

Operatory arytmetyczne

Działanie	Znak
Dodawanie	+
Odejmowanie	-
Mnożenie macierzowe	*
Mnożenie tablicowe	.*
Dzielenie macierzowe	/
Dzielenie tablicowe	./
Potęgowanie macierzowe	^
Potęgowanie tablicowe	.^

Operatory relacji

Wyrażenie	Relacja
$A == B$	A równe B
$A \sim B$	A różne od B
$A < B$	A mniejsze od B
$A > B$	A większe od B
$A \leq B$	A mniejsze równe B
$A \geq B$	A większe równe B

Uwaga !

Należy uważać przy korzystaniu z operatora `==`. Ze względu na to, że reprezentowanie liczb zmiennoprzecinkowych obarczone jest niedokładnością, nawet niewielka różnica między spodziewaną wielkością, a otrzymaną spowoduje, że operator równości da wartość oznaczającą fałsz. Przykładowo przy zastosowaniu operatora `==` w pętli **while** może powstać pętla nieskończona.

Operatory logiczne

Operator	Funkcja logiczna
$A B$	Alternatywa (or)
$A \& B$	Koniunkcja (and)
$\text{xor}(A,B)$	Różnica symetryczna
$\sim A$	Negacja

Funkcje matematyczne

Funkcja	Opis
$\sin(x)$	Sinus
$\cos(x)$	Cosinus
$\tan(x)$	Tangens
$\text{asin}(x)$	Arcus sinus
$\text{acos}(x)$	Arcus cosinus
$\text{atan}(x)$	Arcus tangens
$\text{sqrt}(x)$	\sqrt{x}
$\text{exp}(x)$	e^x
$\log(x)$	$\ln(x)$
$\log_2(x)$	$\log_2(x)$
$\log_{10}(x)$	$\log_{10}(x)$
$\text{abs}(x)$	Moduł liczby x

Funkcje graficzne 2D

- Funkcja **plot** – wykres funkcji jednej zmiennej na podstawie danych wartości funkcji i jej argumentów.

```
plot(y)
plot(x,y)
plot(x,y,s)
plot(x1,y1,x2,y2,...)
plot(x1,y1,s1,x2,y2,s2,...)
```

gdzie:

$y, y1, y2, \dots$ – wektor lub macierz z wartościami wykreślanej funkcji,

$x, x1, x2, \dots$ – wektor lub macierz z wartościami wykreślanej funkcji (gdy nie jest podany, wówczas na osi x wykresu pojawiają się numery kolejnych elementów z macierzy lub wektora $y, y1, \text{ lub } y2, \dots$,

$s, s1, s2, \dots$ – łańcuch zawierający kod koloru, oznaczenie typu linii lub kod koloru i oznaczenie typu linii.

Kody kolorów funkcji <i>plot</i>	
Kod	Kolor
y	żółty
m	karmazynowy
c	cyan
r	czerwony
g	zielony
b	niebieski
w	biały
k	czarny

Kody rodzaju linii funkcji <i>plot</i>	
.	punkty
o	okręgi w punktach
x	x w punktach
+	+ w punktach
*	* w punktach
-	linia ciągła
--	linia kreskowana
:	linia kropkowana
-.	linia kreska-kropka

Przykładowy łańcuch **s**:

`\r-` – linia czerwona ciągła,
`\b--` – linia niebieska kreskowana.

Przykład

```
x=-pi:0.1:pi;
y=cos(x);
plot(x,y)
```

- Polecenie **subplot** – określa kolejność zamieszczania wykresów w oknie wykresów.

`subplot(m, n, p)` – m wykresów w pionie, n wykresów w poziomie numer wykresu, który zostanie narysowany najbliższym wywołaniem funkcji **plot**.

Przykład

```
x1=-pi:0.1:pi;  
y1=cos(x);  
x2=-5:0.1:5;  
y2=x2.^2;  
x3=0:0.1:100;  
y3=2*x3+3;  
subplot(3,1,1);  
plot(x1,y1);  
subplot(3,1,2);  
plot(x2,y2);  
subplot(3,1,3);  
plot(x3,y3);
```

- Polecenie **grid** – służy do wyświetlania siatki wykresu.

grid, grid on – włączają siatkę wykresu,
grid off – włącza siatkę wykresu.

- Polecenia **xlabel** i **ylabel** – włączają etykiety osi x i y.

```
xlabel(tekst)  
ylabel(tekst)
```

tekst – łańcuch zawierający tekst opisujący osie wykresu.

Przykład

```
xlabel('oś x')  
ylabel('oś y')
```

- Funkcja **fplot** – wykres funkcji jednej zmiennej podanej w postaci zależności umieszczonej w m-pliku. Punkty wykresu dobierane są automatycznie.

```
fplot(f,granice)  
fplot(f,granice,n)  
fplot(f,granice,n,kąt)  
fplot(f,granice,n,kąt,podprzedziały)  
[x,y]=fplot(...)
```

f – łańcuch znaków określający nazwę pliku, w którym umieszczona jest wykreślana funkcja,

granice – dwuelementowy wektor z granicami przedziału, w którym ma zostać wykreślona funkcja,

n – parametr określający minimalną liczbę punktów wykresu funkcji (domyślnie 25),

kąt – parametr określający kąt (w stopniach) pomiędzy sąsiednimi odcinkami wykresu,

powyżej którego zwiększana jest liczba punktów próbkowania (domyślnie 10),
`podprzedziały` – parametr określający maksymalną liczbę próbkowania jaka może zostać dodana w gwałtownie zmieniających się miejscach wykresu (domyślnie 20).
`[x,y]=fplot(...)` – wywołanie funkcji **fplot** w tej postaci powoduje utworzenie wektorów: **x** z argumentami funkcji i **y** z wartościami funkcji. Wykres funkcji nie jest tworzony.

Przykład

M-plik z definicją wykreślanej funkcji:

```
function y=funkcja1(x)
y=sin(x);
```

Polecenie rysujące wykres funkcji:

```
fplot('funkcja1',[-pi pi])
```

Funkcje graficzne 3D

- Funkcja **meshgrid** – tworzy macierze ze współrzędnymi punktów wykresu 3D.

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

- Polecenie **surf** lub **mesh** – tworzy wykres 3D – rysuje powierzchnie opisane przez macierze x, y i z. Macierze te mogą być wygenerowane przy pomocy polecenia **meshgrid**.

```
mesh(x,y,z)
mesh(z)
surf(x,y,z)
surf(z)
```

Przykład

Skrypt tworzący wykres funkcji: $f(x,y)=\cos(x)\cdot\sin(x)\cdot e^{-x^2-y^2}$ w przedziale $x\in\langle-\pi,\pi\rangle$, $y\in\langle-\pi,\pi\rangle$.

```
[X,Y]=meshgrid(-pi:0.2:pi,-pi:0.2:pi)
Z=cos(X).*sin(Y).*exp(-X.^2-Y.^2);
mesh(X,Y,Z); %% lub surf(X,Y,Z);
```

Elementy programowania

- Instrukcja warunkowa **if**

```
if wyrażenie_warunkowe_1
    instrukcje_1
elseif wyrażenie_warunkowe_2
    instrukcje_2
else
    instrukcje_n
end
```

Polecenia **elseif** i **else** są opcjonalne.

- Pętla **for**

```
for zmienna_iterowana=start:krok:koniec
    instrukcje
end
```

Przykład

```
for i=1:1:5
    for j=1:1:5
        a(i,j)=sqrt((i-j)/(i+j))
    end
end
```

generuje macierz o rozmiarach 5x5 z elementami $a_{i,j} = \sqrt{\frac{i-j}{i+j}}$

- Pętla **while**

```
while wyrażenie_logiczne
    instrukcje
end
```

Przykład

```
while sin(krok*i)>krok
    tablica(i+1)=sin(krok*i)
    i=i+1
end
```

- Instrukcja **break** – powoduje przerwanie wykonywania pętli.
- Instrukcja **return** – powoduje opuszczenie funkcji lub skryptu i powrót do miejsca wywołania funkcji lub skryptu.

M-pliki

- **M-plik** – plik tekstowy, w którym umieszczone są sekwencje poleceń wykonywanych przez program lub zdefiniowane przez użytkownika funkcje.

Przykład

```
%% Komentarz
x=[0:0.01:2*pi];
y=sin(x);
plot(x,y);
```

Funkcje

W programie MatLab możliwe jest definiowanie funkcji napisanych przez użytkownika. Ogólna postać definicji funkcji:

function

```
[lista_argumentów_wyjściowych]=nazwa_funkcji(parametr1, ...,
parametrN)
ciąg_funkcji
```

- Polecenie **feval** – oblicza wartość funkcji o nazwie określonej łańcuchem.

```
feval(nazwa_funkcji, x1, x2, ..., xn)
```

gdzie:

nazwa_funkcji – łańcuch zawierający nazwę funkcji (może to być również funkcja zdefiniowana przez użytkownika)

x1, x2, ..., xn – argumenty funkcji

Przykład

Polecenie obliczające wartość funkcji **sinus** w przedziale $\langle -\pi, \pi \rangle$:

```
y=feval('sin',[-pi:0.01:pi])
```

Przykład

Funkcja zawierająca wzór na *n*-ty wyraz ciągu a_n :

```
function [a]=geom(n)
a=0.8.^n;
%%koniec definicji funkcji geom w pliku geom.m
```

Funkcja obliczająca sumę *n* wyrazów dowolnego ciągu:

```
function s=suma(N,an)
i=[1:N]
s=sum(feval(an, i));
%%koniec definicji funkcji suma w pliku suma.m
```

Nazwa definiowanej funkcji i m-plik, w którym ta funkcja jest zapisane muszą posiadać takie same nazwy. Na przykład definiując funkcję o nazwie **geom** należy zapisać ją w pliku **geom.m**

Wywołanie funkcji odbywa się przez podanie nazwy funkcji wraz z parametrami wejściowymi (o ile funkcja je posiada).

Przykład

Wywołanie funkcji obliczającej sumę wyrazów dowolnego ciągu:

```
S=suma(15, 'geom');
```

Bibliografia

- Zalewski A., Cegiela R., *Matlab – obliczenia numeryczne i ich zastosowania*, Nakom, Poznań 2003.
- Stachurski M., *Metody numeryczne w programie Matlab*, Mikom, Warszawa 2003.