Praktikumsblatt 3

Lernziele

In diesem Praktikum sollen Sie üben und lernen:

- Vertiefung in Matrizenmanipulation
- Wiederholung verschiedener Schleifenstrukturen
- Schreiben von Funktionen und Skripte

Lösen Sie die Aufgaben im Abschnitt Aufgaben ohne MATLAB, *wirklich ohne* MATLAB!

Aufgaben ohne MATLAB

1. Worin unterscheidet sich in MATLAB eine Funktion von einem Skript?

   Antwort: Ein Skript ist eine Aneinanderreihung von MATLAB-Befehlen. Die Variablen können dabei im Workspace (weiter-) verwendet werden. Ein Skript hat keinen Eingabe- oder Ausgabewert. Funktionen können Variablen als Eingabe enthalten und auch zurückliefern. Nur diese sind nach außen sichtbar. Im allgemeinen werden alle Variablen, die intern angelegt werden, nach Beendigung einer Funktion wieder gelöscht.

2. Was besagt die Fehlermeldung `index out of bounds`? Geben Sie ein Beispiel an.

   Antwort: Sie versuchen Werte einer Variablen zu lesen, die nicht angelegt wurden. Wenn Sie einen Vektor der Größe n oder eine Matrix der Größe $n \times m$ anlegen und danach einen Zugriff mit dem Index $> n + 1$ bzw. in der Zeile $> n + 1$ oder Spalte $> m + 1$ versuchen, erfolgt diese Fehlermeldung.
   Beachten Sie, dass Wertzuweisungen in MATLAB auch auf Bereiche möglich sind, die nicht angelegt wurden, auch auf bestehenden Variablen. Die fehlenden Einträge werden dann einfach mit 0 aufgefüllt. (z.B. `test=ones(3,3)` und dann `test(4,4)=10`).

3. Welche Dimension haben `i, r, c`, nachdem das folgende Skript ausgeführt wurde? Was liefert `find` in den drei Matrizen zurück?

```
A = [1 2 3; 4 5 6; 7 8 9];
i = find(A>5);
[r,c] = find(A>5);
```

   Antwort: Alle sind 4×1 Matrizen, `i` gibt die Indizes (die Matrix als Vektor betrachtet) der Einträge > 5 zurück, in `r` stehen die Zeilen und in `c` die Spalten, wo $A > 5$ ist, d.h., erster Eintrag von `r` und erster Eintrag von `c` ergibt Zeile und Spalte für den ersten Wert > 5.

4. Die folgende MATLAB-Funktion soll das Produkt der beiden Matrizen A und B zurückgeben. Korrigieren Sie den fehlerhaften Code!

```
1 function C = MatrixProdukt(A,B);
2 C = zeros(size(A,1), size(B,2));
3 for i = 1 : size(A,1)
4   for j = 1 : size(B,2)
5     for k = 1 : size(A,2)
6       C(i,j) = C(i,j) + A(i,k) * B(k,i)
7             end
8   end
9 end
```

Antwort: In der 6. Zeile ersetzen Sie `B(k,i)` durch `B(k,j)`.

5. Gegeben sei untenstehendes Skript:

   (a) Beschreiben Sie kurz, was dieses Skript leistet.

   (b) Was passiert, wenn Sie das Skript zunächst für `n=20` und anschließend für `n=10` ausführen? Warum kommt es zu dem Fehler?

   (c) Dies ist der Grund, warum im Normalfall am Anfang eines jeden Skriptes die Befehle `clear all` und `close all` ausgeführt werden sollten (es sei denn es werden explizit Variablen aus dem Workspace benötigt). Korrigieren Sie das obige Skript, dass es auch für den Fall aus Teilaufgabe b) fehlerfrei funktioniert ohne die Funktion `clear` zu verwenden.

```
1 disp(['Plot der Sinus Funktion auf [0,10]']);
2 n = input(['Plot an wievielen Punkten? ']);
3 x = linspace(0,10,n);
4 for i=1:n
5   y(i) = sin(x(i));
6 end;
7 plot(x,y);
```

Antwort: Das Skript plottet die Sinusfunktion durch lineare Interpolation an n Punkten.
Beim ersten Aufruf hat der Zeilenvektor y 20 Einträge. Diese werden nicht gelöscht, die Größe bleibt auch beim zweiten Aufruf bestehen, wobei x aber nur noch 10 Einträge hat. Beim `plot`-Befehl müssen beide Vektoren die gleiche Größe haben!
Fügen Sie nach Zeile 3 die Zeile `y = zeros(size(x));` ein.

Praktische Aufgaben

6. Es sei `A=reshape(exp(i*linspace(0,2*pi,9)),3,3)`. Welches sind die Werte der folgenden MATLAB-Ausdrücke? Überlegen Sie sich das Resultat (mit der Hilfe) und beschreiben Sie es, bevor Sie die Rechnung an der Maschine durchführen. Verwenden Sie als Ausgabeformat `format short`.
   Antwort:

(a) `A'` ... konjugiert transponiert,
(b) `A.'`... transponiert,
(c) `abs(A)` ... 3×3 Matrix mit lauter 1er,
(d) `imag(A)` ... komponentenweise Imaginärteil
(e) `real(A)` ... komponentenweise Realteil
(f) `angle(A)` ... komponentenweise Winkel der komplexen Zahlen
(g) `flipud(A)` ... dreht Spalteneinträge um
(h) `rot90(A)` ... drehen der Einträge um 90° mathematisch positiv
(i) `A(:)` ... liefert die Matrix als Vektor, spaltenweise
(j) `compass(A)` ... plottet die Polarkoordinaten der einzelnen Einträge

7. Erzeugen Sie in einem Skript `trisymA.m`

   (a) Eine 6×6 obere/untere Dreiecksmatrix mit reellen Zufallszahlen zwischen 0 und 1.
   (b) Eine symmetrische 10×10 Matrix S mit ganzzahligen Zufallszahlen zwischen 0 und 8. Suchen Sie die Befehle gegebenenfalls in der Hilfe!

8. Es sei A die folgende Matrix:

$$A = \begin{pmatrix} 4 & 3 & 0 \\ 3 & 6 & 2 \\ 0 & 2 & 4 \end{pmatrix}$$

   Erstellen Sie ein Skript `eigenA.m` und lösen Sie folgende Aufgaben:

   (a) Bestimmen Sie die Eigenwerte der Matrix A.
   (b) Bestimmen Sie zu jedem Eigenwert einen zugehörigen Eigenvektor.
   (c) Berechnen Sie $\det(A - \lambda I)$ und zeigen Sie, dass dies für die Eigenwerte Null ist.
   (d) Zeigen Sie die Gültigkeit von $AQ = QD$, wobei Q diejenige Matrix ist, in deren Spalten die Eigenvektoren stehen, und D die Diagonalmatrix mit den Eigenwerten in der Diagonalen.

9. Schreiben Sie (nur unter Verwendung elementarer Funktionen) eine Funktion `quadEq.m`

```
function [numSol, sol] = quadEq(p,q)
```

   welche die reellen Lösungen der Gleichung $x^2+px+q=0$ bestimmt. Die Eingabewerte sind p und q. Der Rückgabewert `numSol` soll dabei die Anzahl der reellen Lösungen enthalten und `sol` ist ein Vektor mit den Lösungen an sich. Für den Fall, dass es keine Lösung gibt, soll `sol` leer sein, d.h. `sol=[]`.
   Schreiben Sie wie üblich ein Skript, welches die Funktion `quadEq` für verschiedene Werte von `p` und `q` testet.

10. Schreiben Sie jeweils eine Prozedur, welche die Fakultät einer Zahl k

   (a) iterativ mit `for`-Schleife,
   (b) iterativ mit `while`-Schleife und
   (c) iterativ ohne explizite Schleife

(d) rekursiv (für Fortgeschrittene)

berechnet. Vergleichen Sie die Laufzeiten (siehe `tic`, `toc`) für z.B. $k = 20$. Was stellen Sie dabei fest?
Schreiben Sie eine funktion `fakloesung`, die weder einen Eingabewert braucht, noch einen Rückgabewert hat. Dies ermöglicht nun, weitere Funktionen (die gesuchten), unten anzuhängen. **Hinweis**: Um die sehr kurzen Laufzeiten vernünftig vergleichen zu können empfiehlt es sich die Zeit über z.B. $N = 1000$ Auswertungen zu messen.