

Prof. Dr. Frank Noé  
Dr. Christoph Wehmeyer  
Tutoren:  
Katharina Colditz; Anna Dittus;  
Felix Mann; Christopher Pütz

# 0. Übung zur Vorlesung Computerorientierte Mathematik I

Abgabe: Keine Abgabe, Bearbeitung in den Matlab-Tutorien

<http://www.mi.fu-berlin.de/w/CompMolBio/ComaI>

## Aufgabe 0 (*Orientierung*):

Besuchen Sie die Veranstaltungshomepage unter

<http://www.mi.fu-berlin.de/w/CompMolBio/ComaI>

und informieren Sie sich über die Modalitäten der Vorlesung. Lesen Sie zunächst die Kapitel 2 - 4 der unter **Material zur Vorlesung** angegebenen *freundlichen* *MATLAB-Einführung* von J. Behrens und A. Iske. Alles, was dort zu Matlab gesagt wird, funktioniert auch mit dem freien GNU Octave, das in der Uni ebenfalls verfügbar ist und zudem kostenfrei über <http://www.gnu.org/software/octave/> bezogen werden kann.

## Aufgabe 1 (*Erste Schritte in Matlab*):

Sollten Sie noch keinen Fachbereichsaccount haben, lassen Sie sich einen einrichten. Sprechen Sie Ihren Tutor an, wenn Sie Hilfe brauchen.

1. Suchen Sie einen der PC-Räume auf (z.B. Raum 030 im  $\pi$ -Gebäude) und loggen Sie sich auf einem der Rechner ein, indem Sie Ihren Benutzernamen und Ihr Passwort eingeben (Fachbereichsaccount, nicht ZEDAT-Account).
2. Öffnen Sie ein Terminalfenster, indem Sie hier unter KDE (ähnlich bei anderen Fenstermanagern und Benutzeroberflächen) unten links auf das KDE-Logo klicken und über System die Schaltfläche Terminal anwählen.
3. Starten Sie MATLAB, indem Sie `matlab &` im Terminalfenster eingeben. OCTAVE starten sie entsprechend durch Eingabe von `octave`.
4. Geben Sie hinter dem Zeichen `»` nacheinander die folgenden Befehle ein:

```
7*8
a=7
b=8
a-b
a/b
c=a+b
c=a-b
c=a*b
c/a
a
```

5. Geben Sie diese drei Befehle ein:

```
x=[1 2 3];
x
x'
```

Was bewirkt der Apostroph (**Shift + #**) und was das Semikolon?

6. Geben Sie nun die folgenden Befehle ein:

```
y=17;
y
clear all
y
```

Was bewirkt der Befehl `clear all`? Überprüfen Sie Ihre Vermutung, indem Sie `help clear` eingeben.

7. Hilfe zu MATLAB-Befehlen erhalten Sie, indem Sie `help + Name des Befehls` im MATLAB-Kommandofenster eingeben. Geben Sie z.B. einmal `help cos` oder `help sin` ein.
8. Eine ausführlichere Hilfe erhalten Sie, wenn Sie `helpdesk` im MATLAB-Kommandofenster eingeben. Für OCTAVE besuchen Sie die Webseite <http://www.gnu.org/software/octave/doc/interpreter/index.html>.
9. Um herauszukriegen, welche Befehle MATLAB kennt, ist der Befehl `lookfor` sehr nützlich. Wenn Sie z.B. `lookfor logarithm` eingeben, erscheint eine Liste aller MATLAB-Befehle, die irgendwie mit dem Thema *Logarithmus* zusammenhängen. Wichtig: Both, MATLAB and OCTAVE, are native English speakers. Probieren Sie einmal aus, was passiert, wenn Sie einen deutschen Suchbegriff wie z.B. `lookfor Logarithmus` eingeben.

## Aufgabe 2 (Vektoren in Matlab):

1. Benutzen Sie den MATLAB-Editor (oder einen anderen Editor Ihrer Wahl), um Ihr erstes MATLAB-Programm zu erstellen. Klicken Sie hierzu im MATLAB-Fenster auf das weiße Blatt oben links.
2. Schreiben Sie folgende Matrix und folgenden Vektor in MATLAB-Notation.

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}, x = (1 \ 1 \ 1) \quad (0.1)$$

Speichern Sie Ihr erstelltes Programm unter dem Dateinamen `uebung0.m` in Ihrem Heimatverzeichnis und führen Sie es aus. Zum Ausführen geben Sie an der Eingabeaufforderung `test` ein, nachdem sie mittels `cd` in das Verzeichnis gewechselt sind, in dem sich Ihre Datei befindet. (Wenn Sie bereits in dem richtigen Verzeichnis sind, können Sie auch die Taste `F5` benutzen).

3. Im MATLAB-Workspace sollten jetzt die Objekte `A` und `x` existieren. Führen Sie nun folgende Befehle im MATLAB-Kommandofenster aus:

```
A*x'  
x*A  
A*x  
A*A  
A.*A
```

Warum liefert `x*A` ein anderes Ergebnis als `A*x'`? Wieso erzeugt `A*x` eine Fehlermeldung? Was ist wohl der Unterschied zwischen `A*A` und `A.*A` ?

4. Und es wird bunt. Geben Sie die folgenden Kommandos ein, am besten in eine eigene Datei (z.B. mit dem Namen `graph.m`):

```
x = -2:0.1:2;  
f = x.^3;  
plot(x,f)  
figure(2);  
clf;  
g=1/(1+x.^2);  
plot(x,g)
```

Versuchen Sie jede einzelne Zeile zu verstehen und verändern Sie das Programm so, dass der Funktionsgraph von `f` gepunktet dargestellt wird. Tipp: `help plot`.

Warum wird die Funktion `g` nicht geplottet? Finden Sie den Fehler.