

Übungsblatt 5

Ausgabe: 15.11.2017

3-dimensionale Plots

Für die Darstellung von 3-dimensionalen Plots bietet Matlab eine Vielzahl von Funktionen, die unterschiedliche Zwecke erfüllen und sich für unterschiedliche Szenarien eignen. Die Funktion `plot3(x,y,z)` (ein um eine Dimension ergänztes Pendant der Funktion `plot(x,y)`) eignet sich bspw. nur für Visualisierungen von Abbildungen des Typs $f: \mathbb{R} \rightarrow \mathbb{R}^3$. Grund dafür ist, dass `plot3` ausschließlich Linien und keine Flächen darstellt.

Zur Darstellung von Funktionen $g: \mathbb{R}^2 \rightarrow \mathbb{R}$ bzw. $g: [x_{min}, x_{max}] \times [y_{min}, y_{max}] \rightarrow \mathbb{R}$ bietet der Befehl `contour` die Möglichkeit eines Querschnitts- bzw. Konturplots. `contour` müssen allerdings zwei Vektoren x und y sowie eine Matrix $Z = (z_{i,j})$ mit $z_{i,j} = g(x_i, y_j)$ als Eingabeparameter übergeben werden. Um für eine beliebige Funktion $g: \mathbb{R}^2 \rightarrow \mathbb{R}$ eine solche Matrix Z nicht manuell aufstellen zu müssen, gibt es den Befehl `meshgrid`. Der Befehl `[X,Y] = meshgrid(x,y)` konvertiert zwei Vektoren x und y in zwei Matrizen X und Y , deren Zeilen bzw. Spalten Kopien der Ursprungsvektoren sind. Die Matrix Z kann nun über **elementweise** Berechnungen erstellt werden. Eine Darstellung der 2-dimensionalen Parabelfunktion $g(x,y) = x^2 + y^2$ auf dem Gebiet $[-3,3]^2$ mit `contour` sähe wie folgt aus:

```
x = -3:.01:3;
y = -3:.01:3;
[X,Y] = meshgrid(x,y);
Z = X.^2 + Y.^2;
contour(x,y,Z);
```

Um den erzeugten Plot besser interpretieren zu können, empfiehlt es sich, mittels des Befehls `colorbar` eine Höhenlegende hinzuzufügen. Der Befehl `colorbar('north')` bewirkt z.B., dass diese Höhenlegende oben innerhalb des Plots angezeigt wird. Weitere Eingabeargumente findet man unter `help colorbar`.

Aufgabe 19 (3-dimensionale Plots)

- a) Schreiben Sie ein Skript, in dem Sie die „Knollenabbildung“ $f(t): \mathbb{R} \rightarrow \mathbb{R}^3$ gegeben durch

$$f(t) = \begin{pmatrix} r(t) \cdot \cos(20\pi t) \\ r(t) \cdot \sin(20\pi t) \\ t \end{pmatrix} \quad \text{und} \quad r(t) = t \cdot \sin(t)$$

mit dem Befehl `plot3` darstellen. Wählen Sie eine geeignete Anzahl gleichverteilter Stützstellen im Intervall $t \in [0,4]$. Schalten Sie zusätzlich die Gitternetzlinien mit dem Befehl `grid` on ein.

- b) Sei $g(x,y) := \sin(3y - x^2 + 1) + \cos(2y^2 - 2x)$. Schreiben Sie ein Skript, in dem Sie $g(x,y)$ auf dem Gebiet $[x_{min}, x_{max}] \times [y_{min}, y_{max}] = [-2,2] \times [-1,1]$ durch die Erstellung eines `meshgrid` mittels `contour` darstellen. Fügen Sie rechts außen zusätzlich eine Höhenlegende hinzu.

Die Befehle `mesh` und `meshc` erwarten als Eingabe die gleichen Datentypen wie `contour`. Sie liefern drei-dimensionale Plots gemäß den „klassischen“ Vorstellungen eines Geländeplots. `meshc` legt in die x - y -Ebene zusätzlich eine Höhenkarte. Allerdings werden bei diesen Funktionen nur „netzartige“ Ebenen angezeigt. Ausgefüllt bekommt man die Ebenen, indem man anstelle von `mesh` und `meshc` die Befehle `surf` und `surfz` verwendet.

Aufgabe 20 (3-dimensionale Plots – 2)

Sei $h(x,y) := \sin(x) \cdot \cos(y)$.

- a) Schreiben Sie ein Skript, indem Sie $h(x,y)$ auf dem Gebiet $[x_{min}, x_{max}] \times [y_{min}, y_{max}] = [0,8]^2$ durch die Erstellung eines `meshgrid` sowohl mittels `mesh` als auch mittels `meshc` darstellen.
- b) Schreiben Sie ein Skript, indem Sie $h(x,y)$ auf dem Gebiet $[x_{min}, x_{max}] \times [y_{min}, y_{max}] = [0,8]^2$ durch die Erstellung eines `meshgrid` sowohl mittels `surf` als auch mittels `surfz` darstellen.

Für 3-dimensionale Plots können über den Befehl `colormap` verschiedene Farbschemata verwendet werden. Eine Beispieleingabe wäre `colormap('cool')`. Eine Übersicht über alle Farbschemata sowie weitere Optionen findet man unter `help graph3d`.

Wichtig: Beachten Sie, dass Sie trotz diverser Gestaltungsmöglichkeiten für Ihre Plots stets **sinnvolle** Designs verwenden sollten. Gerade bei Präsentationen oder in wissenschaftlichen Arbeiten empfiehlt es sich, für Plots Farbschemata zu verwenden, die gut zu erkennen und eindeutig zu interpretieren sind.

Plots von anonymen Funktionen

Wir haben bisher gesehen, wie man 2- und 3-dimensionale Plots mithilfe von Vektoren und Matrizen erzeugen kann. Man kann allerdings durch die sogenannten ez-Plots (amerikanische Aussprache: „easy-plots“) Darstellungen von anonymen Funktionen erzeugen. Hat man bereits eine 1-dimensionale anonyme Funktion $f(x)$ angelegt, so kann man diese mit dem Befehl `ezplot(f)` darstellen. Das standardmäßige Auswertungsintervall für ez-Plots ist $[x_{\min}, x_{\max}] = [-2\pi, 2\pi]$. Eine beliebige Achsenskalierung kann man über den allgemeineren Befehl `ezplot(f, [xmin xmax])` bzw. `ezplot(f, [xmin xmax ymin ymax])` festlegen.

Für eine 2-dimensionale anonyme Funktion $g(x, y)$ bewirkt der Befehl `ezplot(g)`, dass die implizit definierte Funktion $g(x, y) = 0$ geplottet wird. $g(x, y)$ wird also nach y aufgelöst.

Anders als bei `plot` muss eine Gestaltung des Graphen hier über einen `set`-Befehl separat erfolgen. So bewirkt die Eingabe von

```
h = ezplot('x^2 - y^4');  
set(h, 'color', 'g', 'LineWidth', 10, 'LineStyle', '--');
```

dass die Funktion $x^2 - y^4$ in grüner Farbe mit einer gestrichelten Linie in Stärke 10 geplottet wird. An diesem Beispiel sieht man ferner, dass es nicht notwendig ist, eine anonyme Funktion vor dem Plotten eigens zu initialisieren. Man kann diese als String direkt in `ezplot` eingeben. Beachten Sie aber, dass in diesem Fall auch bereits im Workspace definierte Konstanten als Variablen betrachtet werden.

Aufgabe 21 (ez-Plots anonymer Funktionen)

- Implementieren Sie die Funktion $f(x) := \frac{1}{(x-1)^2} + \frac{3}{(x-2)^2}$ als anonyme Funktion und plotten Sie diese mittels `ezplot` auf dem Intervall $[x_{\min}, x_{\max}] \times [y_{\min}, y_{\max}] = [0, 5] \times [0, 50]$.
- Bei dem Plot aus Aufgabenteil a) erkennt man, dass dieser an manchen Stellen relativ unglatt ist. Dies liegt daran, dass `ezplot` nur eine bestimmte Anzahl von Stützstellen auswählt und es daher bei größeren Sprüngen zu „Unsauberkeiten“ kommen kann. Im 1-dimensionalen Fall ist der Befehl `fplot` syntaktisch gleich zu `ezplot`. `fplot` verwendet bei Sprüngen in der Funktion jedoch eine größere Zahl von Auswertungsstellen. Wiederholen Sie Aufgabenteil a) daher mit dem Befehl `fplot`.
- Implementieren Sie die Funktion $g(x, y) := x^2 + y^2$ und plotten Sie die implizit definierte Funktion $g(x, y) = 1$. Wählen Sie eine angemessene Achsenskalierung. Färben Sie den Graphen rot und legen Sie die Linienstärke auf 2 fest.

Bemerkung: Sollten Ihnen beim Plotten der anonymen Funktionen eine Warnmeldung ausgegeben werden, so haben Sie die anonymen Funktionen vermutlich nicht elementweise definiert. Da `ezplot` intern Vektoren verwendet um die Auswertungen der anonymen Funktionen zu ermitteln, können Sie durch einen elementweise Initialisierung die Erstellung Ihrer Plots beschleunigen.

Wichtig: `ezplot` und `fplot` legen die Anzahl ihrer Auswertungsstellen selber fest. Möchten Sie eine eigene Anzahl an Auswertungsstellen festlegen, so müssen Sie auf die Funktion `plot` zurückgreifen. Hierzu müssen Sie Ihre anonyme Funktion jedoch vorab in einem Vektor auswerten.

Für mehrdimensionale Plots anonymer Funktionen bietet Matlab entsprechende ez-Pendants zu den bereits oben vorgestellten Funktionen `plot3`, `contour`, `mesh/meshc` und `surf/surfc` an (nämlich `ezplot3`, `ezcontour`, `ezmesh/ezmeshc` und `ezsurf/ezsurfc`). Die Argumente dieser Funktionen (außer `ezplot3`) beschränken sich im allgemeinen auf $(f, [xmin xmax ymin ymax])$, wobei f eine anonyme Funktion $f(x, y)$ der Form $f: \mathbb{R}^2 \rightarrow \mathbb{R}$ beschreibt. Bei `ezplot3` können bis zu drei Dimensionen eingetragen werden und für f gilt $f: \mathbb{R} \rightarrow \mathbb{R}^3$. Die Eingabe des Plotgebiets ist ebenfalls wieder optional, Standardgebiet ist $(-2\pi, 2\pi)^2$.

Aufgabe 22 (ez-Plots mehrdimensionaler anonymer Funktionen)

- Wiederholen Sie Aufgabe 19a mit dem Befehl `ezplot3`. Beachten Sie, dass Sie jede Dimension als eine eigene Funktion definieren müssen.
- Wiederholen Sie Aufgabe 20 mit den Befehlen `ezmesh` und `ezmeshc` bzw. `ezsurf` und `ezsurfc`.

Bemerkung: Die auf diesem und dem vorherigen Übungsblatt vorgestellten Befehle zur Darstellung von Funktionen sind die geläufigsten, jedoch bei weitem nicht alle, die Matlab anzubieten hat. Für eine ausführlichere Dokumentation aller Plot-Arten schauen Sie unter <http://de.mathworks.com/help/matlab/2-and-3d-plots.html>.