



# Scilab

## 3D Grafik

J. Koch, 6. Januar 2018

## 1. 3D Grafik

## 1. 3D Grafik

Gitternetz

Schaubild von  $f(x, y)$

Sombrero

Farbpalette

Höhenlinien

Kugel

Vektorfeld

- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.

- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.

- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.

- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.

- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.



- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.

- ▶ Sie können eine Matrix mithilfe eines Gitternetzes visualisieren
- ▶ Sie können die Achsenbeschriftungen anpassen
- ▶ Sie können Schaubilder von Funktionen in zwei Veränderlichen erstellen
- ▶ Sie können die Farbpalette bei Grafiken in 3D verändern
- ▶ Sie können Höhenlinien einer Fläche visualisieren
- ▶ Sie können geschlossene Flächen in 3D (Kugel, Torus, ...) visualisieren
- ▶ Sie können ebene Vektorfelder grafisch darstellen.

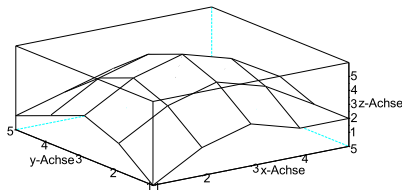
## ▶ Gitternetz

```
—> A = [0 2 3 2 2;
        2 4 5 4 2;
        3 5 6 5 3;
        2 4 5 4 2;
        1 2 3 2 0];
```

```
—> mesh(A)
```

## ▶ Achsenbeschriftung

```
—> a = gca();
—> a.font_size = 8;
—> xlabel('x-Achse','fontsize',8)
—> ticks = a.x_ticks
—> ticks.labels = ['1';'2';'3';'4';'5']
—> ticks.locations = [1;2;3;4;5]
—> a.x_ticks = ticks
```



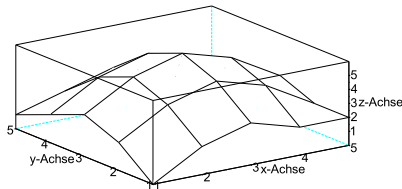
## ► Gitternetz

```
—> A = [0 2 3 2 2;  
        2 4 5 4 2;  
        3 5 6 5 3;  
        2 4 5 4 2;  
        1 2 3 2 0];
```

```
—> mesh(A)
```

## ► Achsenbeschriftung

```
—> a = gca();  
—> a.font_size = 8;  
—> xlabel('x-Achse','fontsize',8)  
—> ticks = a.x_ticks  
—> ticks.labels = ['1'; '2'; '3'; '4'; '5']  
—> ticks.locations = [1;2;3;4;5]  
—> a.x_ticks = ticks
```



## ► Schaubild von $f(x, y)$

—>  $x = -2:2$ ;

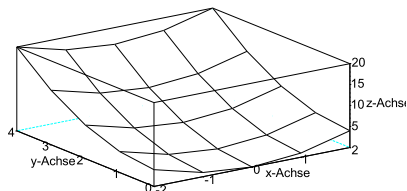
—>  $y = 0:4$ ;

—>  $[X, Y] = \text{meshgrid}(x, y)$

```
X =  
- 2.  - 1.   0.   1.   2.  
- 2.  - 1.   0.   1.   2.  
- 2.  - 1.   0.   1.   2.  
- 2.  - 1.   0.   1.   2.  
- 2.  - 1.   0.   1.   2.  
Y =  
 0.   0.   0.   0.   0.  
 1.   1.   1.   1.   1.  
 2.   2.   2.   2.   2.  
 3.   3.   3.   3.   3.  
 4.   4.   4.   4.   4.
```

—>  $F = X.^2 + Y.^2$ ;

—>  $\text{mesh}(X, Y, F)$ ;

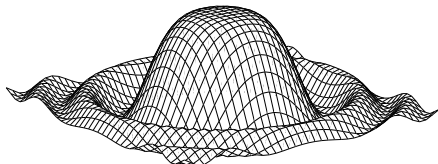


$$f(x, y) = x^2 + y^2$$

```
F =  
 4.   1.   0.   1.   4.  
 5.   2.   1.   2.   5.  
 8.   5.   4.   5.   8.  
13. 10.   9. 10. 13.  
20. 17. 16. 17. 20.
```

## ► Sombrero

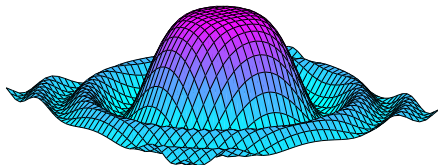
$$f(x, y) = \frac{\sin(x^2 + y^2)}{x^2 + y^2}$$



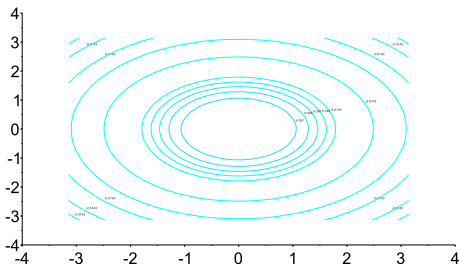
```
—> xy = linspace(-%pi, %pi, 40);  
—> [X,Y] = meshgrid(xy);  
—> F = sin(X.^2+Y.^2)./(X.^2+Y.^2);  
—> mesh(X,Y,F)
```

## ► Farbpalette

$$f(x, y) = \frac{\sin(x^2 + y^2)}{x^2 + y^2}$$



```
—> xy = linspace(-%pi, %pi, 40);  
—> [X,Y] = meshgrid(xy);  
—> F = sin(X.^2+Y.^2)./(X.^2+Y.^2);  
—> surf(X,Y,F)  
—> f = gcf();  
—> f.color_map = coolcolormap(32);
```

$$\frac{\sin(x^2 + y^2)}{x^2 + y^2} = \text{const}$$


```

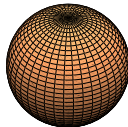
—> xy = linspace(-%pi,%pi,40);
—> [X,Y] = meshgrid(xy);
—> F = sin(X.^2+Y.^2)./(X.^2+Y.^2);
—> contour(xy,xy,F)

```



## ► Kugel

$$S(u, v) = \begin{pmatrix} \sin(u) \cos(v) \\ \sin(u) \sin(v) \\ \cos(u) \end{pmatrix}$$



```
—> u = linspace(0,%pi,40);  
—> v = linspace(0,2*%pi,40);  
—> [U,V] = meshgrid(u,v);  
—> X = sin(U).*cos(V);  
—> Y = sin(U).*sin(V);  
—> Z = cos(U);  
—> surf(X,Y,Z);  
—> f = gcf();  
—> f.color_map = coppercolormap(32);
```

## ► Vektorfeld

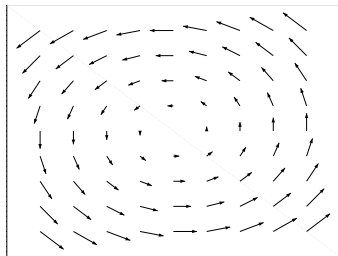
$$\vec{v}(x, y) = \begin{pmatrix} v_x(x, y) \\ v_y(x, y) \end{pmatrix}$$

```
—> x = -2:0.5:2;  
—> y = -2:0.5:2;  
—> [X,Y] = meshgrid(x, y);  
—> Vx = -Y;  
—> Vy = X;  
—> champ(x, y, Vx', Vy');
```

► Matrizen transponieren!

## ► Beispiel

$$\vec{v}(x, y) = \begin{pmatrix} -y \\ x \end{pmatrix}$$



## ► Vektorfeld

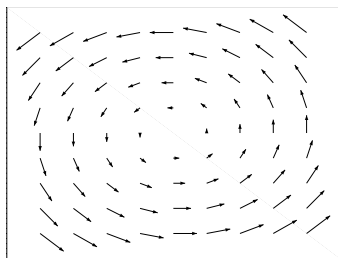
$$\vec{v}(x, y) = \begin{pmatrix} v_x(x, y) \\ v_y(x, y) \end{pmatrix}$$

```
—> x = -2:0.5:2;  
—> y = -2:0.5:2;  
—> [X,Y] = meshgrid(x, y);  
—> Vx = -Y;  
—> Vy = X;  
—> champ(x, y, Vx', Vy');
```

## ► Matrizen transponieren!

## ► Beispiel

$$\vec{v}(x, y) = \begin{pmatrix} -y \\ x \end{pmatrix}$$



1. Erstellen Sie ein Schaubild der Funktion

$$f(x, y) = -4 x e^{-x^2 - y^2},$$

für

$$(x, y) \in [-3, 3]^2.$$

2. Erstellen Sie ein Schaubild des Torus

$$\begin{pmatrix} (2 + \cos(u)) \cos(v) \\ (2 + \cos(u)) \sin(v) \\ \sin(u) \end{pmatrix}$$

für

$$(u, v) \in [0, 2\pi]^2.$$