Mathematik am Computer

5. Übung: Matlab, Teil II

Marcus Grote und Helmut Harbrecht

Universität Basel

23.–26. Oktober 2023

Übersicht

1. Matlab als Programmiersprache
   - Grundlagen
   - Steuerung
2. Graphische Ausgabe
   - Funktionen zeichnen

Variablen

- Variable in Matlab: Speicher für Daten
- Zu “`x = x+1`”: rechts des Gleichheitszeichen wird auf die Variable `x` zugegriffen und die Rechnung durchgeführt, danach wird in der Variable `x` links vom Gleichheitszeichen der errechnete Wert gespeichert.

  $\rightarrow$ Einfach gesagt: “`x = x+1`” inkrementiert `x` um 1.

Beispiele:

Eingabe	Zuweisung x	Zuweisung y
`x = 3;`	3	
`y = 4;`	3	4
`x = x * y;`	12	4
`x = x - y;`	8	4
`y = x^2;`	8	64

Die for-Schleife

Der Vektor `vec` hat die Länge `n`.

```
for k = vec
      Befehle
end
```

1. Zunächst ist $k = vec(1)$, d.h. der erste Wert des Vektors `vec`, und es werden alle Befehle zwischen `for` und `end` mit dem Wert $k = vec(1)$ ausgeführt.
2. Es wird $k = vec(2)$ gesetzt und alle Befehle zwischen `for` und `end` mit dem Wert $k = vec(2)$ ausgeführt, usw.
3. Es werden alle Werte von `vec` durchlaufen, bis einschliesslich $k = vec(n)$.

Die for-Schleife – Beipsiel 1

```
M = [5,-10,7,3,-1.1];
for k = M
     disp('Jetzt ist k:')
     disp(k)
end
```

INPUT

```
Jetzt ist k:
     5
Jetzt ist k:
     -10
Jetzt ist k:
     7
Jetzt ist k:
     3
Jetzt ist k:
     -1.1
```

OUTPUT

Die for-Schleife – Beipsiel 2

Verschachtelte for-Schleifen

```
Wochen = 2;
Tage = 5;
for i = 1:Wochen
     disp(['Woche ' num2str(i) ':'])
     for j = 1:Tage
         disp(['Tag ' num2str(j) ': -'])
     end
end
```

INPUT

OUTPUT

```
Woche 1:
Tag 1: -
Tag 2: -
Tag 3: -
Tag 4: -
Tag 5: -
```

```
Woche 2:
Tag 1: -
Tag 2: -
Tag 3: -
Tag 4: -
Tag 5: -
```

Parametrisierte Kurven zeichnen

Eine (stetige) Funktion $[a, b] \longrightarrow \mathbb{R}^2$, $t \mapsto (u(t), v(t))$ heisst ebene Kurve in Parameter-Form und

$$K = \{(x, y) \in \mathbb{R}^2 \mid x = u(t),\ y = v(t),\ t \in [a, b]\}$$

ist der Graph der Kurve. Zum Beispiel ist

$$K = \{(x, y) \in \mathbb{R}^2 \mid x = \sin(t),\ y = \cos(t),\ t \in [0, 2\pi]\}$$

ein Kreis mit Radius 1 um den Ursprung.

Parametrisierte Kurven zeichnen

K zeichnen:

1. Definiere Spaltenvektor `t` von *t*-Werten, z.B.
```
t = 2*(0 : pi/100 : pi)';
```
2. Definiere Spaltenvektor `x` von *x*-Werten, z.B.
```
x = sin(t);
```
3. Definiere Spaltenvektor `y` von *y*-Werten, z.B.
```
y = cos(t);
```
4. Zeichne Wertetabelle
```
plot(x,y);
```

2. Alternativ:
```
plot(sin(t),cos(t));
```

Funktionen zweier Veränderlichen zeichnen

Grundlagen

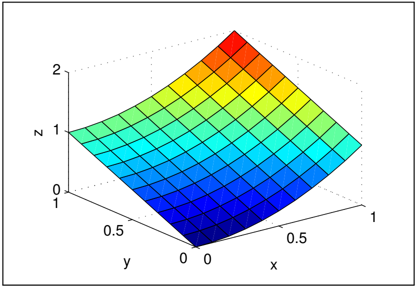
Beispiel: Zeichne

$$z = f(x,y) = x^2 + y, \qquad x \in [0,1], \quad y \in [0,1].$$

INPUT

```
v = 0:0.1:1;
[X,Y] = meshgrid(v,v);
Z = X.^2+Y;
surf(X,Y,Z)
```

OUTPUT



Funktionen zweier Veränderlichen zeichnen

Grundlagen

Es werden Matrizen `X` und `Y` benötigt, so dass die elementweise Auswertung von `X` und `Y` eine Matrix `Z` mit den Funktionswerten liefert, z.B.

$$X = \begin{pmatrix} 0 & 0.5 & 1 \\ 0 & 0.5 & 1 \\ 0 & 0.5 & 1 \end{pmatrix} \qquad Y = \begin{pmatrix} 0 & 0 & 0 \\ 0.5 & 0.5 & 0.5 \\ 1 & 1 & 1 \end{pmatrix}$$

ergibt für $z = f(x, y) = x^2 + y$

$$Z = \begin{pmatrix} 0 & 0.25 & 1 \\ 0.5 & 0.75 & 1.5 \\ 1 & 1.25 & 2 \end{pmatrix}$$

Funktionen zweier Veränderlichen zeichnen

Vorgehensweise

1. Erzeuge Matrizen X und Y, z.B.

```
[X,Y] = meshgrid(0:0.1:1, 0:0.1:1);
```

   Genauer: sind `x` und `y` Vektoren der x- bzw. y-Werte bei denen die Funktion f ausgewertet werden soll, so werden die benötigten Matrizen `X` und `Y` erzeugt durch:

```
[X,Y] = meshgrid(x,y);
```

2. Erzeuge Matrizen Z der Funktionswerte, z.B.

```
Z = X.^2+Y;
```

3. Zeichne Funktion

```
surf(X,Y,Z);
```

Funktionen zweier Veränderlichen zeichnen

Weitere Möglichkeiten:

1. Modifizierung der Schattierung:
   ```
   shading faceted
   ```
   (Matlab-default shading)
   ```
   shading flat
   shading interp
   ```
2. Fläche ohne Gitter:
   ```
   surf(X,Y,Z, 'EdgeColor', 'none');
   ```
3. „beleuchtete“ Fläche:
   ```
   surfl(X,Y,Z);
   ```
4. Nur Gitter:
   ```
   mesh(X,Y,Z);
   ```
5. Höhenlinien:
   ```
   contour(X,Y,Z);
   ```
6. Schattierte Karte:
   ```
   pcolor(X,Y,Z);
   ```

Weitere Plotmöglichkeiten

1. Zeichnen von (parametrisierten) Kurven im Raum

```
plot3(X,Y,Z);
```

2. Zeichnen von 2D-Vektorfeldern, d.h.

   von Funktionen $D \subseteq \mathbb{R}^2 \longrightarrow \mathbb{R}^2$ mit $(x, y) \mapsto (u, v)$

```
quiver(x,y,u,v)
```

3. Zeichnen von 3D-Vektorfeldern, d.h.

   von Funktionen $D \subseteq \mathbb{R}^3 \longrightarrow \mathbb{R}^3$ mit $(x, y, z) \mapsto (u, v, w)$

```
quiver3(x,y,z,u,v,w)
```