

Graphiken in MATLAB

MATLAB besitzt eine Reihe mächtiger Befehle zur Visualisierung von Funktionen und Daten. Einige der **wichtigsten Funktionen** sollen Sie im folgenden näher kennen lernen.

1. Zweidimensionale Graphen

Geben Sie die folgende Befehlsfolge ein:

```
x = -5:0.05:5;
y = sin(x);
plot(x,y)
plot(x,y,'k-.')
y1 = cos(x);
plot(x,y,'k',x,y1,'b--'), hold on
plot([-3*pi/2:pi:3*pi/2], [1, -1, 1, -1], 'ro')
axis([-5, 5, -2, 2])
xlabel('x')
title('Die Sinusfunktion und ihre Ableitung')
legend('f(x) = sin(x)', 'f''(x) = cos(x)')
```

Wissenswertes zum Plotten von Funktionen $y=f(x)$

- Linien- und Punktgraphen können mit dem `plot`-Befehl erzeugt werden. Sind \mathbf{x} und \mathbf{y} n -dimensionale Vektoren, dann zeichnet `plot(x,y)` die Punkte $(\mathbf{x}_i; \mathbf{y}_i)$ für $i=1 \dots n$ in ein Koordinatensystem und verbindet sie mit Linien.
- Als dritter Eingabeparameter des `plot`-Befehls können in einfachen Anführungszeichen entsprechende Kürzel für **Farbe, Punktsymbol und Linienstil** angegeben werden. Eine Liste der Kürzel erhält man mit `help plot`.
- Es können mehrere dieser Eingabetripel in einem `plot`-Befehl verwendet werden. Dann wird alles in ein Graphik gezeichnet. Eine andere Möglichkeit bietet der `hold on`-Befehl. Als dritte Variante kann man die y -Koordinaten zu einer **Matrix zusammenfassen** \mathbf{y} ; Es wird dann zeilenweise in ein gemeinsames Koordinatensystem geplottet.
- Nach dem Erstellen kann nachträglich mit `axis` der Koordinatenausschnitt angepasst werden. Außerdem können mit `title`, `legend`, `xlabel`, `ylabel` und `text` Beschriftungen angebracht werden.

- Logarithmische Plots erhält man mit den Befehlen **semilogx**, **semilogy** und **loglog**.
- Die Graphiken werden in einem eigenen Fenster erzeugt. Mit dem **figure**-Befehl können auch mehrere Fenster angelegt werden.
- Die **Menüleiste** des Fensters bietet verschiedene Möglichkeiten, um die Graphik auszudrucken, abzuspeichern, in eine Datei zu exportieren, zu skalieren sowie nachträglich zu verändern.
- Es gibt noch zahlreiche weitere Funktionen zum Erstellen zweidimensionaler Graphen: z. B. **bar** (Säulendiagramme), **hist** (Histogramme), **pie** (Tortendiagramme), **stairs** (Treppenfunktionen), **quiver** (Vektorfelder), **contour** (Höhenlinien) usw.

2. Dreidimensionale Graphen

Kopieren Sie sich die Dateien **kurve3d.m** und **graph3d.m** in Ihr Verzeichnis; sehen Sie sich zunächst den Inhalt an. Führen Sie die beiden Skripte danach aus.

Geben Sie folgende Befehlsfolge direkt im Command-Window ein:

```
[X,Y,Z] = sphere;
surf(X,Y,Z)
axis equal
colormap cool
shading interp
camlight
lighting phong
```

Wissenswertes zum Plotten von Funktionen $z=f(x,y)$

- Punkte, Polygonzüge und Kurven: Mit dem **plot3**-Befehl kann man – analog zum **plot**-Befehl - dreidimensionale **Linien- und Punktgraphen** zeichnen.
- Sind **x**, **y** und **z** n-dimensionale Vektoren, dann zeichnet **plot3(x,y,z)** die Punkte (**x_i**; **y_i**; **z_i**) für **i = 1...n** in ein Koordinatensystem und verbindet die Punkte mit Linien.

- Graphen von Funktionen zweier reeller Variablen: Um den Graphen einer Funktion über einem Rechteck zu zeichnen, erzeugt man zunächst mit dem **meshgrid**-Befehl ein rechteckiges **Punktegitter** $[x, y]$, für das die Funktionswerte $z=f(x, y)$ berechnet werden sollen. Dabei enthält die Matrix x die x-Koordinaten der Punkte und y die zugehörigen y-Koordinaten.
- Mit Hilfe des erzeugten meshgrids können dann **komponentenweise** die z-Koordinaten berechnet werden z. B. $z = 1 ./ (1+x.^2+y.^2)$.
- Der Graph wird anschließend mit dem Befehl **surf**(x, y, z) (engl.: *Fläche*) oder **mesh**(x, y, z) (engl.: *Gitternetz*) gezeichnet.
- Die 3D-Graphik kann nach Anklicken von **Rotate 3D** (letzter Button *Toolbar*) mit der Maus gedreht werden. Die Blickrichtung kann alternativ mit dem **view**-Befehl geändert werden.
- Das verwendete Farbschema kann mit dem **colormap**-Befehl angepasst werden. Vordefinierte Farbpaletten sind z. B. **gray**, **hot**, **cool**, **spring**, **summer** usw.
- Der Befehl **colorbar** blendet die aktuell verwendete Farbskala ein.
- Flächen können auch „glatt“ dargestellt werden (shading interp).
- 3D-Graphen können beleuchtet werden. Der einfachste Befehl der zur Verfügung steht ist **camlight**. Der dazu verwendete Algorithmus kann mit **lighting** eingestellt werden: der **beste** und gleichzeitig langsamste Modus ist **lighting phong**.