

## Ćwiczenia 3a: Typy danych Notatki

### Typy danych

Do tej pory omawialiśmy podstawowe typy danych MATLABa:

liczby rzeczywiste, wektory i macierze

Oprócz nich spotkamy także

wartości logiczne: true - 1 - nie zero; false - 0

[liczby całkowite, których nie omawiam]

[liczby zespolone, których nie omawiam]

łańcuchy (strings)

macierze komórkowe (cell arrays)

struktury (structures)

### Operatory relacyjne i logiczne

<	mniej niż
>	więcej niż
<=	mniejszy lub równy
>=	większy lub równy
==	równy
~=	nie równy

Uwaga: operator = to operator przypisania:

A = B

przypisuje zmiennej A zawartość zmiennej B;

A == B

daje macierz wartości logicznych

#### Przykład:

A = [1 2; 3 4]

B = [1 2 3]

A == B

A = B

A == B

A = [1 2; 3 4]

B = [1 4; 3 5]

A == B

A <= B

A > B

A = B

A == B

### Operatory logiczne

&	i
	lub
~	nie

**Łańcuchy**

```
S = 'Wprowadz liczbe: '  
disp(S)
```

Specjalne funkcje do obsługi łańcuchów:  
doc strings

**Macierze komórkowe**

Macierze komórkowe to uporządkowane zbiory dowolnych innych obiektów. Do ich tworzenia wykorzystujemy nawiasy klamrowe { } lub komendę cell.

Przykład:

```
A = { [1 2 3], 17, 'krowa'}  
A{1}  
A{2}  
A{3}
```

```
B = cell(3,2)  
B{2,1} = [1 2 3]  
B{2,1}
```

Przykład zastosowania: rejestrujemy 50 odpowiedzi komórki na ten sam bodziec. Za każdym razem komórka generuje inną liczbę iglic. Przechowujemy informację o czasach iglic w macierzy komórkowej, gdzie n-ta komórka jest zwykłym wektorem zawierającym czasy iglic w n-tym powtórzeniu.

**Struktury**

Struktury to obiekty składające się z wielu części, każda dowolnego typu, każda o własnej nazwie. Przykład:

```
kontakt.imie = 'Daniel'  
kontakt.nazwisko = 'Wojcik'  
kontakt.telefon = '0 22 5892 424'  
kontakt
```

```
kontakt(2).imie = 'Jan'      % tak tworzymy macierz struktur
```

**Przykład zastosowania:**

Pliki z danymi doświadczalnymi są często strukturami. Np. pliki ze Spike'a firmy CED eksportowane do Matlaba mają taką strukturę.

*Zobaczyć przykładowy plik z danymi Marka Hunta*

## Ćwiczenia 3b: Grafika Notatki

### Tworzenie wykresów

Generacja danych (np  $x = 0:100$ ; itp)  
 Tworzenie wykresu (np. plot, itp)  
 Plot tools i adnotacje  
 Eksploracja danych  
 Nagrywanie, eksportowanie i drukowanie rysunków  
 Skrypty generujące rysunek  
 Opcje zmieniające generowany wykres.

Zacznijmy od prostego wykresu:

```
x = 0:1:2*pi;
y = sin(x);
plot(x,y)
```

### Przykład:

```
plot(x,y,'r-')
```

łączy punkty czerwoną linią -.

Inne kolory:

```
'c', 'm', 'y', 'r', 'g', 'b', 'w', 'k',
cyan, magenta, yellow, red, green, blue, white, black.
```

Dostępne style linii:

```
solid '-',
dashed '--',
dotted ':',
dash-dotted '-.'
```

Punkty możemy zaznaczać

kropkami '.', kółkami 'o', plusami '+', krzyżykami 'x', albo gwiazdkami '\*'  
 i innymi (help plot).

Porównaj:

```
plot(x,y,'bx')
plot(x,y,'b:x')
```

Kolory, symbole i liczby można kombinować, np. 'r-', 'rx-' albo 'rx:'.

Opcje zmieniające generowany wykres.

b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line

y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

**Rysowanie wielu linii.**

Porównaj:

```
plot(x,y,x,2*y)
plot(x,y,x,2*y,'--')
```

**Dodawanie wykresów**

```
plot(x,y)
hold on
plot(5*x,5*y)
```

Matlab przeskalowuje osie, żeby zmieścił się kolejny wykres

```
plot(x,x)
```

Żeby cofnąć ostatnią zmianę piszemy

```
undo
```

Wyłączamy dodawanie wykresów pisząc

```
hold off
```

Samo hold przełącza między stanami on/off

**Rysowanie macierzy**Jeżeli jednym z argumentów komendy plot jest macierz, Matlab użyje **kolumn** macierzy do wykreślenia zbioru linii

```
>> q = [1 1 1;2 3 4;3 5 7;4 7 10]
```

```
q=
```

```
1 1 1
2 3 4
3 5 7
4 7 10
```

```
>> plot(q)
```

```
>> grid
```

**Przykład 2:**

```
>> x = [0 1 3 6];
```

```
>> plot(x,q)
```

```
>> grid
```

**Przykład 3:**

```
plot(q,x)
```

grid

**Przykład 4** (zarówno x jak i q są macierzami):

```
>> x = [[1 2 3 4]' [2 3 4 5]' [3 4 5 6]']
```

```
x=
```

```
 1  2  3
 2  3  4
 3  4  5
 4  5  6
```

```
>> plot(x,q)
```

```
>> grid
```

### Podstawowe funkcje generujace wykresy

loglog	Both axes logarithmic
semilogx	logarithmic x-axis
semilogy	logarithmic y-axis
box	Axis box for 2-D and 3-D plots
errorbar	Plot graph with error bars
polar	Polar coordinate plot
plotyy	Plot graphs with Y tick labels on the left and right

### Kilka przytecznych komend

shg	% show graphics
clf	% clear figure
cla	% clear axes
close	% zamyka bieżącą figurę
close all	% zamyka wszystkie figury

### Subplots – wykresy wielokrotne

Podziel okno na m wierszy, n kolumn i weź p-ty element. Licz elementy tak jak czytasz:

```
subplot(m,n,p)
t = 0:1:2*pi;
subplot(2,2,1)
plot(cos(t),sin(t))
subplot(2,2,2)
plot(cos(t),sin(2*t))
subplot(2,2,3)
plot(cos(t),sin(3*t))
subplot(2,2,4)
plot(cos(t),sin(4*t))
subplot(2,2,2)
cla
```

### Osie

axis([xmin xmax ymin ymax ])	% sets the axes' minimum and maximum values
axis square	%makes the axes the same length

axis equal	%makes the axes the same scale
axis tight	%sets the axes limits to the range of the data
axis auto	%allows Matlab to choose axes limits
axis off	%removes the axes leaving only the plotted data
axis on	%puts the axes back again
grid on	%draws dotted grid lines
grid off	%removes grid lines
grid	%toggles the grid

### Etykiety

Opisy rysunku:

xlabel('text ')	
ylabel('text ')	
zlabel('text ')	
title('text ')	
text(x,y,'text ')	%places text at position x,y
gtext('text ')	%use mouse to place text

```
t = 0:1:2*pi;
y1 = cos(t);
y2 = sin(t);
plot(t,y1,t,y2)
xlabel('0 \leq \theta < 2\pi')
ylabel('sin \theta, cos \theta')
text(1,cos(1),'cosinus')
text(3,sin(3),'sinus')
```

### Składowe wykresu

- Wykresy są wyświetlane na figurze (figure)
- Potrzebujemy współrzędnych, musimy więc stworzyć osie (axes) zawarte wewnątrz figury
- Wizualna reprezentacja danych zawarta jest w obiektach graficznych takich jak linie i powierzchnie.
- Obiekty te są rysowane w układzie współrzędnych określonym przez osie
- Właściwe dane są przechowywane jako parametry obiektów graficznych
- Do każdego obiektu jest rączka (handle), którą możemy manipulować zmieniając własności obiektu

### Zadanie

W doświadczeniu rejestrowano zewnątrzkomórkowy potencjał elektryczny w dwóch strukturach mózgu, hipokampie i *Nucleus Accumbens*. Załaduj dane znajdujące się w załączonym pliku. Każdej elektrodzie odpowiada jedna zmienna o typie struktury. Pole 'title' zawiera nazwę struktury, gdzie umieszczono elektrodę, pole 'values' zawiera wartości pomiarów pola. Napisz skrypt rysujący na tym samym obrazku 6 wykresów potencjału, oddzielnych dla każdej elektrody (*Wskazówka*: użyj funkcji **subplot**). Każdy wykres powinien być narysowany innym rodzajem krzywej (kolor, rodzaj, rodzaj punktów, itp.). Opisz osie, nadaj tytuł rysunkowi i każdemu wykresowi zgodnie z nazwą kanału. Rejestracje w hipokampie posortuj zgodnie z numerami w nazwach kanałów (title).