

## Übungen zur Vorlesung Einführung in das Programmieren für TM

### Serie 7

Die Aufgaben mit Stern (\*) sind bis zur nächsten Übung vorzubereiten und werden dort abgeprüft. Die übrigen Aufgaben dienen nur Ihrer Übung und mir als zusätzliche Grundlage für den Prüfungstoff in den schriftlichen Tests. Kopieren Sie bitte den Source-Code in ein Unterverzeichnis `serie07` Ihres Home-Verzeichnisses. Überprüfen Sie bitte vor der Übung, ob Ihre Source-Codes mit `matlab` auf der `lva.student.tuwien.ac.at` interpretiert werden können. In den folgenden Aufgaben sollen **Arithmetik, Verzweigungen und Funktionen** geübt werden.

**Aufgabe 61\*.** Die Summe  $r = p + q$  zweier Polynome  $p, q$  ist wieder ein Polynom. Man schreibe eine Funktion `addPolynomials`, die die Summe  $r$  berechnet. Dabei sollen  $p(x) = \sum_{k=1}^m a_k x^{k-1}$  und  $q(x) = \sum_{k=1}^n b_k x^{k-1}$  in Form der Zeilenvektoren  $a \in \mathbb{R}^m$  und  $b \in \mathbb{R}^n$  ihrer Koeffizienten gespeichert werden. Realisieren Sie die Funktion ohne Verwendung von Schleifen mithilfe der MATLAB-Arithmetik. Speichern Sie den Source-Code unter `addPolynomials.m` in das Verzeichnis `serie07`.

**Aufgabe 62\*.** Was macht folgende Funktion?

```
function [y,z] = f(x,y,z)
if (x==1)
    if (y>z)
        tmp=y;
        y=z;
        z=tmp;
    end
else
    if (z>y)
        tmp=y;
        y=z;
        z=tmp;
    end
end
```

Wie könnte man eine Funktion programmieren, die dasselbe leistet, aber nur eine Verzweigung verwendet?

**Aufgabe 63\*.** Matlab stellt Ihnen eine umfangreiche Bibliothek mit zahlreichen nützlichen Funktionen zur Verfügung. Zu jeder Funktion erhalten Sie mit `help funktionsname` eine ausführliche Beschreibung. Erklären Sie die Verwendung von `find` und `max`. Was sind die möglichen Parameter? Was sind die möglichen Rückgabewerte? Überlegen Sie sich einfache Beispiele an Hand derer Sie die Funktionen illustrativ erklären können und halten Sie diese in einem einfachen Skript fest.

**Aufgabe 64\*.** Schreiben Sie eine Funktion `cut`, die zu gegebenem  $k \in \mathbb{N}$  aus einem Vektor  $x \in \mathbb{R}^n$  alle Einträge  $x_j$  mit  $|x_j| \geq k$  streicht. Anstatt Schleifen soll der Befehl `find` verwendet werden. Speichern Sie den Source-Code unter `cut.m` in das Verzeichnis `serie07`.

**Aufgabe 65.** Schreiben Sie eine Funktion `pnorm`, die für  $p \in [1, \infty)$  die  $\ell_p$ -Norm

$$\|x\|_p := \left( \sum_{j=1}^n |x_j|^p \right)^{1/p}$$

eines Vektors  $x \in \mathbb{R}^n$  berechnet. Realisieren Sie die Funktion erstens mittels geeigneter Schleifen und zweitens ohne Verwendung von Schleifen. Im zweiten Fall realisiere man die Summation mittels `sum`.

**Aufgabe 66.** Man schreibe eine Funktion `evalPolynomial`, die den Funktionswert  $p(x) = \sum_{k=1}^n a_k x^{k-1}$  zurückgibt. Dabei werde das Polynom in einem Zeilenvektor  $a \in \mathbb{R}^n$  der Länge  $n$  gespeichert. Falls  $x$  ein Spaltenvektor der Länge  $m$  ist, soll  $p(x)$  ebenfalls ein Spaltenvektor der Länge  $m$  sein. Realisieren Sie die Funktion wieder unter Vermeidung von Schleifen.

**Aufgabe 67.** Schreiben Sie eine Funktion `differentiatePolynomial`, die den Koeffizientenvektor der Ableitung  $p'(x)$  des Polynoms  $p(x) = \sum_{k=1}^n a_k x^{k-1}$  zurückgibt. Realisieren Sie die Funktion wieder ohne Verwendung von Schleifen.

**Aufgabe 68.** Schreiben Sie eine Funktion `rundung`, die für eine gegebene Zahl  $x \in \mathbb{R}$  die Zahl  $n \in \mathbb{Z}$  zurückliefert, die  $x$  am nächsten liegt. Falls  $x$  genau in der Mitte zwischen zwei ganzen Zahlen liegt, werde die größere zurückgegeben.

**Aufgabe 69.** Schreiben Sie eine Funktion `skalarprodukt`, die das Skalarprodukt zweier Vektoren  $x, y \in \mathbb{R}^n$  berechnet, ohne Schleifen zu verwenden. Dabei dürfen  $x$  und  $y$  Spalten- oder Zeilenvektoren sein, die ggf. mittels `reshape` auf passende Form gebracht werden.

**Aufgabe 70.** Schreiben Sie eine Funktion `minabs`, die von zwei Werten  $x, y \in \mathbb{R}$  denjenigen zurückliefert, dessen Absolutbetrag kleiner ist. Der Absolutbetrag wird in Matlab durch `abs` gegeben. Realisieren Sie die Funktion mit und ohne die Verwendung von `min`.