

Matlab–Grundlagen

erstellt für „BerP“ im WiSe 09/10

Jan Käser & Herbert Baaser

23. Januar 2010

Inhaltsverzeichnis

1	Bedienung der Konsole	3
2	Vektoren, Matrizen	5
3	Schleifen	7
4	Matlab-Dateien	9

1 Bedienung der Konsole

`>> %`

Kommentar. Alles, was in dieser Zeile steht, wird von Matlab nicht interpretiert.

`>> ;`

das ausgewertete Ergebnis wird nicht angezeigt

`>> =`

Zuweisungsoperator

`>> help [Befehl]`

Hilfe zu einem Befehl

`>> who`

Zeigt die aktuellen Variablen an

`>> clear X`

löscht die Variable X

`>> clear`

löscht alle Variablen

`>> clear all`

löscht alles: Variablen, Funktionen usw.

`>> clc`

löscht das Eingabe-Fenster

`>> save sitzung`

speichert alle Variablen in eine Datei mit dem Namen „sitzzung“ mit der Endung .mat

`>> load sitzung`

lädt die gespeicherten Variablen der Datei sitzung.mat in den Workspace

`>> quit`

Beendet die Matlabsitzung

2 Vektoren, Matrizen

```
>> n=3
```

Eingabe eines Skalars

```
>> X=[1 2 3 4 5]
```

Eingabe eines Zeilenvektors, Werte werden durch Leerzeichen oder Koma getrennt.

```
>> X=1:5
```

```
<< x= 1 2 3 4 5
```

Erzeugt denselben Zeilenvektor, von 1 bis 5 mit der Schrittweite 1.

```
>> X=1:2:10
```

```
<< x= 1 3 5 7 9
```

Erzeugt einen Zeilenvektor, von 1 bis 10 mit der Schrittweite 2.

```
>> X=[1 2 3 4 5;6 7 8 9 10]
```

Eingabe einer 2 x 5 Matrix, Zeilen werden durch Semikolon getrennt.

```
>> X=zeros(3,2)
```

Erzeugt eine Nullmatrix der Größe 3 x 2.

```
>> X^2
```

Gibt das Quadrat der Matrix X aus.

```
>> X'
```

Die Transponierte der Matrix X.

```
>> C+X
```

Summe der Matrizen C und X.

```
>> C*X
```

Produkt von C und X. Wenn C eine 3x3-Matrix ist, kann X eine 3x1-Matrix (Vektormultiplikation), eine 3x3 Matrix (Matrixmultiplikation) oder eine 1x1-Matrix (elementweise Multiplikation) sein

Diese Operationen können auch komponentenweise ausgeführt werden.

```
>> C.*X  
>> C.^2
```

```
A(1,:)
```

Gibt die erste Zeile der Matrix A aus.

```
A(:,1)
```

Gibt die erste Spalte der Matrix A aus.

Die Werte in einer Matrix sind fortlaufend spaltenweise nummeriert, d.h. bei einer Matrix der Größe 3 x 3 gibt der Befehl

```
>> A(2)
```

den Wert aus, der in der ersten Spalte und zweiten Zeile steht.

3 Schleifen

for Schleife

```
for i=anfang:inkrement:ende
...
end
```

Beispiel:

```
for i=1:0.2:2
    2*i
end
```

Ausgabe:

```
2.0 2.4 2.8 3.2 3.6 4
```

Die Variable *i* erhöht sich bei jedem Schritt um den Wert des *inkrement*, hier 0.2, bis der Endwert erreicht wird.

while Schleife

```
while (bedingung)
    Anweisung
end
```

Führt die Anweisung aus, solange die Bedingungen erfüllt sind.

if Bedingungen

```
if (bedingung1)
    Anweisung1
elseif (bedingung2)
    Anweisung2
else
    Anweisung3
end
```

Wenn bedingung1 wahr ist wird Anweisung1 ausgeführt, wenn nicht, wird geprüft, ob bedingung2 wahr ist. Ist keine Bedingung erfüllt, wird Anweisung3 ausgeführt. Dabei sind `elseif` und `else` optional.

Bedingungen

<code>(A == B)</code>	Überprüft, ob A gleich B
<code>(A < B)</code>	Überprüft, ob A kleiner B ist
<code>(A <= B)</code>	Überprüft, ob A kleiner oder gleich B ist
<code>(A > B)</code>	Überprüft, ob A größer B ist
<code>(A >= B)</code>	Überprüft, ob A größer gleich B ist
<code>(A != B)</code>	Überprüft, ob A nicht gleich B ist.
<code>(A B)</code>	Überprüft, ob die Bedingung A oder B erfüllt ist
<code>(A && B)</code>	Überprüft, ob Bedingung A und B erfüllt ist

break

Eine Schleife kann mit dem Befehl `break` unterbrochen werden. Dieser kann als Anweisung einer `if`-Bedingung in die Schleife eingebaut werden.

continue

Zwingt die aktive Schleife die Ausführung des Befehls der nächsten Iteration derselben Schleife zu überlassen.

return

`return` führt dazu, dass die ganze Funktion stoppt.

4 Matlab-Dateien

Scriptdatei

Matlab-Anweisungen können in einer Datei gespeichert werden. Diese Datei mit der Endung `.m` kann im *Prompt* aufgerufen werden. Alle Anweisungen werden nacheinander aufgerufen, genauso als würde man diese Anweisungen direkt eingeben.

Funktionsdatei

Mit einer Funktionsdatei ist es möglich, eine eigene Funktion zu definieren. Diese kann auch eine vorhandene Matlab-Funktion überlagern. Falls Matlab auf einen Funktionsaufruf $f(x)$ trifft, wird überprüft ob

1. f eine Variable ist
2. f eine Funktion im aktuellen Verzeichnis ist
3. f eine Funktion im Matlab-Pfad ist

Die erste Zeile der Datei Funktionsname.m muss eine der folgenden Formen haben:

```
function Funktionsname(input1, input2, ... ,inputn)
oder
function output1=Funktionsname(input1, input2, ... ,inputn)
oder
function [output1, ... , outputn]=Funktionsname(input1, ... ,inputn)
```

Beispiel:
`summe.m`

```
function output=summe(x, y)
output=x+y
```

Aufruf im Prompt:
`>> summe(3 5)`

Ausgabe:

<< 8

In einer Funktion definierte Variablen sind lokal, d.h. sie sind nur innerhalb einer Funktion gültig.