

Aby nie trzeba było ręcznie wyznaczać obu macierzy Matlab oferuje funkcję, która oblicza obie macierze na podstawie wektorów określających przedział zmienności obu zmiennych.

```
x=0:1:8;  
y=15:2:25;  
[X Y]=meshgrid(x,y)
```

Jeśli współrzędne x i y mają ten sam przedział zmienności (ten sam zakres i ten sam skok) to wystarczy jako parametr podać tylko jeden wektor:

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

Współrzędne z oblicza się podstawiając do wzoru funkcji macierze X i Y. Na przykład:

```
Z=X.^2+2.*X.*Y+Y.^2
```

W ten sposób utworzona została trzecia macierz Z o tym samym wymiarze co macierze X oraz Y. Trójki punktów x y z, będących elementami macierzy X, Y i Z są współrzędnymi punktów w przestrzeni trójwymiarowej.

2. Wykresy trójwymiarowe.

Do tworzenia wykresów trójwymiarowych używa się przeznaczonych do tego funkcji o trzech parametrach wejściowych.

W następnych przykładach wykorzystana zostanie funkcja peaks(), która dobrze nadaje się do demonstracji działania funkcji do grafiki trójwymiarowej (jest to jedna z wbudowanych funkcji matlaba).

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
mesh(X,Y,Z);           %funkcja tworzy wykres w postaci siatki łączącej punkty  
  
surf(X,Y,Z);           %funkcja tworzy wykres w postaci powierzchni łączącej punkty
```

Każda z funkcji ma różne warianty:

```
meshc(X,Y,Z);          %to samo co mesh plus wykres poziomicowy (contour)  
meshz(X,Y,Z);          %to samo co mesh plus pionowe płaszczyzny na końcach wykresu  
surfc(X,Y,Z);          %to samo co surf plus wykres poziomicowy (contour)  
surfl(X,Y,Z);          % to samo co surf plus odbicia światła
```

Do tworzenia rzutowań wykresów na poszczególne płaszczyzny (XY, XZ, YZ) stosuje się funkcję contour()

```
contour(X,Y,Z)
```

Przykład:

```
[X,Y] = meshgrid(-3:.125:3);  
Z = peaks(X,Y);  
subplot(221),mesh(X,Y,Z),xlabel('X'),ylabel('Y'),zlabel('Z')  
subplot(222),contour(X,Y,Z),xlabel('X'),ylabel('Y'),zlabel('Z')  
subplot(223),contour(X,Z,Y),xlabel('X'),ylabel('Z'),zlabel('Y')
```

```
subplot(224),contour(Y,Z,X),xlabel('Y'),ylabel('Z'),zlabel('X')
```

Przykład 2:

```
x = linspace(-1,1,20); y = x;  
[X Y] = meshgrid(x,y);  
R =sin(X.^2 + Y.^2);  
Z = sin(3.*R);  
mesh(X,Y,Z);
```

3. Wyświetlanie komunikatów i wczytywanie danych z klawiatury

```
disp('jakis tekst')      %funkcja wyświetla słowa jakim tekst
```

```
x=input('komunikat')    %wyświetla komunikat i po wpisaniu wartości z klawiatury zapisuje ją jako x
```

Aby zamienić wartości z tekstowych na liczbowe lub odwrotnie należy użyć funkcji `num2str()` lub `str2num()`

```
x=25;  
tekst=['Temperatura otoczenia wynosi ',num2str(x)]
```

```
liczba=str2num('2')
```

Przykład:

```
disp('za chwilę zostaniesz poproszony o podanie liczby');  
x=input('podaj teraz liczbę');  
tekst=['podana liczba to ', num2str(x)]  
disp(tekst)
```

4. Instrukcje warunkowe

Składnia instrukcji warunkowych jest następująca:

```
if warunek1  
    instrukcje  
else if warunek2  
    instrukcje  
else  
    instrukcje  
end
```

Instrukcje złożone tworzy się korzystając z logicznej sumy (&) i logicznego iloczynu (|)

```
if x==0 & y>2
```

```
if x>2 | x<1
```

5. Pętle

```
for zmienna_sterująca = wyrażenie  
    instrukcje
```

```
end
```

```
while warunek
    instrukcje
end
```

Przykłady:

```
for i=1:10
    x(i)=i*i;
end
```

```
x=rand(20,1);
i=1;
while x(i)>0.2
    i=i+1;
end
```

W Matlabie nie ma pętli do ... while, zamiast tego można użyć:

```
while true                %warunek, który jest zawsze prawdziwy
    instrukcje
    if warunek==false; break; end
end
```

6. Instrukcja switch ... case

```
switch wyrażenie
case wartosc1
    instrukcje
case wartosc2
    instrukcje
.
.
.
case wartoscn
    instrukcje
otherwise
    instrukcje
end
```

Przykład:

```
x=input('Podaj liczbę z zakresu 1 do 3');
```

```
switch x
case 1
    disp('wpisałeś 1');
case 2
    disp('wpisałeś 2');
case 3
    disp('wpisałeś 3');
otherwise
```

```
    disp('wpisałeś liczbę spoza zakresu');  
end
```

7. Funkcje

Definiowanie funkcji:

```
function [y1, y2] = nazwa(x1, x2)  
.  
.  
.  
end    %wymagane, jeśli funkcje są zagnieżdżane, w przeciwnym przypadku opcjonalne
```

Funkcje należy zamieszczać w osobnym pliku o nazwie takiej samej, jak pierwsza z funkcji w tym pliku.

Wszystkie zmienne zdefiniowane wewnątrz funkcji mają zasięg lokalny, tzn. nie ma do nich dostępu poza funkcją, a funkcja posiada swoją własną przestrzeń roboczą.

Przykład:

plik fun.m

```
function [y]=fun(x)  
    disp('wartości zmiennych wewnątrz funkcji:');  
    y=2*x  
    a=7  
    b=4  
end
```

plik o dowolnej nazwie:

```
y=5;  
a=2;  
fun(4);  
disp('wartości zmiennych poza funkcją:');  
y  
a  
b    %zdefiniowane tylko wewnątrz funkcji, zatem będzie błąd
```

Jak widać zmienne o tej samej nazwie mają inne wartości wewnątrz i na zewnątrz funkcji.

Aby zmienna była widoczna także poza funkcją należy jej nadać charakter globalny (z takim podejściem należy uważać, ponieważ łatwo można doprowadzić do ciężkich do wykrycia błędów w funkcjonowaniu programu).

global zmienna

Przykład:

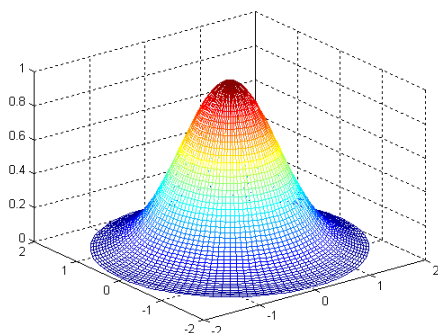
W poprzednim przykładzie proszę dopisać na początku drugiego pliku (bez funkcji):

global b

i zaobserwować wartości zmiennej b, która była definiowana jedynie wewnątrz funkcji.

III. Przykłady do samodzielnego rozwiązania:

1. Narysuj wykres funkcji $z = \exp(-x^2 - y^2)$ w dziedzinie będącej kołem a nie prostokątem. Wykres powinien wyglądać tak jak poniżej.



2. Napisz program, który pobierze od użytkownika trzy liczby: a, b, c a następnie rozwiąże równanie kwadratowe $a \cdot x^2 + b \cdot x + c = 0$.

3. Napisz program, który wykona następujące czynności:

- poprosi użytkownika o podanie zakresu zmiennej wektora x oraz liczby jego elementów, a następnie wczyta te wartości i utworzy wektor x,
- poprosi użytkownika o podanie częstotliwości, amplitudy oraz fazy,
- poprosi użytkownika o wybór koloru wykresu spośród czterech możliwości (niebieski, zielony, czerwony, czarny).
- obliczy w punktach wektora x wartość funkcji sinus o wczytanych parametrach i narysuje wykres tej funkcji we wskazanym kolorze.

4. Napisz program, który prosi o podanie liczby, a następnie obliczy sumę od 1 do wprowadzonej liczby (i wyświetla tę sumę).

5. Napisz program, który wczytuje liczby z klawiatury i oblicza średnią z wprowadzonych dotychczas liczb (po wczytaniu każdej nowej liczby wyświetlana jest średnia). Wczytywanie ma następować do momentu, aż nie zostanie podana liczba 0.

6. Proszę napisać funkcję, która jako parametr przyjmuje macierz oraz numer wiersza macierzy a zwraca sumę elementów macierzy z wiersza o wczytanym numerze.

7. Proszę napisać funkcję, która obliczać będzie silnię z podanej liczby.

8. Proszę napisać program, który będzie zawierał funkcje przeliczające jednostki: metry $\leftrightarrow$ cale, kilogramy $\leftrightarrow$ funty, a następnie w drugim pliku utworzyć interfejs, który zapyta użytkownika, jakiego przeliczenia chce dokonać i poprosi o podanie wartości jednostek.