Matlab

Einführung und Plots

Matlab – Einführung und Plots

Übersicht

Inhaltsübersicht

- Vektoren und Matrizen
    - Eingabe & Zugriff
- Funktionsplots: der Befehl plot
    - Funktionen plotten & beschriften
- Vektorfelder: der Befehl quiver
- Skalarfelder: die Befehle mesh & contour

Matlab – Einführung und Plots

Was bedeutet MATLAB?

Abkürzung für…

MATLAB = MATrix LABoratory (und nicht: MAT = Math!)

- Matlab verwendet vorzugsweise Matrizen und Vektoren zur schnellen Berechnung!
- Messwerte werden als Matrizen oder Vektoren gespeichert

Matrizen in MATLAB

Beispiel:

Wir wollen die folgenden Matrizen implementieren:

$$A = \begin{pmatrix} 1 & -1 & 2 \\ -2 & 3 & 5 \end{pmatrix} \quad \text{und} \quad B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{pmatrix}$$

In MATLAB warden eckige Klammern zum Definieren von Matrizen verwendet:

```
A = [ 1 -1 2; -2 3 5];
B = [1 2; 3 4; 5 6; 7 8];
```

Ein Semikolon trennt die einzelnen Zeilen!

Vektoren in MATLAB

Beispiel:

Vektoren sind im Prinzip Matrizen mit nur einer Zeile oder Spalte…

Sie werden in MATLAB also analog mit eckigen Klammern definert:

```
x = [1;2;3];
y = [1 2 3];
```

x ist ein Spaltenvektor

y ist ein Zeilenvektor

Matrizen in MATLAB

Zugriff auf Matrixeinträge:

Um einzelne Einträge einer Matrix aufzurufen, warden runde Klammern verwendet…

Beispiele:

```
A = [1 2 3; 4 5 6];
A(2,1)
A(2,:)
A(:,3)
```

A(2,1) das Element in der 2. Zeile und 1. Spalte

A(2,:) alle Elemente in der 2. Zeile (: im 2. Argument bedeutet alle Spalten)

A(:,3) alle Elemente in der 3. Spalte (: im 1. Argument = alle Zeilen)

Plots – 1D Funktionen

Vorgehensweise

Wir wollen folgende 1D Funktion plotten: $f : \mathbb{R} \to \mathbb{R}$

Dazu gehen wir, wie beim Zeichnen per Hand vor:

- erstellen einer Wertetabelle
- Wertepaare in Koordinatensystem zeichen

In MATLAB sieht das wie folgt aus:

- definiere einen Vektor mit x-Daten
- berechne einen Vektor mit y-Daten, wobei y = f(x) ist
- zeichen die Daten mit plot(x,y)

Plots – 1D Funktionen

Beispiel:

Wir wollen den Sinus im Intervall $[0,2\pi]$ zeichnen:

```
x = 0:0.05:2*pi;
y = sin(x);
plot(x,y)
```

1. Zeile: Datenvektor x wird definiert (mit Colon Operator)

2. Zeile: es wird sin(x) berechnet und in y gesprichert

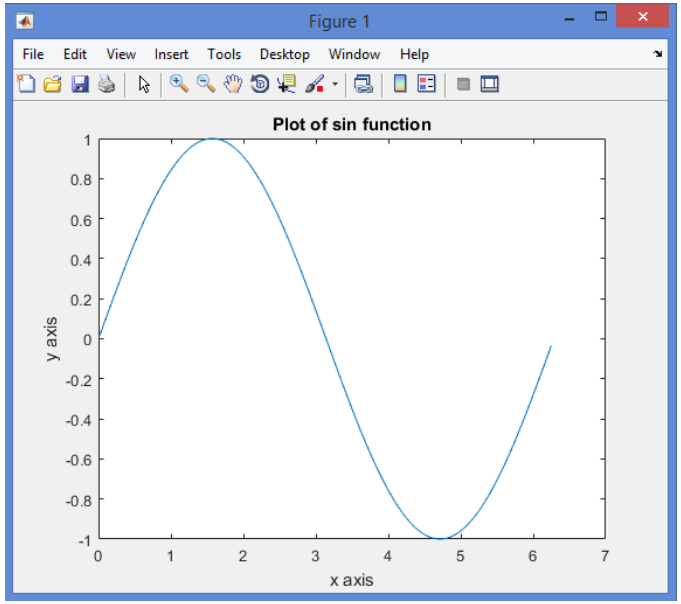
3. Zeile: wir plotten die Wertepaare (x,y)

Matlab – Einführung und Plots

Plots – 1D Funktionen

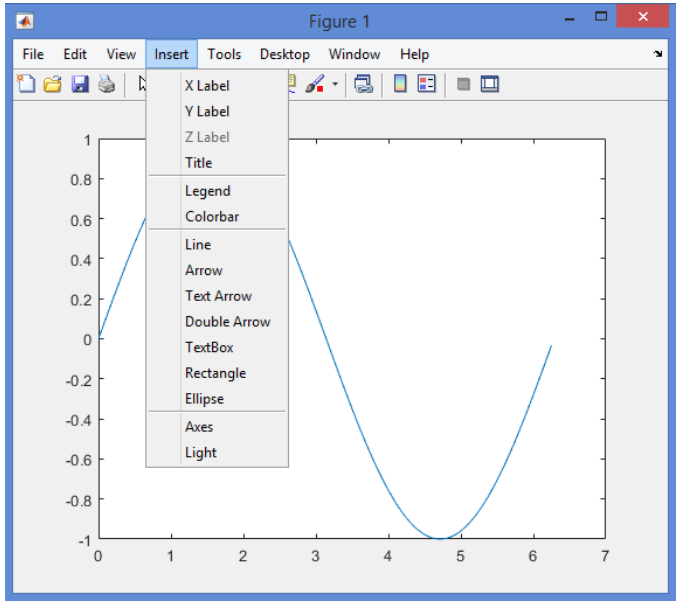
Figure Window

Als Ergebnis des plot Befehls erhalten wir ein Figure Window in dem unser Funktionsgraph gezeichnet ist.



Beschriftungen:

Wir können über das Menü “Insert” Beschriftungen für die Achsen einfügen: x Label und y Label



oder im Command Window:

```
xlabel('x axis')
ylabel('y axis')
title('Plot of sin function')
```

Matlab – Einführung und Plots

Plots – 1D Funktionen

Mehrere Funktionen in einen Plot

Es gibt zwei Möglichkeiten, mehrere Funktionen in einem Plot zu zeichnen…

Direkt mit einem Befehl:

```
x = 0:0.05:2*pi;
plot(x,sin(x),x,cos(x))
```

Hier können für die zweite Funktion auch andere x-Daten verwendet werden.

Nacheinander mit hold on:

```
x = 0:0.05:2*pi
plot(x,sin(x))
hold on
plot(x,cos(x))
hold off
```

hold on: alle plot Befehle werden ergänzt

hold off: beim Plotten werden alte Plots überschrieben

Matlab – Einführung und Plots

Plots – 1D Funktionen

Dot Operator beim Plot von Produkten:

Wir wollen folgende Funktion plotten: $f(x) = \sin(x) \cdot \cos(x)$

MATLAB Befehle:

```
x = 0:0.05:2*pi;
plot(x,sin(x).*cos(x))
```

Achtung:

Der Dot Operator „.*“ mit dem Punkt ist nötig, damit Sinus und Kosinus wirklich elementweise, d.h. für jeden x-Wert mutlipliziert werden.

Sonst erhalten wir folgende Fehlermeldung

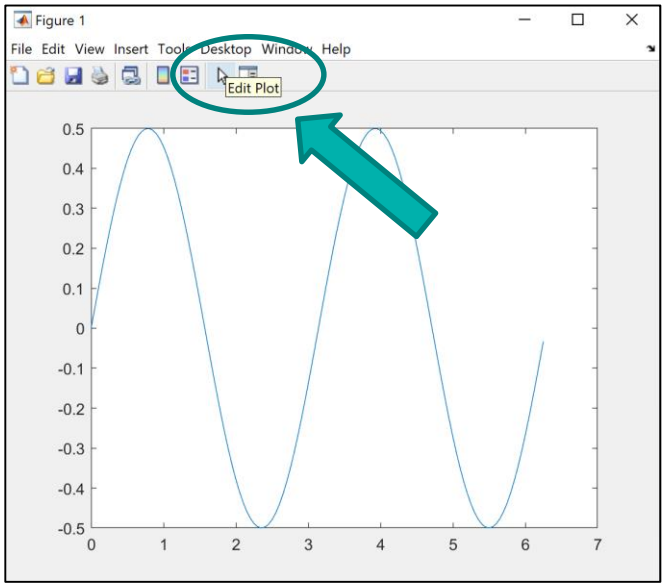
```
Error using  *
Inner matrix dimensions must agree.
```

Matlab – Einführung und Plots

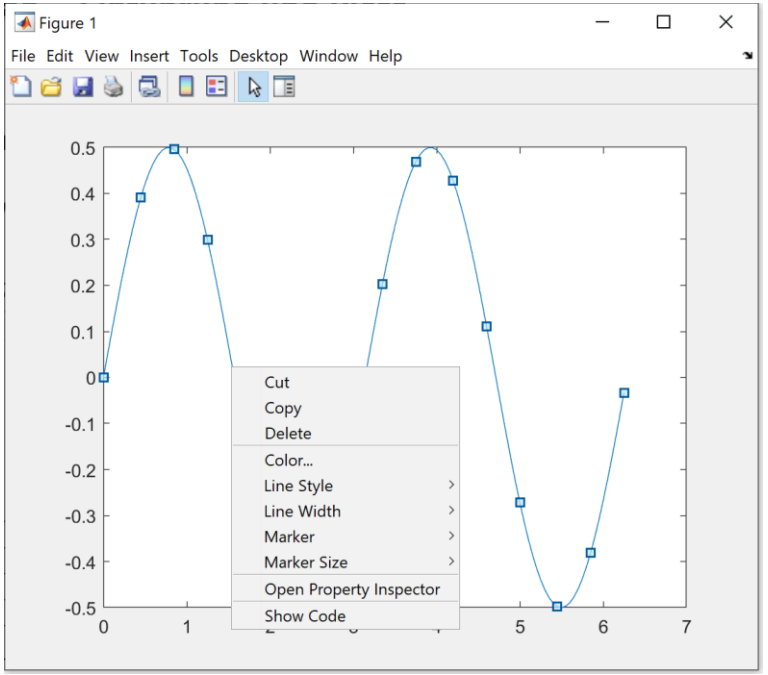
Plots – 1D Funktionen

Eigenschaften ändern:

Auswahlwerkzeug aktivieren…



Graph auswählen & rechte Maustaste…



Matlab – Einführung und Plots

Skalarfelder: Plots

$$f(x,y) = \sin x \cdot \cos y$$

Mesh Grid erzeugen…

Wir wollen ein Skalarfeld plotten…

Wir definieren zuerst den Bereich…

```
>> x = -4:0.1:4;
>> y = -4:0.1:4;
```

Wir benötigen ein Mesh-Grid…

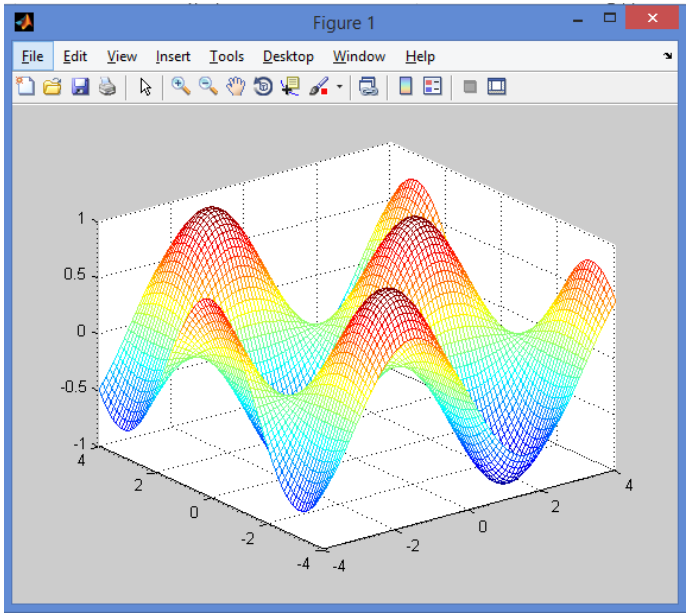
```
>> [X,Y] = meshgrid(x,y);
```

Wir plotten das Skalarfeld mit Mesh…

```
>> mesh(X,Y,sin(X).*cos(Y))
```



Achtung: dot Operator!



Matlab – Einführung und Plots

Skalarfelder: Plots

Mehr zu Meshgrids

Was ist ein Meshgrid und wozu?

Ziel:

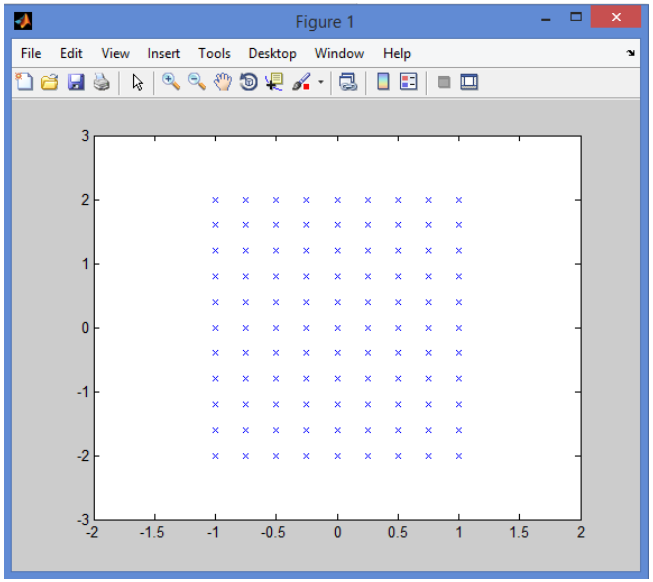
Wir wollen ein Skalar- oder Vektorfeld an diskret verteilten Punkten in der x-y-Ebene plotten.

Zuerst definieren wir den Bereich…

```
>> x = -1:0.25:1;
>> y = -2:0.4:2;
```

Dann erstellen wir die Mesh-Punkte…

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




```
X =

   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000
   -1.0000   -0.7500   -0.5000   -0.2500         0    0.2500    0.5000    0.7500    1.0000

Y =

   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000   -2.0000
   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000   -1.6000
   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000   -1.2000
   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000   -0.8000
   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000   -0.4000
         0         0         0         0         0         0         0         0         0
    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000    0.4000
    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000    0.8000
    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000    1.2000
    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000    1.6000
    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000    2.0000
```

Matlab – Einführung und Plots

Skalarfelder: Plots

Countor Plots

Wir wollen die Höhenlinien des folgenden Skalarfeldes plotten:

$$f(x,y) = \sin x \cdot \cos y$$

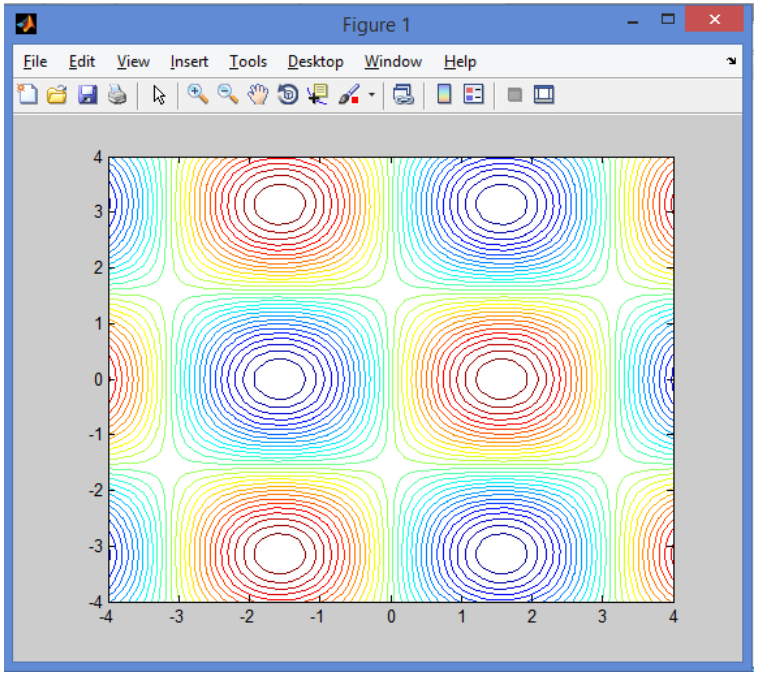
Wir definieren wieder als erstes den Bereich…

```
>> x = -4:0.1:4;
>> y = -4:0.1:4;
>> [X,Y]=meshgrid(x,y);
```

Dann plotten wir die Höhenlinien…

```
>> contour(X,Y,sin(X).*cos(Y),30)
```

optional: Anzahl der Höhenlinien



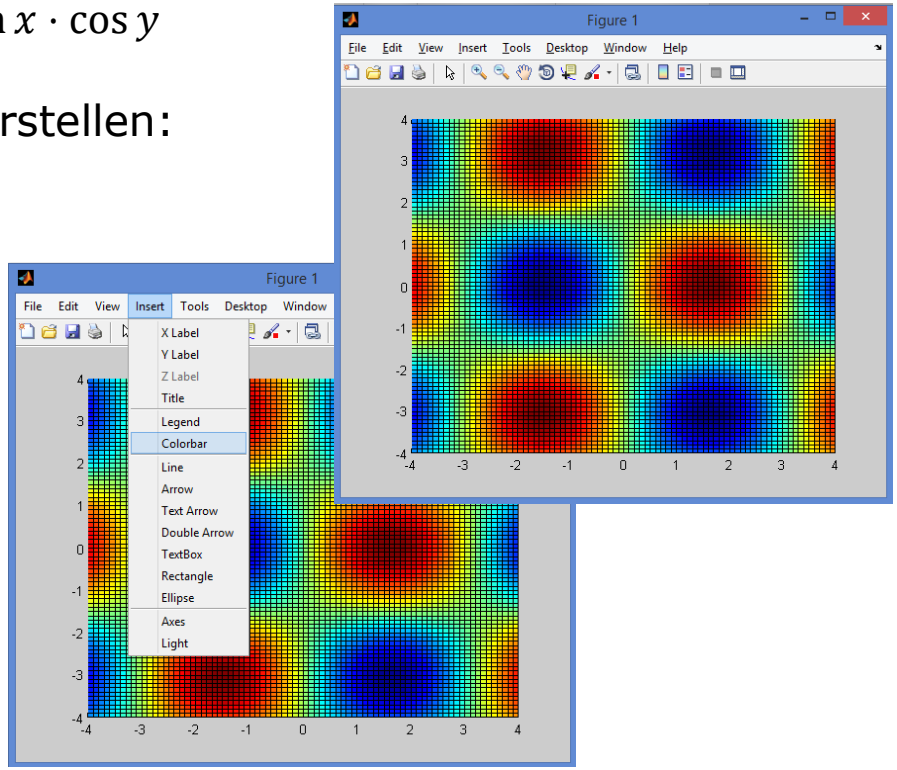
Matlab – Einführung und Plots

Skalarfelder: Plots

Weiterer Plot für Skalarfelder

Wir betrachten das Skalarfeld:

$$f(x,y) = \sin x \cdot \cos y$$



Wir können auch ein „pseudocolor plot" erstellen:

Definition Mesh:

```
>> x = -4:0.1:4;
>> y = -4:0.1:4;
>> [X,Y]=meshgrid(x,y);
```

Plot durchführen:

```
>> pcolor(X,Y,sin(X).*cos(Y))
```

Bemerkung:

Wir können/sollten immer einen Colorbar hinzufügen!

Matlab – Einführung und Plots

Vektorfelder: Plots

Der Befehl quiver

$$F(x,y) = \begin{pmatrix} x \\ x^2 + y^2 \end{pmatrix}$$

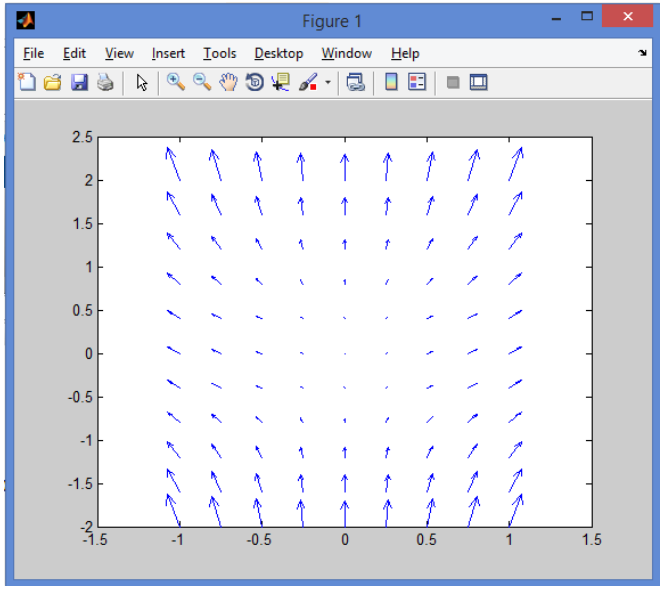
Wir definieren wieder zuerst das Mesh-Grid…

```
>> x = -1:0.25:1;
>> y = -2:0.4:2;
>> [X,Y] = meshgrid(x,y);
```

Anschließend definieren wir einzeln die Komponenten des Vektorfeldes…

```
>> vx = X;
>> vy = X.^2+Y.^2;
```

dot Operator!



Wir plotten das Vektorfeld…

```
>> quiver(X,Y,vx,vy)
```

Matlab – Einführung und Plots

Skalarfeld mit Gradientenfeld

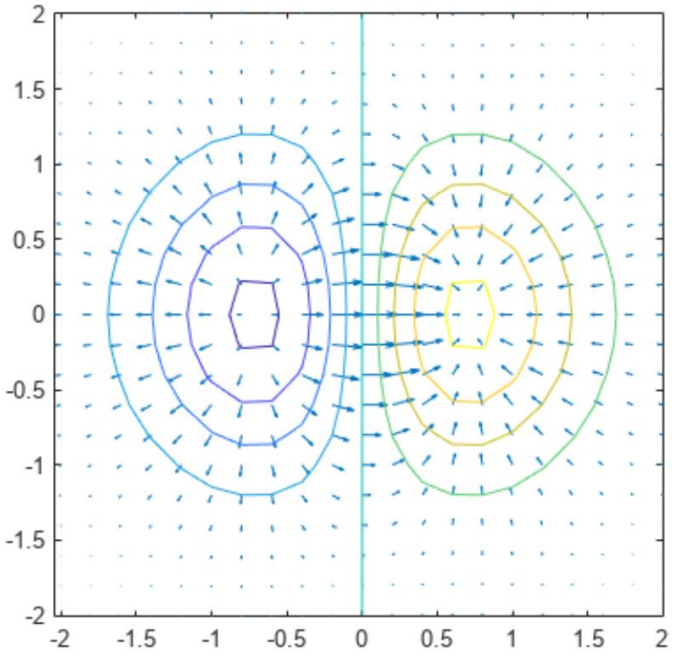
$$z = xe^{-x^2-y^2}$$

Kombination von contour und quiver

Wir wollen einen Höhenplot eines Skalarfeldes erstellen
und die Gradienten an unterschiedlichen Stellen einzeichnen.

```
spacing = 0.2;
[X,Y] = meshgrid(-2:spacing:2);
Z = X.*exp(-X.^2 - Y.^2);
[DX,DY] = gradient(Z,spacing);
```

```
quiver(X,Y,DX,DY)
hold on
contour(X,Y,Z)
axis equal
hold off
```



-2:spacing:2 ersetzt die Eingabe von x und y…

hold on … hold off sorgt dafür, dass beide Plots in eine Figure gemacht werden.