



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

## Kapitel 5

# Differenzialrechnung mehrdimensionaler Funktionen

# Inhalt

## 5.1 Mehrdimensionale Funktionen

Darstellungen, Netzlinien, Niveaulinien

## 5.2 Partielle Ableitungen

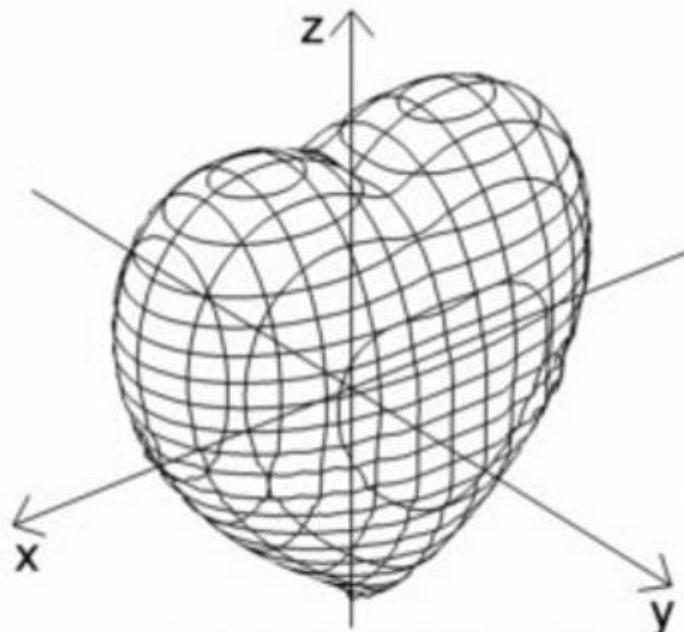
Tangentialebene, Gradient, Richtungsableitung

## 5.3 Extrema

Hesse-Matrix, Extrema, Gradientenabstiegsverfahren

## 5.1 Mehrdimensionale Funktionen

/



$$(x^2+9/4)y^2+z^2-1)^3-x^2z^3-9/80y^2z^3=0$$

$$-3 \leq x, y, z \leq 3$$

Math

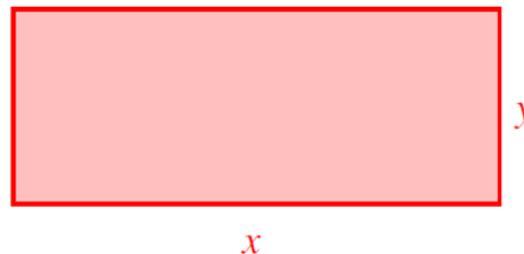
# Warum Funktionen von mehreren Variablen?

Viele Funktionen in Naturwissenschaft, Technik und Informatik hängen von mehreren Variablen ab.

## Einfache Beispiele:

- (a) Der Flächeninhalt eines Rechtecks hängt von Länge und Breite ab:

$$\text{Rechtecksfläche} = f(x, y) = x \cdot y$$



- (b) Die Temperatur in diesem Raum hängt von allen drei räumlichen Koordinaten ab:  $T(x, y, z)$

## Praxisrelevante Beispiele

**Computergrafik:** Darstellung von Flächen im Raum

**Information Retrieval:** Dokumente bestehen aus Tausenden von Termen

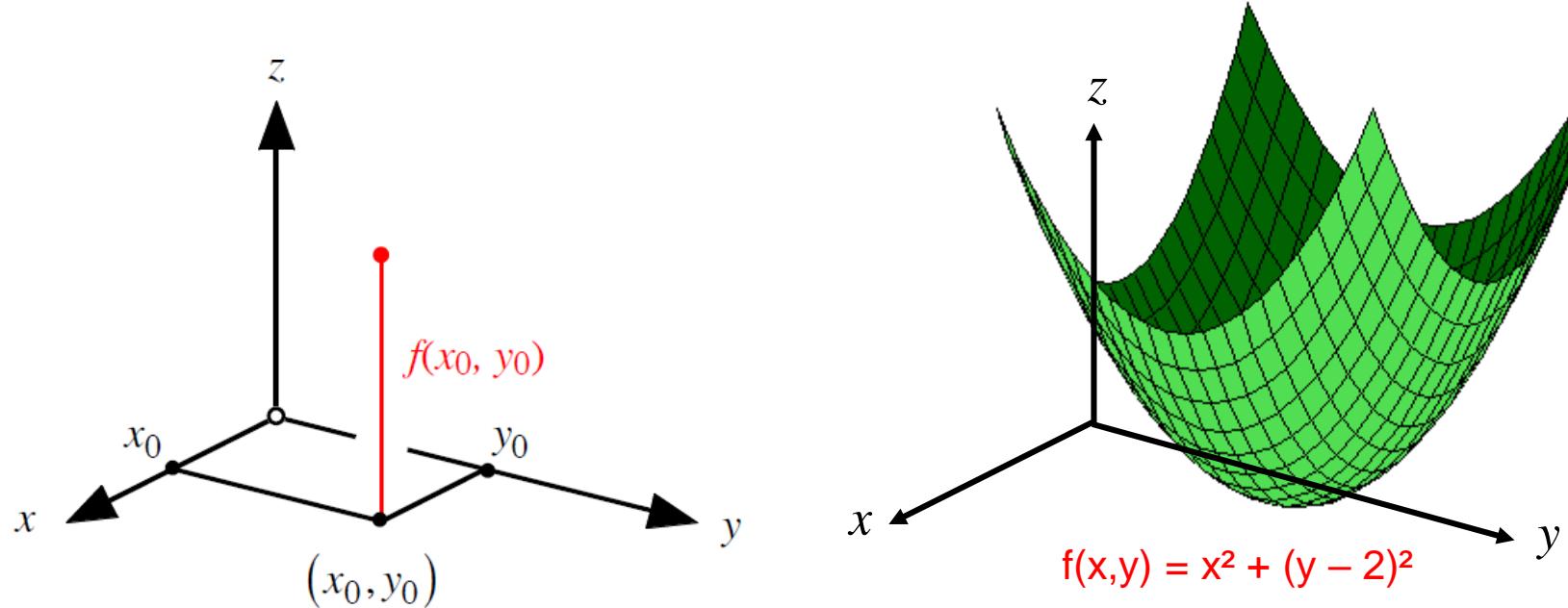
**Bildverarbeitung:** Bilder bestehen aus Millionen von Pixeln

**BWL:** Betriebswirtschaftliche Größen wie Gewinn und Verbrauch hängen von mehreren Variablen ab

**Physik:** Phänomene wie Bewegung, Kräfte oder Felder sind mehrdimensional

## Darstellung von $f(x, y)$

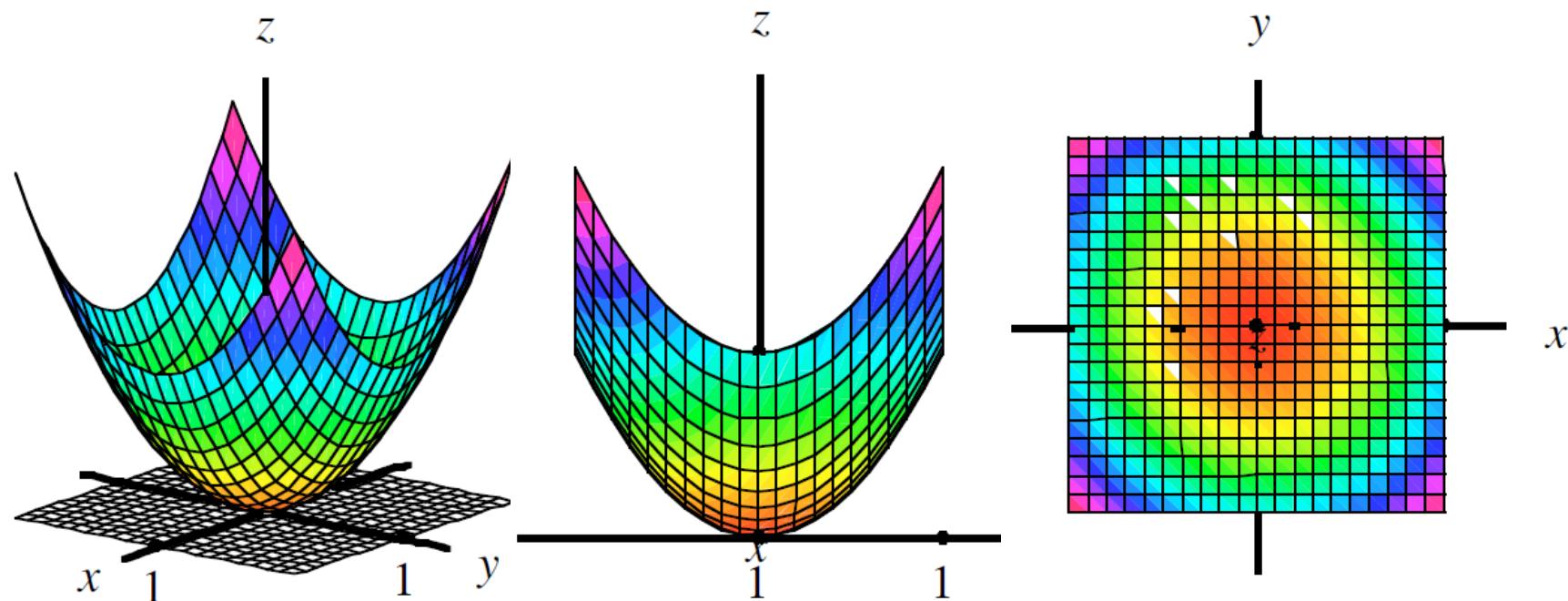
Darstellung von  $z = f(x, y)$  im räumlichen Koordinatensystem:



Der gesamte Graph wird zu einer **Fläche** über der x-y-Ebene.

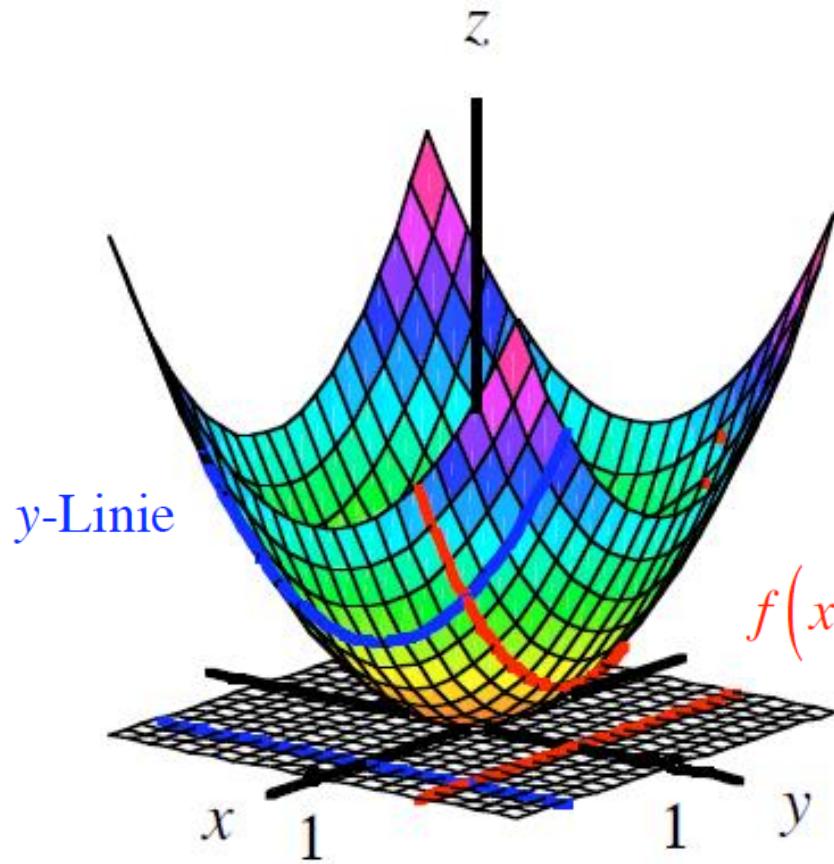
# 1. Beispiel: Paraboloid

**Beispiel:** Die Funktion  $f(x, y) = x^2 + y^2$  in verschiedenen Darstellungen:



**Schrägbild und Sicht von vorne und von oben**

## Netzlinien



**x-Linie:** x ist variabel, y ist fest

**y-Linie:** x ist fest, y ist variabel

*x-Linie*

$$f\left(x, \frac{1}{2}\right) = x^2 + \left(\frac{1}{2}\right)^2 = x^2 + \frac{1}{4}$$

# Niveaulinien

**Niveaulinien** (Höhenlinien) sind horizontale Schnitte auf bestimmten Höhen. Sie werden in der x-y-Ebene dargestellt.

**Beispiel:**  $f(x) = x^2 + y^2$

Niveau 0:  $f(x, y) = x^2 + y^2 = 0$  Niveaulinie: Nullpunkt

Niveau 1:  $f(x, y) = x^2 + y^2 = 1$  Niveaulinie: Einheitskreis

Niveau 2:  $f(x, y) = x^2 + y^2 = 2$  Niveaulinie: Kreis mit Radius  $\sqrt{2}$

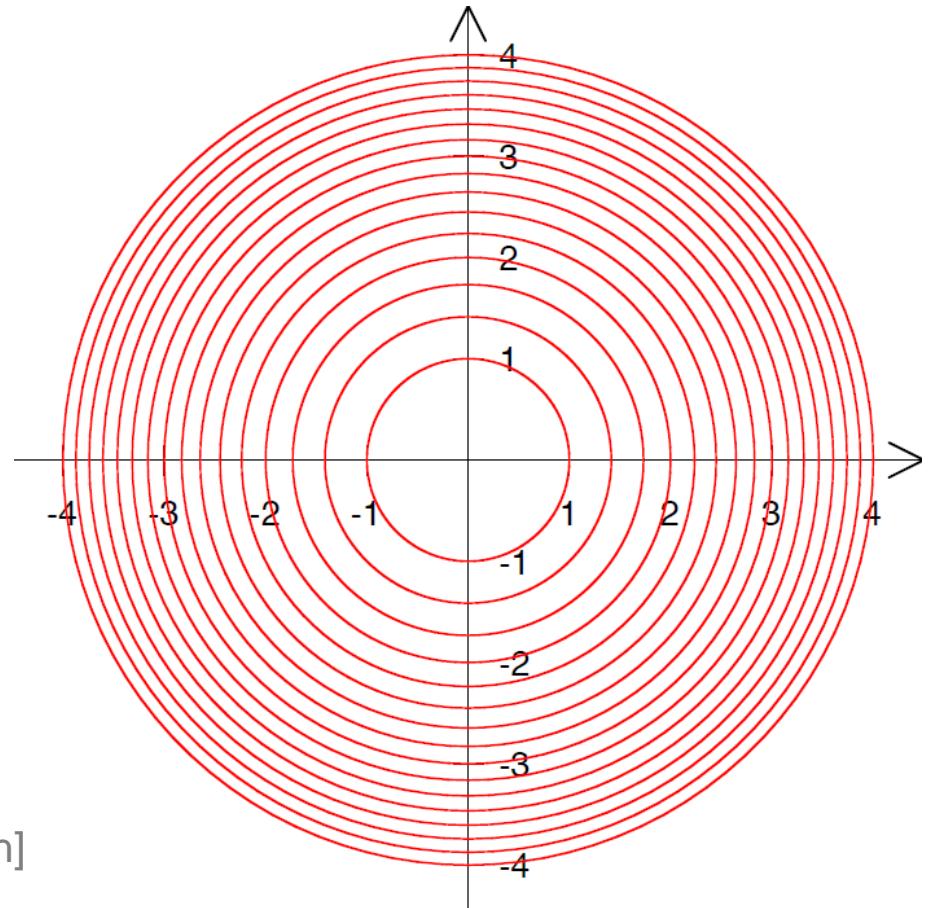
Allgemein:

Niveau  $c$ :  $f(x, y) = x^2 + y^2 = c$  Niveaulinie: Kreis mit Radius  $\sqrt{c}$

## Niveaulinien für die Niveaus 1, 2, ..., 16

Je dichter die Niveaulinien,  
umso steiler ist die Fläche.

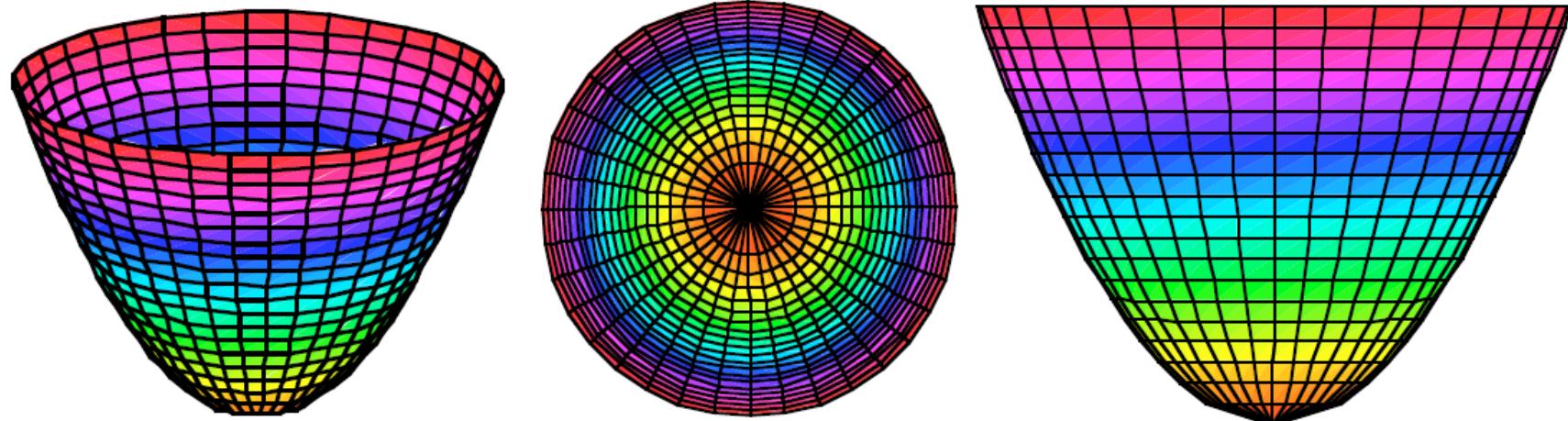
Wenn die Niveaulinien Kreise  
sind, ist die Fläche rotations-  
symmetrisch zur z-Achse.



GeoGebra: ImpliziteKurve[f(x,y) - h]

## Niveaulinien im Schrägbild, Grund- und Aufriss

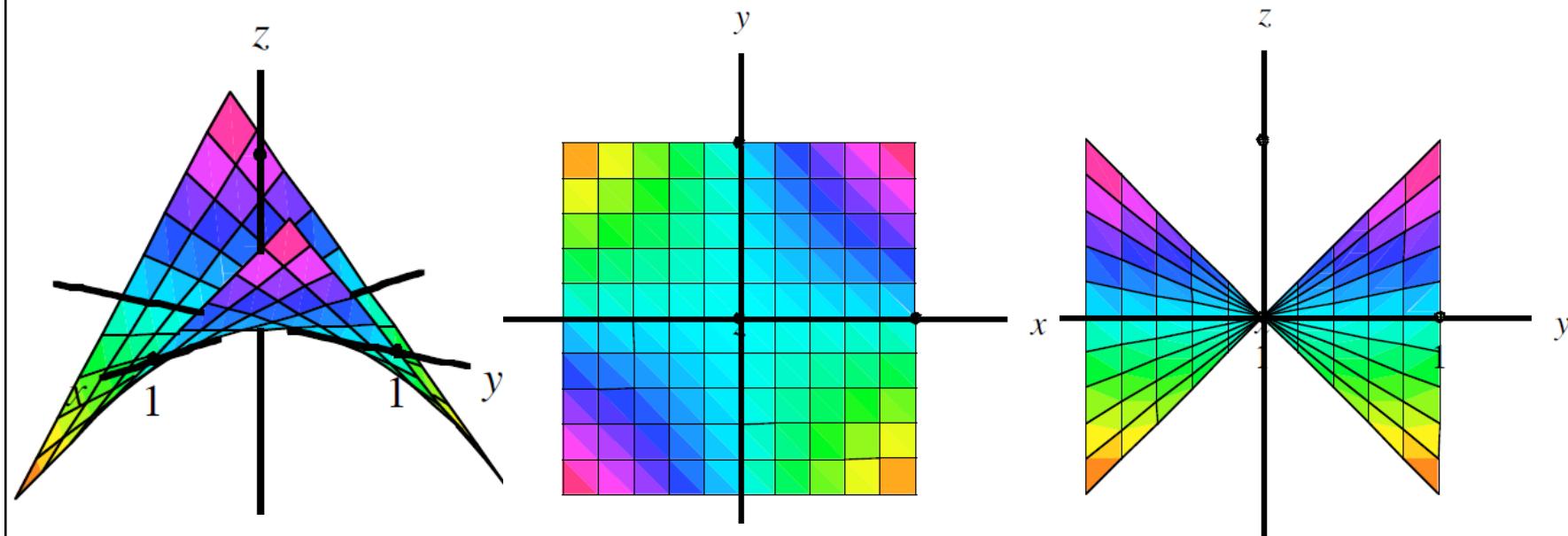
$f(x, y) = x^2 + y^2$  beschreibt ein **Paraboloid**. Im Grundriss sind die Niveaulinien Kreise, im Aufriss horizontale Geraden.



Paraboloid: Schrägbild, Grundriss, Aufriss

## 2. Beispiel: Sattelfläche

**Beispiel:** Die Funktion  $f(x, y) = x \cdot y$  heißt **Sattelfläche**. Sie beschreibt z. B. die Fläche eines Rechtecks.



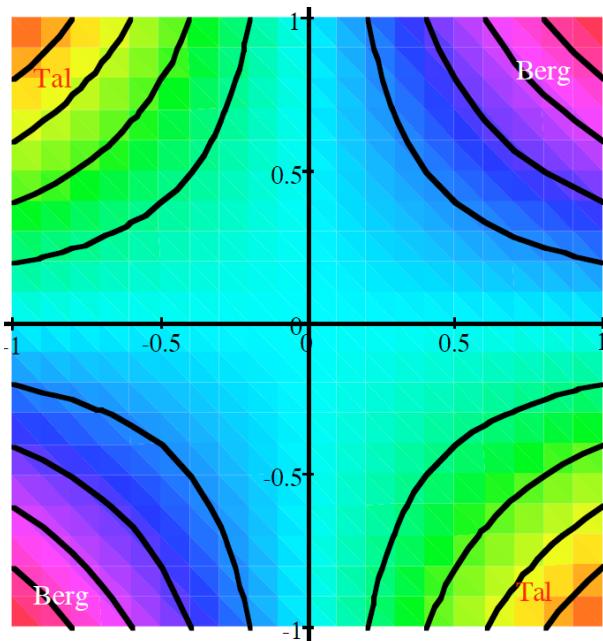
Ansicht, Sicht von oben, Sicht von vorne

## Niveaulinien für die Niveaus $-1, -0.9, \dots, 0.9, 1$

Für die Niveaulinien der Sattelfläche gilt:

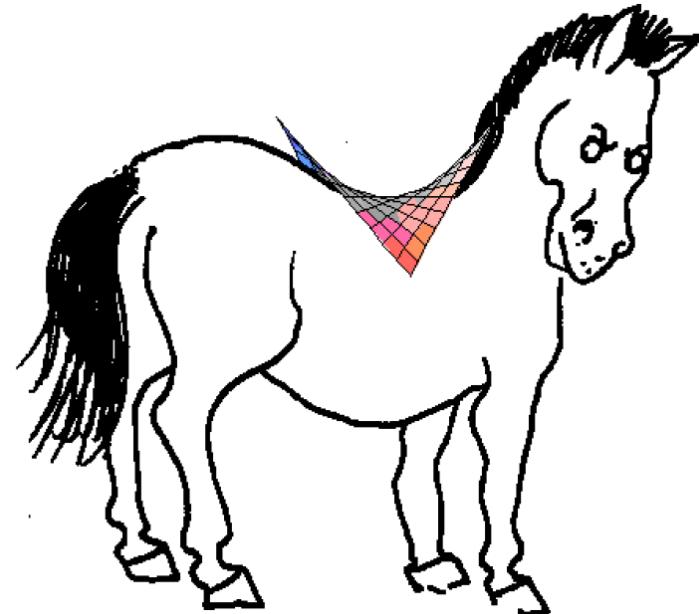
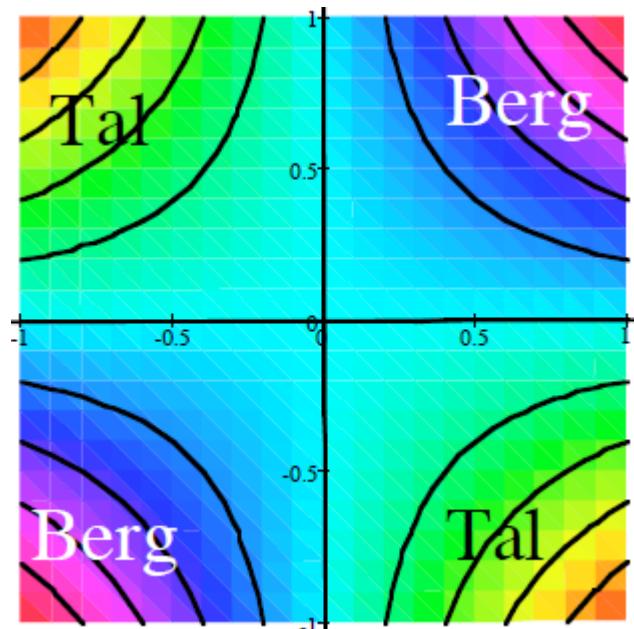
$$\text{Niveau } 0: f(x, y) = x \cdot y = 0 \quad \rightarrow \quad x = 0 \text{ oder } y = 0 \text{ (Achsen)}$$

$$\text{Niveau } c: f(x, y) = x \cdot y = c \quad \rightarrow \quad y = c/x \text{ (Hyperbeln)}$$



## Sattelfläche: Täler, Berge und ein Sattelpunkt

Im 1. und 3. Quadranten haben wir positive Funktionswerte, im 2. und 4. Quadranten negative Funktionswerte. Der Nullpunkt ist ein **Sattelpunkt**: In zwei Richtungen geht es hinauf, in zwei Richtungen hinunter.



## 5.2 Partielle Ableitungen



# Partielle Ableitungen

Trick:  $y = y_0$  setzen und „einfrieren“!

Ableitung von  $f(x, y_0)$  bezüglich  $x$  an der Stelle  $(x_0, y_0)$ :

**Partielle Ableitung nach x** (Ableitung in x-Richtung):

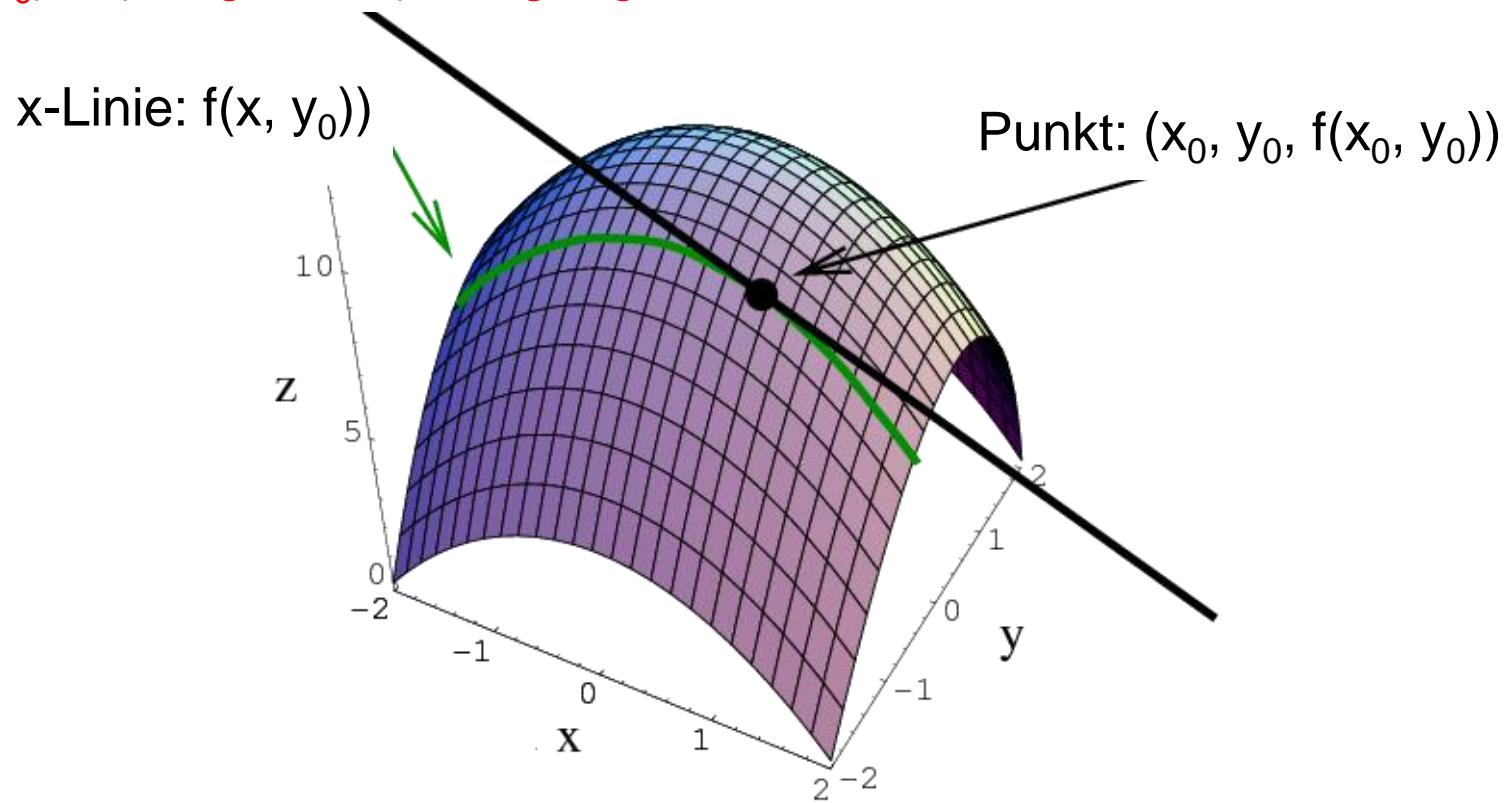
$$f_x(x_0, y_0) = \frac{\partial f}{\partial x}(x_0, y_0) = \lim_{x \rightarrow x_0} \frac{f(x, y_0) - f(x_0, y_0)}{x - x_0}$$

Analog: **Partielle Ableitung nach y** (Ableitung in y-Richtung):

$$f_y(x_0, y_0) = \frac{\partial f}{\partial y}(x_0, y_0) = \lim_{y \rightarrow y_0} \frac{f(x_0, y) - f(x_0, y_0)}{y - y_0}$$

# Geometrische Interpretation der partiellen Ableitung

$f_x(x_0, y_0) = (\text{Tangenten-}) \text{Steigung der } x\text{-Linie}$



## Beispiel und Übung

**Beispiele:** (a)  $f(x, y) = x^2 + y^2 \Rightarrow f_x(x, y) = 2x$   
 $f_y(x, y) = 2y$

(b)  $f(x, y) = x^3 - 3xy^2 \Rightarrow f_x(x, y) = 3x^2 - 3y^2$   
 $f_y(x, y) = -6xy$

**Übungen:** (c)  $f(x, y) = x \cdot y \Rightarrow f_x(x, y) =$   
 $f_y(x, y) =$

(d)  $f(x, y, z) = x \cdot y^2 \cdot z^3 \Rightarrow f_x(x, y, z) =$   
 $f_y(x, y, z) =$   
 $f_z(x, y, z) =$

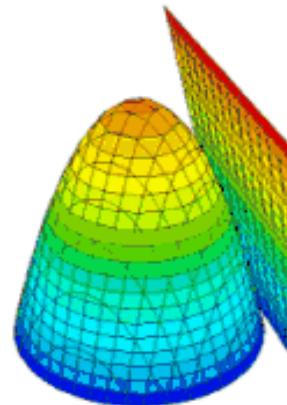
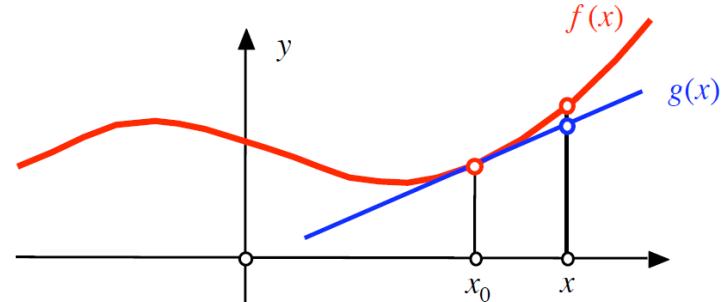
# Tangentialebenen

Erinnerung: **Tangente an Funktion  $f(x)$**

$$g(x) = f(x_0) + f'(x_0)(x - x_0)$$

Analog: **Tangentialebene:**

$$g(x, y) = f(x_0, y_0) + f_x(x_0, y_0)(x - x_0) + f_y(x_0, y_0)(y - y_0)$$

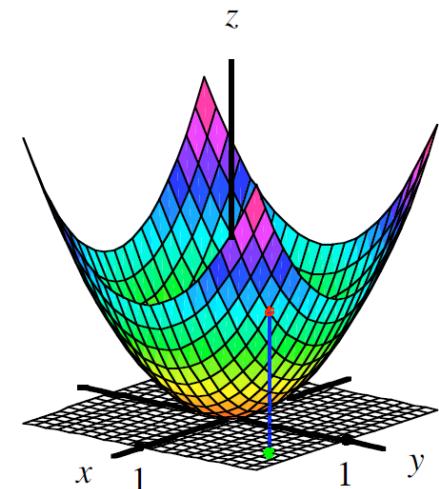


## Beispiel

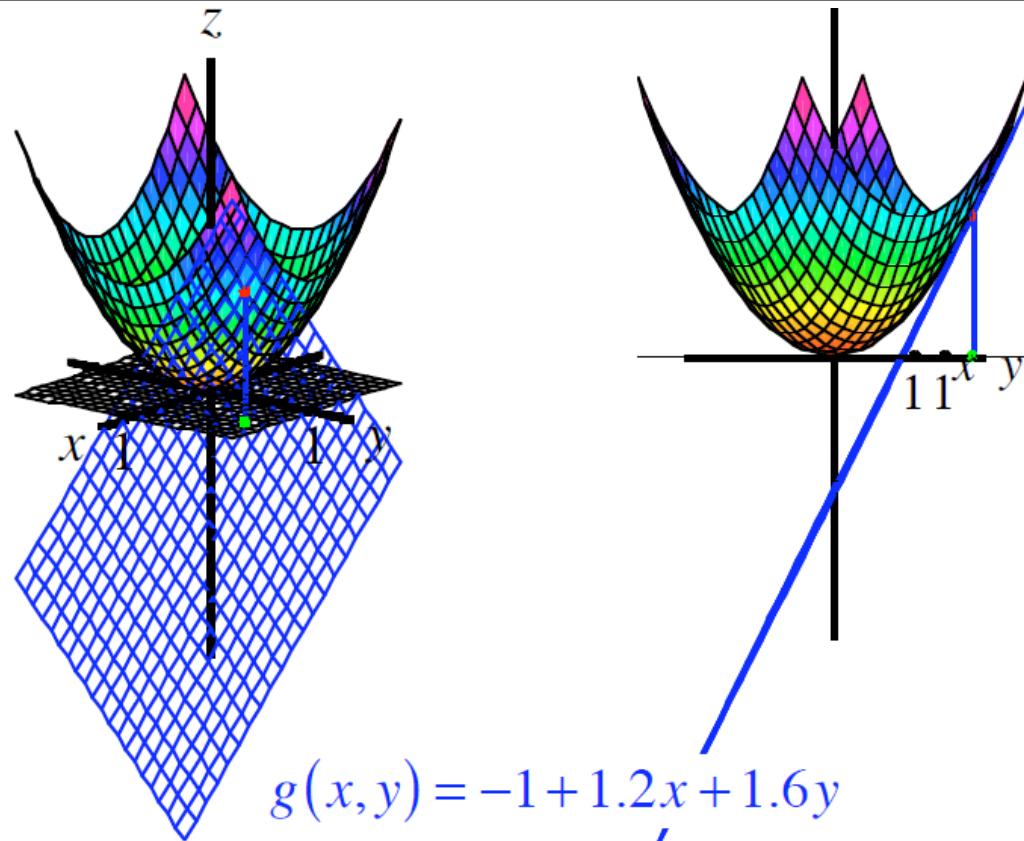
Paraboloid:  $f(x, y) = x^2 + y^2$  mit  $f_x(x, y) = 2x$  und  $f_y(x, y) = 2y$

**Tangentialebene** an der Stelle  $(0.6, 0.8)$ :

$$\begin{aligned}g(x, y) &= f(0.6, 0.8) + f_x(0.6, 0.8)(x - 0.6) + f_y(0.6, 0.8)(y - 0.8) \\&= 0.6^2 + 0.8^2 + 2 \cdot 0.6 \cdot (x - 0.6) + 2 \cdot 0.8 \cdot (y - 0.8) \\&= 0.36 + 0.64 + 1.2 \cdot (x - 0.6) + 1.6 \cdot (y - 0.8) \\&= 1 + 1.2x - 0.72 + 1.6y - 1.28 \\&= -1 + 1.2x + 1.6y\end{aligned}$$



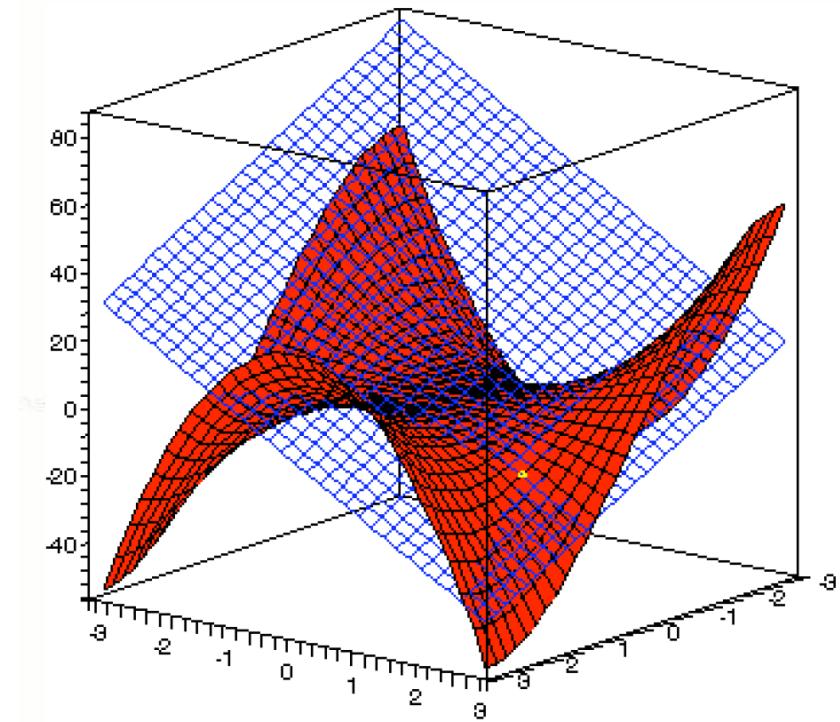
## Beispiel (Fortsetzung)



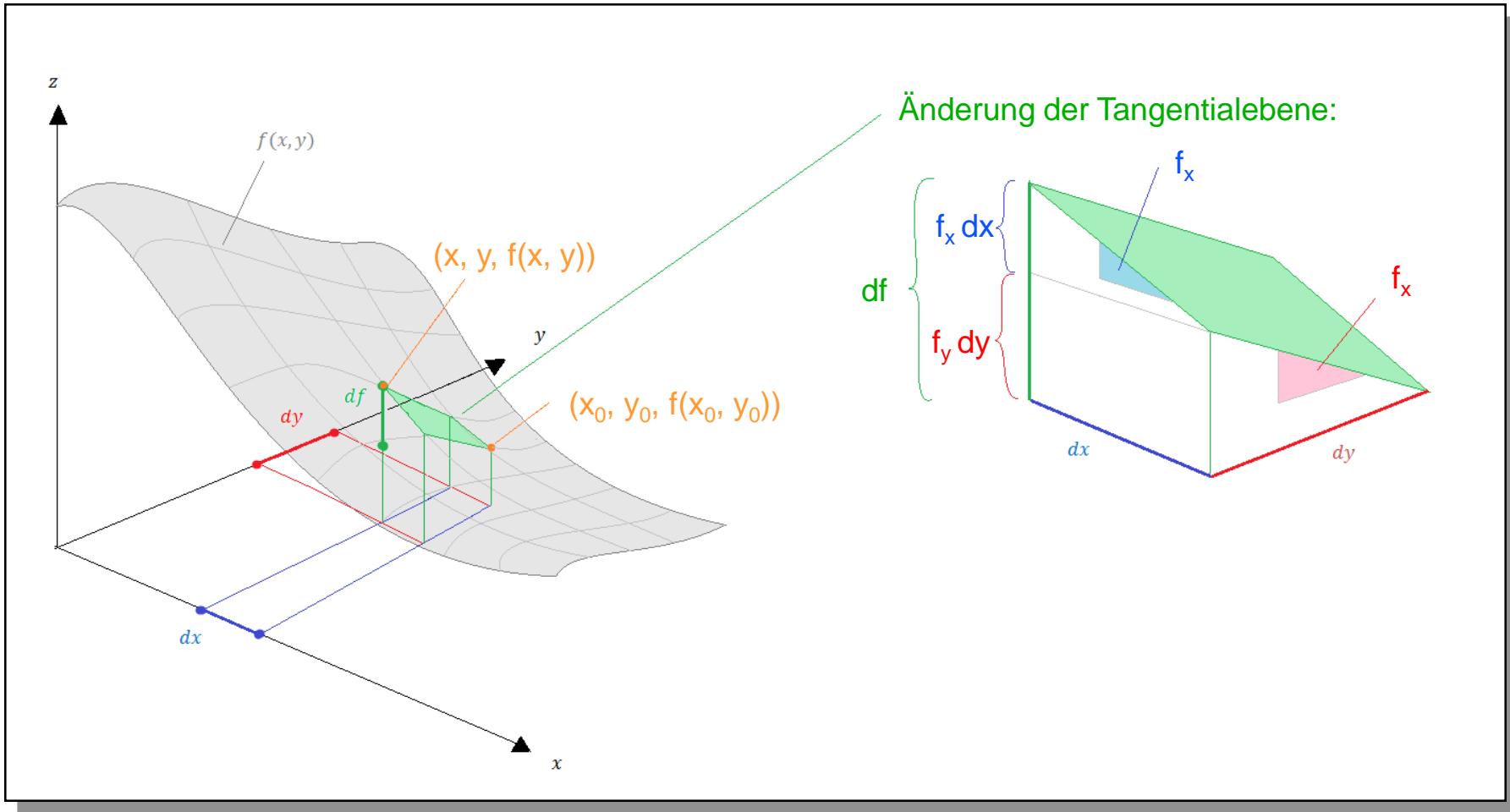
Fläche mit Tangentialebene, verschiedene Ansichten

# Übung

Die Funktion  $f(x, y) = x^3 - 3xy^2$  heißt **Affensattel**. Bestimmen Sie für  $f$  die Tangentialebene an der Stelle  $(1, 2)$ .



# Partielle Ableitungen und Tangentialebene



# Gradient

Der **Gradient** einer Funktion  $f(x, y)$  ist der Vektor ihrer partiellen Ableitungen:

$$\text{grad}(f) = \begin{pmatrix} f_x \\ f_y \end{pmatrix}$$

Statt  $\text{grad}(f)$  schreibt man auch:  **$\nabla f$  (Nabla-Operator)**.

**Beispiel:** Die Funktion  $f(x, y) = x^3 - 3xy^2$  (Affensattel) hat den Gradient

$$\text{grad}(f) = \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} 3x^2 - 3y^2 \\ -6xy \end{pmatrix}$$

An der Stelle  $(1, 1)$  ergibt sich der Gradient:  $\text{grad}(f)(1,1) = \begin{pmatrix} 0 \\ -6 \end{pmatrix}$

# Übung

Berechnen Sie folgende Gradienten an der Stelle (1, 1):

(a)  $\text{grad}(3x^2 - 7y)$

(b)  $\text{grad}(21x^2y)$

(c)  $\text{grad}(x^y)$

# Richtungsableitungen

Wir kennen bereits

- die Ableitung in x-Richtung  
(Steigung einer x-Linie):  $f_x = \frac{\partial f}{\partial x}$
- die Ableitung in y-Richtung  
(Steigung einer y-Linie):  $f_y = \frac{\partial f}{\partial y}$

**Fragen:**

- Wie steil ist es *schräg* zu diesen Richtungen?
- In welcher Richtung ist die Steigung *am größten*?



# Richtungsableitung

Wir beschreiben eine **beliebige Richtung**

durch einen Vektor  $\vec{r} = \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix}$  mit  $|\vec{r}| = 1$ .

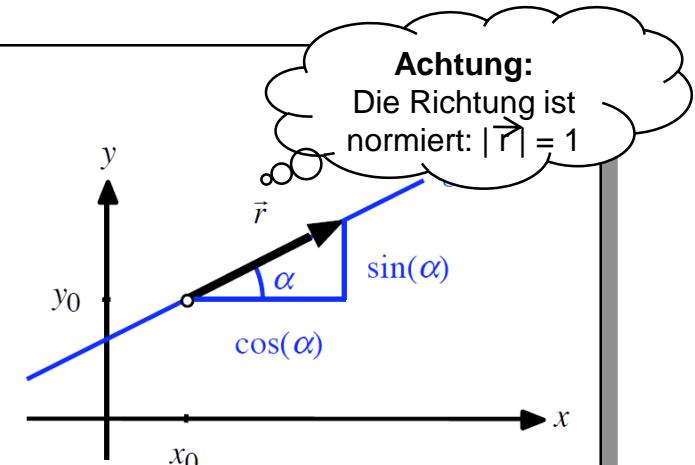
Die **Richtungsableitung** einer Funktion  $f(x, y)$

in Richtung  $\vec{r}$  wird mit  $\frac{\partial f}{\partial \vec{r}}$  bezeichnet.

Sie ergibt sich als **Skalarprodukt** der beiden Vektoren  $\text{grad}(f)$  und  $\vec{r}$ :

**Satz:** Für die Richtungsableitung gilt:

$$\frac{\partial f}{\partial \vec{r}} = \text{grad}(f) \cdot \vec{r} = \begin{pmatrix} f_x \\ f_y \end{pmatrix} \cdot \begin{pmatrix} \cos(\alpha) \\ \sin(\alpha) \end{pmatrix} = f_x \cdot \cos(\alpha) + f_y \cdot \sin(\alpha).$$



# Übung

Berechnen Sie für die Funktion

$$f(x, y) = x^3 \sin(y) + e^x y^2$$

im Punkt  $(1, 0, f(1, 0))$  die Richtungsableitung in Richtung von  $\vec{r} = \begin{pmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{pmatrix}$ .

## Folgerungen

Sei  $\gamma$  der Winkel zwischen den Vektoren  $\text{grad}(f)$  und  $\vec{r}$ . Dann gilt:

$$\frac{\partial f}{\partial \vec{r}} = \text{grad}(f) \cdot \vec{r} = |\text{grad}(f)| \cdot |\vec{r}| \cdot \cos(\gamma)$$

(1.) Für den Fall  $\gamma = 0$  folgt:  $\gamma = 0 \Rightarrow \cos(\gamma) = 1 \Rightarrow \frac{\partial f}{\partial \vec{r}}$  maximal

D. h.: Die Richtungsableitung ist maximal, wenn  $\gamma = 0$  ist, wenn also  $\vec{r}$  in die Richtung von  $\text{grad}(f)$  zeigt.

**Folgerung 1:** Der Gradient zeigt in Richtung des steilsten Anstiegs.

(2.) Für den Fall  $\gamma = \pi/2$  ( $= 90^\circ$ ) folgt:  $\gamma = \pi/2 \Rightarrow \cos(\gamma) = 0 \Rightarrow \frac{\partial f}{\partial \vec{r}} = 0$

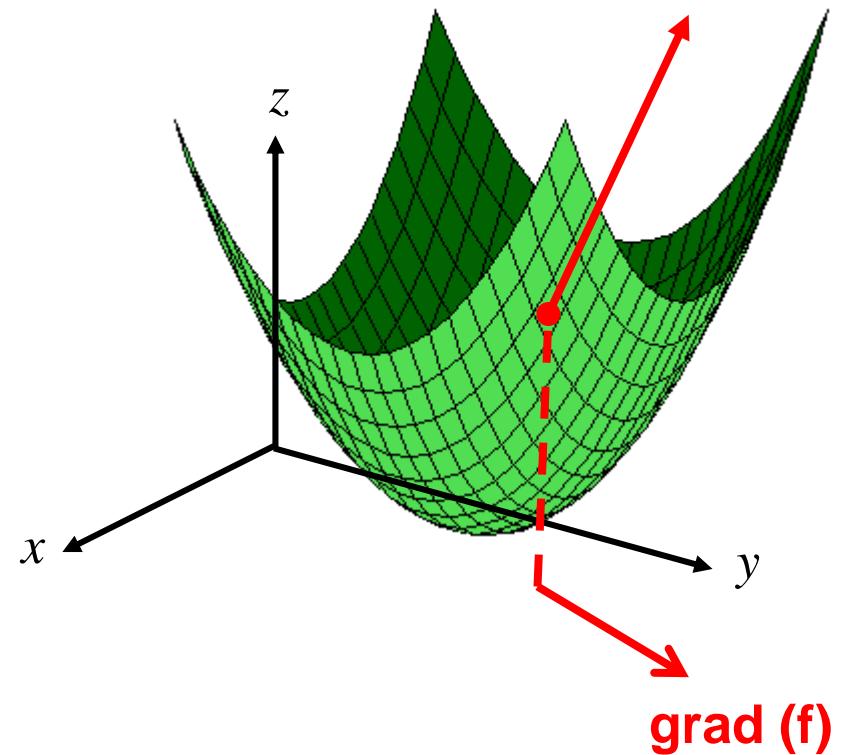
**Folgerung 2:** Der Gradient steht senkrecht auf den Niveaulinien.

# Eigenschaften des Gradienten

(1.) Der Gradient zeigt in Richtung des steilsten Anstiegs.

Sein Betrag ist die Steigung in dieser (steilsten) Richtung.

(2.) Der Gradient steht senkrecht auf den Niveaulinien.



## Beispiel: $f(x, y) = xy$

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

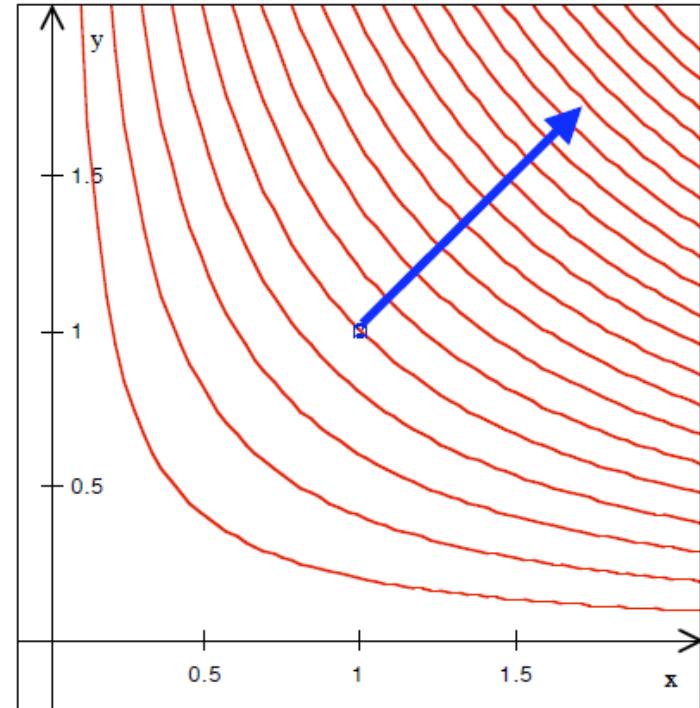
Gradient:

$$\text{grad}(f) = \begin{pmatrix} f_x \\ f_y \end{pmatrix} = \begin{pmatrix} y \\ x \end{pmatrix}$$

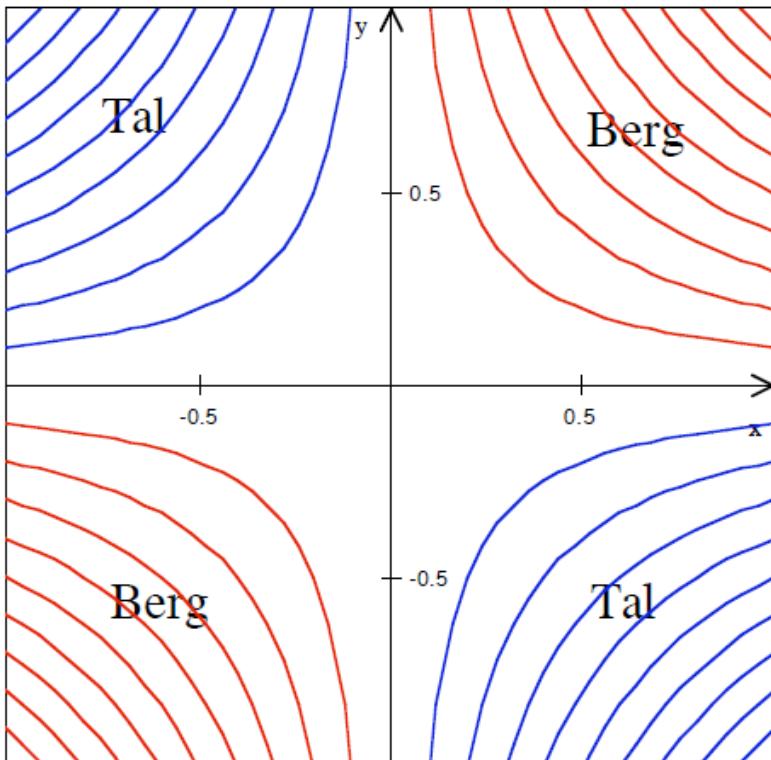
Im Punkt  $(1, 1)$ :

$$\text{grad}(f)(1, 1) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

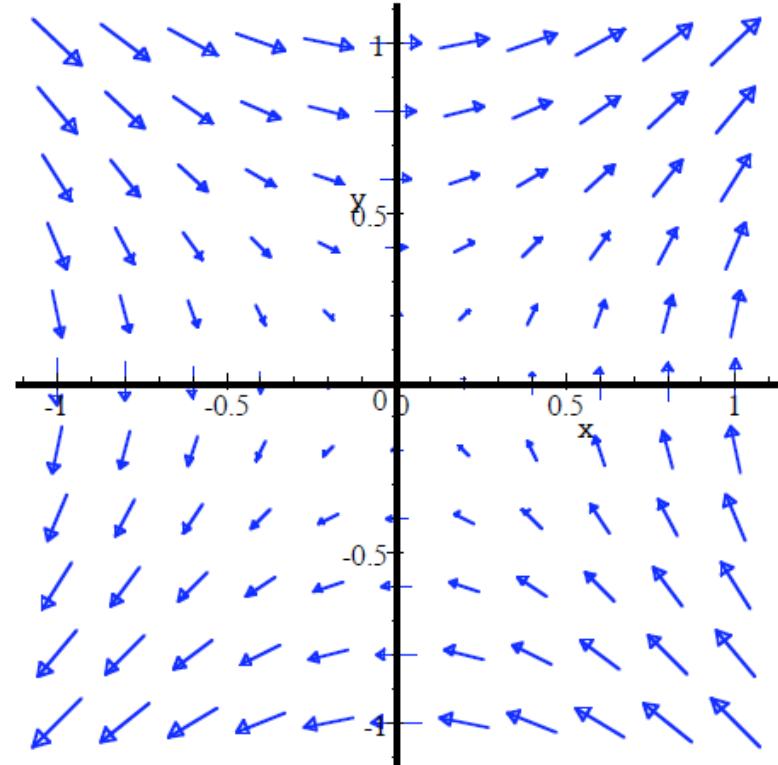
- zeigt in Richtung des steilsten Anstiegs,  
Steigung in dieser Richtung =  $\sqrt{2}$
- steht senkrecht auf der Niveaulinie



# Gradientenfeld von $f(x, y) = xy$



Niveaulinien



Gradientenrichtung

## Übung

Berechnen Sie die Richtung des größten Anstiegs der Funktion

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

im Punkt  $(1, 1, f(1, 1))$  und die Steigung in dieser Richtung.

## Höhere partielle Ableitungen

Die partiellen Ableitungen  $f_x$  und  $f_y$  können erneut nach x oder y abgeleitet werden.

Zweimalige partielle Ableitung nach x:

$$f_{xx} = \frac{\partial^2 f}{\partial x^2}$$

Einmal nach x und einmal nach y abgeleitet:  $f_{xy} = \frac{\partial^2 f}{\partial x \partial y}$

Zweimalige partielle Ableitung nach y:

$$f_{yy} = \frac{\partial^2 f}{\partial y^2}$$

## Beispiel

$$f(x,y) = x^4 y^7$$
$$f_x(x,y) = 4x^3 y^7 \quad f_y(x,y) = 7x^4 y^6$$
$$f_{xx}(x,y) = 12x^2 y^7 \quad f_{xy}(x,y) = 28x^3 y^6 \quad f_{yy}(x,y) = 42x^4 y^5$$

```
graph TD; f[x^4 y^7] --> fx[4x^3 y^7]; f --> fy[7x^4 y^6]; fx --> fxx[12x^2 y^7]; fx --> fxy[28x^3 y^6]; fy --> fyy[42x^4 y^5];
```

**Satz:** Die Reihenfolge der partiellen Ableitungen spielt keine Rolle.

$$f_{xxx}(x,y) = 24x y^7 \quad f_{xxy}(x,y) = 84x^2 y^6 \quad f_{xyy}(x,y) = 168x^3 y^5 \quad f_{yyy}(x,y) = 210x^4 y^4$$

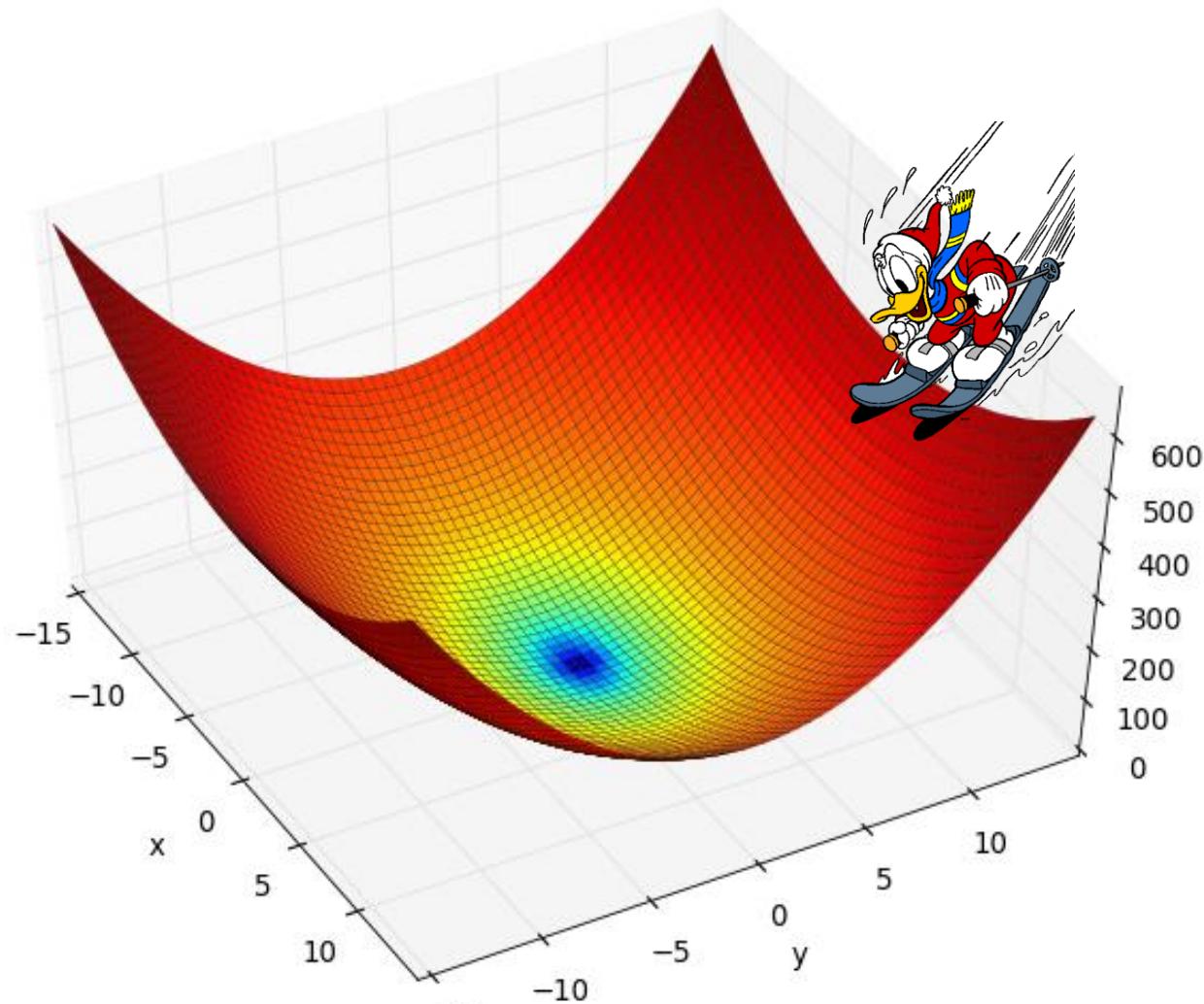
# Übung

Bestimmen Sie  $f_x$ ,  $f_y$ ,  $f_{xy}$ ,  $f_{yx}$ :

(a)  $f(x, y) = x^2y + xy^3 + 5xy$

(b)  $f(x, y) = \cos(2x) \cdot e^{5y}$

## 5.3 Extrema



# Wdh.: Extrema von $f(x)$

Erinnerung an Funktionen  $f(x)$   
einer Variablen:

Kriterium für lokale Extrema:

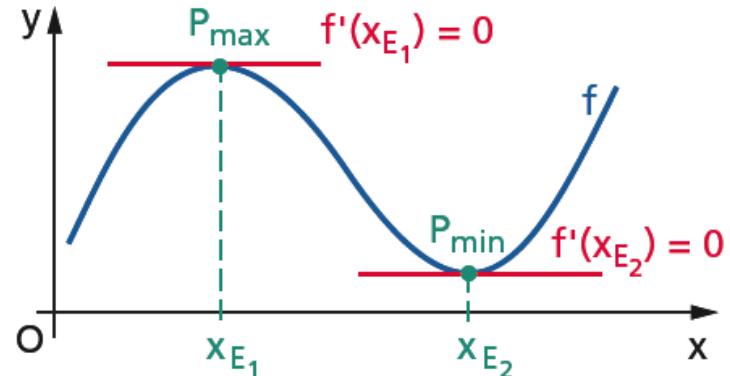
- Notwendig:

$f'(x_E) = 0$  („waagerechte Tangente“)

- Hinreichend:

$f''(x_E) > 0$  („linksgekrümmt“)  $\Rightarrow$  Minimum in  $x_E$

$f''(x_E) < 0$  („rechtsgekrümmt“)  $\Rightarrow$  Maximum in  $x_E$



## Extrema von $f(x, y)$ : notwendig

Für Funktionen  $f(x, y)$  von *zwei* Variablen gelten analoge Kriterien.

### Notwendiges Kriterium für lokale Extrema:

Wenn eine Funktion  $f(x, y)$  an der Stelle  $(x_0, y_0)$  ein lokales Extremum hat, dann gilt an dieser Stelle  $f_x = 0$  und  $f_y = 0$ , also

$$\text{grad}(f) = \vec{0} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Anstelle von  $f'$  müssen jetzt also *beide* ersten Ableitungen  $f_x$  und  $f_y$  gleich Null sein. Um mögliche Extrempunkte zu finden, muss daher ein Gleichungssystem mit zwei Gleichungen gelöst werden.

## Hesse-Matrix

Anstelle von  $f''$  benötigen wir für das hinreichende Kriterium die *vier* zweiten Ableitungen:  $f_{xx}$ ,  $f_{xy}$ ,  $f_{yx}$  und  $f_{yy}$ .

Diese werden in der **Hesse-Matrix** zusammengefasst:

$$H = \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix}$$

Entscheidend für die Art des Extremums ist die **Determinante** der Hesse-Matrix:

$$\Delta = \det \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix} = f_{xx} \cdot f_{yy} - f_{xy}^2$$

# Übung

Bestimmen Sie die Determinante der Hesse-Matrix von  $f(x, y) = x^2y^3$ .

$$f_x = \quad \Rightarrow \quad f_{xx} = \quad , \quad f_{xy} =$$

$$f_y = \quad \Rightarrow \quad f_{yy} = \quad , \quad f_{yx} =$$

$$\Delta = \det \begin{pmatrix} f_{xx} & f_{xy} \\ f_{yx} & f_{yy} \end{pmatrix} = f_{xx} \cdot f_{yy} - f_{xy}^2$$

## Extrema von $f(x, y)$ : hinreichend

### Hinreichendes Kriterium für lokale Extrema:

Wenn für eine Funktion  $f(x, y)$  an der Stelle  $(x_0, y_0)$  gilt  $\text{grad}(f) = 0$ , dann liegt in  $(x_0, y_0)$  unter folgenden Bedingungen folgendes vor:

$\Delta > 0$  und  $f_{xx} > 0 \Rightarrow$  isoliertes Minimum

$\Delta > 0$  und  $f_{xx} < 0 \Rightarrow$  isoliertes Maximum

$\Delta < 0 \Rightarrow$  Sattelpunkt

$\Delta = 0 \Rightarrow$  keine Aussage möglich

**Vorgehen bei der Extremwertsuche:** Zunächst werden die Punkte bestimmt, an denen  $\text{grad}(f) = 0$  ist (Gleichungssystem lösen). Anschließend werden diese Punkte in  $\Delta$  und ggf. in  $f_{xx}$  eingesetzt.

## 1. Beispiel

**1. Beispiel:**  $f(x, y) = x \cdot y$  (**Sattelfläche**)

Bestimmung der Punkte mit  $\text{grad}(f) = 0$ , also mit  $f_x = 0$  und  $f_y = 0$ :

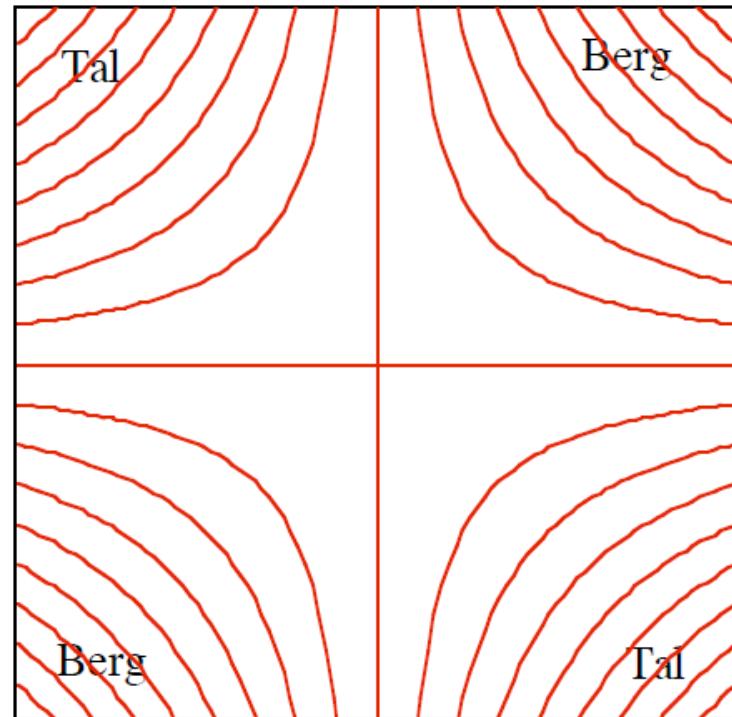
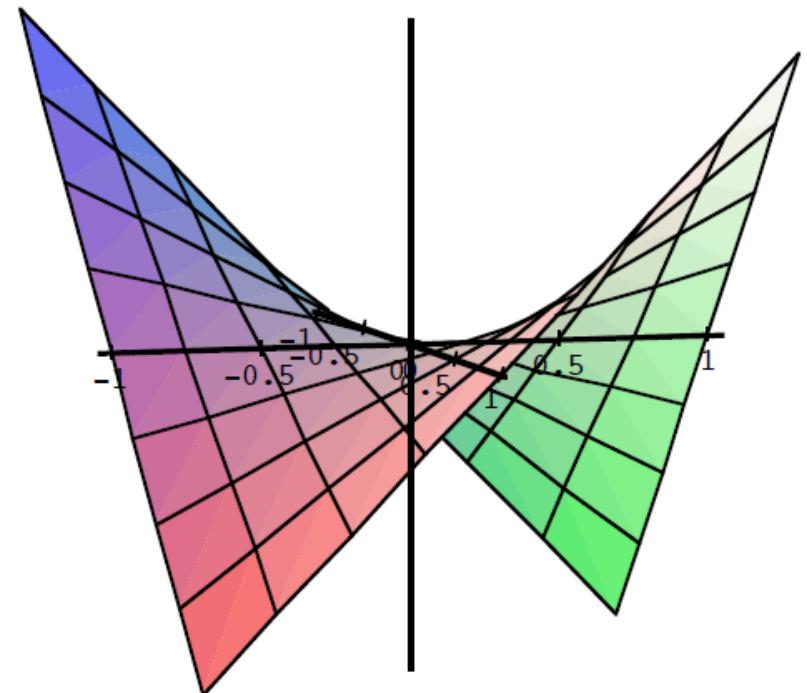
$$\begin{array}{l} f_x = y = 0 \\ f_y = x = 0 \end{array} \quad \Rightarrow \quad \text{Punkt: } (0, 0)$$

Untersuchen des Punkts  $(0, 0)$  mit  $\Delta$  (und ggf.  $f_{xx}$ ):

$$\begin{array}{ll} f_{xx} = 0 & f_{xy} = 1 \\ f_{yx} = 1 & f_{yy} = 0 \end{array} \quad \Rightarrow \quad \Delta = \det \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} = 0 \cdot 0 - 1^2 = -1$$

Wegen  $\Delta < 0$  liegt im Ursprung  $(0, 0)$  ein **Sattelpunkt** vor.

# 1. Beispiel



$$f(x,y) = xy, \text{ Sattelpunkt im Ursprung}$$

## 2. Beispiel

2. Beispiel:  $f(x, y) = x^2 + y^2$  (Paraboloid)

Bestimmung der Punkte mit  $\text{grad}(f) = 0$ , also mit  $f_x = 0$  und  $f_y = 0$ :

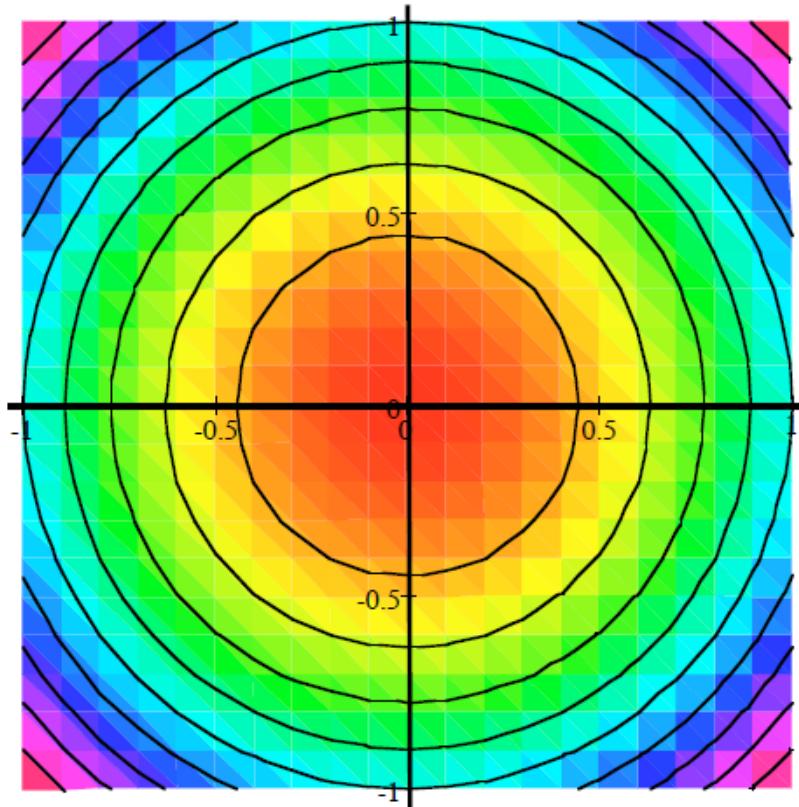
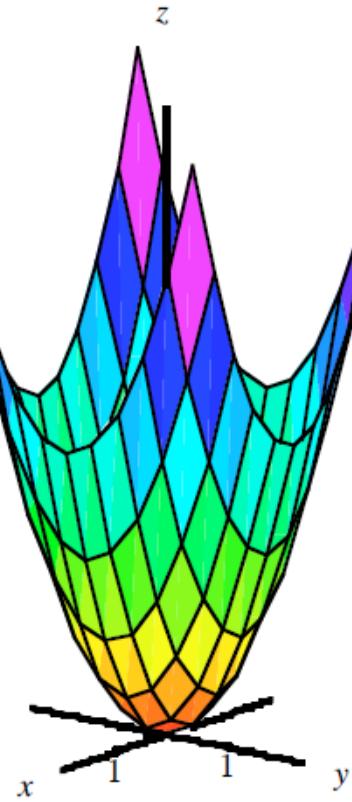
$$\begin{aligned} f_x &= 2x = 0 & \Rightarrow & \quad x = 0 \\ f_y &= 2y = 0 & \Rightarrow & \quad y = 0 \end{aligned} \quad \left. \right\} \quad \Rightarrow \quad \text{Punkt: } (0, 0)$$

Untersuchen des Punkts  $(0, 0)$  mit  $\Delta$  (und ggf.  $f_{xx}$ ):

$$\begin{aligned} f_{xx} &= 2 & f_{xy} &= 0 \\ f_{yx} &= 0 & f_{yy} &= 2 \end{aligned} \quad \left. \right\} \quad \Delta = \det \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} = 2 \cdot 2 - 0^2 = 4 > 0$$

Wegen  $\Delta > 0$  und außerdem  $f_{xx}(0, 0) = 2 > 0$  befindet sich in  $(0, 0)$  ein isoliertes **Minimum**.

## 2. Beispiel



$$f(x,y) = x^2 + y^2, \text{ isoliertes Minimum im Ursprung}$$

# Übung

Untersuchen Sie folgende Funktion auf Extrema und Sattelpunkte:

$$f(x, y) = \frac{1}{3}x^3 - x^2 + y^3 - 12y$$

# Gradientenabstiegsverfahren

## Probleme in der Praxis:

- Gleichungen lösen, höhere Ableitungen berechnen:  
oft schwierig, aufwändig, unmöglich
- Kriterium liefert manchmal keine Aussage  
→ Ausweg: Numerische Näherung!

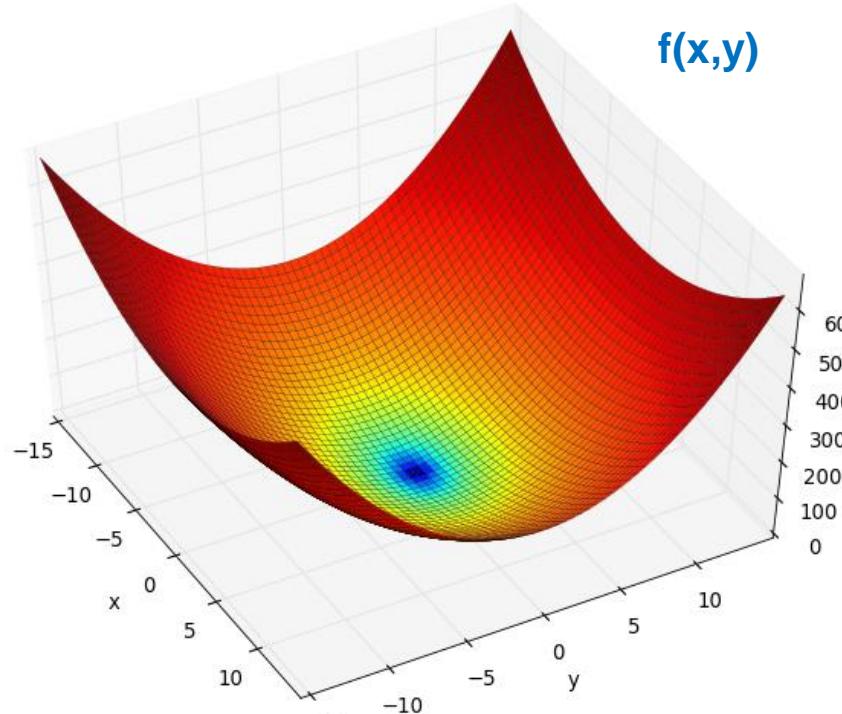
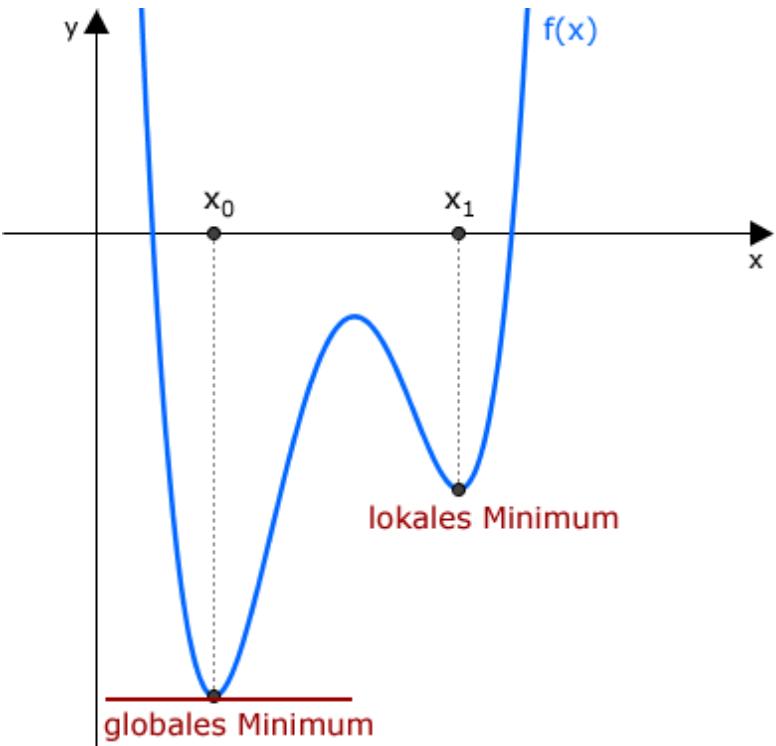
## Gradientenabstiegsverfahren:

- auch: „Verfahren des steilsten Abstiegs“, „steepest descent“
- fundamentales Optimierungsverfahren, 1847 von Cauchy vorgestellt
- moderne Anwendungen, u. a. in der künstlichen Intelligenz

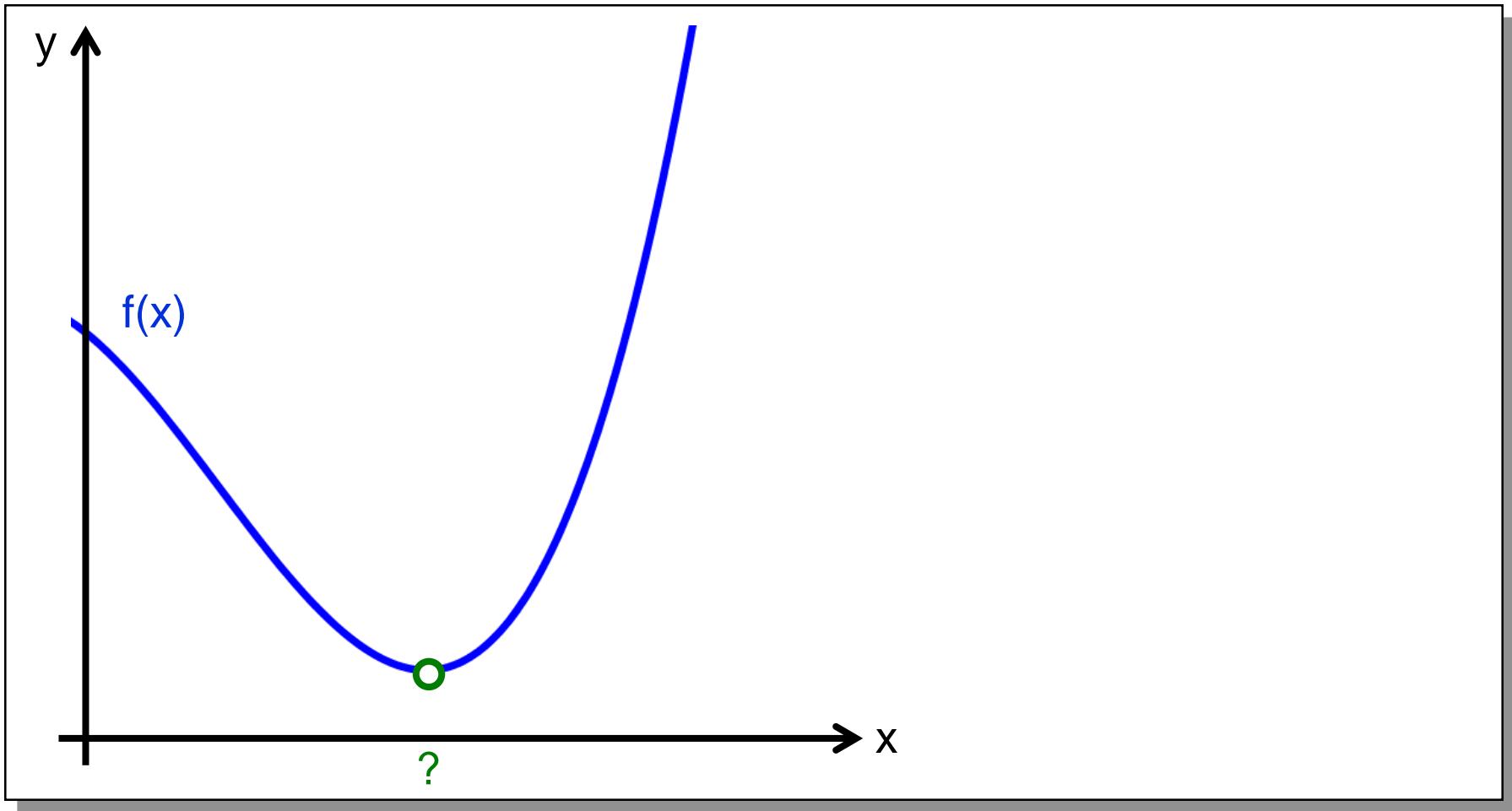


# Gradientenabstiegsverfahren

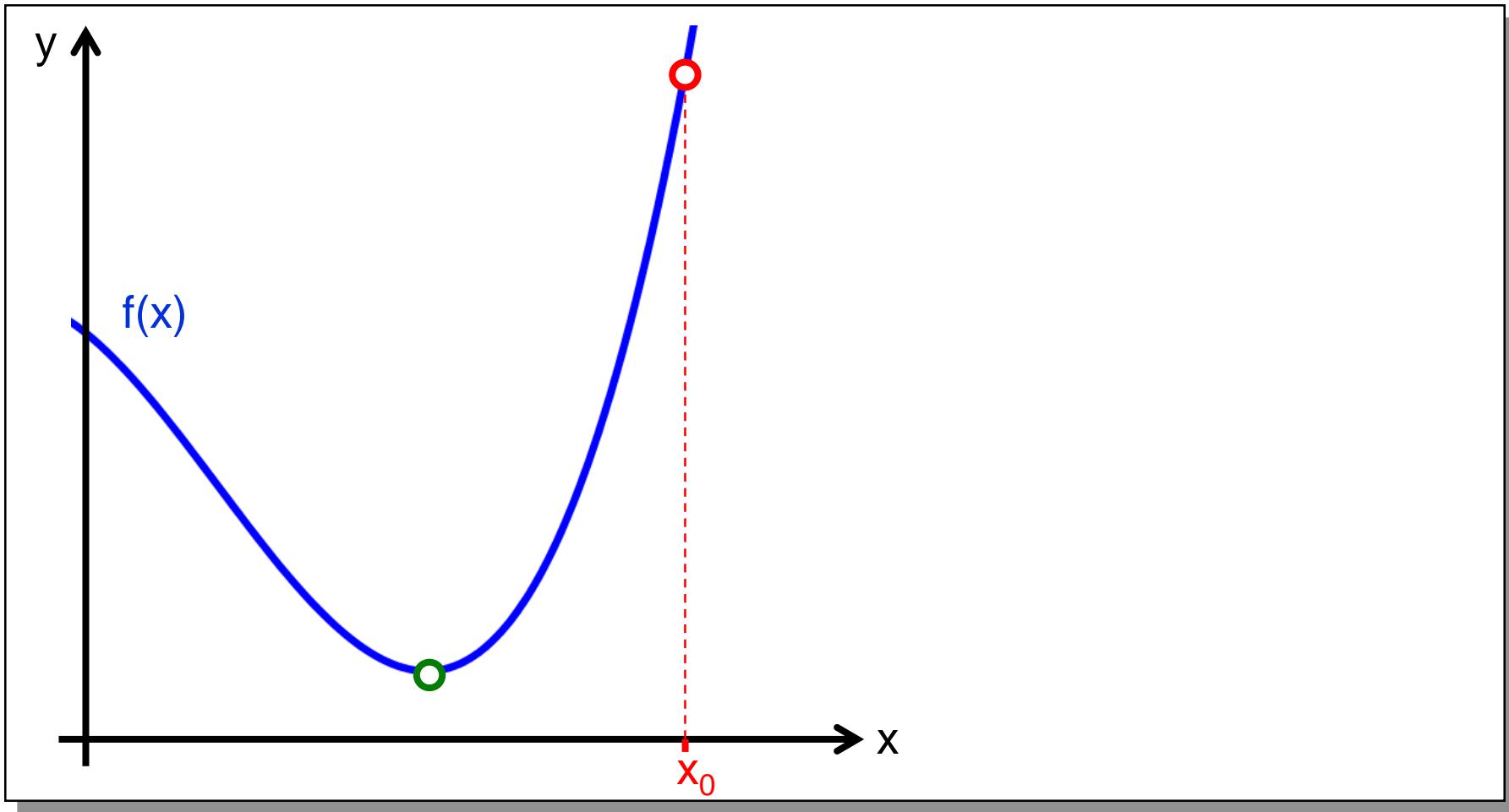
Ziel: numerische Berechnung eines lokalen Minimums



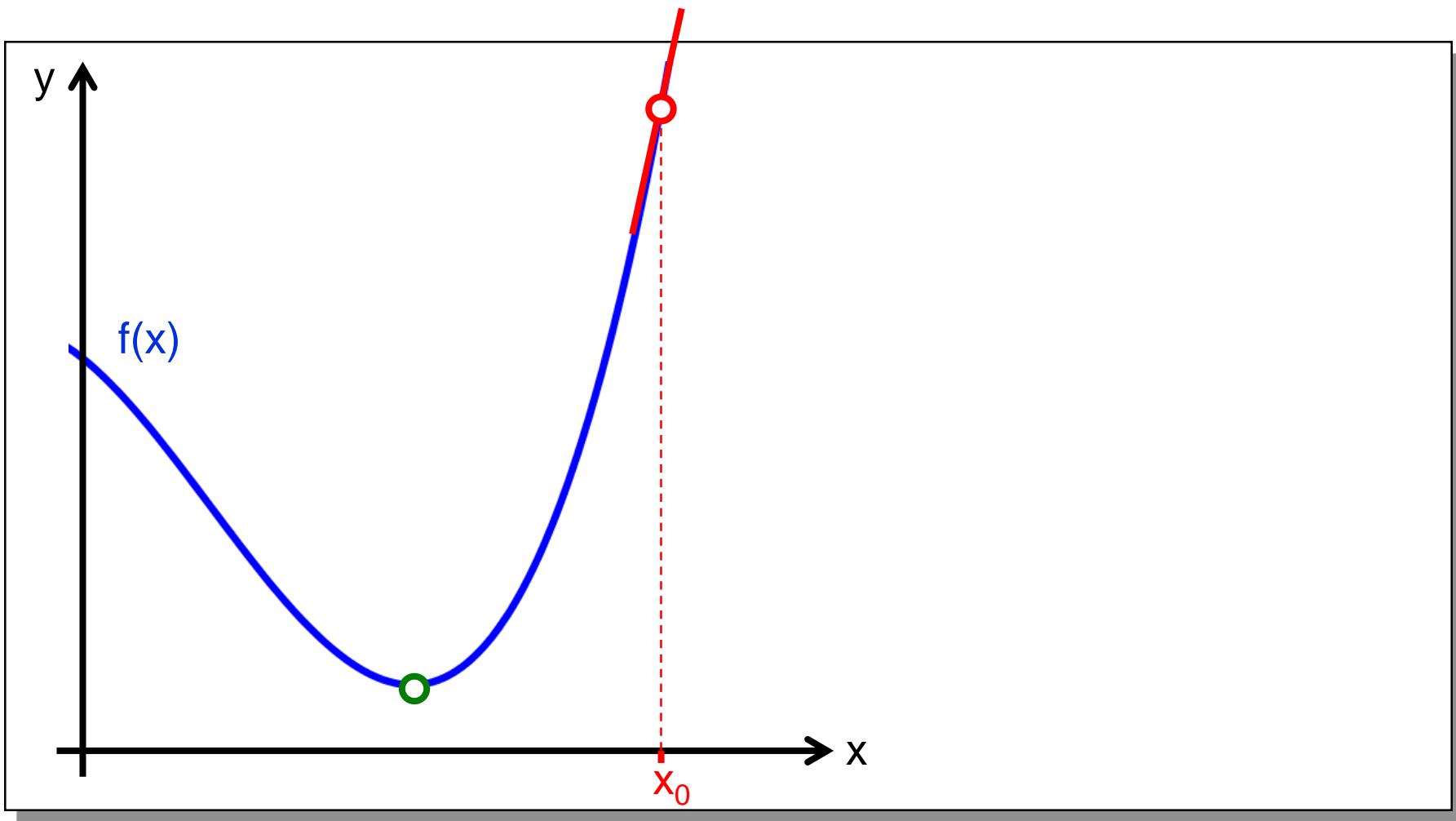
## Gesucht: Minimum!



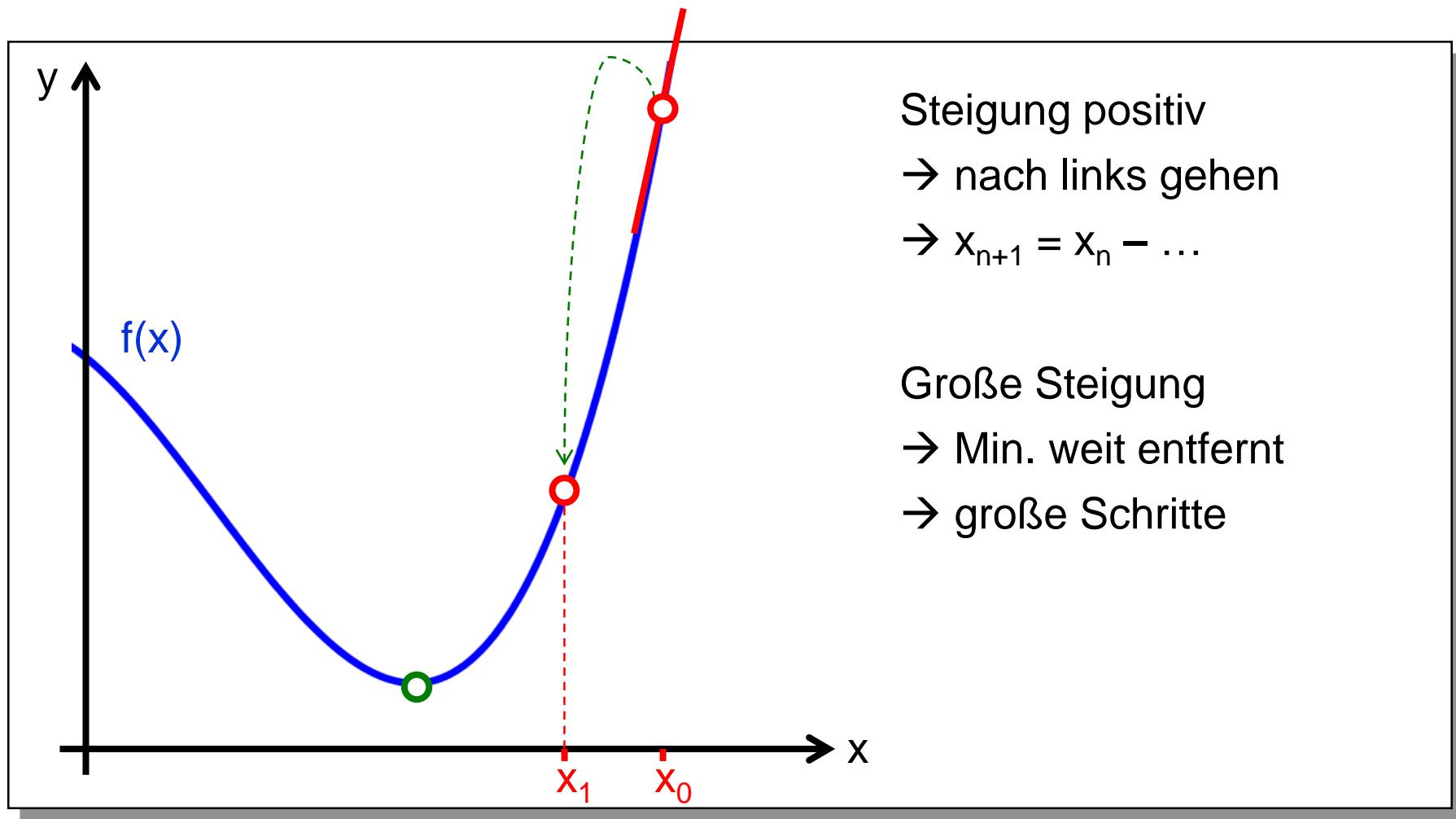
## Startstelle wählen



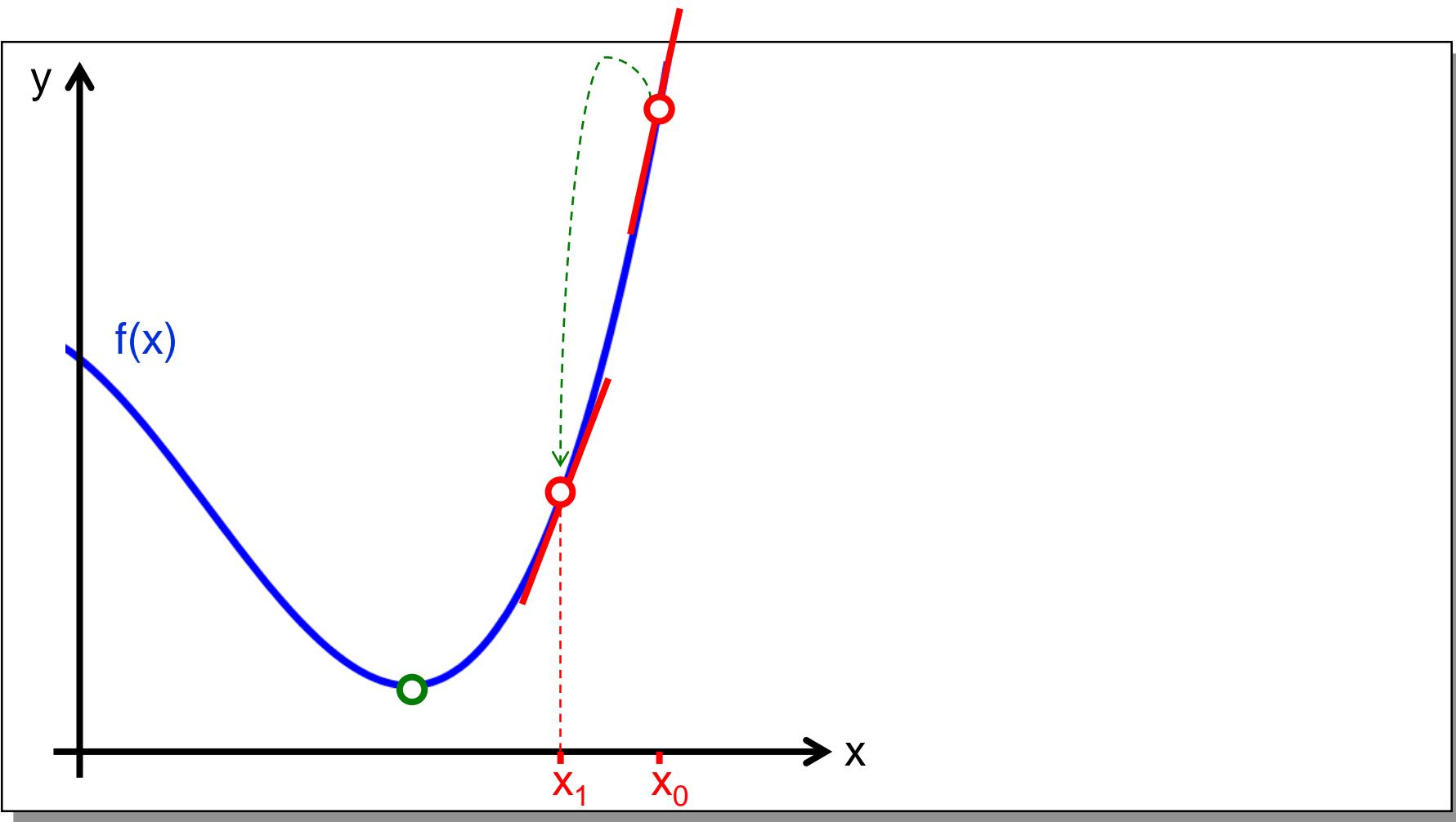
# Steigung berechnen



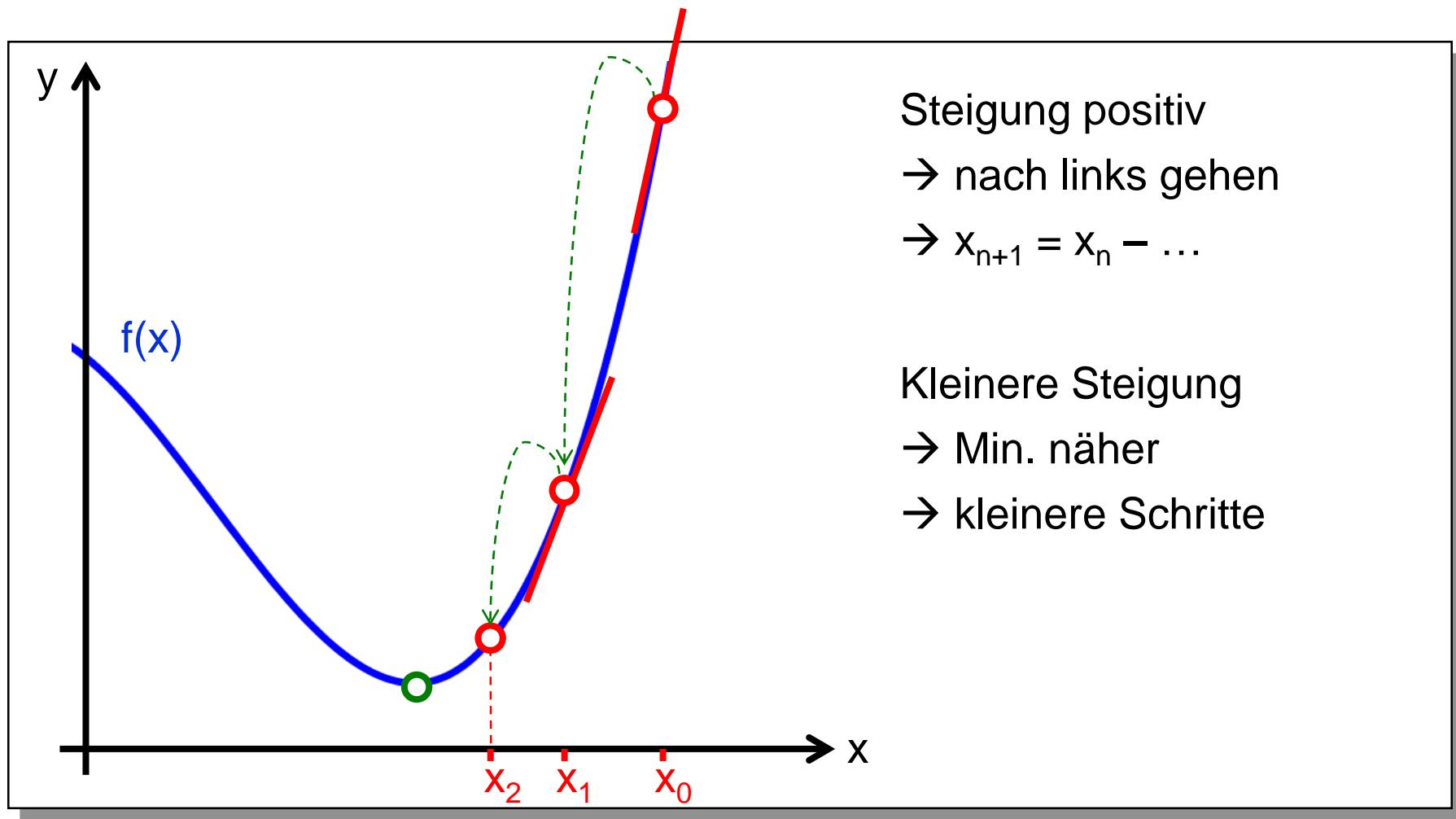
# Richtung und Schrittweite?



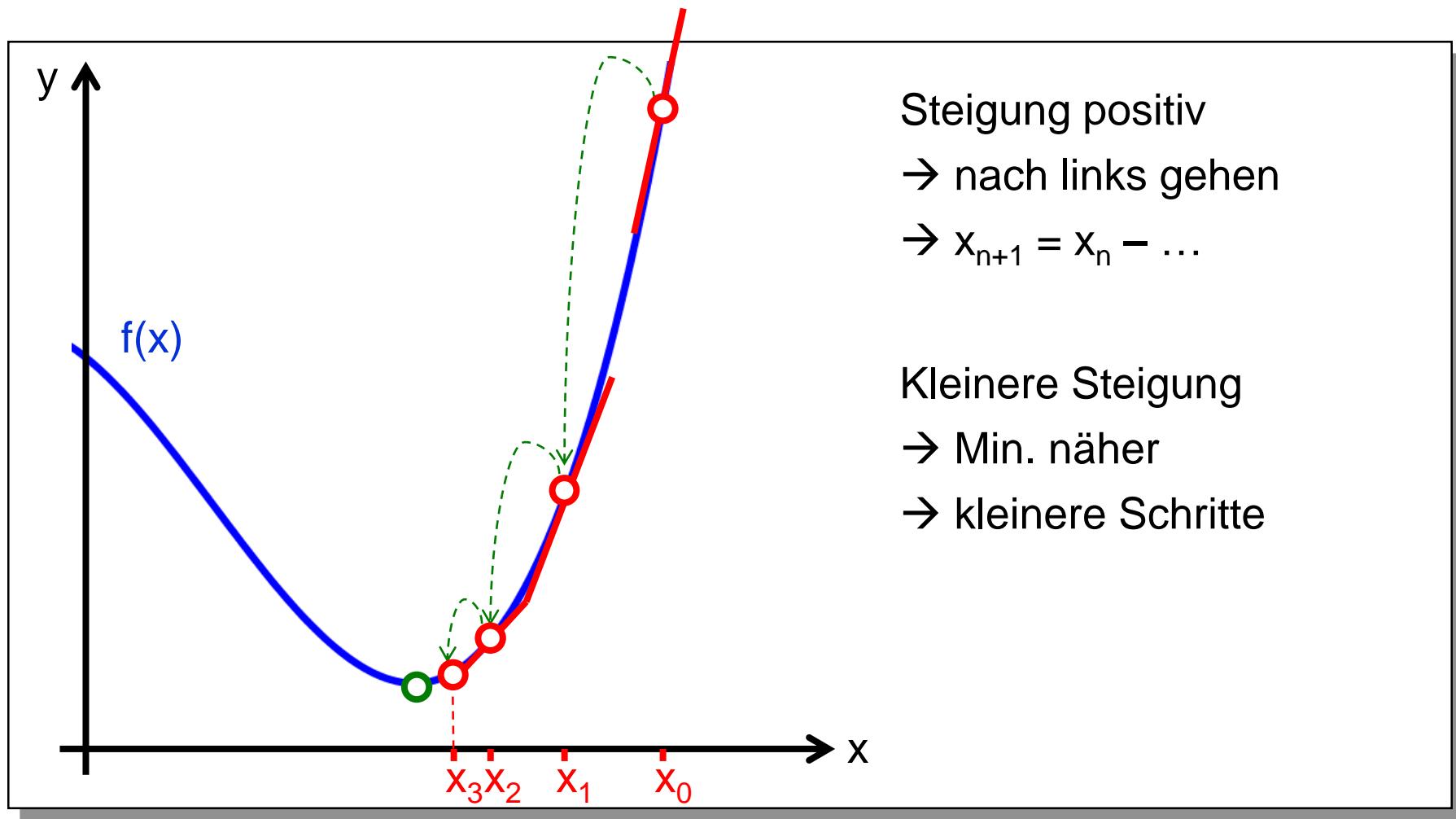
## Und wieder Steigung berechnen ...



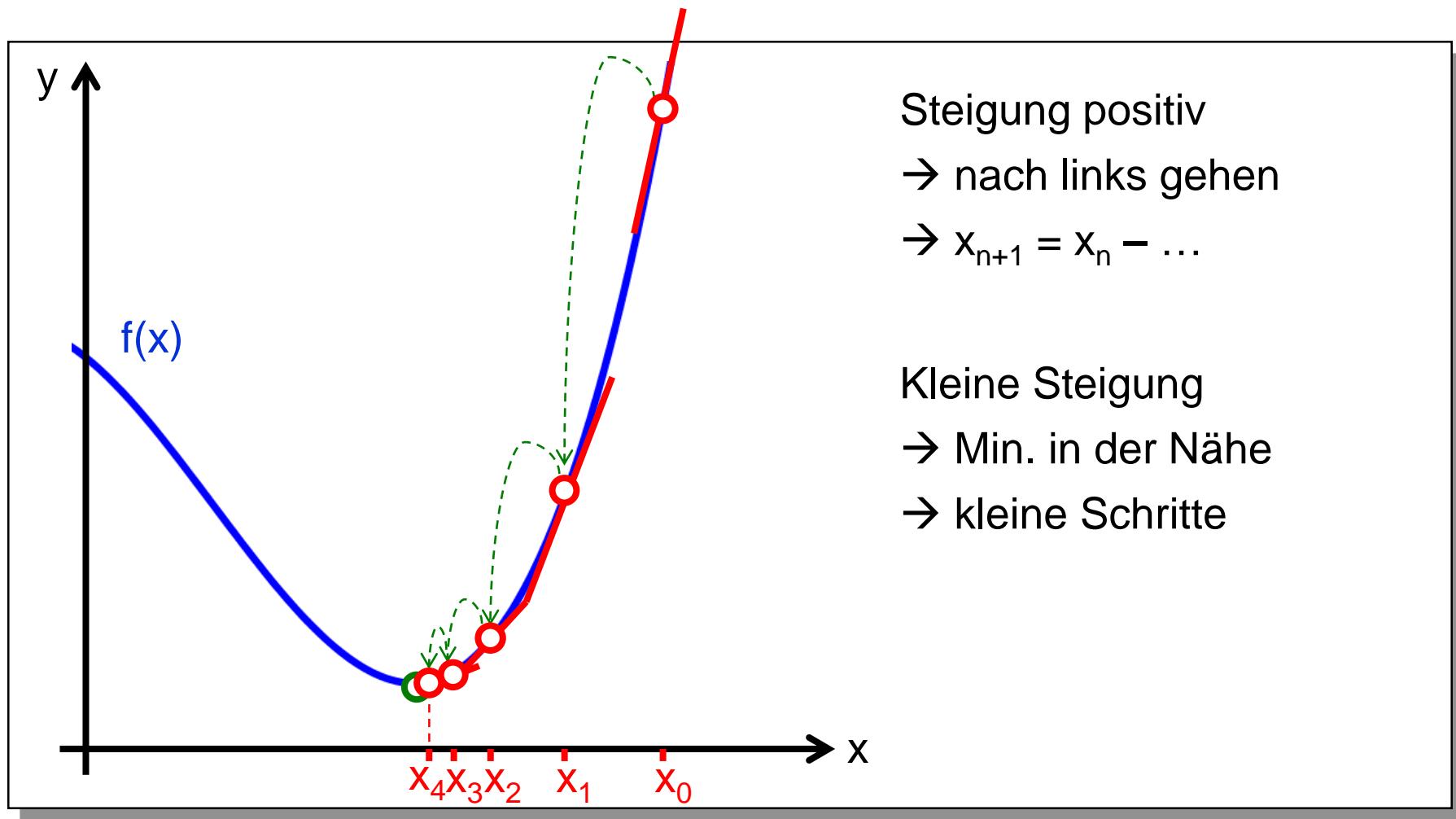
# Richtung und Schrittweite?



Usw.



## Usw. bis Steigung klein genug



# Gradientenabstiegsverfahren (eindimensional)

- **Startwert**  $x_0$  beliebig wählen

- **Richtung?**

Steigung positiv  $\rightarrow$  nach links:  $x_{n+1} = x_n - \dots$

Steigung negativ  $\rightarrow$  nach rechts:  $x_{n+1} = x_n + \dots$

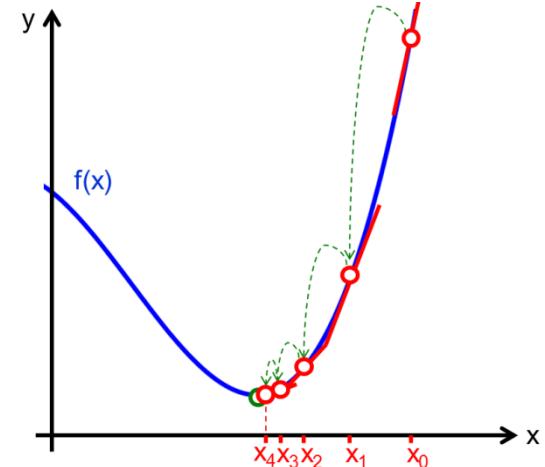
- **Schrittweite?**

Große Steigung  $\rightarrow$  Min. weit entfernt  $\rightarrow$  große Schritte

Kleine Steigung  $\rightarrow$  Min. in der Nähe  $\rightarrow$  kleine Schritte

$\rightarrow$  Schrittweite proportional zur Steigung!

- **Iterationsvorschrift:**  $x_{n+1} = x_n - s \cdot f'(x_n)$



## Beispiel

**Funktion:**  $f(x) = x^3 - 2x^2 + 2$

Ableitung:  $f'(x) = 3x^2 - 4x$

Wähle: Startwert:  $x_0 = 2$   
Schrittweite:  $s = 0.1$

**Gradientenabstieg:**

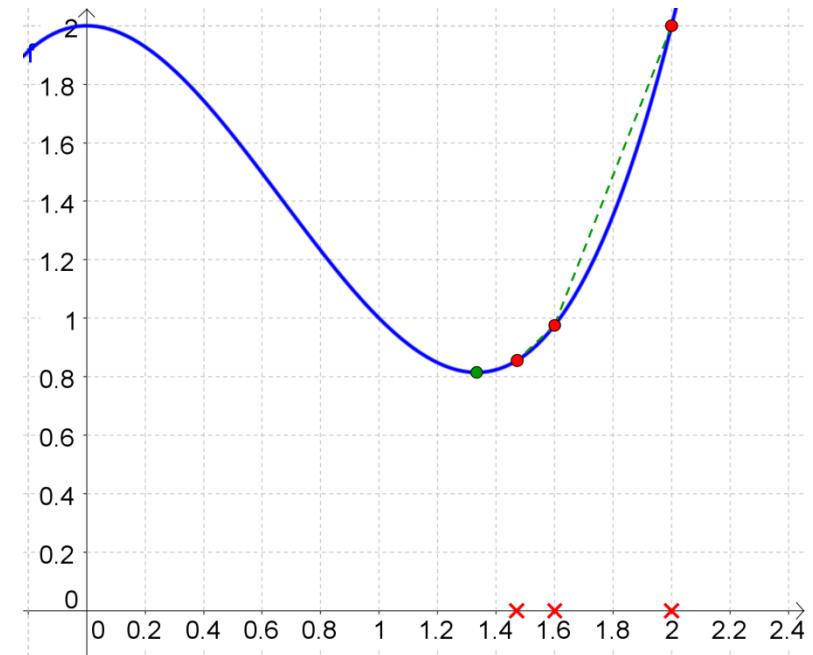
$$\begin{aligned}x_{n+1} &= x_n - s \cdot f'(x_n) \\&= x_n - 0.1 \cdot (3x_n^2 - 4x_n)\end{aligned}$$

Ergebnis der ersten Iterationen:

$$x_0 = 2$$

$$x_1 = 2 - 0.1 \cdot (3 \cdot 2^2 - 4 \cdot 2) = 1.6$$

$$x_2 = 1.6 - 0.1 \cdot (3 \cdot 1.6^2 - 4 \cdot 1.6) = 1.472$$



# Taschenrechner-Trick

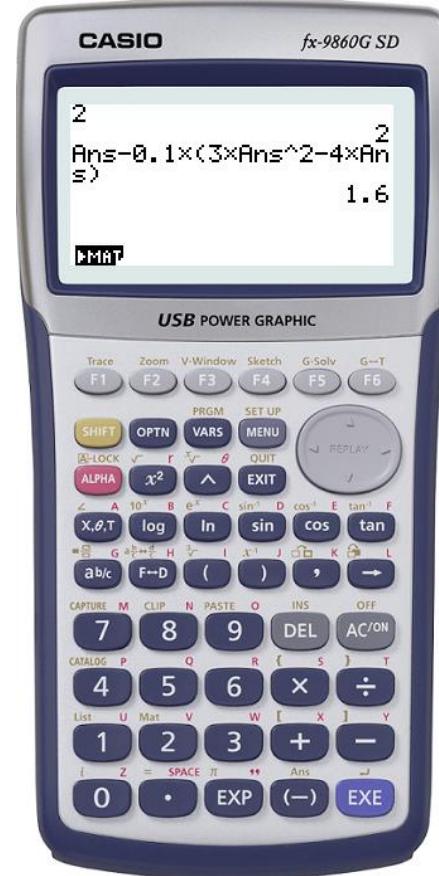
Der TR speichert  $x_n$  im ANS-Speicher!

$$\begin{aligned}x_{n+1} &= x_n - 0.1 \cdot (3x_n^2 - 4x_n) \\&= [\text{ANS}] - 0.1 \cdot (3 \cdot [\text{ANS}]^2 - 4 \cdot [\text{ANS}])\end{aligned}$$

**TR-Eingabe:**

$$2 =$$

$$\text{Ans} - 0.1 \times (3 \times \text{Ans}^2 - 4 \times \text{Ans}) =$$



# Übung

Berechnen Sie ein Minimum der Funktion

$$f(x) = x^4 - 4x^3 + 2x^2 + 4x + 1$$

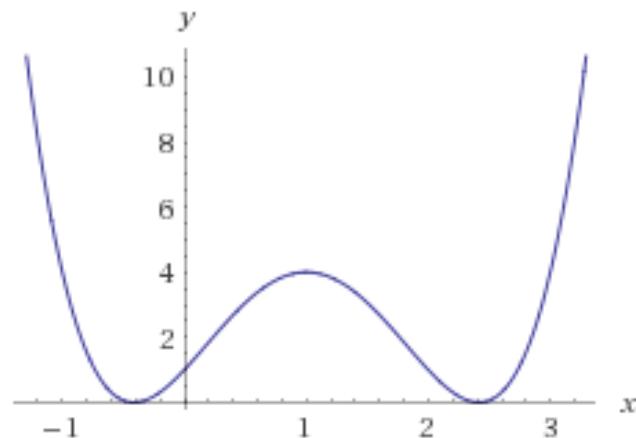
mit dem Gradientenabstiegsverfahren (mit  $x_0 = 3$  und  $s = 0.1$ ).

$$x_{n+1} = x_n - s \cdot f'(x_n)$$

$$= x_n - 0.1 \cdot (4x_n^3 - 12x_n^2 + 4x_n + 4)$$

Ergebnis:  $x = 2.4142$

Bei Startwert  $x_0 = 0$ :  $x = -0.41421$



# Programmierung des Gradientenabstiegsverfahrens



The screenshot shows a Java development environment with two main windows.

**Java Editor:** The code editor displays the file `G:\JavaEditor\Gradientenabstieg eindimensional\Gradientenabstiegeindim.java`. The code implements the gradient descent method for a one-dimensional function. It includes methods for the function and its derivative, and a main loop for the optimization process.

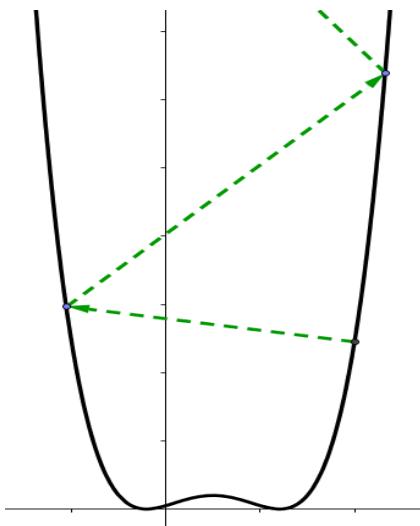
```
64 private double s = 0.1; // Schrittweitenfaktor
65
66 public double f(double x){ // Funktion
67     return x*x*x*x-4*x*x*x+2*x*x+4*x+1;
68 }
69
70 public double f_strich(double x){ // Ableitung
71     return 4*x*x*x-12*x*x+4*x+4;
72 }
73
74
75 public void jButton1ActionPerformed(ActionEvent evt) {
76     double x = jTextField1.getDouble(); // Eingabe x
77     while (Math.abs(f_strich(x))>0.01) { // Solange |f'| zu groß...
78         while (f(x-s*f_strich(x))>=f(x)) {s=s/2;} // ggf. s halbieren
79         x = x - s*f_strich(x); // neuer x-Wert
80         jTextArea1.append(String.valueOf(x)+"\n"); // Ausgabe x
81     }
82 }
```

**Graphical User Interface:** A window titled "Gradientenabstiegeindim" contains input fields and a button. The input field "Startwert x0:" contains the value 3. Below it is a "Berechne" button. To the right, a list labeled "xi:" shows the iterative steps of the algorithm, starting at 1.4 and continuing down to 2.413803087698099.

xi:
1.4
1.6944
2.1159865278464
2.4528243262498988
2.3884941234450148
2.428529446013336
2.405275056213035
2.4194413622101334
2.411030444957549
2.4161062507477644
2.4130718673236786
2.4148963679401647
2.413803087698099

# Steuerung der Schrittweite

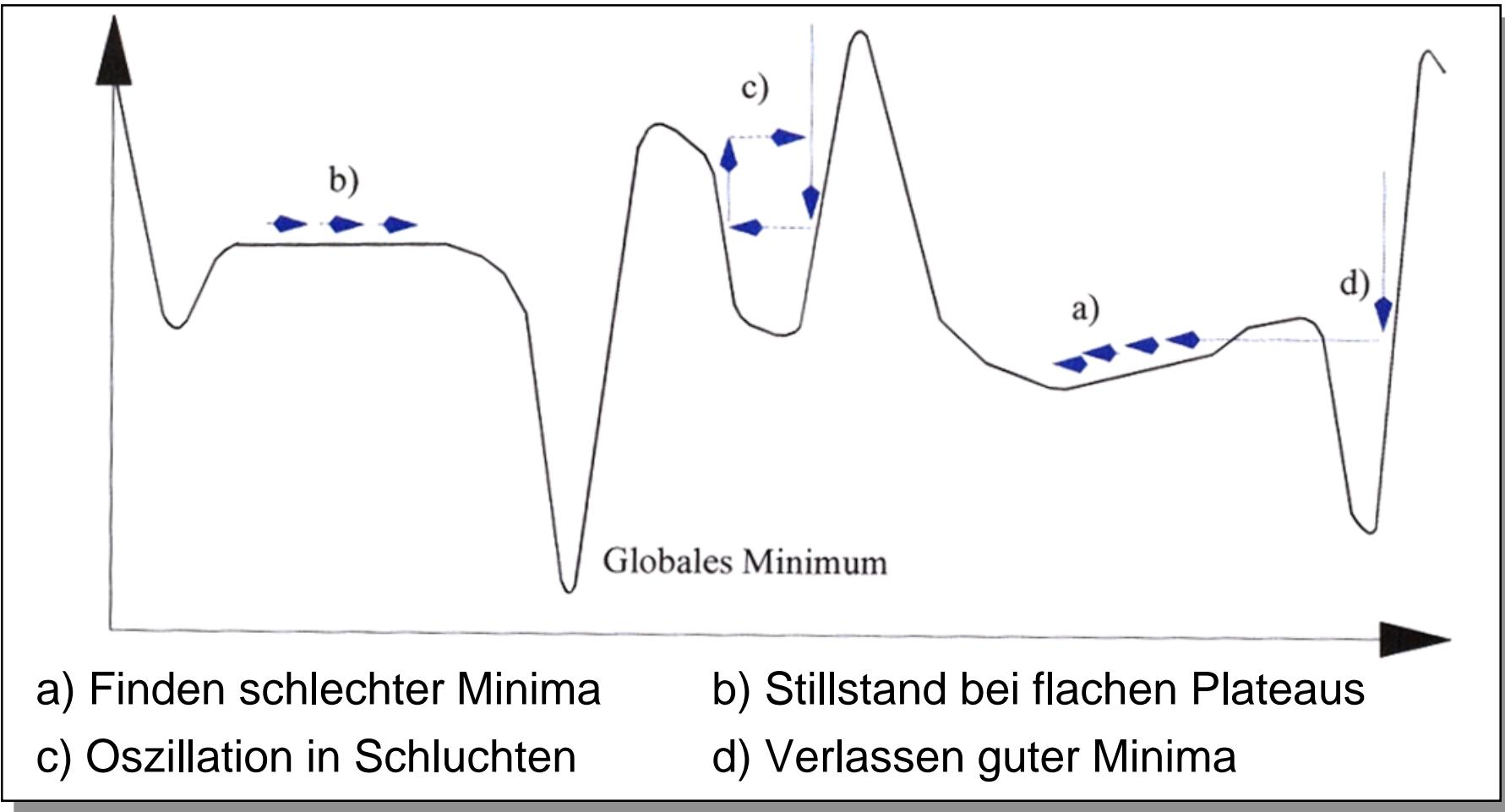
**Problem:** Divergenz bei zu großer Schrittweite:



**Lösung:** Solange  $f(x_{n+1}) \geq f(x_n)$  Schrittweitenfaktor **s verkleinern**:

```
while (f(x - s*f_strich(x)) >= f(x)) {s = s/2}
```

## Weitere mögliche Probleme



- a) Finden schlechterer Minima  
c) Oszillation in Schluchten

- b) Stillstand bei flachen Plateaus  
d) Verlassen guter Minima

# Wie findet der Skifahrer im Nebel zurück ins Tal?



# Minima mehrdimensionaler Funktionen

## Verallgemeinerung des Gradientenabstiegs:

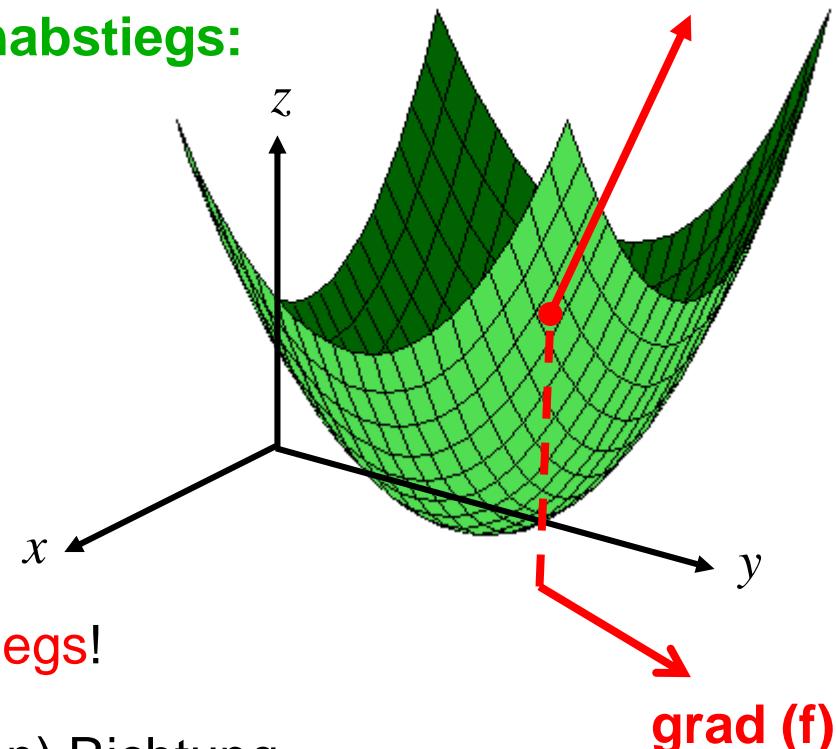
Wie finden wir ein Minimum bei einer *mehrdimensionalen* Funktion?

### Erinnerung: Gradient

= Vektor der partiellen Ableitungen

- zeigt in Richtung des **steilsten Anstiegs!**

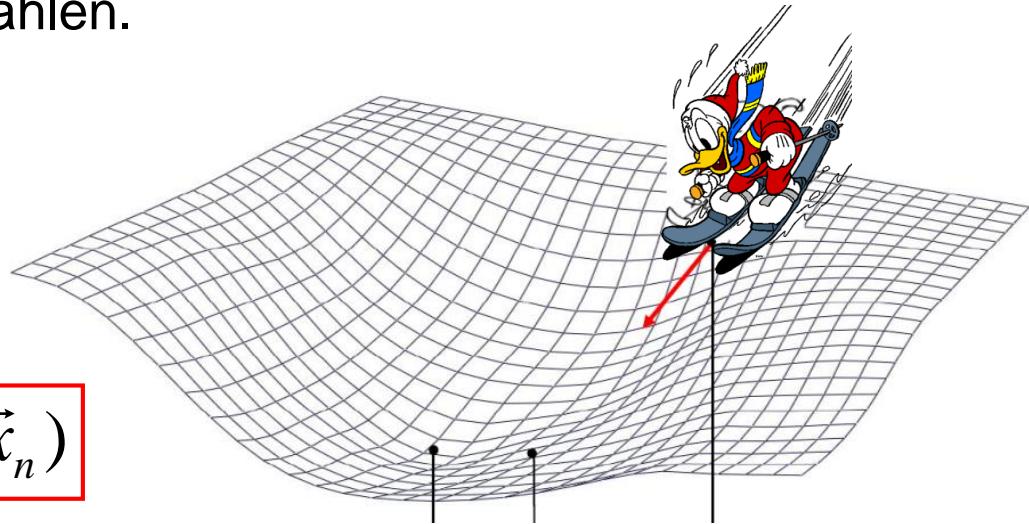
- Betrag = Steigung in dieser (steilsten) Richtung



# Gradientenabstiegsverfahren (mehrdimensional)

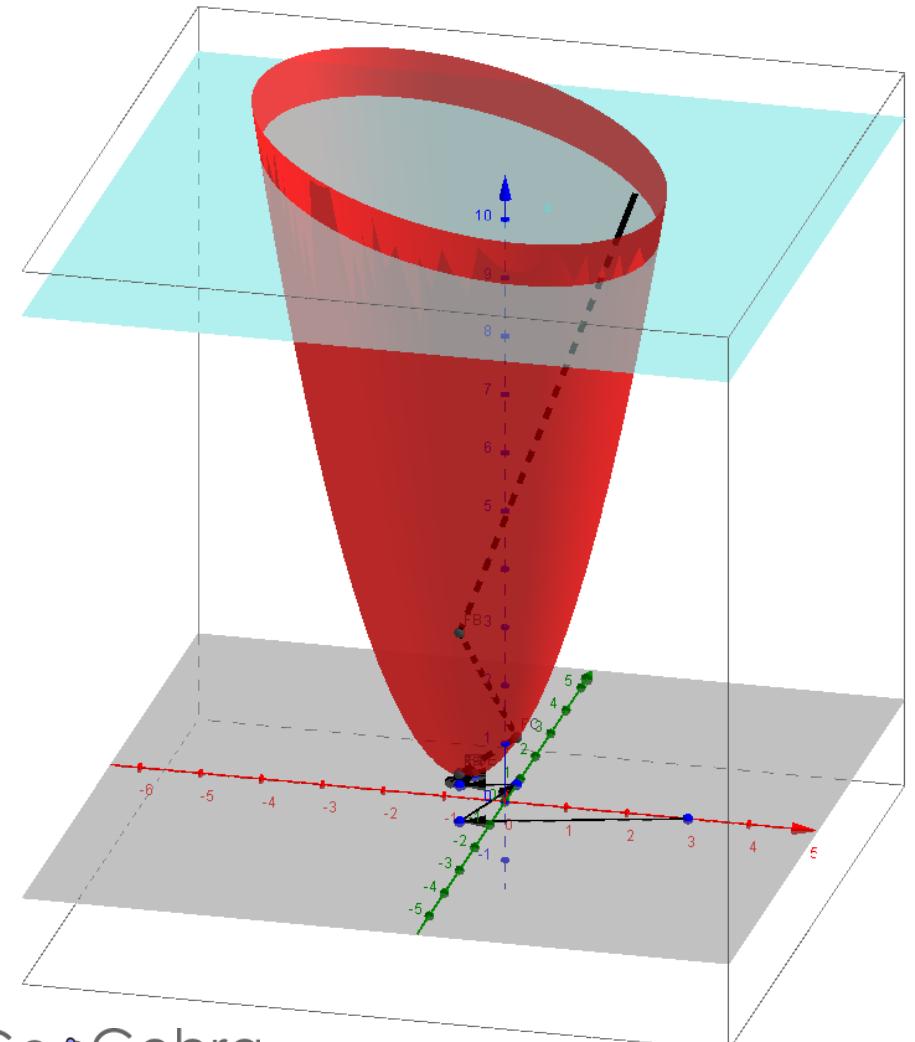
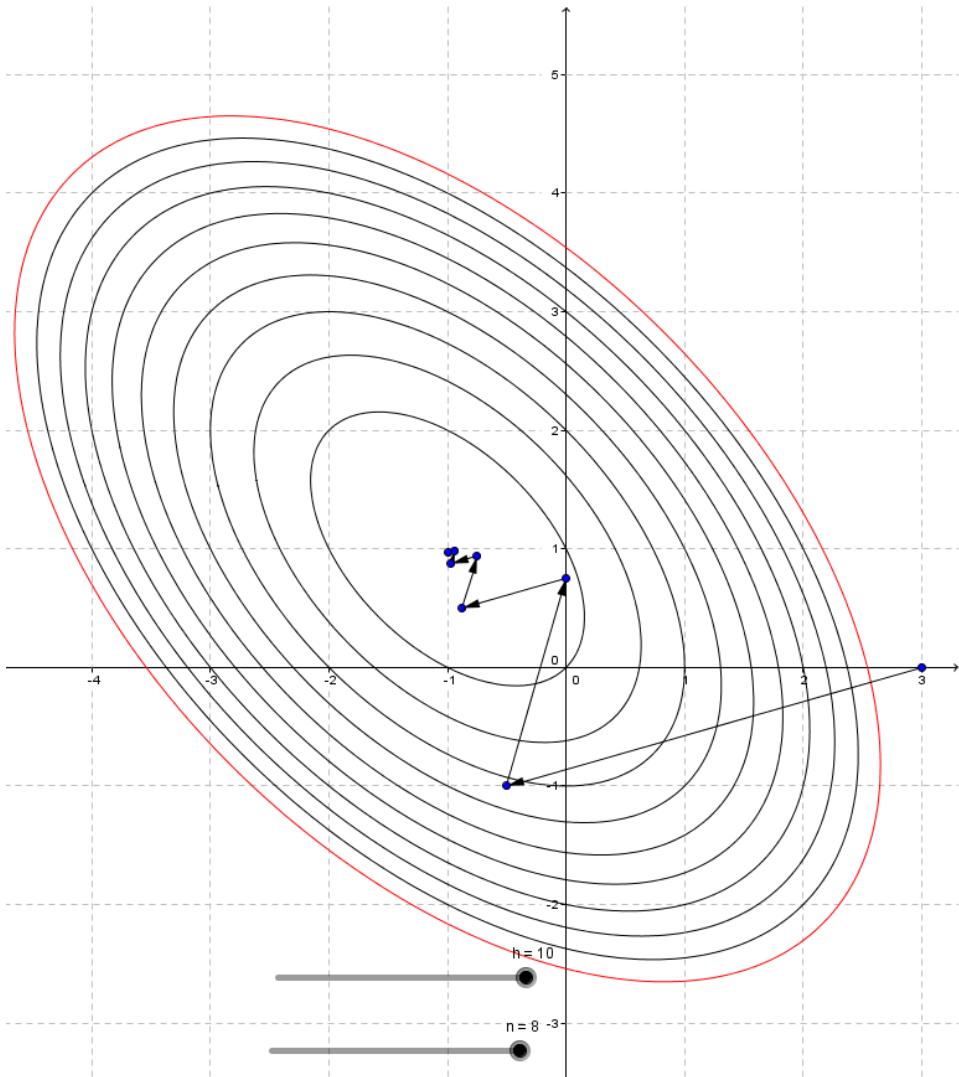
- **Startpunkt**  $\vec{x}_0 = (x_0, y_0)$  wählen.
- In **Richtung** des **negativen** Gradienten forschreiten  
(größter **Abstieg!**):

$$\vec{x}_{n+1} = \vec{x}_n - s \cdot \text{grad } f(\vec{x}_n)$$



- **Wiederholung** bis keine numerische Verbesserung.

# Visualisierung mit GeoGebra



GeoGebra

# Programmierung (mehrdimensional)

The screenshot shows a Java development environment with two windows. On the left is a code editor titled "G:\Javaeditor\Gradientenabstieg Mehrdimensional\Gradientenabstiegzweidim.java". The code implements a gradient descent algorithm for a 2D function. It defines a function `f` and its partial derivatives `f_x` and `f_y`. It also calculates the magnitude of a vector and handles user input for the start point `x0` and `y0`. The right window is titled "Gradientenabstiegzweidim" and displays the iterative steps of the gradient descent process, showing the current values of `x` and `y` in a table.

```
82
83     public double f(double x, double y) { // Funktion
84         return x*x + y*y + x*y + x - y + 1;
85     }
86
87     public double f_x(double x, double y){ // part. Abl. nach x
88         return 2*x + y + 1;
89     }
90
91     public double f_y(double x, double y){ // part. Abl. nach y
92         return 2*y + x - 1;
93     }
94
95     public double betrag(double x, double y){ // Betrag eines Vektors
96         return Math.sqrt(x*x+y*y);
97     }
98
99     public void jButton1ActionPerformed(ActionEvent evt) {
100         double x = jTextField1.getDouble(); // Eingaben x und y
101         double y = jTextField2.getDouble();
102         double s = 0.1; // Startschrittweite
103         while (betrag(f_x(x,y), f_y(x,y))>0.001) { // Solange |grad| zu groß...
104             while (f(x-s*f_x(x,y), y-s*f_y(x,y)) >= f(x,y)) { // ggf. s halbieren
105                 s=s/2;
106             }
107             x = x - s*f_x(x,y); // neues x und y
108             y = y - s*f_y(x,y);
109             jTextArea1.append(String.valueOf(x)+"\n"); // Ausgabe x und y
110             jTextArea2.append(String.valueOf(y)+"\n");
111         }
112     }

```

Startpunkt x0:	y0:
3	0

xi:	yi:
-0.8306732096724078	0.8407461963113623
-0.8486131873690624	0.8574582757859961
-0.8646363774738496	0.8724302583761818
-0.8789521278166978	0.8858394194826152
-0.8917456442016198	0.8978461000062542
-0.9031811253619213	0.9085949925411955
-0.9134043995436566	0.918216433987322
-0.9225451630336575	0.9268276634932233
-0.9307188967762483	0.9345340204722035
-0.938028519468219	0.9414300683245846
-0.9445658224070337	0.9476006369003711
-0.950412721615664	0.9531217816818633
-0.9556423554607175	0.9580616608915624
-0.9603200504577303	0.96248133759023
-0.9645041737420865	0.966435484381427
-0.968246887431812	0.9699730762483229
-0.9715948175702819	0.9731379427556864
-0.9745896483317942	0.975693190377285
-0.9772686505692082	0.9785023202871037
-0.9796651524840769	0.9807683714780907
-0.9818089591350706	0.9827955930959796
-0.9837267266176545	0.9846091471385492
-0.9854422960079785	0.9862315473116372
-0.9869769915375465	0.9876829370030644
-0.9883498869303436	0.988981338295486
-0.9895780433738235	0.9901428749737711
-0.9906767221964359	0.9911819721986604
-0.9916595749770147	0.9921115352566299
-0.9925388135072748	0.9929431095560314
-0.9933253617614229	0.9936870238209674



## Anwendungen

- **Optimierung** von differenzierbaren, **mehrdimensionalen Funktionen** ohne Nebenbedingungen:

$$f(x, y, \dots) = \min!$$

- **Lösen von nichtlinearen Gleichungssystemen:** Die Lösung von

$$g_1(x, y, \dots) = 0$$

$$g_2(x, y, \dots) = 0$$

....

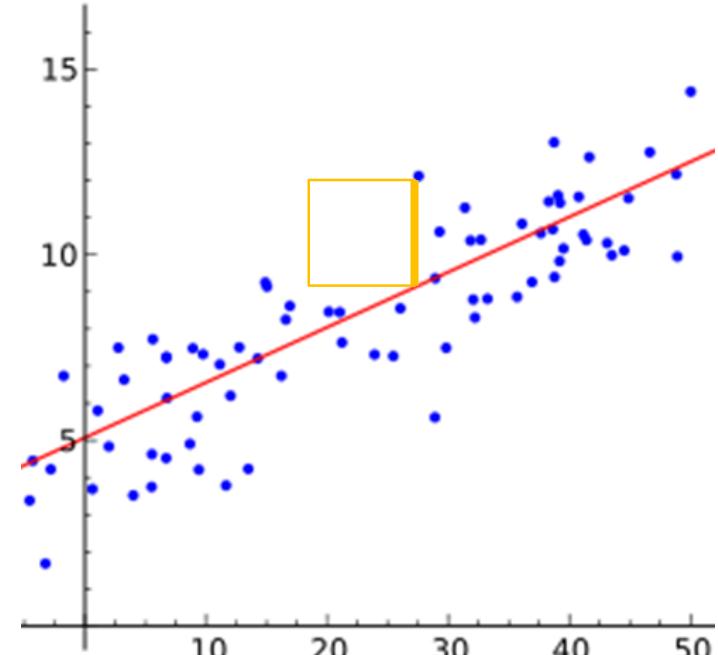
ist äquivalent zum Minimum von  $f(x, y, \dots) = \sum_i g_i^2(x, y, \dots)$ .

## Anwendung: Regression

### Beispiel:

- Welche Funktion  $y = m x + b$  beschreibt die Messpunkte  $(x_i, y_i)$  am besten?
- Summe der Quadrate der Abstände soll minimal werden:

$$f(m, b) = \sum_i (y_i - (m x_i + b))^2$$



- Minimierung der „Fehlerfläche“  $f(m, b)$  durch Gradientenabstieg!

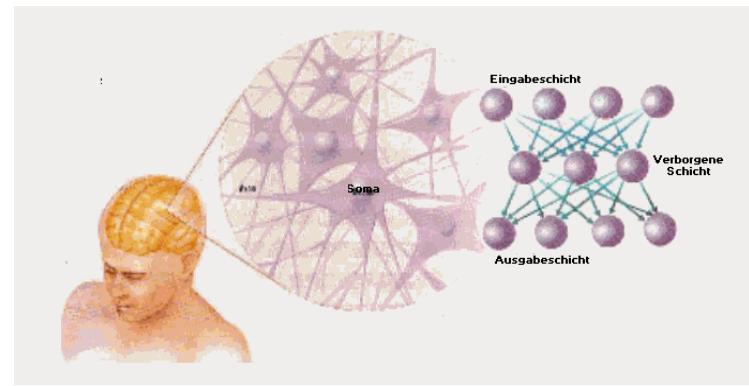
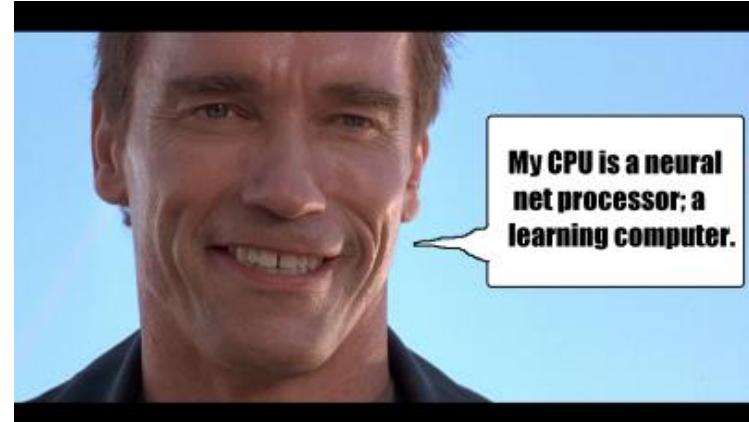
# Wie können Maschinen lernen?



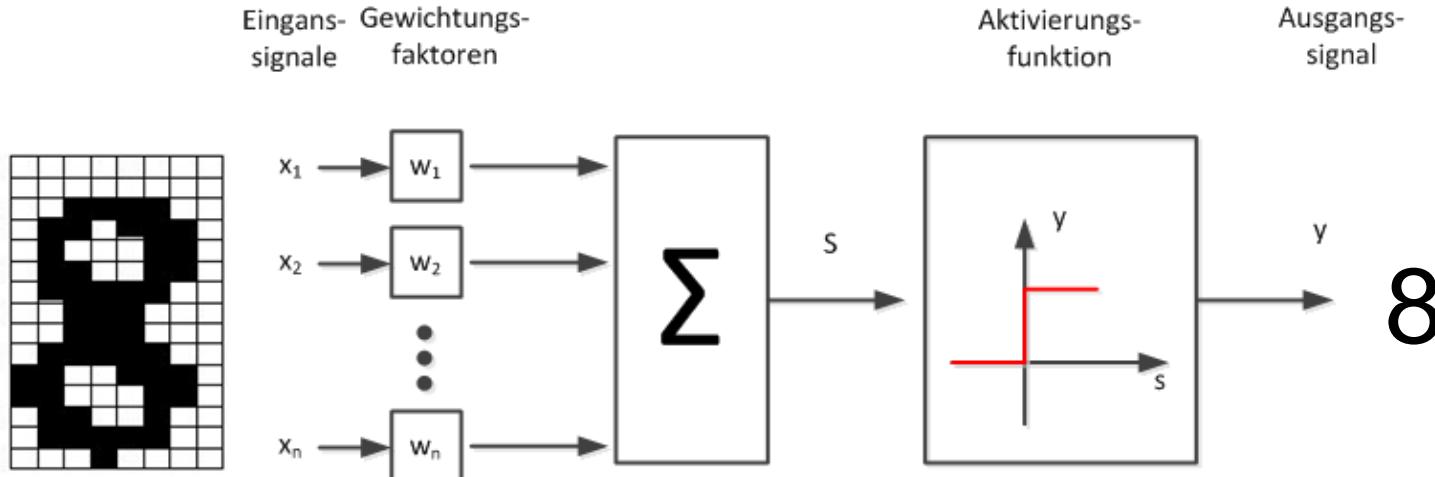
# Anwendung: Lernen bei neuronalen Netzen

## Neuronale Netze :

- von der Natur inspirierte Methode, Computer lernfähig zu machen.
- Anwendungen:
  - Sprach- und Schrifterkennung
  - Optimierungen in der Industrie
  - Künstliche Intelligenz
  - ...
- Mathematisch: Gerichteter Graph, Knoten = künstliche Neuronen



# Künstliches Neuron



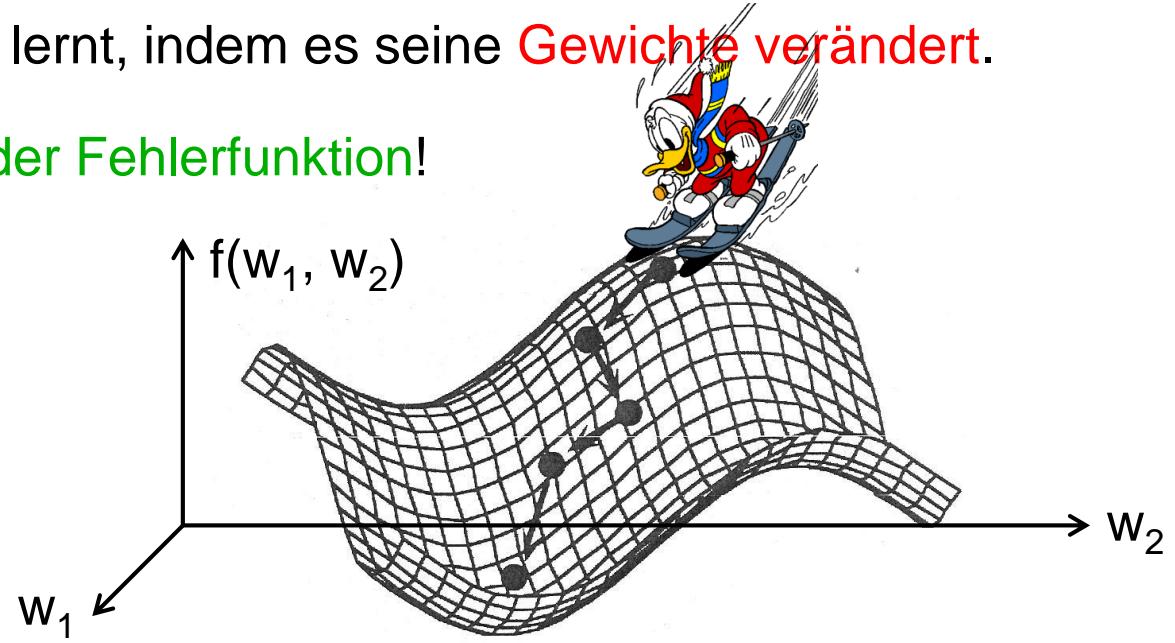
- **Trainingsphase:** Trainingseingaben → Vergleich mit erwarteten Ausgaben.  
Dabei passieren noch Fehler.
- **Ausführungsphase:** Lösung unbekannter Probleme (z.B: geschriebene Ziffern erkennen und diese als Output ausgeben).

# Lernen durch Fehlerminimierung

- Fehler des Neuron-Outputs ist Funktion der Gewichte:

$$f(w_1, w_2, \dots)$$

- Das neuronale Netz lernt, indem es seine Gewichte verändert.
- Gesucht: Minimum der Fehlerfunktion!
- Gradientenabstieg auf Fehlerfläche!





Just after learning the "steepest descent" method  
in optimization class...