

# PODSTAWY INFORMATYKI 1

## MATLAB CZ. 3

**TEMAT: Program Matlab: Instrukcje sterujące, grafika.**

### Wyrażenia logiczne

Wyrażenia logiczne służą do porównania wartości zmiennych o tych samych rozmiarach. W wyrażeniach logicznych mogą występować operatory relacyjne i logiczne.

Operatory relacyjne		Operatory logiczne	
operator	znaczenie	operator	znaczenie
<b>x == y</b>	<b>x = y</b>	<b>x   y</b>	x lub y (OR)
<b>x ~= y</b>	<b>x ≠ y</b>	<b>x &amp; y</b>	x i y (AND)
<b>x &lt; y</b>	<b>x &lt; y</b>	<b>~x</b>	nie x (NOT)
<b>x &gt; y</b>	<b>x &gt; y</b>		
<b>x &lt;= y</b>	<b>x ≤ y</b>		
<b>x &gt;= y</b>	<b>x ≥ y</b>		

Jeśli porównywane są skalary i wyrażenie logiczne jest prawdziwe to zwracana jest wartość **1**, jeśli fałszywe - wartość **0**.

Jeśli porównywane są macierze lub wektory o tych samych rozmiarach, to porównywanie wykonywane jest element po elemencie i zwracana jest macierz zawierająca wartości **1** lub **0** na odpowiednich pozycjach (zależnie od wyniku porównania).

### Instrukcja warunkowa if

Instrukcja warunkowa **if** może występować w następujących postaciach:

```
if wyrażenie  
    instrukcje  
end
```

```
if wyrażenie  
    instrukcje  
elseif wyrażenie  
    instrukcje  
else  
    instrukcje  
end
```

**Wyrażenie** jest to wyrażenie logiczne, **instrukcje** zostaną wykonane jeśli **wyrażenie** jest prawdziwe.

### Przykład:

```
% Skrypt sprawdzający ile liczb spośród trzech
% wprowadzonych znajduje się w przedziale <5,10>
ilosc=0;
x=input('Liczba nr 1: ');
if (x>=5) & (x<=10)
    ilosc = ilosc + 1;
end
x=input('Liczba nr 2: ');
if (x>=5) & (x<=10)
    ilosc = ilosc + 1;
end
x=input('Liczba nr 3: ');
if (x>=5) & (x<=10)
    ilosc = ilosc + 1;
end
disp('Ilość liczb w przedziale: ')
disp(ilosc);
```

### Pętla for

Ogólna postać instrukcji **for**:

```
for zmienna = macierz_wartości
    instrukcje
end
```

Działanie pętli **for** polega na przypisywaniu **zmiennej** kolejnych kolumn **macierzy\_wartości**.

**Macierz\_wartości** ma najczęściej jedną z dwóch postaci:

- **min:max**
- **min:krok:max**

### Przykład:

```
% Skrypt sprawdzający ile liczb spośród trzech
% wprowadzonych znajduje się w przedziale <5,10>
ilosc=0;
for i = 1:3
    x=input('Podaj liczbę: ');
    if (x>=5) & (x<=10)
        ilosc = ilosc + 1;
    end
end
disp('Ilość liczb w przedziale: ')
disp(ilosc);
```

## Tworzenie i usuwanie okien graficznych

Grafika w Matlabie wyświetlana jest w oknach graficznych. Jednocześnie może być otwartych kilka okien graficznych, ale tylko jedno z nich jest oknem aktywnym (czyli takim, w którym wyświetlane są wyniki działania funkcji graficznych). Jeśli żadne okno graficzne nie jest otwarte, to funkcje graficzne **automatycznie tworzą nowe**. Do obsługi okien graficznych wykorzystywane są funkcje:

- figure** - tworzy nowe okno graficzne,
- figure(n)** - tworzy nowe okno graficzne o numerze **n** (jeśli okno o tym numerze już istnieje, to staje się aktywnym),
- close** - zamyka aktywne okno graficzne,
- close all** - zamyka wszystkie okna graficzne,
- clf** - czyści zawartość aktywnego okna graficznego,
- subplot(m,n,p)** - dzieli okno graficzne na mniejsze prostokątne okienka umieszczając je w **m**-wierszach i **n**-kolumnach, **p** - numer aktywnego okienka, w każdym okienku można umieścić odrębny wykres.

## Grafika dwuwymiarowa (2D)

Do wyświetlania grafiki dwuwymiarowej (krzywych) służy funkcja **plot**. W zależności od liczby argumentów wywołanie funkcji może mieć różną postać:

- plot(x,y)** - rysuje wykres  $y = f(x)$ ,
- plot(y)** - rysuje wykres elementów wektora **y**, przyjmując za **x** kolejne liczby całkowite począwszy od **1**,
- plot(x,y,s)** - rysuje wykres  $y = f(x)$  z określeniem sposobu rysowania linii,
- plot(x1,y1,x2,y2,...)** - rysuje w jednym oknie wiele wykresów,
- plot(x1,y1,s1,x2,y2,s2,...)** - rysuje w jednym oknie wiele wykresów z określeniem sposobu rysowania linii każdego z nich.

Łańcuch znaków (**s**) opisujący sposób rysowania linii może określać: rodzaj linii, jej kolor oraz oznaczenie punktów. Możliwe wartości zestawiono w poniższej tabeli:

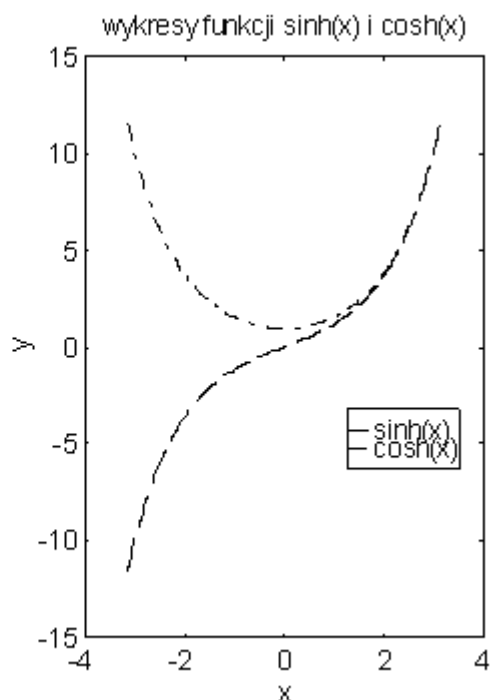
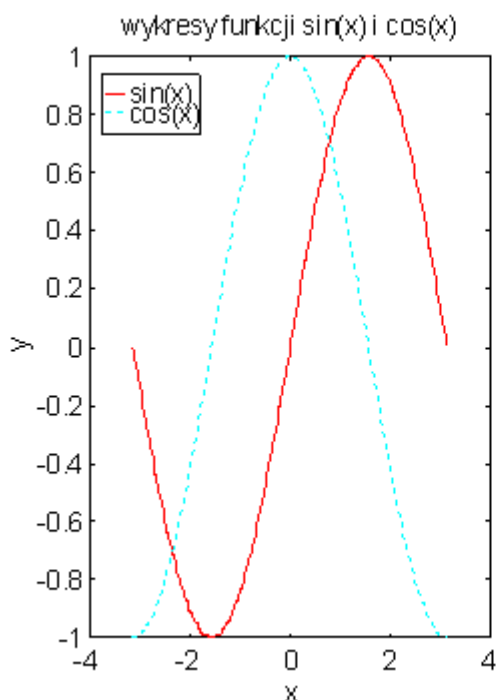
Znak	Rodzaj linii	Znak	Kolor linii	Znak	Ozn. punktów
-	ciągła (domyślna)	<b>y</b>	żółty	<b>+</b>	krzyżyk
--	kreskowana	<b>m</b>	karmazynowy	<b>*</b>	gwiazdka
:	kropkowana	<b>c</b>	turkusowy	<b>.</b>	kropka
-. .	kreska-kropka	<b>r</b>	czerwony	<b>o</b>	kółko
		<b>g</b>	zielony	<b>x</b>	iks
		<b>b</b>	niebieski	<b>s</b>	kwadrat
		<b>w</b>	biały	<b>d</b>	romb
		<b>k</b>	czarny	<b>p</b>	gwiazdka

Wykresy mogą być opisywane przy wykorzystaniu przedstawionych poniżej funkcji:

- xlabel (tekst)** - wyświetla **tekst** opisujący oś **x** aktywnego wykresu,
- ylabel (tekst)** - wyświetla **tekst** opisujący oś **y** aktywnego wykresu,
- title (tekst)** - wyświetla **tekst** jako tytuł aktywnego wykresu,
- text (x,y, tekst)** - wyświetla łańcuch znaków **tekst** w miejscu określonym przez współrzędne **x** i **y**,
- legend (s1,s2,...)** - wyświetla legendę, **s1** - opis pierwszego wykresu, **s2** - opis drugiego wykresu, itd.

### Przykład:

```
x=-pi:0.01:pi;  
y1=sin(x);  
y2=cos(x);  
y3=sinh(x);  
y4=cosh(x);  
subplot(1,2,1);  
plot(x,y1,'-r',x,y2,':b');  
xlabel('x');  
ylabel('y');  
legend('sin(x)', 'cos(x)');  
title('wykresy funkcji sin(x) i cos(x)');  
subplot(1,2,2);  
plot(x,y3,'--g',x,y4,'-.w');  
xlabel('x');  
ylabel('y');  
legend('sinh(x)', 'cosh(x)');  
title('wykresy funkcji sinh(x) i cosh(x)');
```



## Funkcja fplot

Do narysowania wykresu dowolnej funkcji można wykorzystać funkcję **fplot**:

```
fplot(function, [x1 x2])
```

gdzie: **function** - łańcuch znaków zawierający nazwę funkcji,  
**x1, x2** - granice przedziału argumentów funkcji.

Wywołanie funkcji **fplot** jest następujące:

```
[x,y]=fplot(function, [x1 x2])
```

Funkcja ta nie rysuje wykresu, a tylko przygotowuje wektory **x** i **y** do narysowania wykresu funkcją **plot(x,y)**.

### Przykład:

```
[x,y]=fplot('2*sin(x)*cos(x/2)', [-pi pi]);  
plot(x,y);
```

## Grafika trójwymiarowa (3D)

Rysowanie wykresów funkcji trzech zmiennych (krzywej) umożliwia funkcja **plot3** o składni:

```
plot3(x,y,z,s)
```

gdzie: **x, y, z** - wektory o jednakowej długości określające współrzędne punktów,  
**s** - łańcuch znaków opisujący wygląd linii.

Narysowanie wykresu powierzchniowego (czyli wykresu  $z = f(x,y)$ ) jest nieco bardziej skomplikowane. Pierwszym krokiem jest wygenerowanie specjalnej siatki na płaszczyźnie **XY** w węzłach której określone będą wartości funkcji w osi **z**. Służy do tego funkcja **meshgrid** o składni:

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

gdzie: **X, Y** - wektory zawierające wartości w punktach ograniczających płaszczyznę **XY**,  
**x, y** - macierze określające współrzędne, dla których będą liczone wartości **z**.

Po przygotowaniu siatki opisanej macierzami **x** i **y** należy obliczyć wartości macierzy **z**:  $z = f(x,y)$ . Do narysowania powierzchni wykorzystywane są przedstawione poniżej funkcje. Powstały wykres składa się z czworokątów, których wierzchołki leżą w punktach o współrzędnych opisanych macierzami **x, y, z**.

**mesh(x,y,z,c)** - rysuje powierzchnię w postaci kolorowej siatki o polach wypełnionych kolorem tła, elementy macierzy **c** określają kolory linii poszczególnych pól,

**mesh(x,y,z)** - rysuje powierzchnię przy **c=z**,

**surf(x,y,z,c)** - rysuje różnokolorową powierzchnię,

**surf(x,y,z)** - rysuje różnokolorową powierzchnię przyjmując **c=z**.

### Przykład:

```
[x,y]=meshgrid(-2:0.1:2,-2:0.1:2);  
z=peaks(x,y);  
mesh(x,y,z);  
colorbar;
```

```
[x,y]=meshgrid(-3:0.2:3,-3:0.2:3);  
z=sinh(x).*cosh(x/2).*log(abs(y)+0.75);  
surf(x,y,z);
```

### LITERATURA:

1. B. Mrozek, Z. Mrozek: *MATLAB i Simulink: poradnik użytkownika*. Helion, Gliwice, 2004.
2. M. Wciślik: *Wprowadzenie do systemu MATLAB*. Politechnika Świętokrzyska, Kielce, 2003.
3. A. Zalewski, R. Cegiela: *Matlab - obliczenia numeryczne i ich zastosowania*. Wydawnictwo Nakom, Poznań, 2000.
4. J. Brzózka, L. Dorobczyński: *Programowanie w Matlab*. Wydawnictwo Mikom, Warszawa, 1998.
5. B. Mrozek, Z. Mrozek: *MATLAB - Uniwersalne środowisko do obliczeń naukowo-technicznych*. Wydawnictwo PLJ, Warszawa, 1996.
6. A. Kamińska, B. Pańczyk: *Ćwiczenia z ... Matlab, Przykłady i zadania*. Wydawnictwo MIKOM, Warszawa, 2002.