Mathematik am Computer

MATLAB, Teil IV

M. Grote, H. Harbrecht und M. Schmidlin

Universität Basel

02.11 – 05.11.2026

Übersicht

1 Funktionen zeichnen

2 Parametrisierte Kurven zeichnen

3 Funktionen zweier Veränderlichen zeichnen

Funktionen einer Veränderlichen zeichnen

Matlab zeichnet keine Funktionen, sondern Wertetabellen als Polygonzug!

① Definiere Spaltenvektor `x` von x-Werten, z.B.

```
x = (0 : pi/100 : pi)';
```

② Definiere Spaltenvektor `y` von y-Werten einer Funktion, z.B.

```
y = sin(x);
```

③ Zeichne Wertetabelle

```
plot(x,y);
```

④ Alternativ:

```
plot(x,sin(x));
```

Beachte: Die Funktion, die gezeichnet werden soll, muss elementweise auf `x` anwendbar sein!

Funktionen einer Veränderlichen zeichnen

Um mehrere Funktionen zu zeichnen, geht man wie folgt vor:

① Definiere Spaltenvektor `x` von x-Werten, z.B.

```
x = (0 : pi/100 : pi)';
```

② Definiere Spaltenvektoren `y` und `z` von y-Werten von Funktionen, z.B.

```
y = sin(x); z = cos(x);
```

③ Zeichne Wertetabellen

```
plot(x,y,x,z);
```

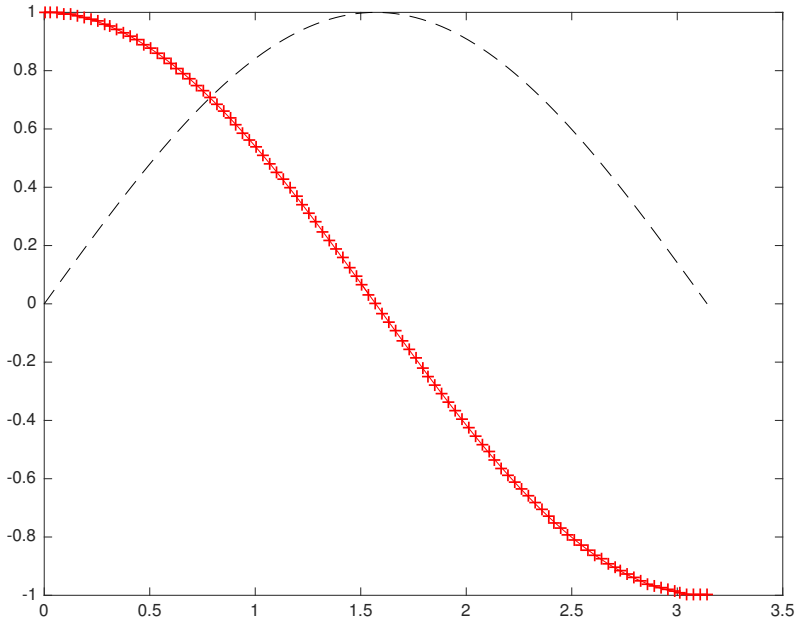
④ Alternativ:

```
plot(x,[y,z]);
plot(x,sin(x),x,cos(x));
```

Plotoptionen und Modifizierung von Graphen

① Weitere Optionen beim Zeichnen von Kurven, z.B.

```
plot(x,sin(x),'k--',x,cos(x),'r-+')
```



Plotoptionen und Modifizierung von Graphen

② Modifizierung der Achsen:

```
axis([xmin xmax ymin ymax])
axis equal
xlim([xmin xmax])
```

usw.

③ Beschriftung:

```
title('Überschrift')
xlabel('x-Achse')
legend('Graph 1','Graph 2','Graph 3')
```

④ Weitere Modifizierungen direkt am Graphen möglich!

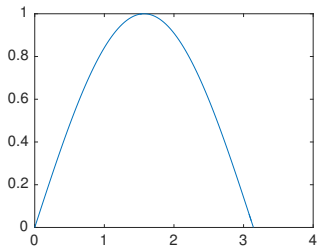
Der subplot-Befehl

Um mehrere Plots in einem Fenster darstellen zu lassen, benutzen wir den Befehl

```
subplot(m,n,k)
```

Beispiel:

```
subplot(2,2,1)
plot(x,sin(x))
```



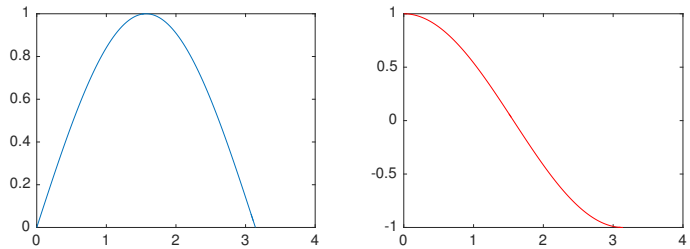
Der subplot-Befehl

Um mehrere Plots in einem Fenster darstellen zu lassen, benutzen wir den Befehl

```
subplot(m,n,k)
```

Beispiel:

```
subplot(2,2,1)
plot(x,sin(x))
subplot(2,2,2)
plot(x,cos(x),'r')
```



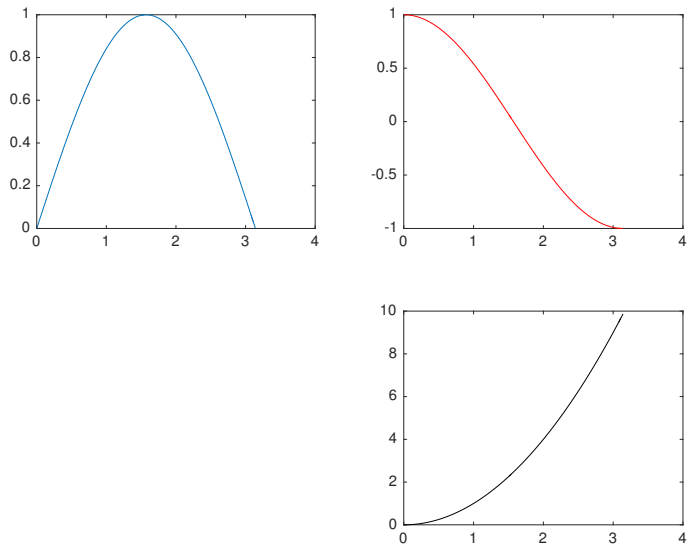
Der subplot-Befehl

Um mehrere Plots in einem Fenster darstellen zu lassen, benutzen wir den Befehl

```
subplot(m,n,k)
```

Beispiel:

```
subplot(2,2,1)
plot(x,sin(x))
subplot(2,2,2)
plot(x,cos(x),'r')
subplot(2,2,4)
plot(x,x.^2,'k')
```



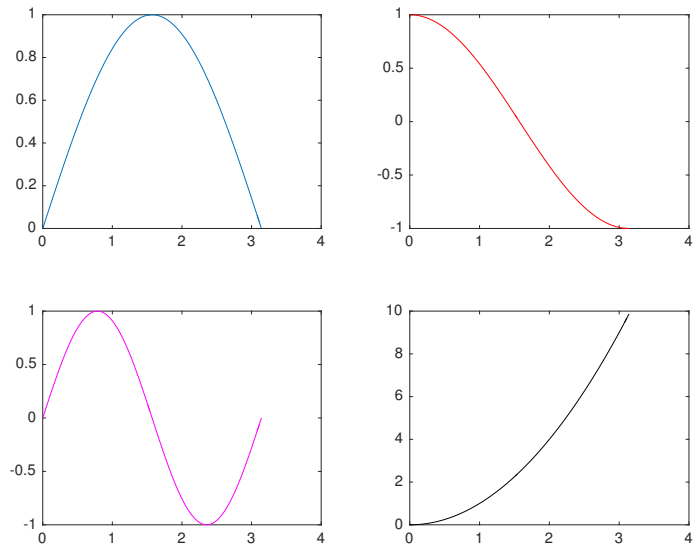
Der subplot-Befehl

Um mehrere Plots in einem Fenster darstellen zu lassen, benutzen wir den Befehl

```
subplot(m,n,k)
```

Beispiel:

```
subplot(2,2,1)
plot(x,sin(x))
subplot(2,2,2)
plot(x,cos(x),'r')
subplot(2,2,4)
plot(x,x.^2,'k')
subplot(2,2,3)
plot(x,sin(2*x),'m')
```



Der figure-Befehl

Der Befehl `figure(n)` öffnet ein Fenster mit der Nummer *n*.

Alle weiteren Plot-Befehle werden in diesem Fenster ausgeführt

Mit den Befehlen `figure(1)`, `figure(2)`, usw. kann man zwischen den Fenstern 1, 2, usw. beliebig wechseln.

Auch Plots in Subplot-Grids lassen sich mit `title('...')` einzeln betiteln; Dem gesamten Subplot-Grid kann mittels `sgtitle('...')` ein Titel hinzugefügt werden.

Mit `close n` und `close all` lassen sich ein bestimmtes bzw. alle Fenster wieder schliessen.

Grundlagen

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.

Vorgehensweise

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);
```

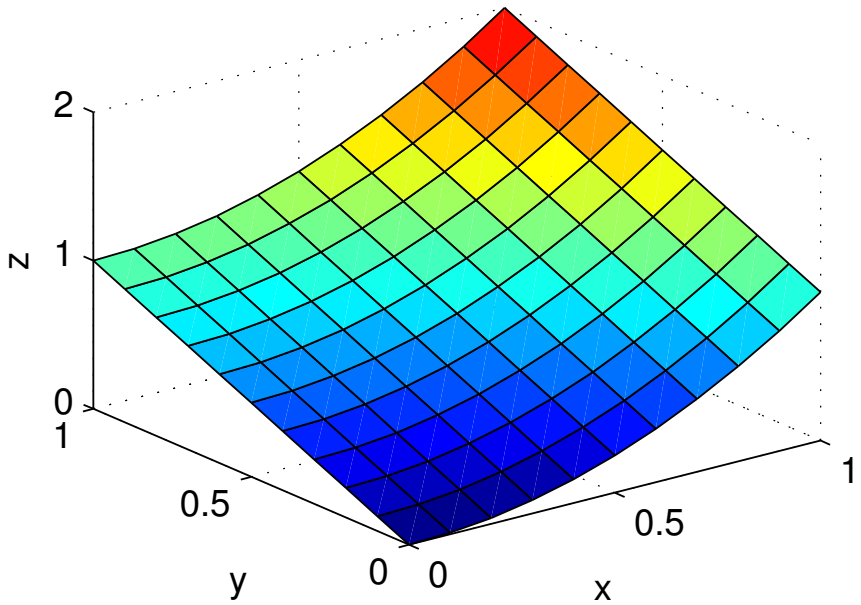
⑤ Alternativ:

```
plot(sin(t),cos(t));
```

Grundlagen

Beispiel: Zeichne

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



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 $z = f(x, y) = x^2 + y$

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

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);
```

Weitere Möglichkeiten

① Modifizierung der Schattierung:

```
shading faceted
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 Möglichkeiten

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)
```