

1 Podstawowe operacje

Pakiet MATLAB umożliwia obliczenia numeryczne i symboliczne, zarówno w sposób interaktywny, jak i skryptowy. W poniższych przykładach przedstawione są najprostsze jego zastosowania:

```
>>4*5+3
ans = 23
>>sqrt(9)
ans = 3
>>tan(pi/4)
ans = 1
```

Pakiet Matlab nie wymaga deklarowania zmiennych:

```
>>a=4*5+3
a = 23
>>b=sqrt(a)
b = 4.7958
>>c='ala ma kota'
c = ala ma kota
```

Postawienie średnika na końcu wiersza powoduje, że wprowadzona zmienna nie jest wyświetlana na ekranie:

```
>>a=4*5+3;
>>b=sqrt(a);
>>c='ala ma kota';
```

Ogromną zaletą Pakietu jest łatwe i szybkie definiowanie tablic (również wielowymiarowych). Poniżej przedstawiono, jak do zmiennej "v" przypisać można wektor pięcioelementowy:

```
>>v=[ 3 , 5 , 7 , 9 , 11 ];
```

Identyczny wektor zdefiniować można używając innego polecenia:

```
>>v=3:2:11;
```

Bardzo wygodne jest też odwoływanie się do wybranych elementów wektora, na przykład instrukcja:

```
>>v(2:4)=[ 1 , 0 , -1 ];
```

spowoduje, że do komórek 2,3,4 wektora "v" wpisane będą odpowiednio wartości 1,0,-1. Aby wyświetlić ostateczny wektor "v" wystarczy wpisać

```
>>v
v = 3 1 0 -1 11
```

Tablice dwuwymiarowe (macierze) definiujemy rozdzielając kolejne wiersze średnikami:

```
>>A=[ 1 , 2 , 3 ; 4 , 5 , 6 ; 7 , 8 , 9 ];
```

Odwołania do poszczególnych elementów są intuicyjne - polecenie

```
>>A(2,3)= -5 ;
```

oznacza, że elementowi macierzy "A" znajdującemu się w drugim wierszu i trzeciej kolumnie przypisana będzie wartość -5.

Niewątpliwie jedną z najważniejszych zalet pakietu są szybkie operacje macierzowe. Przyjmijmy, że

```
>>B=[ 9 , 8 , 7 ; 6 , 5 , 4 ; 3 , 2 , 1 ];
```

wówczas operacje takie, jak suma, różnica, a nawet iloczyn macierzy "A" i "B" mogą być wyliczone przy użyciu pojedynczej instrukcji:

```
>>SUMA=A+B;
```

```
>>ROZNICA=A-B;
```

```
>>ILOCZYN=A*B;
```

W wielu przypadkach przydatne jest również mnożenie poelementowe macierzy (muszą one mieć te same rozmiary)

```
>>ILOCZYN_PE=A.*B;
```

Transpozycję macierzy uzyskuje się poprzez dodanie na końcu macierzy apostrofu. Po wykonaniu poniższej instrukcji

```
>>B=A' ;
```

macierz "B" będzie równa transpozycji macierzy "A".

W kolejnym przykładzie wyliczona jest norma ("a") i wyznacznik ("b") macierzy "A":

```
>>a=norm(A);
```

```
>>b=det(A);
```

W celu sprawdzenia jakie zmienne znajdują się obecnie w pamięci możemy użyć polecenia "who":

```
>>who
```

```
Your variables are:
```

```
A B ILOCZYN ILOCZYN_PE ROZNICA SUMA a b c v
```

Usunięcie (wyczyszczenie) zmiennej następuje po wykonaniu instrukcji "clear" wraz z nazwą zmiennej. W celu usunięcia zmiennej "b" należy zatem zastosować polecenie:

```
>>clear b
```

Istnieje polecenie umożliwiające usunięcie wszystkich zmiennych znajdujących się w pamięci:

```
>>clear all
```

Aby dowiedzieć się więcej o danym poleceniu możemy skorzystać z instrukcji "help" wraz z tym poleceniem. Na przykład instrukcja:

```
>>help clc
```

spowoduje wyświetlenie na ekranie informacji o poleceniu "clc".

Ćwiczenie 1: Sprawdź co oznaczają instrukcje "format", "length" oraz "size".

Ćwiczenie 2: Przeanalizuj różnice w operacjach:

```
>>A^2
```

oraz

```
>>A.^2
```

gdzie "A" jest macierzą zdefiniowaną powyżej.

2 Wizualizacja

Rysowanie wykresów funkcji jednej zmiennej realizuje się poleceniem "plot":

```
>>x=-5:0.1:5;  
>>y=sin(x);  
>>plot(x,y);
```

Wykresy trójwymiarowe (dla funkcji dwóch zmiennych) tworzyć można w oparciu o polecenia "mesh", "surf", "waterfall" i inne. W poniższym przykładzie rysowany jest wykres funkcji $f(x, y) = 4 - x^2 - y^2$ w obszarze $D = \{(x, y) \in \mathbb{R}^2 : -3 < x < 3 \wedge -2 < y < 4\}$

```
>>[X,Y]=meshgrid(-3:0.1:3,-2:0.1:4);  
>>Z=4-X.^2-Y.^2;  
>>mesh(X,Y,Z);
```

Polecenie "meshgrid" pozwala wygenerować dwie macierze "X" i "Y" zawierające współrzędne "x" i "y" punktów, w których liczone będą wartości funkcji (zebrane w macierzy Z).

Ćwiczenie 3: Sprawdź co oznaczają instrukcje "grid", "title", "xlabel", "ylabel", "legend", "clf" oraz "hold".

Ćwiczenie 4: Sprawdź co oznaczają instrukcje "meshc", "meshz" oraz "shading".

3 M-pliki, czyli tworzenie skryptów

W przypadku złożonych obliczeń bardzo wygodnie jest stworzyć plik (z rozszerzeniem "m" - np.: pierwszy_program.m), w którym zamieścić można cały zestaw poleceń w kolejnych liniach (ewentualnie rozdzielając polecenia przecinkami lub średnikami). Do stworzenia i edycji pliku można zastosować edytora, w który wyposażony jest pakiet Matlab. W celu jego uruchomienia należy wprowadzić polecenie:

```
>>edit pierwszy_program.m
```

Po ewentualnym ustawieniu odpowiednich ścieżek, uruchomienie skryptu wymaga wpisania w linii komend nazwy pliku (bez rozszerzenia):

```
>>pierwszy_program
```

Przerwanie wykonywania skryptu możemy wywołać poprzez wciśnięcie "Ctrl + C".

Ćwiczenie 5: Stwórz plik o nazwie "rysuj.m", w którym zawarte będą instrukcje mające na celu wykreślenie funkcji dwóch zmiennych opisanej powyżej.

Ćwiczenie 6: Sprawdź co oznaczają instrukcje "input" oraz "disp".

4 Instrukcje warunkowe

Instrukcje warunkowe wygodnie będzie przedstawić w oparciu o przykładowy skrypt, obliczający pierwiastki rzeczywiste trójkątnu kwadratowego:

```
a = 1; b = 3; c = -4;
delta = b^2 - 4*a*c;
if delta > 0
    x = [ (-b-sqrt(delta))/(2*a) , (-b+sqrt(delta))/(2*a) ];
elseif delta == 0
    x = [ -b/(2*a) ];
else
    x = [];
end
disp(x);
```

Zastosowanie słów "elseif" oraz "else" jest opcjonalne, poprawnie zapisana instrukcja warunkowa musi zawierać jedynie zwroty "if" oraz "end".

Ćwiczenie 7: Sprawdź co oznacza instrukcja "switch".

5 Pętle

Również w przypadku pętli wygodnie będzie przedstawić ich konstrukcje w oparciu o prosty skrypt, wyznaczający silnię wybranej liczby naturalnej "n"

```
n=5;
silnia=1;
for i=1:n
    silnia=silnia*i;
end
disp(silnia);
```

Należy tu pamiętać, że zmienna "i" po zakończeniu pętli będzie miała wartość równą "n".

Drugim typem pętli jest pętla "while". Jej zastosowanie ilustruje poniższy przykład, pozwalający na wyznaczenie precyzji obliczeń

```
epsilon=1;
while ( epsilon+1 > 1)
    epsilon=epsilon/2;
end
epsilon=epsilon*2;
disp(epsilon);
```

Warto tu zaznaczyć, że w pakiecie Matlab zdefiniowana jest stała "eps" opisująca dokładność obliczeń (podobnie jak np.: "pi").

Ćwiczenie 8: Sprawdź co oznaczają instrukcje "break" oraz "continue".

6 Funkcje

Funkcje wygodnie jest tworzyć w oddzielnych plikach o nazwie takiej samej jak nazwa funkcji (również z rozszerzeniem "m"). Prosty przykład funkcji przedstawiony jest poniżej. Funkcja obliczająca sumę i iloczyn dwóch liczb zapisana jest w pliku "sumilo.m":

```
function [ suma , iloczyn ] = sumilo ( x , y)
suma = x + y ;
iloczyn= x * y ;
```

Wywołanie funkcji w programie (lub z linii komend) ma postać:

```
a=3; b=5;
[ s , i ] = sumilo ( a , b);
disp(s);
disp(i);
```

Zmienne przekazywane do funkcji są kopiowane, zatem operacje wykonywane wewnątrz funkcji nie wpływają na wartości zmiennych w programie głównym.

Ćwiczenie 9: Sprawdź co oznacza instrukcja "global".

Ćwiczenie 10: Zapoznaj się funkcjami "min", "max" oraz "sort".

7 Obliczenia symboliczne

Do obliczeń symbolicznych niezbędne jest zdefiniowanie zmiennych symbolicznych:

```
>>syms a b c x y
```

Wykonanie powyższego polecenia nadaje zmiennym "a", "b", "c", "x", "y" typ symboliczny.

Do rozwiązywania równań (lub układów równań) stosuje się instrukcję "solve". Na przykład równanie kwadratowe można rozwiązać w następujący sposób:

```
>> solve('a*x^2+b*x+c')
ans =
 1/2/a*(-b+(b^2-4*a*c)^(1/2))
 1/2/a*(-b-(b^2-4*a*c)^(1/2))
```

Pakiet umożliwia wykonywanie różniczkowania funkcji (również wielu zmiennych)

```
>> diff('x^3*y^2-x*y',x)
ans =
 3*x^2*y^2-y
```

lub

```
>> diff('x^3*y^2-x*y',y)
ans =
 2*x^3*y-x
```

oraz całkowania, zarówno w przypadku nieoznaczonym

```
>> int('sin(x)')
ans =
    -cos(x)
```

jak i oznaczonym

```
>> int('sin(x)',0,pi/2)
ans =
    1
```

gdzie 0 i pi/2 są granicami całkowania.

Ćwiczenie 11: Sprawdź co oznaczają instrukcje "subs", "limit", "pretty", "simple" oraz "vpa".

Zadania

Zadanie 1

Napisz funkcję obliczającą wartość dwumianu Newtona:

$$\binom{n}{k} = \frac{n!}{(n-k)!k!}$$

dla zadanych liczb naturalnych "n" i "k".

Wskazówka: Można wykorzystać przedstawiony wcześniej fragment kodu liczący silnię.

Zadanie 2

Porównaj czas potrzebny na wykonanie iloczynu macierzowego w sposób bezpośredni (potrójna pętla z operacjami na elementach macierzy), z czasem w jakim operacja ta wykonywana jest automatycznie, na poziomie całej macierzy. Niech wymiar macierzy będzie równy 1000.

Wskazówka: Instrukcje "tic" i "toc" pozwalają określić łączny czas (w sekundach) operacji wykonywanych pomiędzy ich wywołaniami, na przykład formuła:

```
>>tic; v=1:0.0001:1000; czas=toc;
```

spowoduje, że do zmiennej "v" przypisany będzie odpowiedni wektor, a zmiennej "czas" przypisany będzie czas wykonania tej operacji.