

Berechnungen mit MATLAB

Eine Vielzahl mathematischer Probleme kann mit dem Programm MATLAB gelöst werden. Im Folgenden wird eine kurze Einführung in MATLAB gegeben. Weitergehende Hinweise für das Arbeiten mit diesem Programm findet man in der Fachliteratur, z. B. in dem Buch:
Stein, U.: Programmieren mit MATLAB. Hanser, München 2017

1 Reelle Zahlen

In der Dezimaldarstellung einer Zahl wird das Komma durch einen Punkt ersetzt. Das Semikolon am Ende der Zuweisung unterdrückt die Bildschirmausgabe des Zahlenwertes:

```
a = 1.234;
```

Namen von Variablen müssen mit einem Buchstaben beginnen. MATLAB unterscheidet zwischen Groß- und Kleinschreibung; die Variable A hat nicht den Wert, welcher der Variablen a zugewiesen wurde.

Das Malzeichen darf nicht weggelassen werden. MATLAB kennt keine griechischen Buchstaben; der Wert von π ist in MATLAB bekannt.

Nach einem Zeichen % wird der weitere Text in der betreffenden Zeile als Kommentar behandelt, mit dem keine Berechnung durchgeführt wird.

```
w = 2 * pi * f; % Kreisfrequenz
```

Bei Funktionen wird das Argument in Klammern gesetzt.

Die trigonometrischen Funktionen sin, cos und tan verlangen Winkelangaben im Bogenmaß (rad); Winkelangaben in Grad erhält man mit der Variablen b2g:

```
b2g = 180 / pi;  
wg = b2g * wr; % wg: Winkel in Grad
```

Es gibt aber auch die Winkelfunktionen sind, cosd und tand, bei denen der Winkel in Grad steht; der Buchstabe d weist auf *degree* (Grad) hin.

Beim Arkustangens wird der Hauptwert (in rad) geliefert. Es gibt auch einen Arkustangens für zwei Argumente, bei dem die Umrechnung von Koordinatenangaben intern vorgenommen wird.

Math. Ausdruck	MATLAB-Ausdruck	Ergebnis
$\sqrt{2}$	sqrt(2)	1.4142
$\ln 0,5$	log(0.5)	-0.6931
$\lg 0,0001$	log10(1e-4)	-4.0000
$e^{\ln 0,5}$	exp(log(0.5))	0.5000
$\sin 30^\circ$	sin(30/b2g)	0.5000
$\cos \pi$	cos(pi)	-1.0000
$\arctan 3$	atan(3)	1.2490
Punkt (-1; -3)	atan2(-3,-1)	-1.8925
$\sinh 0,5$	sinh(0.5)	0.5211
$\operatorname{artanh} 0,8937$	atanh(0.8937)	1.4400

2 Ein- und Ausgabe

Die Eingabe einer Variablen über die Tastatur kann einen Hinweis-Text enthalten, der in Hochkommata eingeschlossen wird:

```
f = input('Frequenz f = ');
```

Die Ausgabe kann ebenfalls einen Hinweis-Text enthalten. Bei der Ausgabe einer Zahl wird auch die Ausgabeform mit der zugehörigen Anzahl der Zeichen in Hochkommata eingeschlossen:

```
disp([' w = ', sprintf('%5.4e ',w,' 1/s')]);
```

Für die Ausgabeform gibt es die Möglichkeiten:

- d: ganze Zahl;
- e: Gleitpunktzahl;
- f: Festpunktzahl;
- s: Zeichenkette.

3 Komplexe Zahlen

Eine komplexe Zahl wie z. B. $z = 3 + j4$ kann in drei Formen eingegeben werden: $3+4i$; $3+4*j$; $3+j*4$; die übliche mathematische Schreibweise wird von MATLAB nicht akzeptiert.

Von einer komplexen Zahl können folgende Werte abgerufen werden:

```
real(z) ... Realteil
imag(z) ... Imaginärteil
abs(z) ... Betrag
angle(z) ... Winkel
conj(z) ... konjugiert komplexe Größe z*
```

Eine Eingabe der komplexen Größe in der P-Form kennt MATLAB nicht, deshalb muss sie umschrieben werden:

```
zb = input('Betrag = ');
wg = input('Winkel in Grad = ');
wb = wg / b2g; % Winkel im Bogenmaß
z = zb * cos(wb) + j * zb * sin(wb);
```

Bei komplexen Rechnungen sollten die Variablen i und j nicht mit anderen Werten belegt werden, damit keine unerwünschten Effekte oder Fehler auftreten.

4 Matrizen

Eine Matrix wird in eckigen Klammern eingegeben. Am Ende jeder Zeile muss ein Semikolon stehen.

Beispiel

Die Lösung des linearen Gleichungssystems

$$\begin{aligned} 3x + 4y &= 22 \\ 4x + 5y &= 28 \end{aligned}$$

ist mit MATLAB folgendermaßen möglich:

```
M = [3 4; 4 5]; Rs = [22; 28];
L = linsolve(M, Rs);
```

Auch L ist eine Matrix, auf deren Elemente 2 und 4 mithilfe von runden Klammern zugegriffen werden kann:

```
x = L(1);
y = L(2);
```

5 Kontrollstrukturen

MATLAB kennt vier Arten von Kontrollstrukturen:

- if-Anweisung
- case-Anweisung
- for-Schleife
- while-Schleife

Wir wollen uns im Rahmen dieser kleinen Einführung lediglich mit der if-Anweisung und der for-Schleife befassen.

Bei der einfachen if-Anweisung folgt dem if eine Bedingung und dann eine Gruppe von Anweisungen; das Ganze wird mit einem end abgeschlossen.

Beispiel

Betrag der Zahl x

```
x = input('x = ');
if x < 0; x = -x;
end;
```

Bei diesem einfachen Beispiel ist also $x < 0$ die Bedingung und $x = -x$; die Gruppe der Anweisungen.

Sollen zwei Bedingungen angegeben werden, so können sie durch die Konjunktion (und) mit dem Zeichen & oder durch die Alternative (oder) mit dem Zeichen | miteinander verknüpft werden. Die Zeichen \leq (kleiner gleich) bzw. \geq (größer gleich) werden in MATLAB als \leq bzw. \geq nebeneinander geschrieben. Bei der Abfrage auf Gleichheit müssen zwei Gleichheitszeichen gesetzt werden.

Die einfache if-Anweisung kann durch elseif und else ergänzt werden.

Beispiel: Vorzeichen einer Zahl

```
if x < 0; s = -1,
elseif x == 0; s = 0,
else s = 1, end
```

Bei der for-Anweisung durchläuft eine Variable mehrere Werte von Anfangswert bis zum Endwert: for k=1:3; ... Anweisungen ... , end

Beispiel: Summe der Zahlen von 1 bis 10

```
s = 0;
for k=1:10;
    s = s + k;
end
```

Die Schrittweite steht nach dem Startwert.

Beispiel: Summe ungerader Zahlen von 1 bis 19

```
s = 0;
for k=1:2:19;
    s = s + k;
end
```

6 Symbolische Rechnung

Bei der symbolischen Rechnung wird nicht mit Zahlen, sondern mit Variablen gerechnet. die dem Programm nach dem Codewort `syms` mitgeteilt werden müssen. Die symbolische Rechnung wird in Hochkommata eingeschlossen.

Das Ergebnis der symbolischen Rechnung ist ein symbolischer Ausdruck, in den gegebene Zahlenwerte, die dem Programm vorher mitgeteilt wurden, mit der Anweisung `subs` eingebracht werden können.

7 Programmierung

Die MATLAB-Anweisungen können im Command Window direkt eingegeben und gestartet werden.

Soll eine Befehlsfolge als Programm gespeichert werden, so muss es in eine m-Datei umgewandelt werden. Die m-Dateien sind ASCII-Dateien, die in einem Verzeichnis gespeichert werden; so kann z. B. das Programm `quad.m` zur Lösung einer quadratischen Gleichung von der homepage heruntergeladen werden.

Beim Aufruf einer m-Datei im Command Window wird nur der Dateiname ohne die Erweiterung `„.m“` eingegeben.

8 Beispiele

Das Beispiel 8.1 zeigt ein Programm, mit dem die Gleichung $ax^2 + bx + c = 0$ gelöst werden kann. Dabei arbeitet MATLAB mit komplexen Zahlen; so hat z. B. für $a = 1$; $b = 2$ und $c = 3$ die quadratische Gleichung zwei konjugiert komplexe Lösungen.

Beispiel 8.1

Lösung der Gleichung $ax^2 + bx + c = 0$

```
syms a b c x Lv;
a=input('a = ');
b=input('b = ');
c=input('c = ');
Lv=solve('a*x^2 + b*x + c = 0', x);
x1=subs(Lv(1));
x2=subs(Lv(2));
```

Im nächsten Beispiel wird gezeigt, wie die nichtlineare Gleichung aus der Aufgabe 2.14 gelöst werden kann.

Beispiel 8.2

In diesem Programm müssen die Werte der Variablen in die symbolische Lösung von `solve` substituiert werden.

```
syms R1 R2 R3 Rx;
clc
R1=1000;
R2=2000;
R3=3000;
disp(, );
eqn=Rx==R1+1/(1/R2+1/(R3+Rx));
x = solve(eqn,Rx);
y=subs(x(1));
z=subs(x(2));
erg=double(y);
if z>y; erg=double(z); end;
disp([,Widerstand Rx = ,sprintf(, %4.4f ,,erg,]);
```

Diese Lösung kann als Programm `k2auf14.m` von der homepage heruntergeladen werden.