

## Übungsblatt 1

Ausgabe: 16.10.2017

### Erste Schritte und Grundbefehle

Beim Start von Matlab werden vier Fenster angezeigt:

- **Command Window:** Hier werden die Befehle und Variablenzuweisungen eingegeben. Alle Anweisungen müssen Sie mit <return> bestätigen.
- **Current Folder:** Hier sieht man das aktuelle Verzeichnis.
- **Workspace:** Hier sind die momentan benutzten Variablen und Parameter einzusehen.
- **Command History:** Hier werden die ausgeführten Eingaben im Command Window aufgelistet. Durch Doppelklick auf einen dieser Befehle wird dieser erneut ausgeführt (Alternative auf Tastatur: ↑ und ↓).

Sie können alle Ihre Anweisungen im Command Window eingeben. Dabei werden die vier Grundrechenarten (+, −, \*, /) wie gewohnt verwendet. Bei der Multiplikation darf allerdings das Multiplikationszeichen \* nicht weggelassen werden. Potenzieren (z.B.  $a^2$ ) wird **vor** einer Multiplikation oder Division ausgewertet. Danach kommen Addition und Subtraktion. Durch Klammersetzung kann man diese Reihenfolge ändern.

Ein Semikolon am Ende der Eingabezeile bewirkt, dass Matlab das Ergebnis zwar auswertet, aber nicht auf dem Bildschirm ausgibt.

Braucht man einen Ausdruck wiederholt, so kann man ihn als Variable bzw. Konstante definieren. Matlab speichert den Wert dann automatisch im Workspace. Folgende Regeln sind bei der Definition von Variablen zu beachten:

- Ein Variablenname darf keine Sonderzeichen außer dem Unterstrich enthalten.
- Das erste Zeichen muss ein Buchstabe sein.
- Der Name darf nicht mehr als 19 Zeichen enthalten (der Rest wird abgeschnitten).

Variablen im Workspace können wieder gelöscht werden mit den Anweisungen:

```
clear a      % löscht die Variable a
clear       % löscht alle Variablen
```

Prozentzeichen nach einem eingegebenen Befehl dienen Kommentaren. Matlab ignoriert alles, was in der Zeile nach dem Prozentzeichen steht. Kommentare sind besonders für die später eingeführten eigenen m-Files von Bedeutung, da sie deren Lesbarkeit (vor allem für andere) deutlich verbessern.

### Aufgabe 1 (Rechnen und Umgang mit Variablen)

- Geben Sie nacheinander folgende Anweisungen ein. Überlegen Sie vorher, wie das Ergebnis lautet. Was gibt Matlab aus?  
 $a=3$ ,  $b=8$ ;  
 $(a+5)/4^3$  und  $a+5/4^3$   
 $y = 3*a$   
 $y = x+2$
- Welche der folgenden Variablennamen sind nicht zulässig und warum?  
 $anzahl$ ,  $Produkt_a*b$ ,  $3\_Monatskarte$ ,  $dauer\_test2$
- Löschen Sie die Variable  $a$ . Welche Fehlermeldung ergibt danach die Eingabe von  $y = 3*a$ ? Löschen Sie anschließend alle gespeicherten Variablen.
- Der Befehl `help` ist sehr hilfreich, wenn man Näheres zu einer Matlab-Anweisung erfahren möchte. Geben Sie den Befehl `help who` ein, der Ihnen nähere Informationen zu dem Kommando `who` gibt. (Eine Liste **aller** Hilfe-Themen erhält man durch Eingabe von `help`.)

### Formate und Funktionen

Es gibt in Matlab – wie auch in jedem Taschenrechner – bereits eingebaute Funktionen. Beispiele hierfür sind die Wurzelfunktion  $\sqrt{x}$ , trigonometrische Funktionen wie  $\sin(x)$  oder  $\cos(x)$  oder die Exponentialfunktion  $\exp(x)$ . Eine Liste aller eingebauten elementaren Funktionen mit kurzer Beschreibung erhält man durch `help elfun` (kurz für *Elementare Funktionen*). Wichtig ist dabei:

- Das Argument von trigonometrischen Funktionen wird von Matlab immer im Bogenmaß interpretiert.
- Matlab unterscheidet Groß- und Kleinschreibung. Daher gibt bspw. `Sin(x)` eine Fehlermeldung zurück.

Matlab gibt standardmäßig ein ganzzahliges Ergebnis als Zahl ohne Nachkommastellen und ein nicht-ganzzahliges Ergebnis als Dezimalzahl mit 4 Nachkommastellen aus. Man kann das Ausgabeformat der numerischen Ergebnisse ändern. Mehr Nachkommastellen erhält man durch Änderung des Formats von `short` auf `long`. Diese Änderung geschieht durch Eingabe von:

`format long;      % ändert das Ausgabeformat auf 15 Nachkommastellen`

### Aufgabe 2 (Erster Umgang mit Formaten)

Ändern Sie das Format auf `long` und bearbeiten Sie Aufgabe 1a erneut.

Sehr große oder kleine Zahlen werden automatisch in Gleitkommadarstellung ausgegeben. Eine Beschreibung der verschiedenen numerischen Formate erhalten Sie mit `help format`. Die verschiedenen Darstellungsformate von Maschinenzahlen (d.h. von Zahlen, wie sie im Rechner dargestellt werden), werden in der Vorlesung Algorithmische Mathematik eingeführt. Die Maschinengenauigkeit von Matlab ist `eps`. Die Variable `eps` gibt den Abstand der Zahl 1 zur nächst größeren (Gleitkomma-)Zahl an, die darstellbar ist (siehe ebenfalls AlgoMath-Vorlesung).

**Wichtig:** Bei Änderung des Ausgabeformats ändert sich nur die **Ausgabe**, nicht aber die interne Darstellung (und damit die Genauigkeit) der Zahlen.

### Aufgabe 3 (Umgang mit Formaten)

- Lassen Sie sich die Zahlen `pi` und `1/1000` in den Formaten `short`, `long`, `shorte` und `longe` ausgeben.
- Berechnen Sie den natürlichen Logarithmus von 1.37 auf 15 Nachkommastellen genau und lassen Sie sich den Kosinus von  $\pi/4$  in einfacher Gleitkommadarstellung (`shorte`) ausgeben.
- Finden Sie nur durch Wahl eines geeigneten Ausgabeformats eine möglichst einfache (approximative) Bruchdarstellung von 0.607489.
- Finden Sie ein geeignetes Ausgabeformat zur Berechnung von Geldbeträgen. Initialisieren Sie die Variable `a=0.238` und lassen Sie sich die Werte `a`, `2*a` und `3*a` ausgeben. Wie sind die Ergebnisse zu erklären?
- Berechnen Sie `1/exp(36)`, `1/exp(36)+1`, `1/exp(37)`, `1/exp(37)+1` und `eps`. Wie sind die unterschiedlichen Zahldarstellungen bei gleichbleibendem Format zu erklären?

Weitere Funktionen, die man im Command Window erklären und anwenden kann, sind Polynomfunktionen oder Komposita einfacher Funktionen, sogenannte *anonyme Funktionen*. Die Parabelfunktion  $f(x) = x^2$  kann in Matlab durch Eingabe von `f = @(x) x^2` definiert werden. Durch `@(x)` wird festgelegt, welche Variable in der Funktion `f` als Argument verwendet werden soll. Eine Indizierung über `f(x) = x^2` ist **nicht** möglich, da Matlab ohne das `@`-Argument unklar ist, ob `x` eine Variable oder eine bereits definierte Konstante darstellt. Zweidimensionale Funktionen können durch die Festlegung von mehreren Variablen im `@`-Argument erzeugt werden. So ist z.B. die Kreisfunktion  $g(x,y) = x^2 + y^2$  darstellbar durch:

`g = @(x,y) x^2 + y^2`

Analog erfolgt eine Erweiterung auf höherdimensionale Funktionen.

### Aufgabe 4 (Umgang mit anonymen Funktionen)

Berechnen Sie alle Ihre Ergebnisse auf vier Nachkommastellen.

- Initialisieren Sie die Funktion  $f(x) = x^2 + \sin(x^2 + \pi/2)$  und berechnen Sie die Werte  $f(0)$  und  $f(\pi)$ .
- Initialisieren Sie die Funktion  $h(x,y,z) = \ln(2x^2 - 3y^2 + 5z + e)$  und berechnen Sie die Werte  $h(1,2,3)$  und  $h(3,2,1)$ .

Es ist möglich, anonyme Funktionen auch abschnittsweise zu definieren. Für die Indikatorfunktion  $\chi_{[a,b]}$  lässt sich die Hut-Funktion  $h(x) = (x-2) \cdot \chi_{[2,3]} + (4-x) \cdot \chi_{(3,4]}$  z.B. darstellen durch:

`h = @(x) (x-2) * and(x>=2, x<=3) + (4-x) * and(x>3, x<=4)`

Der Befehl `and(x>=a, x<=b)` erlaubt dabei die Darstellung der Indikatorfunktion  $\chi_{[a,b]}$ . Die Operatoren „>“ und „>“ sind entsprechend für offene Intervalle zu verwenden.

### Aufgabe 5 (Umgang mit abschnittsweise definierten anonymen Funktionen)

Initialisieren Sie die abschnittsweise definierte Funktion  $h(x) = (x-1) \cdot \chi_{[1,3]} + (5-x) \cdot \chi_{(3,5]}$ . Verifizieren Sie Ihre Eingabe mit dem Befehl `ezplot(h, [0,6])`, der Ihre Funktion auf dem Intervall  $[0,6]$  plottet.

**Bemerkung:** Mit weiteren Möglichkeiten zur Visualisierung von Funktionen werden wir uns in den kommenden Wochen beschäftigen. Im Laufe der Präsenzübungen werden wir stets die für die Aufgaben benötigten Funktionsbefehle (oder die `help`-Befehle, um diese einzusehen) angeben, sofern diese bis dahin noch nicht eingeführt wurden.