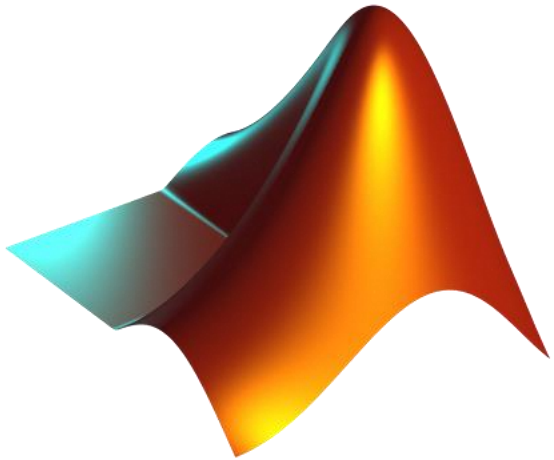
Programowanie w zastosowaniach inżynierskich



Wprowadzenie do środowiska MATLAB

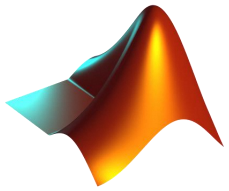
Zmienne i typy danych

Tablice i macierze

Wykresy 2D i 3D

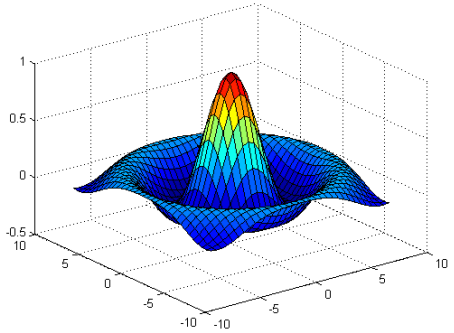
Literatura

- Mrozek B., Mrozek Z., *MATLAB i Simulink. Poradnik użytkownika*, Helion, Gliwice 2017.
- Pratap R., *Matlab dla naukowców i inżynierów*, PWN, 2015
- Sradomski W., *MATLAB. Praktyczny podręcznik modelowania*, Helion, Gliwice 2015.
- Treichel W., *Matlab w działaniu. Ćwiczenia i zadania*, Witkom, 2021
- e-learning na stronie matlab.mathworks.com
  - MATLAB Onramp – podstawy środowiska MATLAB, 14 mod., 2 godz.
  - MATLAB Fundamentals – wprowadzenie do analizy i wizualizacji danych, modelowania i programowania w MATLAB-ie, 18 modułów, 16.5 godz.
  - Simulink Onramp – podstawy modelowania w Simulinku, 14 mod., 2 godz.
  - Simscape Onramp – modelowanie układów fizycznych, 9 mod., 1.5 godz.

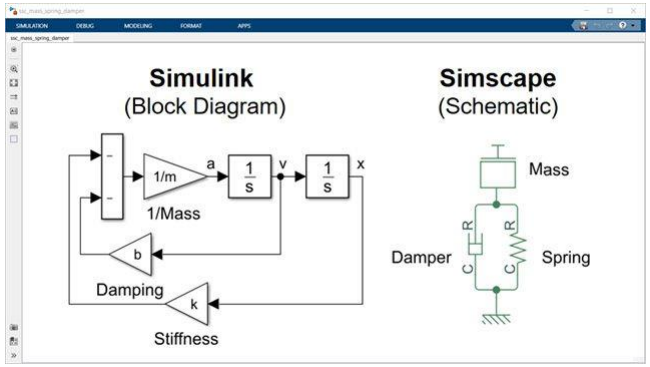


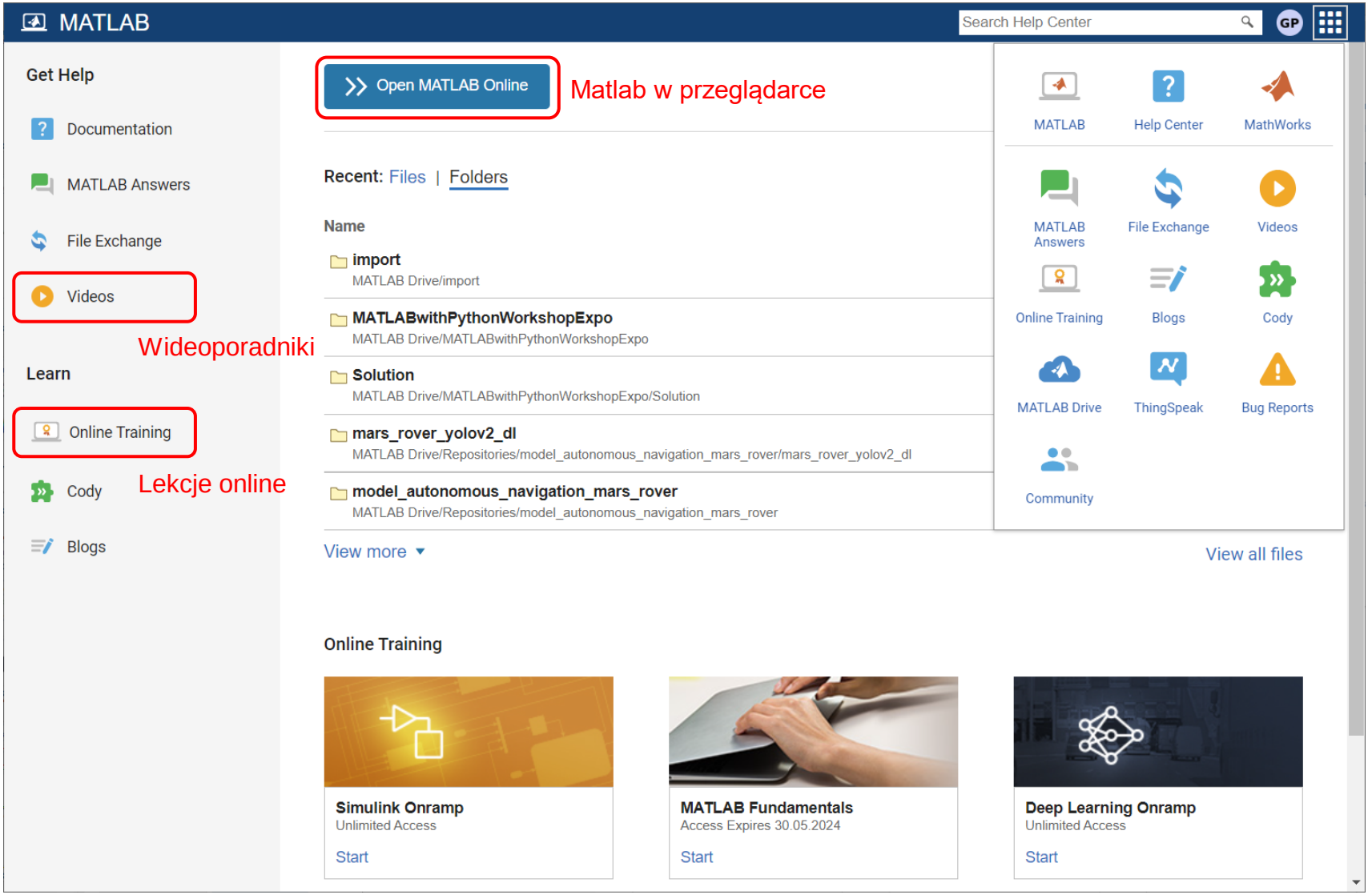
MATLAB – *MATrix LABoratory*, program komputerowy początkowo przeznaczony do obliczeń macierzowych, obecnie interaktywne środowisko do wykonywania obliczeń naukowych i inżynierskich oraz tworzenia zaawansowanych symulacji.

Najważniejsze cechy programu Matlab

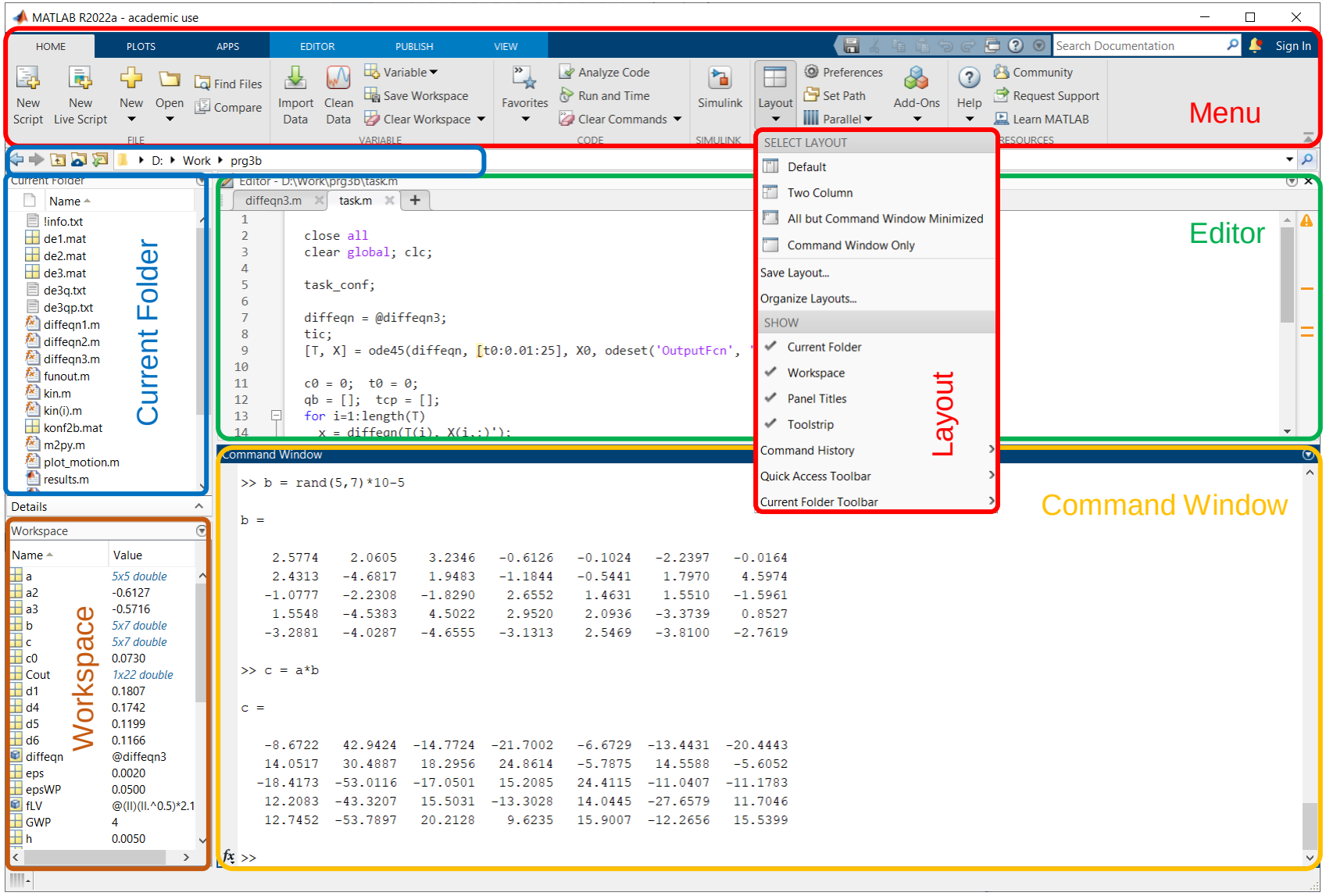


- Duża liczba funkcji bibliotecznych i możliwość rozbudowy przez użytkownika.
- Wbudowany język programowania wysokiego poziomu.
- Możliwość tworzenia dwu i trójwymiarowych wykresów oraz graficznej wizualizacji wyników (statyczne rysunki oraz animacje).
- Możliwość odbierania i wysyłania danych z/do urządzeń zewnętrznych.
- Dodatkowe biblioteki (toolboksy) do rozwiązywania specjalistycznych problemów z różnych dziedzin (matematyka, automatyka, elektronika, sztuczna inteligencja, itp.).
- Graficzne narzędzia do projektowania i symulacji fizycznych systemów z wykorzystaniem schematów i diagramów blokowych (Simulink, Simscape).




MATLAB
Search Help Center
GP
Get Help
Documentation
MATLAB Answers
File Exchange
Videos
Wideoporadniki
Learn
Online Training
Lekcje online
Cody
Blogs
Open MATLAB Online
Matlab w przeglądarce
Recent: Files | Folders
Name
import
MATLAB Drive/import
MATLABwithPythonWorkshopExpo
MATLAB Drive/MATLABwithPythonWorkshopExpo
Solution
MATLAB Drive/MATLABwithPythonWorkshopExpo/Solution
mars_rover_yolov2_dl
MATLAB Drive/Repositories/model_autonomous_navigation_mars_rover/mars_rover_yolov2_dl
model_autonomous_navigation_mars_rover
MATLAB Drive/Repositories/model_autonomous_navigation_mars_rover
View more
View all files
MATLAB
Help Center
MathWorks
MATLAB Answers
File Exchange
Videos
Online Training
Blogs
Cody
MATLAB Drive
ThingSpeak
Bug Reports
Community
Online Training
Simulink Onramp
Unlimited Access
Start
MATLAB Fundamentals
Access Expires 30.05.2024
Start
Deep Learning Onramp
Unlimited Access
Start

Interfejs Matlaba (R2022a)



Podstawowe elementy interfejsu

Menu – zestaw wszystkich opcji programu zebranych w postaci wstążki o zmiennej zawartości dostosowanej do aktualnie wybranego elementu interfejsu.

Command Window – okno umożliwiające interaktywną pracę z programem, każda wprowadzona komenda jest wykonywana, a jej wynik zostaje wyświetlony poniżej.

Editor – edytor tekstu umożliwiający edycję programów/skryptów użytkownika, skrypty mogą być uruchomione przez wprowadzenie ich nazwy w Command Window, lub z wykorzystaniem opcji dostępnych na wstążce menu.

Current Folder – zawartość aktualnie wybranego folderu (pod prawym przyciskiem dostępne podstawowe operacje na plikach).

Workspace – wykaz danych utworzonych w aktualnej sesji (dane są usuwane po zakończeniu pracy programu); podwójne kliknięcie pozwala na podgląd zawartości.

Układ elementów interfejsu może być dowolnie dostosowany przez użytkownika (przeciąganie elementów myszą), układy domyślne, opcje zapisu/odczytu układu oraz widoczne elementy interfejsu mogą być ustawiane w menu Layout.

Zmienne i podstawowe typy danych

Zmienna – element języka programowania używany do przechowywania danych. Podczas pracy programu wartości zmiennej mogą ulegać modyfikacji.

Wybrane typy danych

- numeryczne: `double`, `single` – liczby rzeczywiste (podwójna i pojedyncza precyzja), `int8`, `int16`, `int32`, `int64` – liczby całkowite (8, 16, 32 i 64 bitowe);
- tekstowe: `char` – znak (wartość podawana w apostrofach); `string` – łańcuch/ciąg znakowy (wartość podawana w cudzysłowach);
- logiczne: `logical` – wartości zapisywane jako liczbowe, interpretowane jak "prawda" (każda liczba różna od zera) i "fałsz" (wartość zero);

Uwaga: każda wartość numeryczna domyślnie traktowana jest jako `double`.

Tworzenie zmiennej

```
nazwa = wartość
```

`nazwa` – unikalny ciąg znaków, dozwolone litery, cyfry, znak podkreślenia, zaczyna się od litery.

`wartość` – dowolna wartość dozwolona w programie Matlab, jej typ określa typ zmiennej.

Przykłady

```
x = 123, cena = 550.27, nazwisko = "Kowalski"
```

Tablice i macierze

Tablica (array) – kolekcja elementów tego samego typu zorganizowanych w strukturze o określonej liczbie wierszy i kolumn (możliwa większa liczba wymiarów).

Macierz (matrix) – tablica zawierająca wartości numeryczne; macierz o jednym wierszu lub kolumnie jest nazywana **wektorem**.

Tworzenie tablic

```
nazwa = [w11, w12, … w1m; w21, w22, … w2m; … wn1, wn2, … wnm]
```

Wartości tablicy podawane są wierszami w nawiasach kwadratowych, elementy wiersza oddzielane są przecinkami lub spacjami, wiersze oddzielane są średnikami.

Uwaga: w Matlabie każda wartość traktowana jest jako tablica, wartość prosta (pojedyncza liczba, tekst, itp.) jest tablicą o jednym wierszu i jednej kolumnie.

Funkcje specjalne

- `zeros(n)`, `zeros(n,m)` – tablica wypełniona zerami
- `ones(n)`, `ones(n,m)` – tablica wypełniona jedynkami
- `rand(n)`, `rand(n,m)` – tablica wypełniona wartościami losowymi
- `eye(n)`, `eye(n,m)` – macierz jednostkowa (jedynki na głównej przekątnej)
- `a:b`, `a:b:k` – tablica jednowierszowa od `a` do `b` z domyślnym krokiem 1 lub krokiem określonym przez `k` (maksymalna wartość nie przekracza `b`).

Indeksowanie tablic

- `nazwa_tablicy(i)` – odwołanie do `i`-tego elementu (numerowanie kolumnowe)
- `nazwa_tablicy(i,j)` – odwołanie do elementu z `i`-tego wiersza i `j`-tej kolumny

Uwaga: indeksy mogą być tablicami w takim przypadku odwołanie dotyczy wielu elementów, wierszy lub kolumn; w przypadku tablic o liczbie wymiarów większej od 2 liczba indeksów rośnie.

Symbole specjalne

- dwukropek (`:`) – wszystkie elementy, wiersze, kolumny tablicy
- `end` – ostatni element, wiersz, kolumna tablicy

Modyfikowanie elementów tablicy

```
nazwa_tablicy(index) = wartość
```

- jeżeli `index` wskazuje pojedynczy element zostaje on zastąpiony przez `wartość`
- jeżeli `index` jest tablicą wszystkie elementy będą zastąpione przez `wartość`
- jeżeli `wartość` jest tablicą o wymiarach zgodnych wymiarami `indeksu` wszystkie elementy zostaną zastąpione przez odpowiadające im elementy `wartości`

Indeksowanie tablic – przykłady

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

```
>> b = [10 20 30]
b =
     10   20   30
```

```
>> c = [100; 200]
c =
     100
     200
```

```
>> a(1,3)
ans =
       3
```

```
>> a(3)
ans =
       2
```

```
>> a(2,:)
ans =
       4 5 6
```

```
>> a(1,3) = 30
a =
     1   2   30
     4   5    6
```

```
>> a(1,:) = 10
a =
     10   10   10
      4    5    6
```

```
>> a(end,1:2) = b(2:3)
a =
     10   10   10
     20   30    6
```

Uwaga: jeżeli uruchamiane wyrażenie nie będzie przypisane do zmiennej Matlab automatycznie utworzy zmienną "ans" (answer).

Operatory relacyjne i logiczne

Operatory relacyjne

Operator	*Opis*
`==`	Równy
`>=`	Większy lub równy
`>`	Większy
`<=`	Mniejszy lub równy
`<`	Mniejszy
`~=`	Różny
`isequal`	Równość tablic

Operatory logiczne

Operator	*Opis*
`&`	Koniunkcja (AND)
`|`	Alternatywa (OR)
`~`	Negacja (NOT)

`x`	`y`	`x & y`	`x | y`	`~x`
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0
1	1	1	1	0

Uwaga I: wynikiem działania operatorów relacyjnych jest wartość logiczna (0 – fałsz, 1 – prawda).

Uwaga II: w przypadku porównywania tablic wynikiem operatorów relacyjnych jest tablica logiczna o wymiarze równym tablicy porównywanej zawierająca rezultat porównania poszczególnych elementów.

Uwaga III: tablice mogą być indeksowane tablicami logicznymi jako wynik zwracane są wszystkie elementy, które odpowiadają indeksom o wartości 1 (prawda).

Indeksowanie tablic wartościami logicznymi

```
>> v = [ -2 0 4 -3 5 ]
   v =
       -2    0    4   -3    5
>> v > 0
   ans =
        1×5 logical array
        0    0    1    0    1
>> v(v>0)
   ans =
        4    5
>> v(v>0) = 0
   v =
       -2    0    0   -3    0
>> v(v<0)=-v(v<0)
   v =
        2    0    0    3    0
```

```
>> m = [ 1 -7 3 9; -6 6 0 -2 ]
   m =
        1   -7    3    9
       -6    6    0   -2
>> idx = ( m < -5 ) | ( m > 5 )
   idx =
          2×4 logical array
          0    1    0    1
          1    1    0    0
>> m(idx) = sign(m(idx))*5
   m =
        1   -5    3    5
       -5    5    0   -2
>> m(~idx) = 0
 m =
        0   -5    0    5
       -5    5    0    0
```

- **sign** zwraca znak wartości (1 lub -1)

Operatory tablicowe i macierzowe

Operatory tablicowe operują na pojedynczych elementach tablicy.

Operatory macierzowe wykonują operacje zgodnie z regułami algebry liniowej.

Operator		*Opis*
+		Suma tablic/macierzy
-		Różnica tablic/macierzy
.*	*	Iloczyn tablicowy (`A.*B=[A(i,j)*B(i,j)]`), mnożenie macierzy
./	/	Prawostronne dzielenie tablic (`A./B=[A(i,j)/B(i,j)]`) , prawostronne dzielenie macierzy (rozwiązanie układu równiań `xA=B`, A i B muszą mieć tą samą liczbę kolumn)
.\	\	Lewostronne dzielenie tablic (`A.\B=[B(i,j)/A(i,j)]`), lewostronne dzielenie macierzy (rozwiązanie układu równań `Ax=B`, A i B muszą mieć tą samą liczbę wierszy)
.^	^	Potęgowanie tablicowe (`A.^B=A(i,j)^B(i,j)`), potęgowanie macierzy
.'	'	Transpozycja (wiersze są przekształcane w kolumny), transpozycja algebraiczna (różnica występuje tylko dla macierzy zespolonych)

Uwaga: część operacji wykonywana jest tak samo w przypadku tablic i macierzy, takie operatory mają pojedynczy symbol.

Operatory tablicowe – przykłady

```
>> a = [1 2 3; 4 5 6];
>> b = [3 2 1; 0 -1 -2];
```

```
>> c = a + b
c =
     4   4   4
     4   4   4
```

```
>> c = a + 2
c =
     3   4   5
     6   7   8
```

```
>> c = a .* b
c =
     3    4     3
     0   -5   -12
```

```
>> c = a.^2
c =
     1    4    9
    16   25   36
```

```
>> c = b.^a
c =
     3    4    1
     0   -1   64
```

```
>> c = a'
c =
     1   4
     2   5
     3   6
```

Uwaga: jeżeli wyrażenie zostanie zakończone średnikiem, to jego wynik nie będzie wyświetlony w Command Window.

Operatory macierzowe – przykłady

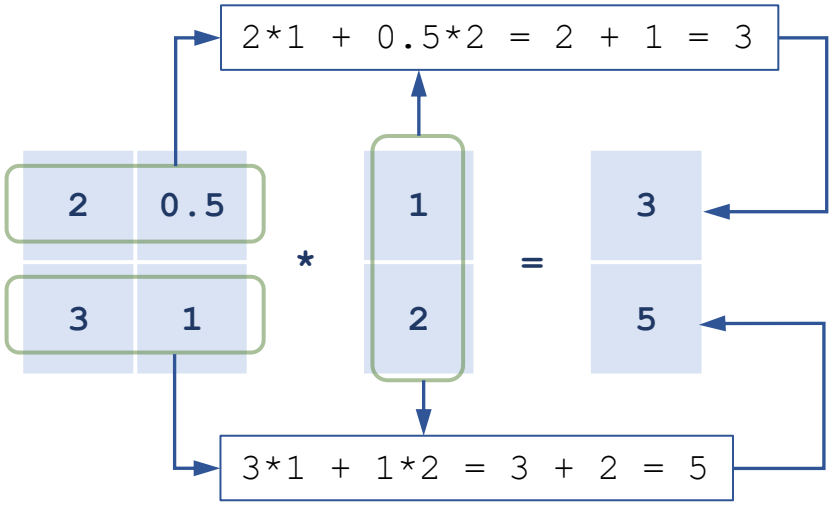
```
a = [2 0.5; 3 1];
b = [1; 2];
c = [1 2 3; 4 5 6];
```

Uwaga: mnożenie macierzy jest wykonalne, gdy liczba kolumn pierwszej jest równa liczbie wierszy drugiej

```
>> d = c * 3
d =
     3    6    9
    12   15   18
```

```
>> d = a * b
d =
     3
     5
```

```
>> d = a * c
d =
     4    6.5    9
     7   11     15
```



Wybrane operacje na tablicach

Łączenie i modyfikowanie tablic

- `horzcat(A,B,…,C)` – łączy tablice w poziomie (alternatywa `[A,B,…C]`), liczba wierszy we wszystkich tablicach musi być równa
- `vertcat(A,B,…,C)` – łączy tablice w pionie (alternatywa `[A;B;…C]`), liczba kolumn we wszystkich tablicach musi być równa
- `reshape(A,d)` – przekształca `A` przestawiając elementy tak żeby utworzyły tablicę o wymiarach określonych przez wektor `d` (liczba elementów nie może ulec zmianie)
- `A(index)=[]` – usuwa element określony przez index

Rozmiar i kształt tablic

- `length(A)` – długość największego wymiary tablicy
- `size(a)` – rozmiar tablicy (wektor dla tablic wielowymiarowych)
- `ndims(A)` – liczba wymiarów tablicy
- `numel(A)` – liczba elementów tablicy
- `isscalar(A)` – określa czy `A` jest wartością skalarną (np. pojedyncza liczba)
- `isvector(A)`, `iscolumn(A)`, `isrow(A)` – określa czy A jest wektorem (dowolnym, kolumnowym, wierszowym)
- `ismatrix(A)` – określa czy `A` jest macierzą

Wykresy 2D

Funkcje rysujące wykresy 2D wymagają przygotowania danych w postaci dwóch wektorów zawierających punkty z dziedziny i odpowiadające im wartości.

Wybrane funkcje wykreślające wykresy 2D

- `plot(X,Y,LineSpec)` – wykres liniowy
- `stairs(X,Y,LineSpec)` – wykres schodkowy
- `bar(X,Y)`, `barh(X,Y)` – wykres słupkowy (pionowy i poziomy)
- `histogram(X,n)` – wyświetla histogram danych zgromadzonych w `X` z zastosowaniem podziału na `n` grup (ilość elementów w każdej grupie)
- `bubblechart(X,Y,s)` – wykres bąbelkowy (w punktach określonych przez `X` i `Y` rysowane jest kółko o rozmiarze `s`, argument `s` może być wektorem)

Uwagi:

- *funkcja wymagające określenia dziedziny i zbioru wartości akceptują jeden argument, który traktowany jest jako zbiór wartości, dziedzinę stanowią wówczas numery próbek,*
- *w funkcji `plot` argument `Y` może być macierzą wielokolumnową, każda kolumna zostanie potraktowana jako osobna seria danych (funkcja wykreśli kilka wykresów).*
- *w funkcji `plot` dziedzina i zbiór wartości mogą być podawane wielokrotnie (jako kolejne argumenty), każdy zbiór danych zostanie wykreślony jako kolejny wykres.*

Wykresy 2D – parametry linii

Parametr `LineSpec` pozwala określić styl i kolor linii używanej do wykreślenia wykresu oraz rodzaj markerów używanych do oznaczenia punktów. Parametr podawany jest jako łańcuch znakowy (tekst w cudzysłowach) zawierający zestaw symboli.

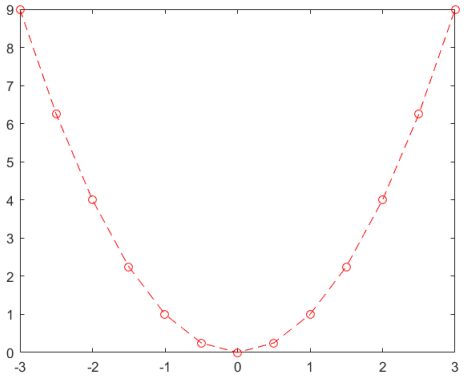
Styl linii	
Symbol	***Opis***
`-`	Linia ciągła
`--`	Linia przerywana
`:`	Linia kropkowa
`-.`	Linia kreska-kropka

Markery	
Symbol	***Opis***
`o`	Kółko
`+`	Plus
`*`	Gwiazdka
`.`	Kropka
`x`	Krzyż

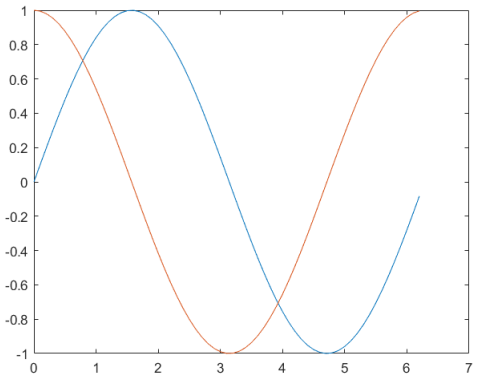
Kolory	
Symbol	***Opis (R,G,B)***
`r`	Czerwony (1,0,0)
`g`	Zielony (0,1,0)
`b`	Niebieski (0,0,1)
`c`	Cyjan (0,1,1)
`m`	Magenta (1,0,1)
`y`	Żółty (1,1,0)
`k`	Czarny (0,0,0)
`w`	Biały (1,1,1)

Wykresy 2D – przykłady

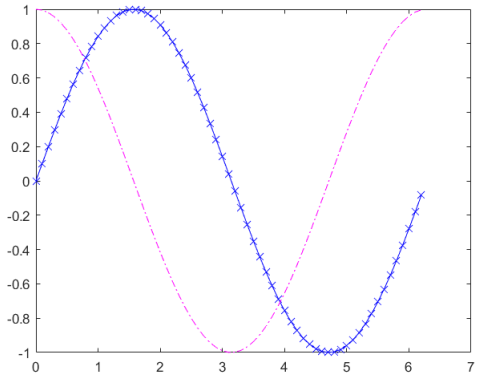
```
>> x = -3:0.5:3;
>> y = x.^2;
>> plot(x,y,"--or")
```



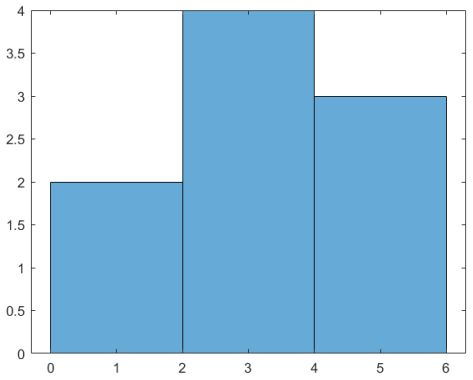
```
>> x=0:0.1:2*pi;
```



- `>> y=[sin(x); cos(x)];`
- `>> y1=sin(x); y2=cos(x);`
- `>> plot(x,y1,"-xb", x,y2,"-.m")`



- `>> x = [ 2.5, 4, 2.2, 3, 6, 5, 0, 1, 3.5];`
- `>> histogram(x,3)`



Wykresy 3D

Funkcje rysujące wykresy 3D wymagają przygotowania danych w postaci trzech macierzy z których każda zawiera odpowiednią współrzędną kreślonego wykresu.

Wybrane funkcje wykreślające wykresy 3D

- `surf(X,Y,Z)` – wykres powierzchniowy (oczka wypełnione)
- `mesh(X,Y,Z)` – wykres powierzchniowy (oczka niewypełnione)
- `contour(X,Y,Z)` – wykres poziomicowy

Funkcje przygotowujące dziedzinę wykresu 3D

- `[X,Y]=meshgrid(x,y)`

  generuje siatkę punktów opisujących dziedzinę wykresu 3D, `x, y` to wektory zawierające wartości współrzędnych siatki a `X, Y` to wygenerowane na ich podstawie macierze ze współrzędnymi punktów dziedziny, funkcja meshgrid może być wywoływana z jednym parametrem jeśli `y` jest równe `x`

- `[X1,X2,…]=ndgrid(x1,x2,…)`

  generuje siatkę punktów opisujących dziedzinę wykresu w przestrzeni N wymiarowej, znaczenie parametrów `x1,x2,…` oraz `X1,X2,…` analogiczne jak w `meshgrid`

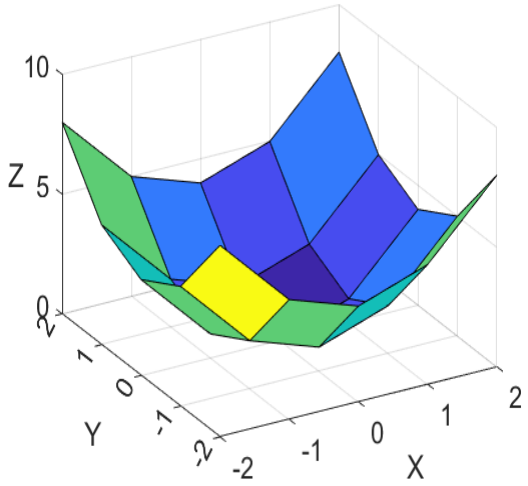
```
>> [X, Y] = meshgrid(-2:2)

X =
     -2     -1      0      1      2
     -2     -1      0      1      2
     -2     -1      0      1      2
     -2     -1      0      1      2
     -2     -1      0      1      2
Y =
     -2     -2     -2     -2     -2
     -1     -1     -1     -1     -1
      0      0      0      0      0
      1      1      1      1      1
      2      2      2      2      2

>> Z = X.^2 + Y.^2

Z =
      8      5      4      5      8
      5      2      1      2      5
      4      1      0      1      4
      5      2      1      2      5
      8      5      4      5      8

>> surf(X, Y, Z)
```



Dziedzina (złożenie X i Y)

```
(-2,-2) (-1,-2) (0,-2) (1,-2) (2,-2)
(-2,-1) (-1,-1) (0,-1) (1,-1) (2,-1)
(-2, 0) (-1, 0) (0, 0) (1, 0) (2, 0)
(-2, 1) (-1, 1) (0, 1) (1, 1) (2, 1)
(-2, 2) (-1, 2) (0, 2) (1, 2) (2, 2)
```