

**Für alle Matlab-Novizen: Kurzer Leitfaden zum Selbststudium daheim**

- (a) Loggen Sie sich mit Ihrem ZID-Nutzernamen (an der KFU: Gruppe BZEDVZ) und Passwort am PC ein.
- (b) Starten Sie Matlab indem Sie bei *Programme/Dateien durchsuchen* den Suchstring *matlab* eingeben. und dann *Matlab2019* auswählen.
- (c) Schauen Sie sich das Einführungsvideo "Getting Started with MATLAB" via youtube<sup>1</sup> **daheim** an.
- (d) Laden Sie sich das **E-Book** „MATLAB 7 : Eine Einführung“ von C.Überhuber [Ueber05] als PDF herunter.  
Probieren Sie die Bsp. 2.1-2.7 aus den Kapiteln 2.1-2.3 des Buches in dem *Command Window* aus.
- (e) Legen Sie ein neues M-File via den Menüeintrag *Editor* → *New* → *Script* an. Kopieren Sie **einige** der in obiger Aufgabe benutzen Matlab-Kommandos (nicht die Ausgabe der Ergebnisse mitkopieren!!) in das sich öffnende *Editor*-Fenster.  
Speichern Sie dieses M-File (Achtung: in **eigenes** Verzeichnis speichern!!) unter dem Namen *try\_1.m*. Starten Sie die Abarbeitung der im File beschriebenen Kommandos über den Menüeintrag *Run*. Die Ausgabe wird im Kommandofenster getätigkt. Erhalten Sie die korrekten Ausgaben oder erscheint eine Fehlermeldung?
- (f) Ihr kleines Matlabskript werden Sie jetzt Debuggen (Fehler suchen, Variablenwerte kontrollieren):
  - Im Editorfenster: Platzieren Sie die Maus in Zeile 2 Ihres Skriptes und setzen Sie via Menüeintrag *Breakpoint* → *Set/Clear* einen Breakpoint (Unterbrechung ds Programmes).
  - Starten Sie die Ausführung Ihres Skriptes via Menüeintrag *Run*.
  - Mit dem Menüpunkt *Step* können Sie das Skript zeilenweise abarbeiten und im Fenster des Workspace die Änderungen der Variablenwerte beobachten.
  - Der Menüpunkt *Continue* führt das Skript bis zum nächsten Breakpoint oder bis zum Skriptende aus.
  - Der Menüpunkt *Quit Debugging* beendet das Debugging.

Erhalten Sie die korrekten Werte Ihrer Variablen?

- (g) Erschließen Sie selbständig den Gebrauch der Matlab-Funktionen *log* und *exp* durch Benutzung von *help* und *doc* (z.B. *doc exp*).
- (h) Schauen Sie sich einige Matlab-Demos an:  
*Home* → *Help* → *Examples* → *Graphics* → *2-D Plots*  
Öffnen Sie das zugehörige M-File im Editor durch einen Klick auf das entsprechende Link (*Open this Example*) in Kopfzeile der Demobeschreibung und führen Sie das File aus.  
Kopieren Sie zusammengehörende Teile des M-Files (= Cell) in das Kommandofenster und führen Sie diese Matlab-Kommandos dort aus.  
Führen Sie **einzelne Cells** aus dem Editor heraus aus indem Sie durch Anklicken einer Skriptzeile die Zelle aktivieren (Zelle wird gelb) und danach über *Run Section* dieses Teilskript abarbeiten lassen.

Weiter auf Seite 2.

## Literatur

- [Ueber05] C. Überhuber, S. Katzenbeisser and D. Praetorius. *MATLAB 7: Eine Einführung*. Springer-Verlag, Wien, 2005. E-book<sup>2</sup>.

<sup>1</sup><http://www.youtube.com/watch?v=tqjZ80PwqB0&feature=c4-overview-vl&list=PL7CAABC40B2825C8B>

<sup>2</sup><https://link.springer.com/book/10.1007%2Fb138723>

## Nun die ersten Aufgaben (Blatt 1 - Abgabe 11.12.2023):

### 1. Symbolisches Paket:

- (a) Gegeben ist die Funktion  $f(a, b, c) = 2(a^2 - b^3)(b - c^2)(e^{c-a} - 1)$ .  
Lösen Sie den Ausdruck  $f(a, b, c) == 0$  jeweils nach  $a$ ,  $b$  bzw.  $c$  mittels `solve` auf.
- (b) Lösen Sie  $x^4 - 9x^3 - 389x^2 - 639x + 9100 = 0$  symbolisch (`syms`, sowohl `solve` als auch `vpasolve` probieren).  
Zeichnen Sie die Funktion mit `ezplot` oder `fplot`.
- (c) Lösen Sie symbolisch die Gleichung  $ax^2 + bx + c = 0$ .  
Lösen Sie symbolisch die Gleichung  $x^2 - x - 6 = 0$ .
- (d) Bestimmen Sie symbolisch die Schnittpunkte der beiden Kreise (`solve`, `vpasolve`)

$$(x + 4)^2 + (y - 1)^2 - 49 = 0 \quad \text{Mittelpunkt } (-4, 1) \text{ und Radius } 7$$

$$(x - 3)^2 + (y + 1)^2 - 16 = 0 \quad \text{Mittelpunkt } (3, -1) \text{ und Radius } 4$$

und geben Sie diese Schnittpunkte als Gleitkommazahlen mittels `eval` aus.

### 2. Integration und Differentiation.

- (a) Berechnen Sie  $\int e^{-x}(x - 3) \cos(x) dx$ .
- (b) Berechnen Sie  $\int_{-2}^5 e^{-x}(x - 3) \cos(x) dx$ .
- (c) Berechnen Sie die erste und zweite Ableitung von
- $f(x) = \frac{e^x + a}{x}$  und von
  - $g(x) = \frac{x^2 + \tan(x)}{e^x + x} + 3x^2$ .
- (d) Nutzen Sie `limit` und berechnen Sie  $\lim_{x \rightarrow \infty} g''(x)$ .

Hinweise: `syms`, `int`, `diff`, `pretty`, `limit`, `char`, `fplot`

### 3. Die Funktion $f(x) = x^7 - 3x^5 + 2x^4 - 14x^3 + 9x - 5$ ist gegeben.

- (a) Berechnen Sie  $f'(x)$  und  $f''(x)$ . Visualisieren Sie  $f(x)$ ,  $f'(x)$  und  $f''(x)$  im Intervall  $[-3, 3]$ . Speichern Sie die Grafik in ein File `bsp_3.jpg`.
- (b) Bestimmen Sie alle reellen Nullstellen  $x_s$  von  $f(x)$  (`vpasolve` und `imag`).
- (c) Berechnen Sie die potentiellen Extremstellen  $x_E$ , d.h.  $f'(x_E) = 0$ . Sind diese Minima oder Maxima?

Hinweise: `diff`, `ezplot`, `plot`, `hold on, saveas`, `solve`, `vpasolve`, `eval`, `imag`

### 4. Überprüfen Sie, ob die folgenden Rechenregeln der Mathematik auch im Computer mit **nicht symbolischen** Variablen gelten, d.h. Deklaration und Definition via `a = 0.00187`. Berechnen Sie jeweils die Differenz aus beiden Seiten der Gleichung und geben Sie diese im Kommandofenster aus. Eine Auswahl math. Funktionen ist unter §5 Kernbichler<sup>3</sup> und in [Ueber05, §12.3] zu finden.

$$\begin{aligned} e^a e^b &\stackrel{?}{=} e^{a+b} \\ \log_2(a+b) &\stackrel{?}{=} \frac{\ln(a+b)}{\ln 2} \\ \cos(a-b) &\stackrel{?}{=} \cos(a) \cos(b) + \sin(a) \sin(b) \\ \arcsin a &\stackrel{?}{=} \text{arccsc } 1/a \\ \sinh a + 2 \ln a + \cosh a - \ln b &\stackrel{?}{=} e^a + \ln(\frac{a^2}{b}) \end{aligned}$$

<sup>3</sup><http://itp.tugraz.at/LV/kernbich/AppSoft-1/Kapitel/appsoft1-kapitel-5.pdf>

Testen Sie mit selbstgewählten Werten für  $a$  und  $b$  (Definitionsbereiche der math. Funktionen beachten!). Ein Vorschlag wäre  $a = 0.187$ ,  $b = 98$ , ein anderer Vorschlag  $a = 0.9876$ ,  $b = 1.762 \cdot 10^{-4}$  ( $= 1.762 \cdot 10^{-4} = 0.0001762$ ).

Ist eine Differenz ungleich Null immer gleichbedeutend mit einem Fehler in Ihrem Skript?  
(Stichworte dafür: absolute vs. relative Genauigkeit, Maschinen- $\varepsilon$ <sup>4</sup> bei Gleitpunktzahlen)

Hinweise: `exp, log, log10, pi, sin, asin, acot, sqrt, cosh, sinh`

5. Schreiben Sie eine Matlabfunktion `bsp_5_fkt` welche das arithmetische, das geometrische und das harmonische Mittel<sup>5</sup>, beim Aufruf zu fünf übergebender (INPUT) Zahlen berechnet und die drei Mittel an das aufrufende Programm zurückgibt (OUTPUT).

Testen Sie Ihre Funktion sowohl im Command Window interaktiv als auch aus einem Scriptfile `bsp_5.m` (Hauptfile) heraus, in welchem die Eingabedaten vor dem Aufruf der Funktion (z.B., `[a,g,h] = bsp_5_fkt(v,w,x,y,z)`) gesetzt werden und die Mittelwerte nach dem Aufruf durch das Hauptfile ausgegeben werden.

Daten: Testen Sie Ihre Funktion mit sinnvollen Daten, z.B. `[1,3,5,12,20]`.

(\*) Schreiben Sie Ihre Funktion derart, daß die INPUT-Parameter auch Vektoren (und Matrizen) sein können (siehe [Ueber05, §5.3.7] und §4.1.2 im Kernbichlerskript<sup>6</sup>). Testen Sie dies durch einen zweiten Aufruf Ihrer Funktion aus dem Hauptfile heraus, wobei Sie gleichgroße (Zeilen-) Vektoren als INPUT verwenden.

Hinweise: `function, .* , .^ , ./`

Abgabe der Lösungen:

Die Abgabe der Lösungen (\*.m-Files und Grafiken) muß über Moodle<sup>7</sup>. erfolgen.

Die Filenamen **müssen** dem Schema `bsp_nummer`, gefolgt von der Filextension, entsprechen. Z.B. sind in Beispiel 2 die Files `bsp_2.m` und `bsp_2.jpg` abzugeben. Andere Filebezeichner zählen nicht als abgegebene Files.

Abzugebende Files (keine Unterverzeichnisse!):

```
bsp_1.m
bsp_2.m
bsp_3.m      bsp_3.jpg
bsp_4.m
bsp_5.m      bsp_5_fkt.m
```

<sup>4</sup><https://de.wikipedia.org/wiki/Maschinengenauigkeit>

<sup>5</sup><https://de.wikipedia.org/wiki/Mittelwert>

<sup>6</sup><http://itp.tugraz.at/LV/kernbich/AppSoft-1/Kapitel/appsoft1-kapitel-4.pdf>

<sup>7</sup><http://moodle.uni-graz.at>