TU Chemnitz – Professur Numerische Mathematik 9. April 2015
Prof. Dr. Oliver Ernst
Dipl.-Math. Björn Sprungk

Numerische Mathematik

Sommersemester 2015

1. Übung: Einführung in MATLAB

In dieser Übung sollen Sie erste Schritte in MATLAB tätigen und sich dabei selbst eine eigene Hilfe erarbeiten. Laden Sie dazu die bereitgestellten MATLAB-Skripte `Intro.m`, `Intro_Matrix.m`, `Intro_Kontrollstrukturen.m`, `my_func.m` und `Intro_Grafik.m` von der Homepage der Vorlesung herunter und bearbeiten Sie die nachstehen Aufgaben.

Generell sei auch auf die Online-Dokumentation von MATLAB verwiesen:

http://de.mathworks.com/help/matlab/

Aufgabe 1 (MATLAB als Taschenrechner)

Öffnen Sie das Skript `Intro.m` in MATLAB. Lesen Sie die Kommentare (Text nach dem Symbol %) aufmerksam durch und führen Sie die einzelnen Befehle in der Kommandozeile aus. Ersetzen Sie die auftretenden „???“ im Skript durch entsprechende Befehle bzw. Kommentare.

Aufgabe 2 (Arbeiten mit Matrizen)

Öffnen Sie das Skript `Intro_Matrix.m`. Führen Sie alle angegebenen Befehle im Kommandofenster aus und ersetzen Sie alle „???“ im Skript durch entsprechende Kommentare.

(a) Was bewirken die Befehle `>> [1 2 3; 4 5 6 7]` und `>> A = eye(5); A(3, 3:5) = ones(2,3)` in MATLAB und wieso?

(b) Es sei `A = ones(4)` zugewiesen. Überlegen Sie sich, was der Befehl

```
A(5,6) = 42
```

bewirken könnte. Geben Sie ihn anschließend ein.

(c) Finden Sie heraus, was die Befehle `fliplr` und `flipud` machen. Erzeugen Sie danach möglichst einfach die Matrix

$$B = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{7\times 7}$$

(d) Berechnen Sie mit MATLAB die Lösung $X \in \mathbb{R}^{3\times 3}$ der folgenden Matrixgleichung

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 2 & 3 \end{pmatrix}^2 X^\top + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 3 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Aufgabe 3 (Kontrollstrukturen)

Öffnen Sie das Skript `Intro_Kontrollstrukturen.m`. Lesen Sie es und führen Sie alle Befehle darin aus. Ersetzen Sie wieder alle „???“ durch erklärende Kommentare.

(a) Erzeugen Sie unter Verwendung der `for`-Schleife und der `if`-Verzweigung die Matrix

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 0 & 6 & 7 & 8 & 9 \\ 0 & 0 & 10 & 11 & 12 \\ 0 & 0 & 0 & 13 & 14 \\ 0 & 0 & 0 & 0 & 15 \end{pmatrix}.$$

(b) Ein Wort zur Effizienz MATLAB: programmieren Sie, wenn möglich vektorisiert! Vergleichen Sie dazu mittels `tic` und `toc` die Laufzeit des vektorwertigen Befehls mit der `for`-Schleife:

```
x = [0 : pi/100 : 100*pi];
y = zeros(size(x));
tic; for(j in 1:length(x), y(j) = sin(x(j)); end; toc
tic; y = sin(x); toc
```

Aufgabe 4 (Funktionen)

(a) Man kann in MATLAB einfache Funktionen als *anonyme* Funktionen mittels "@"definieren:

```
Funktionsvariable = @(Argumente) Befehl;
```

Probieren Sie die folgenden Befehle in MATLAB aus

```
f = @(x,p) x.^p;
f(2,3)
f([1 2 3], 2)
```

(b) Kompliziertere Funktionen – die aus mehreren Befehlen bestehen oder mehrere Ausgaben haben – können mittels eines *Funktions-files* implementiert werden. Öffnen Sie dazu `my_func.m` und lesen Sie die Datei aufmerksam. Probieren Sie dann in der Kommandozeile

```
my_func(4)
[a,b] = my_func(7.5)
```

(c) Implentieren Sie anschießend eine Funktion, die die Wurzel einer Zahl a mittels des Newton-Verfahrens, also der Iteration

$$x_{k+1} = \frac{1}{2}\left(x_k + \frac{a}{x_k}\right),$$

berechnet. Als zweites Argument soll die Funktion eine Toleranz `tol` haben. Die Newton-Iteration soll mit $x_0 = a$ beginnen und erst abbrechen, wenn $|x_{k+1} - x_k| \leq \texttt{tol}$. Neben der angenährten Wurzel von a soll auch die Anzahl der Iterationen ausgegeben werden. Testen Sie Ihre Funktion für $a = 9, \texttt{tol} = 10^{-3}$ und $a = 2, \texttt{tol} = 10^{-8}$.
Zusatzfrage: Was passiert, wenn Sie a negativ wählen? (Schauen Sie sich dazu ggf. die Iterierten an.)

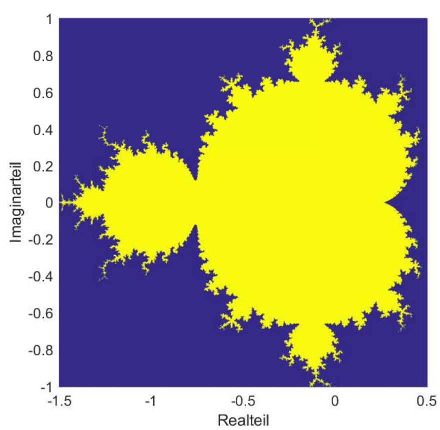
Aufgabe 5 (Grafik)

Öffnen Sie das Skript `Intro_Grafik.m`. Führen Sie nach dem Lesen der Kommentare die Befehle **einzeln** aus und schreiben Sie für die „???“ entsprechende Erklärungen in das Skript.

Zeichnen Sie nun die Mandelbrot-Menge. Diese ist definiert als jene Teilmenge der komplexen Zahlen $c \in \mathbb{C}$, für welche die Iteration

$$z_{n+1} = z_n^2 + c, \qquad z_0 = 0,$$

beschränkt bleibt.



Diskretisieren Sie zur Darstellung der Mandelbrot-Menge die Teilmenge $\{z \in \mathbb{C} : \mathrm{Re}(z) \in [-1.5, 0.5],\ |\mathrm{Im}(z)| \leq 1\}$ der komplexen Ebene mit Schrittweite 10^{-3} in jede Richtung. Führen Sie die Iteration 100mal aus und prüfen Sie, ob der Betrag der letzten Iterierten keiner als 2 ist. Die Punkte, die dies erüllen gehören zur Mandelbrot-Menge.

Hinweise:

- Nutzen Sie `mesh` zum Plotten, sowie `complex` und `real` um komplexe Zahlen aus Paaren von Reellen und reelle Zahlen aus Wahrheitswerten zu generieren.

- Arbeiten Sie soweit möglich vektorwertig. Lassen Sie keine Schleife über die Diskretisierungspunkte laufen!