

Matlab-Kochrezept

Variablen

Grundsätzlich werden in Matlab keine Variablen deklariert, der Speicherbedarf wird „on the fly“ alloziert. Das Gleiche gilt auch für Felder. Der default Datentyp ist *double*. Es existieren jedoch auch die Typen *int8*, *int16*, *int32*, *uint8*, *uint16* und *uint32*. ACHTUNG es sind auf diese Typen nicht alle Rechenoperationen definiert!

Gültigkeitsbereich

Grundsätzlich sind Variablen nur in ihrem Workspace sichtbar, d.h. entweder auf oberster Kommandozeilen Ebene oder innerhalb der Funktionen, in der sie verwendet werden. Mit der Direktive *global* kann eine Variable auch in mehreren Modulen (Funktionen) sichtbar sein.

Beispiele für Variablen und Zuweisung:

| | |
|---------------------------|---|
| a=1; | |
| b=a; | |
| c=[1 2 3 4 5]; | explizite Zuweisung an einen Zeilen-Vektor |
| c(3)=16; | indizierte Zuweisung |
| d=[1 2 3; 4 5 6; 7 8 9]; | zweidimensionales Array |
| result=c(1)+d(2,3); | Indizierung |
| a=zeros(5,10); | Initialisiert ein Array mit 5 Zeilen und 10 Spalten |

Vektoren und Felder

ACHTUNG: Arrays werden von 1 beginnend indiziert!

Der : „Colon-Operator“ definiert eine Folge von Zahlen. Die Syntax lautet:

low:step:high

die Schrittweite *step* ist optional.

1:10 erzeugt die Zahlenfolge 1 2 3 4 5 6 7 8 9 10

Beispiele:

| | |
|------------|--|
| a=[1:100]; | Belegt die Komponenten des Vektors mit den Zahlen 1 bis 100. |
|------------|--|

a(30:45)=1;

b=zeros(100,1000);

b(:,512)=1;

Belegt die 512. Spalte des Feldes mit 1.

Memo: zeros, (), [], :

Visualisierung von Daten-Vektoren

Zur Visualisierung von eindimensionalen Daten werden die Befehle Plot und bar verwendet. Die Syntax für plot lautet:

plot(a, 'format')

Der Vektor **a** wird entsprechend dem String **format** visualisiert, der aus bis zu drei Zeichen bestehen kann:

Matlab HELP:

Various line types, plot symbols and colors may be obtained with PLOT(X,Y,S) where S is a character string made from one element from any or all the following 3 columns:

| | | | | | |
|---|---------|---|------------------|----|---------|
| b | blue | . | point | - | solid |
| g | green | o | circle | : | dotted |
| r | red | x | x-mark | -. | dashdot |
| c | cyan | + | plus | -- | dashed |
| m | magenta | * | star | | |
| y | yellow | s | square | | |
| k | black | d | diamond | | |
| | | v | triangle (down) | | |
| | | ^ | triangle (up) | | |
| | | < | triangle (left) | | |
| | | > | triangle (right) | | |
| | | p | pentagram | | |
| | | h | hexagram | | |

Folgende Befehle werden zur Beschriftung des Graphen verwendet:
xlabel(), *ylabel()*, *title()*, *legend()*.

grid on|off schaltet ein Gitter zur besseren Lesbarkeit der Daten an. Mit *hold on|off* können mehrere Datensätze simultan dargestellt werden.

Darstellung zweidimensionaler Daten

Eine indizierte Faschfarbendarstellung einer Matrix kann durch den Befehl *images()* (= image scale) ermöglicht. Dabei werden die Pixel mit den 64 aus der Farbtabelle verfügbaren Farben dargestellt. Die Werte zwischen dem minimalen und maximalen Intensitätswert werden auf die 64 Farben aufgeteilt.

Beispiel:

```
a=imread('face.tif');  
imagesc(a);  
axis image;  
colormap(gray)  
colorbar  
title('This is a face');
```



Eine Reliefdarstellung wird durch den Befehl *mesh()* erzeugt.

Aufgaben:

- Lesen sie das File *DNAchip.tif* mit *imread()* ein.
- Stellen sie das Array mittels *imagesc()* und *mesh()* dar. ACHTUNG der Befehl *mesh* arbeitet nur mit input-Daten des Typs *double*. Führen sie vorher eine Typ-Konversion durch.
- Stellen sie simultan eine Zeile und eine Spalte des Arrays dar, beschriften sie die Graphik.
- Wählen sie den linken oberen Sektor aus (8x4 Spots). Dazu speichern sie diesen Teil der Matrix in einer Sub-Matrix ab.
- Darstellung mittels *imagesc()* und *mesh()*.
- Stellen sie ein Intensitätsprofil jeder Spalte dar, dazu wird die Spalte mit der höchsten Intensität ausgewählt.

Funktionen: *mesh()*, *plot()*, *imagesc()*, *clear*, *close*, *figure*, *hold*, *imread()*