

1. Zmienne i wyrażenia.

Wynik obliczeń wartości wyrażenia jest dostępny jako wartość zmiennej, której to wyrażenie przypisano. Dla polecenia w postaci:

```
>> zmienna = wyrażenie
```

po zatwierdzeniu klawiszem Enter, zostanie obliczona wartość wyrażenia i przypisana zmiennej.

Zmienna będzie wprowadzona do przestrzeni roboczej Matlaba. Jeśli zmienna już istniała, to przyjmie nową wartość. Na ekranie pojawi się odpowiedź:

```
zmienna =  
wartość wyrażenia
```

oraz znak zachęty >> pozwalający na dalszy dialog. Możliwe jest także wykonanie polecenia w postaci:

```
>> wyrażenie
```

Po naciśnięciu klawisza Enter również i w tym przypadku zostanie obliczona wartość wyrażenia. Natomiast wynik obliczeń będzie przypisany standardowej zmiennej roboczej **ans**. Odpowiedź na takie polecenie jest następująca:

```
ans =  
wartość wyrażenia
```

2. Formaty liczb.

Wykaz wszystkich dostępnych formatów uzyskuje się poleceniem **help format**. Możliwa jest także zmiana ustawienia formatów wyprowadzanych wartości liczbowych za pomocą następujących poleceń:

format short	- 5 cyfr, reprezentacja stałoprzecinkowa,
format long	- 15 cyfr, reprezentacja stałoprzecinkowa,
format short e	- 5 cyfr, reprezentacja zmiennoprzecinkowa,
format long e	- 15 cyfr, reprezentacja zmiennoprzecinkowa,
format rat	- wypisanie liczby w postaci ułamka,
format compact	- pomijanie pustych linii przy wyświetlaniu,
format loose	- wprowadzanie pustych linii przy wyświetlaniu,
format	- powrót do formatu domyślnego wypisywania liczb.

3. Zmienne w przestrzeni roboczej.

W Matlabie nie stosuje się deklaracji zmiennych. Zmienne wybranego typu określa użytkownik. Program automatycznie rozpoznaje typ zmiennej. Wyniki wykonanych operacji matematycznych lub funkcji zależą od rozpoznanego typu.

Wprowadzenie zmiennej **a** do przestrzeni roboczej i przypisanie jej wartości wektora, np. 3-elementowego, wykonuje się następująco (elementy wektora oddziela się spacjami lub przecinkami):

```
>> a = [1 2 3]
```

Potwierdzeniem wykonania komendy jest wyświetlenie na ekranie nazwy zmiennej i jej nowej wartości w postaci:

```
a =  
1 2 3
```

Macierz wprowadza się, oddzielając elementy tego samego wiersza spacjami lub przecinkami, a wiersze oddziela się średnikami, np.:

```
>> A = [1 2 3; 8 5 -4; 4 1 -2]  
A =  
1 2 3  
8 5 -4  
4 1 -2
```

Zmiennym można również przypisać ciągi znaków, czyli łańcuchy.

```
>> s = 'Technika'
s =
Technika
```

Wykaz zmiennych przechowywanych w przestrzeni roboczej uzyskuje się przez podanie polecenia **who** lub **whos**.

Polecenie **workspace** otwiera dodatkowe okno z nagłówkiem **Workspace** i, podobnie jak **whos**, podaje pełną informację o zmiennych zgromadzonych w przestrzeni roboczej. Dodatkowo, umożliwia ono pełną edycję tablic. Okno to da się otworzyć także przez wybór opcji **Desktop/Workspace**.

Zmienne wprowadzone do przestrzeni roboczej mogą być zapisywane do zewnętrznych plików poleceniem **save**. Wartości zapisane w takim pliku mogą być wykorzystywane w kolejnych sesjach Matlabu, po wydaniu polecenia **load**.

Zapisywanie zmiennych z przestrzeni roboczej realizuje się również, wykorzystując pliki binarne z rozszerzeniem **.mat**. Do MAT-pliku można wpisać jedną lub kilka zmiennych. Zapis binarny, w odróżnieniu od zapisu w kodzie ASCII, pozwala na zachowanie pełnej dokładności liczb binarnych.

W środowisku Windows operację zapisywania zmiennych wprowadzonych do przestrzeni Matlabu można wykonać poprzez uaktywnienie opcji **File/Save Workspace As**. Otwiera się wówczas okienko dialogowe, w którym należy napisać sesję, tj. nazwę pliku przechowującego wszystkie zmienne przestrzeni roboczej.

Pojedynczą zmienną można zachować w MAT-pliku następująco:

```
>> save L1 L1
```

gdzie pierwsze **L1** to nazwa MAT-pliku (bez rozszerzenia), zaś drugie **L1** to nazwa zmiennej.

Często zachodzi potrzeba usunięcia kilku zmiennych lub wszystkich zmiennych z przestrzeni roboczej. Jest to konieczne na przykład po ukazaniu się komunikatu:

```
>> out of memory
```

oznaczającego brak wolnej pamięci. Wszystkie zmienne z przestrzeni roboczej usuwa polecenie **clear**, jeśli jest użyte samodzielnie. Wybrane zmienne można usunąć, wydając polecenie **clear**:

```
>> clear lista_nazw-zmiennych
```

4. Znaki i nazwy specjalne.

Znaki specjalne są przeznaczone do wprowadzania danych i komentarzy oraz pisania wyrażeń i poleceń. Matlab zawiera pewną liczbę wartości i zmiennych specjalnych, które są reprezentowane przez nazwy specjalne.

Pełne zestawienie znaków specjalnych Matlabu, wraz z objaśnieniami, ilustruje tabela 1. Można je uzyskać za pomocą polecenia **help ops**.

Tabela 1. Znaki specjalne

Symbol	Opis znaku specjalnego
:	generowanie wektorów, indeksowanie macierzy
..	katalog macierzysty
;	koniec wiersza, wstrzymanie wypisania wypowiedzi
...	kontynuacja polecenia w następnej linii
.	kropka dziesiętna, element operatorów arytmetycznych używany przy wskazywaniu
%	początek komentarza
=	przypisanie wartości
,	separator indeksów, argumentów funkcji, poleceń
[]	używane przy tworzeniu tablic, listy argumentów wyjściowych funkcji i w operacjach łączenia macierzy
'	wprowadzanie łańcuchów, operatorów transpozycji lub sprzężenia macierzy
!	wykonanie komendy systemu operacyjnego
{ }	zaznaczanie wyrażeń liczonych w pewnej kolejności i listy argumentów wej. funkcji

Zestawienie wybranych nazw stosowanych w Matlabie, jako zmienne i stałe specjalne, wraz ich opisem, zamieszczono w tabeli 2.

Tabela 2. Nazwy stałych i zmiennych specjalnych

Nazwa	Opis zmiennej lub stałej
<code>pi</code>	3,1415926535897...
<code>eps</code>	dokładność obliczeń
<code>i, j</code>	jednostka urojona
<code>nargin</code>	liczba argumentów wejściowych funkcji
<code>nargout</code>	liczba argumentów wyjściowych funkcji
<code>flops</code>	licznik operacji zmiennoprzecinkowych
<code>realmin</code>	najmniejsza dostępna liczba rzeczywista
<code>realmax</code>	największa dostępna liczba rzeczywista
<code>computer</code>	nazwa komputera, na którym jest zainstalowany Matlab
<code>inf</code>	nieskończoność
<code>nan</code>	wartość nieokreślona (<i>Not a Number</i>)
<code>ans</code>	zmienna robocza - przypisuje się jej automatycznie wartość wyrażenia nie skojarzonego z żadną zmienną
<code>clock</code>	aktualna data i czas jako elementy wektora
<code>now</code>	aktualna data w postaci liczby
<code>date</code>	aktualna data w postaci łańcucha
<code>tic, toc</code>	funkcje do odmierzenia czasu komputera
<code>etime</code>	podaje wartość wybranego przedziału czasu
<code>cputime</code>	upływający czas działania komputera

5. Funkcje arytmetyczne i trygonometryczne.

Zestawienie wybranych elementarnych funkcji matematycznych, dostępnych w Matlabie zawiera tabela 3. Podano w niej także opis obliczeń wykonywanych przez te funkcje.

Tabela 3. Funkcje matematyczne i trygonometryczne

Nazwa	Opis funkcji
<code>abs</code>	wartość bezwzględna, moduł liczby zespolonej, wektor wartości znaków łańcucha wg kodu ASCII
<code>acos, acosh</code>	arcus cosinus, arcus cosinus hiperboliczny
<code>acot, acoth</code>	arcus cotangens, arcus cotangens hiperboliczny
<code>acs, acsh</code>	arcus cosecans, arcus cosecans hiperboliczny
<code>angle</code>	kąt fazowy liczby zespolonej w radianach $[-\pi, +\pi]$
<code>besseli</code>	modyfikowana funkcja Bessela 1-go rodzaju
<code>besselj</code>	funkcja Bessela 1-go rodzaju
<code>besselk</code>	modyfikowana funkcja Bessela 2-go rodzaju
<code>bessely</code>	funkcja Bessela 2-go rodzaju
<code>beta</code>	funkcja Beta
<code>betains</code>	niepełna funkcja Beta
<code>betaln</code>	logarytm funkcji Beta
<code>cart2pol</code>	zmiana współrzędnych prostokątnych na biegunowe
<code>cart2sph</code>	zmiana współrzędnych prostokątnych na sferyczne
<code>ceil</code>	zaokrąglenie w kierunku $+\infty$
<code>conj</code>	liczba sprzężona do liczby danej w argumencie
<code>cos, cosh</code>	cosinus, cosinus hiperboliczny
<code>cot, coth</code>	cotangens, cotangens hiperboliczny
<code>csc, csch</code>	cosecans, cosecans hiperboliczny
<code>ellipj</code>	funkcja eliptyczna Jacobiego
<code>ellipke</code>	całka eliptyczna
<code>erf, erfc, erfcx, erfinv</code>	funkcje błędu Erf

<code>exp</code>	e do potęgi danej przez argument
<code>expint</code>	całka z $(\exp(-f)/t) dt$ w granicach od x do ∞
<code>fix</code>	zaokrąglenie w kierunku zera
<code>floor</code>	zaokrąglenie w kierunku $-\infty$
<code>gamaln</code>	logarytm z funkcji Gamma
<code>gamma, gammainc</code>	funkcja Gamma, niepełna funkcja Gamma
<code>gcd</code>	największy wspólny dzielnik
<code>imag</code>	część urojona liczby zespolonej
<code>lcm</code>	najmniejsza wspólna wielokrotność
<code>legendre</code>	funkcja Legendre'a
<code>log</code>	logarytm naturalny (przy podstawie e)
<code>log10</code>	logarytm przy podstawie 10
<code>log2</code>	logarytm przy podstawie 2
<code>pol2cart</code>	zmiana współrzędnych biegunowych na prostokątne
<code>pow2</code>	potęgowanie przy podstawie 2
<code>rat, rate</code>	rozkład na iloraz dwóch liczb całkowitych
<code>real</code>	część rzeczywista liczby zespolonej
<code>rem</code>	reszta z dzielenia
<code>round</code>	zaokrąglenie do najbliższej liczby całkowitej
<code>sec, sech</code>	secans, secans hiperboliczny
<code>sign</code>	znak funkcji
<code>sin, sinh</code>	sinus, sinus hiperboliczny
<code>sph2cart</code>	zmiana współrzędnych sferycznych na prostokątne
<code>sqrt</code>	pierwiastek kwadratowy
<code>tan, tanh</code>	tangens, tangens hiperboliczny

6. Zapis sekwencji poleceń.

Wprowadzanie z klawiatury kolejnych kroków realizowanego algorytmu lub skomplikowanych wyrażeń matematycznych jest uciążliwe. Wpisywany zestaw poleceń wygodniej jest zapisać w zewnętrznym pliku tekstowym o dowolnej nazwie z rozszerzeniem ***.m**. Do tworzenia i modyfikowania M-plików można wykorzystać wbudowany edytor Matlaba lub dowolny edytor tekstowy.

W edytorze Matlaba można utworzyć nowy M-plik, wybierając z menu opcję **File/New/M-File** lub poprzez kliknięcie w ikonę paska narzędziowego **New M-File**. W rezultacie utworzy się okno edytora, w którym można zapisać tekst M-pliku. Przy modyfikacji M-pliku już istniejącego, należy wybrać opcję **File/Open...** lub kliknąć w ikonę paska narzędziowego **Open file**. Wywołanie wbudowanego edytora można też zrealizować wpisując z okna głównego Matlaba **edit** i nazwę pliku poddawanego edycji, bądź samo **edit**, w przypadku tworzenia nowego M-pliku.

7. Typy danych.

W Matlabie typy danych dzielą się ogólnie na:

- **Typy fundamentalne** - tworzenie zmiennych wybranego typu lub/i zmianę typu istniejącej zmiennej realizuje się za pomocą funkcji: **double**, **sparse**, **char**, **cell**, **struct**, **uint8**. Można przyjąć, że funkcje te są jednocześnie nazwami typów danych (klas) dostępnych w Matlabie.
- **Typy definiowane przez użytkownika** - tworzenie zmiennych, czyli obiektów według typów określonych przez użytkownika dokonuje się za pomocą funkcji **class**.

Fundamentalne typy danych:

- **Tablice dwuwymiarowe** - złożone są z elementów rzeczywistych lub zespolonych, zapisywanych zawsze w podwójnej precyzji.
- **Macierze pełne** - dla tego typu macierzy w pamięci wewnętrznej jest przechowywana wartość każdego ich elementu składowego (w odróżnieniu od macierzy rzadkich).
- **Macierze rzadkie** - ten typ macierzy charakteryzuje się zapamiętywaniem jedynie wartości elementów niezerowych oraz ich indeksów.
- **Tablice znakowe** - elementami takich tablic są znaki lub ciągi znaków, czyli łańcuchy.

- **Tablice wielowymiarowe** - elementami tablic wielowymiarowych mogą być liczby rzeczywiste i zespolone, znaki, bloki lub struktury. W celu określenia elementu takiej tablicy używa się więcej, niż dwóch indeksów - pierwszy indeks jest numerem wiersza, drugi jest numerem kolumny, trzeci i wyższe odpowiadają numerom kolejnych wymiarów.
- **Tablice blokowe** - ich elementami są bloki, które zawierają inne tablice. W blokach można znowu umieszczać tablice o dowolnych wymiarach i typach.
- **Struktury** - struktury, podobnie jak tablice blokowe, mogą przechowywać inne tablice o dowolnych wymiarach i elementach. Struktury różnią się od tablic blokowych sposobem odwoływania się do elementów składowych. Pola struktur mają swoje nazwy własne.
- **uint8** - tablica, której elementy są ośmiobitowymi liczbami całkowitymi, jest stosowana do przechowywania liczb z zakresu 0÷255.

Informację o wymiarach i typach zmiennych można uzyskać poleceniem **whos**.

8. Operatory.

Wyrażenie będące zapisem jakiegoś algorytmu buduje się z zastosowaniem różnych operatorów określających operacje, jakie będą wykonywane na argumentach wchodzących w skład wyrażenia.

Priorytet działania operatorów jest następujący:

- **operatory arytmetyczne**,
- **operatory relacji**,
- **operatory logiczne**.

Priorytet operatorów można zmieniać, ujmując wybrane wyrażenie w nawiasy okrągłe. Gdy stosuje się dwa operatory o tym samym priorytecie, wówczas działania wykonywane są kolejno w kierunku od lewej do prawej.

W Matlabie za pomocą operatorów przeprowadza się 2 rodzaje operacji:

- **operacje macierzowe** - na przykład iloczyn **x*y** oznacza, że obliczenia zostaną przeprowadzone zgodnie z zasadami rachunku macierzowego,
- **operacje tablicowe** - na przykład iloczyn **x.*y** realizuje mnożenie elementów wektorów lub macierzy o tych samych indeksach, tj. **(x(i,j)*y(i,j))**.

Tabela 4. Operatory działań

Symbol operacji macierzowej	Nazwa operacji	Symbol operacji tablicowej
+	dodawanie	+
-	odejmowanie	-
*	mnożenie	.*
^	potęgowanie	.^
/	dzielenie prawostronne	./
\	dzielenie lewostronne	.\
'	sprzężenie macierzy	
.'	transpozycja macierzy	
kron	iloczyn tensorowy Kroneckera	

Tabela 5. Operatory relacji i operatory logiczne

Symbol	Operator relacji	Symbol	Operator logiczny
<	mniejszy od	&	AND
<=	mniejszy lub równy		OR
>	wiekszy od	~	NOT
>=	wiekszy lub równy	xor	EXCLUSIVE OR
==	równy		
~=	różny od		

9. Porównywanie łańcuchów.

Łańcuchy można porównywać, korzystając np. z funkcji **strcmp** lub **strncmp**. Operatory relacji (**>**, **>=**, **<**, **<=**, **==**, **~=**) działają w ten sposób, że dokonują porównania odpowiednich wartości znaków w kodzie ASCII. Możliwe jest zatem porównywanie łańcucha lub tablicy znaków z wartością skalarną.

10. Ćwiczenia.

- ♦ Wyświetl pomoc dla polecenia `sqrt`:
`>> help sqrt`
- ♦ Przypisz zmiennej `a` wartość 100:
`>> a = 10`
- ♦ Wyświetl wartość zmiennej `a`:
`>> disp(a)`
- ♦ Przypisz zmiennej `b` wartość 3.14:
`>> b = 3.14`
- ♦ Wyświetl listę zmiennych znajdujących się w pamięci Matlab'a:
`>> who`
- ♦ Wyświetl szczegółową listę zmiennych wraz z ich rozmiarem i typem danych:
`>> whos`
- ♦ Usuń z pamięci wszystkie zmienne:
`>> clear all`
- ♦ Wyświetl ponownie listę zmiennych:
`>> who`
- ♦ Wyczyść okno poleceń z poprzednio wyświetlanego tekstu:
`>> clc`
- ♦ Utwórz zmienną `x`, sprawdź jej wartość, a następnie oblicz \sqrt{x} .
- ♦ Utwórz zmienne `a = -0,002`, `b = 20,0342 \cdot 10^{-12}`, a następnie sprawdź ich wartości.
- ♦ Przypisz zmiennej `y` wartość 7 bez zwracania wyniku obliczeń.
- ♦ Zapisz w jednym wierszu 3 polecenia: `x = 1, y = 2, z = 3`.
- ♦ Oblicz wartość funkcji `sin(x)` dla `x` zmieniającego się od $-\pi$ do π z krokiem 0,1.
`>> x = -pi:0.1:pi;`
`>> sin(x)`
- ♦ Przypisz zmiennej `s` łańcuch `Matlab`, a następnie dokonaj jego konwersji na wektor kodów ASCII.
`>> s = 'Matlab'`
`>> a = double(s)`
- ♦ Dokonaj konwersji odwrotnej, tj zamień wektor kodów ASCII na łańcuch znaków.
`>> b = char(a)`
- ♦ Wyświetl liczbę znaków łańcucha przypisanego zmiennej `b`.
`>> length(b)`
- ♦ Zdefiniuj zmienne `k = 2` i `pi = 5`.
`>> k = 2; pi = 5;`
- ♦ Oblicz wyrażenie `x = kπ`.
`>> x = k * pi`
- ♦ Zapisz zmienne `k` i `pi` w pliku o nazwie `wynik.mat`.
`>> save wynik k pi`
- ♦ Przywróć domyślną wartość stałej `pi`.
`>> clear pi`
- ♦ Oblicz ponownie `x = kπ` i porównaj wynik z wynikiem wcześniejszym.
- ♦ Wczytaj zmienne z pliku `wynik.mat`, oblicz `x` i porównaj wynik z wynikiem wcześniejszym.
`>> load wynik, x = k * pi`

11. Zadania.

Zadanie 1.

Wykonaj polecenia:

- a) wyświetl wszystkie zmienne lokalne,
- b) zapisz zmienne w pliku dane,
- c) usuń wszystkie zmienne,
- d) wyświetl zawartość katalogu roboczego (`dir`).

Zadanie 2.

Oblicz:

- a) $e^{2\sin(2\pi)}$,
- b) $\cos\left(\frac{\pi}{3}\right)^4$,
- c) $\ln\sqrt{5}$.