

## Beilage zur Serie 6

### MATLAB (MATrix LABoratory)

Matlab ist ein Programm, das nicht für symbolische (wie z.B. in Maple), sondern primär für die numerische Lösung von Problemen verwendet wird. Es wurde Ende der 1970er Jahre von Cleve Moler an der Universität New Mexico entwickelt. Mitte der 1980er Jahre gründete Moler zusammen mit Jack Little und Steve Bangert *The MathWorks* und machte Matlab zu einem kommerziellen Produkt, welches sich seither stetig weiterentwickelt. Wie bereits der Name suggeriert sind die Objekte, mit denen Matlab arbeitet Matrizen. Vektoren und Skalare sind dabei Matrizen mit den speziellen Dimensionen  $1 \times n$ ,  $m \times 1$  oder  $1 \times 1$ . Matlab bietet eine praktische graphische Desktopumgebung, auf welcher man einerseits verschiedene wichtige Dinge wie Code, Plots und Variablen im Überblick behalten und andererseits wie gewohnt per Mausklick und Tastenkombinationen arbeiten kann. In Matlab kann man einerseits (wie in Maple) im Commandwindow arbeiten, oder aber man kann sich sogenannte Skripte und Funktionen erstellen, in denen man den Code bearbeiten und ihn danach ausführen kann.

### Verfügbarkeit und Verwendung von MATLAB an der Universität Basel

Studierende der Universität Basel erhalten über `asknet.unibas.ch` Semesterlizenzen. Des weiteren gibt es auch Open Source-Alternativen wie Octave oder FreeMat. Im Mathematikstudium an der Uni Basel wird Matlab hauptsächlich für Vorlesungen zur Numerik im Bachelor und Master eingesetzt. Weitere nützliche Informationen zu Matlab findet man z.B. auf der Praktikumshomepage, der Universitäts- oder Institutsbibliothek, oder auf <http://www.mathworks.ch/>.

### Starten von MATLAB unter Linux

Tippe im Terminal `ml MATLAB` um die verfügbaren Softwarepakete zu laden. Starte danach MATAB mit dem Befehl `matlab`.

### Benutzung von MATLAB

Der Arbeitsablauf ist wie folgt:

- Starte Matlab.
- Öffne ein leeres m-file und gib ihm einen Namen.

- Schreibe die Befehle in das File.
- Führe das Skript aus. Klicke dazu auf den grünen Pfeil oben am Fenster oder tippe den Namen des Skripts in das Commandwindow ein.

## Matlab-Befehle

- Um die Ausgabe zu unterdrücken, schreibe ein Semikolon hinter jeden Befehl (;). Falls das Resultat gewünscht wird, lasse dieses weg.
- Um mehr Informationen zu einem Befehl zu erhalten, kannst du `help 'Befehl'` benutzen, oder dir mit `doc 'Befehl'` die Matlab-Dokumentation anschauen.
- Mit `clc` löscht man alle vorherigen Eingaben und Ausgaben im Commandwindow.
- Der Befehl `clear all` löscht alle alten Variablen.
- Kommentare: alles was hinter einem `%` steht, wird vom Compiler nicht gesehen.
- Mit `=` können wir Variablen definieren sowie damit rechnen

```
>> x = 1;
>> y = 1;
>> z = x + y
```

```
z =
```

```
2
```

- Vektoren und Matrizen definiert man wie folgt

```
>> v = [1, 2]
```

```
v =
```

```
1 2
```

```
>> w = [1; 2]
```

```
w =
```

```
1
```

```
2
```

```
>> A = [1, 2; 3, 4]
```

```
A =
```

```
1 2
```

```
3 4
```

- Es gibt auch noch eine kürzere Weise Vektoren zu definieren:  
`1:5` steht für den Vektor `[1 2 3 4 5]`  
`5:-1:1` für `[5 4 3 2 1]`  
`0:0.25:1` für `[0 0.25 0.5 0.75 1]` usw.  
 Diese Schreibweise erstellt immer Zeilenvektoren, die entsprechenden Spaltenvektoren erhält man durch Transposition mit `'`, beispielsweise `[1:5]'`.
- Mit Matrizen und Vektoren kann man auch rechnen. Dabei muss man darauf achten wie verschiedene Operationen auf die Strukturen wirken. Gewisse Operationen wirken elementweise, d.h. sie wirken so als würde man sie auf jedes Element einzeln anwenden. Die Operationen sind dann `.*` oder `.^` statt `*` und `^`. Man muss ebenfalls darauf achten, dass die Strukturen die korrekten Dimensionen haben, ansonsten ist die Operation nicht definiert.
- Einen Plot erstellt man in Matlab indem man erst einen Vektor `x` definiert, der die gewünschte Auflösung oder Feinheit der Abszissenachse hat (z.B. `x = -3:0.01:3;`) und einen Vektor `y` der entsprechend viele Funktionswerte enthält (z.B. `y = x.^2;`). Diese beiden Vektoren kann man anschliessend gegeneinander plotten mit `plot(x,y)`. Dies funktioniert nur dann, wenn die beiden Vektoren dieselbe Länge haben. In der Matlab-Hilfe findet man weitere Informationen dazu.