilany napięciem stałym lub sinusoidalnie zmiennym	79
4.5. Szeregowy obwód RLC zasilany napięciem stałym lub sinusoidalnie zmiennym	81
4.6. Szeregowy obwód RL zasilany z prostownika jednopółkowego	85
4.7. Model silnika obcowzbudnego szeregowego prądu stałego	87
5. MODELOWANIE WYBRANYCH ZAGADNIEŃ ZA POMOCĄ ŚRODOWISKA GRAFICZNEGO SIMULINK	91
5.1. Wprowadzenie do modelowania w SIMULINK-u	91
5.2. Obliczanie podstawowych parametrów schematu zastępczego sieci elektroenergetycznych	98

5.2.1. Parametry schematu zastępczego trójfazowego układu zasilającego	98
5.2.2. Parametry schematu zastępczego linii przesyłowej	99
5.2.3. Parametry schematu zastępczego transformatora trójfazowego	101
5.2.4. Parametry schematu zastępczego silnika indukcyjnego	104
5.3. Pomiary wielkości elektrycznych	108
5.3.1. Wiadomości teoretyczne	108
5.3.2. Zastosowanie programu Simulink do pomiarów wielkości elektrycznych	112
5.4. Pomiary mocy w układach trójfazowych	116
5.4.1. Wiadomości teoretyczne	116
5.4.2. Pomiary mocy z wykorzystaniem SIMULINK-a	120
5.5. Rozruch silnika asynchronicznego	126
5.5.1. Wiadomości teoretyczne	126
5.5.2. Modelowanie rozruchu silnika asynchronicznego klatkowego	127
5.6. Kompensacja mocy biernej	135
5.6.1. Wiadomości teoretyczne	135
5.6.2. Model układu kompensacji mocy biernej	138
5.7. Prostownik mostkowy 6D	141
5.7.1. Wiadomości teoretyczne	141
5.7.2. Model prostownika 6D	146
5.8. Symulacje stanów pracy transformatora dwuuzwojeniowego trójfazowego	152
5.8.1. Obliczenia parametrów schematu zastępczego	153
5.8.2. Symulacja biegu jałowego transformatora	157
5.8.3. Symulacja próby zwarcia transformatora	158
5.8.4. Symulacja stanu pracy przy obciążeniu znamionowym	159
6. OBLICZANIE ROZPŁYWÓW MOCY W SIECIACH ELEKTRO-ENERGETYCZNYCH ZA POMOCĄ PAKIETU MATPOWER	162
6.1. Teoretyczne podstawy obliczania rozplywu mocy w sieciach elektroenergetycznych	162
6.2. Opis pakietu MATPOWER	164
6.2.1. Format plików zawierające dane	164
6.2.2. Obliczenia rozplywu mocy	167
6.2.3. Przykładowe obliczenie rozplywu mocy dla testowej sieci 14-węzłowej IEEE	168
7. LITERATURA	173
8. DODATEK: BIBLIOTEKA „POWER SYSTEM BLOCKSET”	175
8.1. Wiadomości podstawowe	175
8.2. Opis wybranych bloków biblioteki „Power System Blockset”	176

PRZEDMOWA

Skrypt jest przeznaczony dla studentów wydziałów elektrycznych wyższych szkół technicznych rozpoczynających pracę z programem MATLAB.

W skrypcie zaprezentowane zostały rozwiązania podstawowych zagadnień z dziedziny metod numerycznych, elektrotechniki i elektroenergetyki uzyskane za pomocą metod i narzędzi dostępnych w programie MATLAB i nakładki SIMULINK *Version 6.0.0.88 Release 12*. Skrypt obejmuje zagadnienia, które zawiera program przedmiotów *Metody numeryczne* oraz *Modelowanie i symulacja w elektrotechnice*, prowadzonych na Wydziale Elektrotechniki, Elektroniki, Informatyki i Automatyki Politechniki Łódzkiej.

W pierwszej części opisane zostały zasady programowania oraz przedstawiono wybrane funkcje biblioteczne MATLAB-a. Druga część zawiera opis metod stosowanych przy przybliżaniu funkcji, podczas rozwiązywania układów równań liniowych, znajdowania zer funkcji i wielomianów oraz wyznaczania wartości całek oznaczonych i rozwiązywania równań różniczkowych. Trzecia część przedstawia wykorzystanie opisanych wyżej metod do modelowania wybranych zagadnień z dziedziny elektrotechniki. W dalszych częściach skryptu zaprezentowane zostało modelowanie obiektowe wybranych zagadnień z dziedziny elektroenergetyki za pomocą nakładki SIMULINK, umożliwiającej symulację stanów dynamicznych w układach sterowania. Ostatnia część zawiera opis pakietu MATPOWER służącego do obliczania rozpiętów mocy w sieciach elektroenergetycznych.

Podczas pisania skryptu wykorzystano doświadczenia Autorów zdobyte w trakcie prowadzenia zajęć dydaktycznych z przedmiotów „Modelowanie w elektrotechnice” oraz „Metody numeryczne” dla studentów Wydziału Elektrotechniki, Elektroniki, Informatyki i Automatyki na kierunku Elektrotechnika.

Autorzy wyrażają podziękowania Panu dr hab. inż. Mirosławowi Parolowi za cenne i wnikliwe uwagi, które przyczyniły się do podniesienia wartości merytorycznej skryptu.

Autorzy składają również podziękowania Panom dr inż. Andrzejowi Kanickiemu oraz dr inż. Józefowi Wiśniewskiemu za pomoc udzieloną podczas pisania niniejszego skryptu.

Ewa Dyka, Przemysław Markiewicz, Roman Sikora

Łódź, marzec 2006 r.

1. PODSTAWOWE WIADOMOŚCI O ŚRODOWISKU MATLAB

1.1. Wiadomości wstępne

Pakiet MATLAB jest szeroko stosowanym narzędziem wspomagającym rozwiązywanie wszelkiego rodzaju problemów inżynierskich. Dostępne są wersje przystosowane do pracy w środowisku *Windows* na komputerach klasy PC jak i wersje do pracy w środowisku *Unix* na stacjach roboczych.

Pakiet składa się z interpretera języka symbolicznego, pakietu standardowych bibliotek, pakietu bibliotek dodatkowych służących do rozwiązywania szczegółowych problemów z wybranych dziedzin nauki (*Toolbox*) oraz nakładki SIMULINK wraz z własnymi bibliotekami (tzw. „*blockset*”).

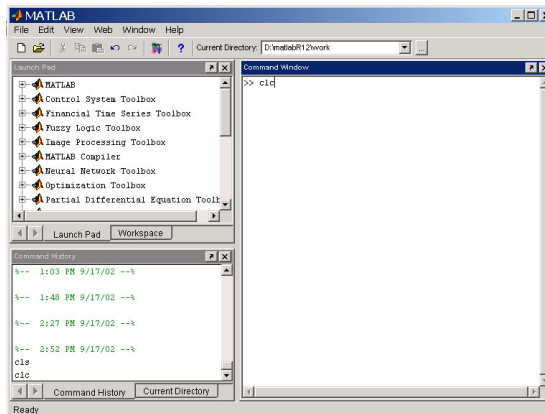
Program MATLAB pozwala na pracę w dwóch trybach:

- interaktywnym
- skryptowym

Tryb interaktywny polega na kolejnym pisaniu i wykonywaniu na bieżąco poszczególnych komend w oknie poleceń (*Command Window*). Jest to wygodny sposób w przypadku bardzo prostych zadań jak np. wykonywanie obliczeń arytmetycznych lub znajdowanie wartości funkcji elementarnych.

Tryb skryptowy jest zasadniczym trybem pracy w środowisku MATLAB. Polega on na kolejnym wykonywaniu, linijka po linijce, poszczególnych komend przez wbudowany interpreter języka programowania. Komendy te umieszcza się w pliku MATLAB-a typu *M-File*. Pliki tego rodzaju posiadają rozszerzenie *m* (*nazwa_pliku.m*).

Bezpośrednio po uruchomieniu programu pojawia się okno główne i można rozpocząć pracę w trybie interaktywnym.



Rys. 1-1. Widok okna głównego MATLAB-a

W lewej części okna głównego widoczne są dwa okna pomocnicze wyświetlające odpowiednio:

Launch Pad	<i>załadowane biblioteki</i>
Workspace	<i>zestaw przechowywanych w pamięci zmiennych</i>
Command history	<i>listę wcześniej wykonanych komend</i>
Current directory	<i>zawartość bieżącego katalogu</i>

W prawej części okna głównego znajduje się okno poleceń (**Command Window**), w którym wykonywane są polecenia oraz wyświetlane są wyniki obliczeń i komunikaty o błędach.

1.2. Pomocnicze polecenia okna Command Window

W rozdziale przedstawione zostaną wybrane komendy usprawniające pracę w oknie poleceń MATLAB-a (**Command Window**).

cd

Polecenie **cd** z parametrami służy do zmiany bieżącego katalogu, natomiast bez parametrów do wyświetlania aktualnego katalogu.

Składnia polecenia **cd** ma postać:

```
cd ścieżka  
cd
```

Przykład:

- zmiana katalogu
» cd c:/test/katalog1
- przejsie do nadrzednego katalogu
» cd..
- wyświetlenie bieżącego katalogu
» cd

clear

Polecenie **clear** powoduje usunięcie z obszaru roboczego (**Workspace**) znajdujących się tam zmiennych. Polecenie **clear** może posiadać następujące składnie:

```
clear nazwa_zmiennej
```

```
clear functions
clear all
```

Polecenie **clear all** powoduje usunięcie wszystkich zmiennych, funkcji oraz łączy ze zbiorami MEX z obszaru roboczego.

clc

Polecenie **clc** czyści okno poleceń MATLAB-a (*Command Window*).

dir

Polecenie **dir** wyświetla zawartość bieżącego katalogu. Jest to polecenie analogiczne do polecenia wykorzystywanego w DOS-ie.

disp

Polecenie **disp** służy do wyświetlania określonej zmiennej w oknie poleceń (*Command Window*).

Przykład:

```
>> A=[1 2 3;4 5 6]
```

```
A =
```

```
     1     2     3
     4     5     6
```

```
>> disp(A)
```

```
     1     2     3
     4     5     6
```

format

Polecenie **format** umożliwia określenie formatu liczb wyświetlanych na ekranie. W MATLAB-ie domyślnym formatem jest *format short*, który pozwala na wyświetlenie czterech cyfr po przecinku. Wyświetlenie czternastu cyfr po przecinku zapewnia *format long*.

Przykład:

```
>> 1/7
```

```
ans =
```

```
    0.1429
```

```
>> format long
```

```
>> 1/7
```

```
ans =  
0.14285714285714
```

help

Polecenie **help** użyte bez parametrów wyświetla listę katalogów MATLAB-a, zdefiniowanych w ścieżkach przeszukiwań, natomiast polecenie **help** z parametrem powoduje wyświetlenie pomocy dotyczącej wskazanego polecenia.

- Polecenie **help** bez parametru:

Przykład:

```
» help  
  
HELP topics:  
  
MATLAB\general - General purpose commands.  
MATLAB\ops - Operators and special characters.  
MATLAB\lang - Programming language constructs.  
MATLAB\elmat - Elementary matrices and matrix manipulation.  
MATLAB\elfun - Elementary math functions.  
  
For more help on directory/topic, type "help topic".
```

- Polecenie **help** z parametrem:

```
Help nazwa_polecenia
```

Przykład:

```
» help sin  
  
SIN Sine.  
SIN(X) is the sine of the elements of X.
```

home

Polecenie **home** powoduje wyczyszczenie okna poleceń MATLAB-a oraz ustawienie kursora w jego lewym górnym rogu.

matlabpath

Polecenie **matlabpath** wyświetla listę ścieżek przeszukiwań. Polecenie to nie posiada argumentów wejściowych.

Przykład:

```
» matlabpath

MATLABPATH

D:\MATLAB\sn
D:\MATLAB\toolbox\MATLAB\general
D:\MATLAB\toolbox\MATLAB\ops
```

path

Polecenie **path** służy do wyświetlania i dodawania ścieżek przeszukiwań. Użycie polecenie **path** bez parametrów ma działanie analogiczne jak polecenie **matlabpath**.

W przypadku, gdy chcemy dodać ścieżkę przeszukiwań należy zastosować składnię:

```
path(path, 'ścieżka_do_katalogu')
```

pwd

Polecenie **pwd** wyświetla zawartość bieżącego katalogu; jest ono analogiczne do polecenia **cd** bez parametrów.

Przykład:

```
» pwd

ans =
D:\MATLAB\bin
```

ver

Polecenie **ver** wyświetla informacje o wykorzystywanej wersji MATLAB-a i zainstalowanych bibliotekach dodatkowych.

Przykład:

```
>> ver
To get started, select "MATLAB Help" from the Help menu.
MATLAB Version 6.0.0.88 (R12) on PCWIN
MATLAB License Number: 191257
-----
MATLAB Toolbox           Version 6.0 (R12) 06-Oct-2000
Simulink                 Version 4.0 (R12) 16-Jun-2000
MATLAB Compiler          Version 2.1 (R12) 26-Jul-2000
Control System Toolbox   Version 5.0 (R12) 01-Sep-2000
DSP Blockset             Version 4.0 (R12) 01-Sep-2000
```

who

Polecenie **who** wyświetla listę zmiennych przechowywanych w obszarze roboczym (*Workspace*).

Przykład:

```
» who

Your variables are:

a           b
```

whos

Działanie polecenia **whos** jest podobne do polecenia **who**, ale wyświetla rozszerzone informacje o przechowywanych zmiennych.

Przykład:

```
» whos
  Name      Size      Bytes  Class
  a         2x3         48  double array
  b         2x3         48  double array

Grand total is 33 elements using 138 bytes
```

1.3. Skrypty i funkcje użytkownika

Skrypty

Skrypt jest plikiem typu *M-File* (o rozszerzeniu *m*), w którym umieszcza się program obliczeniowy napisany zgodnie z regułami obowiązującymi w MATLAB-ie.

Obliczenia można również przeprowadzać w trybie interaktywnym, jednakże jest on mało efektywny i dlatego powinien być wykorzystywany jedynie do prostych działań.

Jeżeli skrypt zapisany został w katalogu roboczym MATLAB-a (katalog *Work*), wówczas wykonanie obliczeń w nim zawartych (bez potrzeby jego otwierania) następuje po napisaniu w oknie poleceń (*Command Window*) nazwy skryptu (bez rozszerzenia) i wciśnięciu klawisza *ENTER*. Otwarty skrypt można również uruchomić korzystając z poleceń paska narzędzi edytora skryptów (*Editor - Debug - Run*).

W przypadku umieszczenia skryptu w innym katalogu niż roboczy, należy podać ścieżkę do tego katalogu korzystając z polecenia *path*

(np. `path(path,'G:\katalog')`), lub po jej zmapowaniu ustawić ją w oknie **Command Window (File - Set Path...)**.

Przykład prostego skryptu zostanie przedstawiony poniżej.

Przykład:

Rysowanie funkcji *sinus* w przedziale $\langle 1; 2 \rangle$ z krokiem $0,01$.

```
x=1:0.01:2;  
y=sin(x);  
plot(x,y)
```

Należy zwrócić uwagę na zastosowanie znaku średnika. Znak średnika umieszczony na końcu wiersza powoduje, że w oknie poleceń (**Command Window**) nie są wyświetlane zarówno pośrednie obliczenia jak i wynik końcowy. Aby wyświetlić określoną zmienną lub jedną z jej wartości należy skorzystać z polecenia **disp**.

Komentarze

Znak „%” na początku wiersza oznacza, że wiersz ten jest komentarzem lub, że polecenia w nim zawarte mają zostać pominięte podczas wykonywania programu.

Komentarz zamieszczony na początku skryptu (do momentu pojawienia się pierwszego pustego wiersza lub wiersza bez znaku komentarza) traktowany jest jako tekst pomocy danego skryptu.

Przykład:

- Kod skryptu *plik.m*:

```
%komentarz  
%komentarz  
  
%x=2  
y=5
```

- Help skryptu *plik.m*:

```
>> help plik  
komentarz  
komentarz
```

- Wykonanie skryptu *plik.m*:

```
>> plik
y =
    5
```

Funkcje

W MATLAB–ie istnieje możliwość zadeklarowania własnej funkcji. Definicję własnej funkcji umieszcza się w skrypcie, którego nazwa powinna być taka sama jak nazwa funkcji.

Składnia polecenia jest następująca:

```
function zmienna = nazwa_funkcji(argument_1,..., argument_n)
instrukcje
```

Przykład:

- Deklaracja funkcji obliczającej sumę trzech liczb (skrypt o nazwie *suma.m*):

```
function y=suma(a,b,c)
y=a+b+c;
```

- Kod skryptu o nazwie *dodawanie.m* wykorzystującego deklarowaną funkcję:

```
a=2;
b=3;
c=4;
wynik=suma(a,b,c)
```

- Wykonanie skryptu:

```
>>dodawanie
wynik =
    9
```

Zmienne deklarowane w funkcji są zmiennymi lokalnymi, czyli po zakończeniu działania funkcji usuwane są z pamięci, co powoduje, że nie można się już do nich odwołać. Zadeklarowanie zmiennych jako globalnych spowoduje, że po zakończeniu działania funkcji zmienne te będą ciągle dostępne.

Zmienne deklarowane są jako globalne przy pomocy instrukcji **global**:

```
global zmienna_1 zmienna_2 ... zmienna_N
```

Należy zadeklarować wykorzystywane w funkcji zmienne globalne również w skrypcie.

input

Polecenie **input** wykorzystywane jest do wprowadzania z klawiatury wartości zmiennych.

Składnia polecenia **input** jest następująca:

```
zmienna = input('tekst_zachęty')
```

Przykład:

- Obliczanie sumy dwóch liczb a i b wprowadzanych z klawiatury (skrypt o nazwie *plik.m*):

```
a=input('Podaj wartość liczby a = ');
b=input('Podaj wartość liczby b = ');

wynik=a+b
```

- Wykonanie skryptu *plik.m* następuje po podaniu wartości a i b :

```
>> plik
Podaj wartość liczby a = 2
Podaj wartość liczby b = 5

wynik =
      7
```

1.4. Operacje na macierzach i tablicach wartości

W MATLAB–ie podstawowym typem danych są macierze. Wielkości skalarne można traktować jako macierze o wymiarach 1×1 , a po umieszczeniu ich w tablicach wartości można wykonywać na nich dowolne operacje matematyczne.

Zastosowanie rachunku macierzowego pozwala na prostsze rozwiązywanie różnorodnych problemów matematycznych, np. rozwiązywanie układu n równań.

Operatory matematyczne

Postać ogólną macierzy można zapisać w następujący sposób:

$$\mathbf{A} = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{m1} & \dots & a_{mn} \end{bmatrix}_{m \times n}$$

$m \times n$ jest wymiarem macierzy \mathbf{A}

Działania na macierzach lub tablicach wartości sprowadzają się do odpowiednich działań na poszczególnych elementach macierzy lub tablicy wartości. Zastosowanie wszystkich operatorów matematycznych w obydwu przypadkach jest identyczne z wyjątkiem mnożenia, dzielenia i potęgowania.

Zastosowanie operatorów porównania i operatorów logicznych powoduje przypisanie każdemu niezerowemu elementowi danej macierzy wartości logicznej 1 i każdemu elementowi zerowemu wartości logicznej 0. W wyniku otrzymuje się, zgodnie z wymaganym działaniem, odpowiednią macierz o elementach zerojedynkowych.

Tabela 1-1 Operatory matematyczne

Operatory arytmetyczne	
+, -	dodawanie, odejmowanie
*	mnożenie
/	dzielenie (dzielenie prawostronne)
\	dzielenie (dzielenie lewostronne)
^	potęgowanie
'	transpozycja macierzy (tablicy)
=	znak równości
.	operator tablicowy
Operatory porównania	
==	równe tożsamościowo
~=	różne
<	mniejsze
>	większe
<=	mniejsze równe
>=	większe równe
Operatory logiczne	
	alternatywa
&	koniunkcja
xor()	różnica symetryczna
-	negacja

Wprowadzanie formuł i wartości liczbowych

MATLAB umożliwia także dokonywanie operacji arytmetycznych dla poszczególnych elementów macierzy, przy wykorzystaniu tzw. operatorów tablicowych. W celu odróżnienia działań dokonywanych na macierzach od działań dokonywanych na tablicach wartości, w przypadku tablic wartości należy zawsze po zmiennej umieścić kropkę przed znakiem mnożenia, dzielenia i potęgowania (np. $3*x.^2 - x.*\sin(x) + x./\cos(x)$). W przypadku dzielenia kropkę stosuje się również przed znakiem dzielenia (np. $2./x$).

Elementy macierzy lub tablicy wartości umieszcza się zawsze w nawiasach kwadratowych, natomiast nawiasy zwykłe są zarezerwowane dla argumentów poleceń lub funkcji MATLAB-a. Poszczególne argumenty oddzielane są przecinkami i jeżeli nie stanowią zmiennej lub liczby, umieszczane są w apostrofach (np. $plot(x, y, 'r*')$).

W MATLAB-ie macierze należy oznaczać dużymi literami, natomiast wektory bądź tablice wartości mogą być oznaczane małymi lub dużymi literami. Tę samą zmienną zapisaną raz małą a raz dużą literą program traktuje jak dwie różne zmienne.

Deklaracje macierzy

Macierz można zadeklarować w sposób następujący: podać elementy macierzy, wygenerować elementy macierzy, zdefiniować macierz wykorzystując zależności określające jej elementy lub skorzystać z biblioteki MATLAB-a.

- Tworzenie macierzy przez podanie jej elementów

Macierz tworzy się poprzez umieszczenie jej poszczególnych elementów, oddzielonych spacją, w nawiasach kwadratowych.

$$\mathbf{A} = [a_{11} \quad a_{12}; a_{21} \quad a_{22}]$$

Przykład:

```
» A=[1 1;2 2]
```

```
A =
     1     1
     2     2
```

Znak średnika (";") lub wciśnięcie klawisza ENTER oznacza początek następnego wiersza macierzy.

W przypadku, gdy liczba elementów macierzy jest duża (lub polecenie nie mieści się w jednej linii) należy wykorzystać znak "...", który powoduje, że tekst znajdujący się w następnej linii jest częścią tekstu poprzedniej linii.

Przykład:

```
» A=[1 1; ...
     2 2]
```

```
A =
     1     1     2     2
```

Deklaracja pustej macierzy:

```
>> A=[]
A =
     []
```

- Definiowanie macierzy poprzez generowanie jej elementów

Definiowanie macierzy poprzez generowanie jej elementów realizowane jest w następujący sposób:

```
A = wartość_początkowa : krok : wartość_końcowa
```

Przykład:

```
>> A=1:1:10
A =
     1     2     3     4     5     6     7     8     9    10
```

Powyższe polecenie powoduje utworzenie 10-elementowej macierzy wierszowej.

Wykorzystując elementy tablicy wartości x można wygenerować tablicę wartości y , której elementy określone są w sposób następujący: $y_{ij}=f(x_{ij})$.

Przykład:

```
>> x=1:1:5
x =
     1     2     3     4     5
>> y=x.^2
y =
     1     4     9    16    25
```

- Generowanie macierzy na podstawie zależności określającej jej elementy

Przykład:

Elementy macierzy *Hilberta* zdefiniowane są za pomocą zależności: $A(i,j)=1/(i+j-1)$. Elementy te można wygenerować wykorzystując dwie pętle *for*. Dla stopnia macierzy $n=5$ program obliczeniowy jest następujący:

```

>> n=5
for i=1:n
    for j=1:n
        A(i,j)=1/(i+j-1);
    end
end
disp(A)

A =
    1.0000    0.5000    0.3333    0.2500    0.2000
    0.5000    0.3333    0.2500    0.2000    0.1667
    0.3333    0.2500    0.2000    0.1667    0.1429
    0.2500    0.2000    0.1667    0.1429    0.1250
    0.2000    0.1667    0.1429    0.1250    0.1111

```

Średnik na końcu linii powoduje pominięcie wyświetlania pośrednich obliczeń wykonywanych w danej linii. Polecenie ***disp(A)*** wyświetla wynik.

Macierz Hilberta można także pobrać z biblioteki MATLAB-a pisząc polecenie ***hilb(n)***, gdzie *n* jest stopniem macierzy.

Wyznacznik macierzy

Wyznacznik macierzy kwadratowej (*macierz*) obliczany jest przy pomocy funkcji ***det(macierz)***.

Dla macierzy stopnia drugiego zależności są następujące:

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \quad |\mathbf{B}| = b_{11}b_{22} - b_{12}b_{21}$$

Przykład:

```
>> B=[1 2; 3 4]
```

```
B =
     1     2
     3     4
```

```
>> det(B)
```

```
ans =
    -2
```

Wymiar macierzy

Polecenie ***size(macierz)*** służy do określania wymiaru macierzy.

```
[w, k] = size (A)  wyświetla ilość wierszy w i kolumn k macierzy A
w = size (A,1)    wyświetla ilość wierszy w macierzy A
```

```
k = size (A,2)    wyświetla kolumn k macierzy A
```

Przykład:

```
» A=[ 1 4 6;2 3 4; 4 8 10]
```

```
A =
```

```
     1     4     6
     2     3     4
     4     8    10
```

```
» size(A)
```

```
ans =
     3     3
```

length

Polecenie **length**(wektor) podaje wymiar wektora. W przypadku, gdy argumentem wejściowym jest macierz, wyświetlany jest większy z wymiarów macierzy.

Przykład:

▪ wymiar wektora

```
>> A=[1 2 3 4 5 6 7]
```

```
A =
```

```
     1     2     3     4     5     6     7
```

```
>> length(A)
```

```
ans =
     7
```

▪ wymiar macierzy

```
» A=[1 2 3 4 5;1 3 4 5 6]
```

```
A =
```

```
     1     2     3     4     5
     1     3     4     5     6
```

```
» length(A)
```

```
ans =
     5
```

Odwołania do dowolnego elementu macierzy

Aby odwołać się do danego elementu $A(i,j)$ macierzy \mathbf{A} , należy podać jego współrzędne; tzn. numer wiersza i oraz kolumny j , na przecięciu których znajduje się ten element.

Przykład:

```
>> A=[1 2 3 4 5;6 7 8 9 10;11 12 13 14 15]
```

```
A =
```

```
     1     2     3     4     5
     6     7     8     9    10
    11    12    13    14    15
```

```
>> A(2,3)
```

```
ans =
```

```
     8
```

Transpozycja macierzy

Transpozycja macierzy, czyli zamiana wierszy i kolumn, realizowana jest za pomocą operatora " ' ".

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \quad \mathbf{A}' = \begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \\ a_{13} & a_{23} \end{bmatrix}$$

Przykład:

```
>> A=[1 2 3; 4 5 6]
```

```
A =
```

```
     1     2     3
     4     5     6
```

```
>> A'
```

```
ans =
```

```
     1     4
     2     5
     3     6
```

Odwracanie macierzy

Odwracanie macierzy (*macierz*⁻¹) realizowane jest zgodnie z poniższymi zależnościami (przedstawionymi dla macierzy stopnia drugiego), za pomocą funkcji **inv**(*macierz*).

$$\mathbf{B} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \qquad \mathbf{B}^{-1} = \frac{\mathbf{B}^{D*}}{|\mathbf{B}|}$$

$$\mathbf{B}^{D*} = \begin{bmatrix} \mathbf{B}_{11}^* & \mathbf{B}_{12}^* \\ \mathbf{B}_{21}^* & \mathbf{B}_{22}^* \end{bmatrix}^T$$

$$[\mathbf{B}_{11}^*] = (-1)^2 b_{22} \quad [\mathbf{B}_{12}^*] = (-1)^3 b_{21} \quad [\mathbf{B}_{21}^*] = (-1)^3 b_{12} \quad [\mathbf{B}_{22}^*] = (-1)^4 b_{11}$$

$$\mathbf{B}^{D*} = \begin{bmatrix} b_{22} & -b_{12} \\ -b_{21} & b_{11} \end{bmatrix} \qquad |\mathbf{B}| = b_{11}b_{22} - b_{12}b_{21}$$

$$\mathbf{B}^{-1} = \frac{\mathbf{B}^{D*}}{|\mathbf{B}|} = \begin{bmatrix} \frac{b_{22}}{b_{11}b_{22} - b_{12}b_{21}} & \frac{-b_{12}}{b_{11}b_{22} - b_{12}b_{21}} \\ \frac{-b_{21}}{b_{11}b_{22} - b_{12}b_{21}} & \frac{b_{11}}{b_{11}b_{22} - b_{12}b_{21}} \end{bmatrix}$$

Przykład:

```
>> A=[1 2;3 4]
```

```
A =
     1     2
     3     4
```

```
>> inv(A)
```

```
ans =
-2.0000    1.0000
 1.5000   -0.5000
```

Dodawanie oraz odejmowanie macierzy

Dodawanie oraz odejmowanie macierzy lub tablic wartości o jednakowych wymiarach dokonywane jest zgodnie z zależnościami: