

**ComputerMathematik-LAK WS 24**  
**Computerpraktikum zum Abgabetermin 14.10.2024, 23:55 (Kontrolle ab 21.10.)**

**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 *Matlabxxx* 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 getätigt. 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 entsprechen 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=tqjZ80PwqBU&feature=c4-overview-vl&list=PL7CAABC40B2825C8B>

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

## Nun die ersten Aufgaben (Blatt 0 - Abgabe 14.10; Kontrolle ab 21.10.2024):

### 1. Symbolisches Paket:

- (a) Gegeben ist die Funktion  $f(a, b, c) = 2(a + b^2 + c^3) \cdot (e^{\frac{a+b}{2}} - 1)$ .  
Lösen Sie den Ausdruck  $f(a, b, c) == 0$  jeweils nach  $a$ ,  $b$  bzw.  $c$  mittels `solve` auf.
- (b) Lösen Sie  $x^3 - 64 \cdot x^2 - 256 \cdot x + 1024 = 0$  symbolisch (`syms`, sowohl `solve` als auch `vpsolve` probieren. Bei `solve` könnte die Option `MaxDegree` nützlich sein.).  
Zeichnen Sie die Funktion mit `ezplot` oder `fplot`.
- (c) Lösen Sie symbolisch die Gleichung  $x^2 + px + q = 0$ .  
Lösen Sie symbolisch die Gleichung  $x^2 + 4x - 6 = 0$ .
- (d) Bestimmen Sie symbolisch die Schnittpunkte des Kreises mit der Ellipse (`solve`, `vpsolve`)

$$\begin{aligned}(x - 2)^2 + (y - 3)^2 - 25 &= 0 && \text{Mittelpunkt } (2, 3) \text{ und Radius } 5 \\(x + 4)^2 + y^2 - 36 &= 0 && \text{Mittelpunkt } (-4, 0) \text{ und Radius } 6\end{aligned}$$

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

### 2. Integration und Differentiation.

- (a) Berechnen Sie  $\int e^{-x}(x^2 + 1) \cos(x) dx$ .
- (b) Berechnen Sie  $\int_{-2}^5 e^{-x}(x^2 + 1) \cos(x) dx$ .
- (c) Berechnen Sie die erste und zweite Ableitung von
- $f(x) = \frac{\log_{10}(a - x)}{x^2}$  und von
  - $g(x) = \frac{x * \cos(x) + x}{e^{2x} - \ln(x)} + 3 * x^2$ .
- (d) Nutzen Sie `limit` und berechnen Sie  $\lim_{x \rightarrow \infty} g''(x)$ .
- (e) Lösen Sie die Aufgabe 2 *Stau, a)1*) (unter [https://www.mathago.at/wp-content/uploads/Downloads/KL22\\_PT2\\_HTL\\_AMT\\_AB\\_H2\\_AU.pdf](https://www.mathago.at/wp-content/uploads/Downloads/KL22_PT2_HTL_AMT_AB_H2_AU.pdf) zu finden) mithilfe von Matlab. (`subs` um einen Wert einzusetzen).

Hinweise: `syms`, `solve`, `vpsolve`, `eval`, `int`, `diff`, `pretty`, `char`, `fplot`, `ezplot`, `subs`

### Abgabe der Lösungen:

Die Abgabe der Lösungen (\*.m-Files und Grafiken) muß über Moodle<sup>3</sup>. erfolgen.

Die Filenamen **müssen** dem Schema `bsp_nummer`, gefolgt von der Fileextension, 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 (auch als ein zip-File möglich):

`bsp_1.m`  
`bsp_2.m`

### Arbeiten von Zuhause:

Sie können unter <https://it.uni-graz.at/de/services/software/virtuelle-software/> eine *remote-Version* von Matlab verwenden. Diese können Sie zuhause ausführen. **Vorsicht:** Sie müssen Ihre Datei am Ende auf Ihrem eigenen Computer speichern, sie arbeiten dabei auf einem Uni-System, und dort wird auch die Datei gespeichert.

<sup>3</sup><https://moodle.uni-graz.at/course/view.php?id=119911>