

NumProg WS 20/21 : Tutorübung 11

1. Fixpunktiteration: Splitting-Verfahren
2. Jacobi- und Gauss-Seidel Verfahren
3. Verfahren den steilsten Abstiegs
4. Newton-Verfahren

Fixpunktiteration

Ein **Fixpunkt** ist ein Punkt x einer Funktion f , sodass $f(x) = x$ gilt.

Der Punkt bildet über die Funktion auf sich selbst ab.

Fixpunktiteration

Ein **Fixpunkt** ist ein Punkt x einer Funktion f , so dass $f(x) = x$ gilt.

Der Punkt bildet über die Funktion auf sich selbst ab.

Allgemein versuchen Fixpunktverfahren mit einem

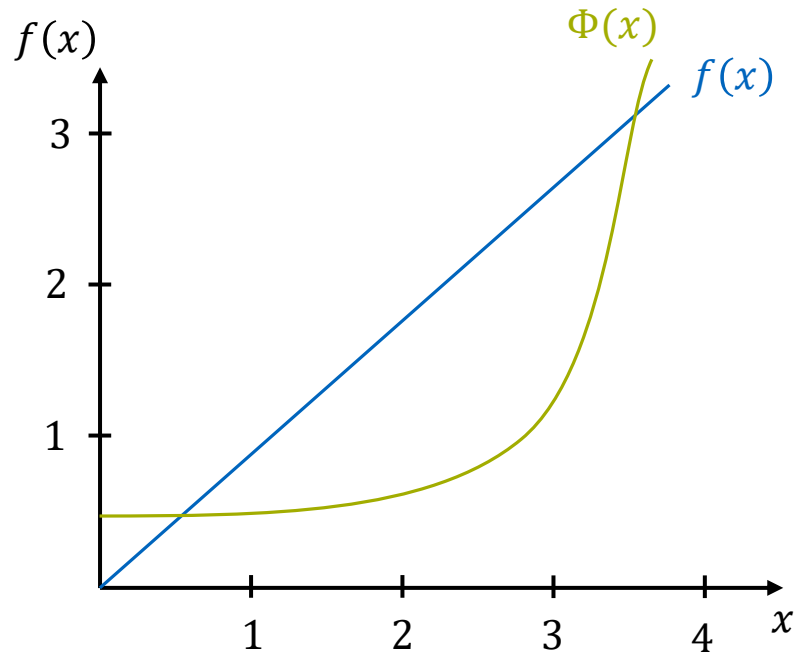
- **Fixpunkt** x^* ,
- der **Iterationsvorschrift** $\Phi(x)$ und
- einem **Startwert** x_0

die Formel $x^* = \Phi(x^*)$ mit Gleichungen der folgenden Form zu lösen:

$$x_{i+1} = \Phi(x_i)$$

Fixpunktiteration

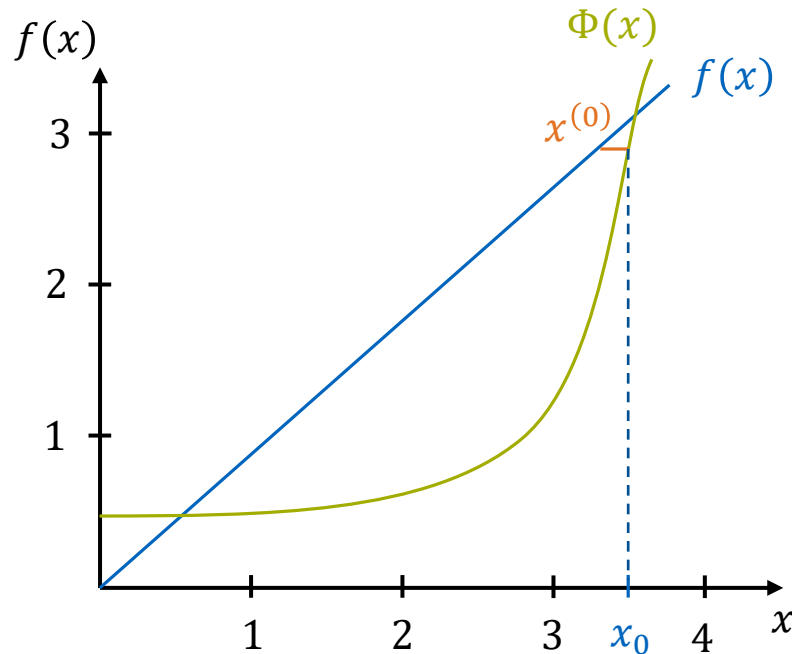
Graphisch sind von der Iterationsfunktion $\Phi(x)$ die Fixpunkte der Funktion $f(x)$ genau auf den **Schnittpunkten** von $\Phi(x)$ und $f(x)$



Man bildet eine „Treppe“, mit der man zwischen $f(x)$ und $\Phi(x)$ hin und herspringt.

Fixpunktiteration

Graphisch sind von der Iterationsfunktion $\Phi(x)$ die Fixpunkte der Funktion $f(x)$ genau auf den **Schnittpunkten** von $\Phi(x)$ und $f(x)$

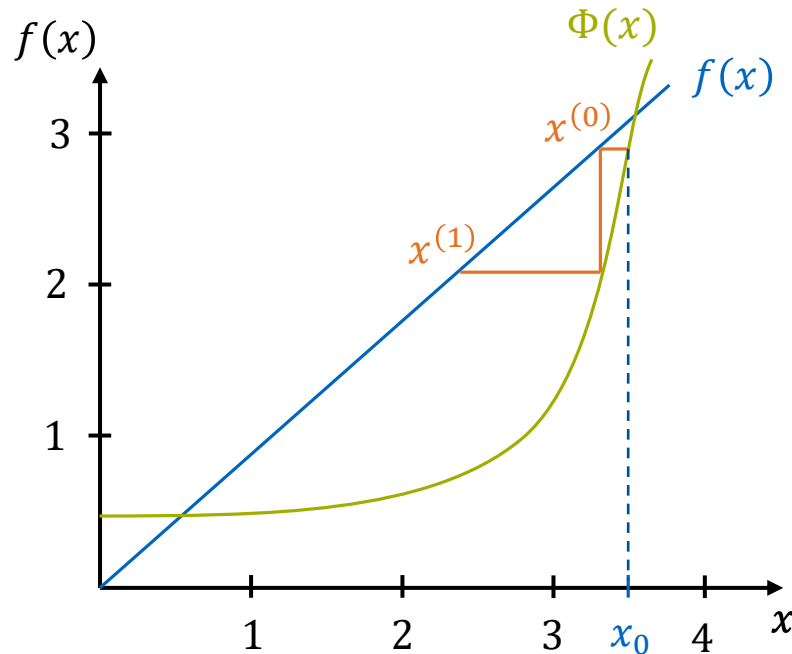


Man bildet eine „Treppe“, mit der man zwischen $f(x)$ und $\Phi(x)$ hin und herspringt.

Man beginnt bei x_0 auf $\Phi(x)$ und zieht eine gerade Linie nach links oder rechts bis man auf $f(x)$ trifft.

Fixpunktiteration

Graphisch sind von der Iterationsfunktion $\Phi(x)$ die Fixpunkte der Funktion $f(x)$ genau auf den **Schnittpunkten** von $\Phi(x)$ und $f(x)$



Man bildet eine „Treppe“, mit der man zwischen $f(x)$ und $\Phi(x)$ hin und herspringt.

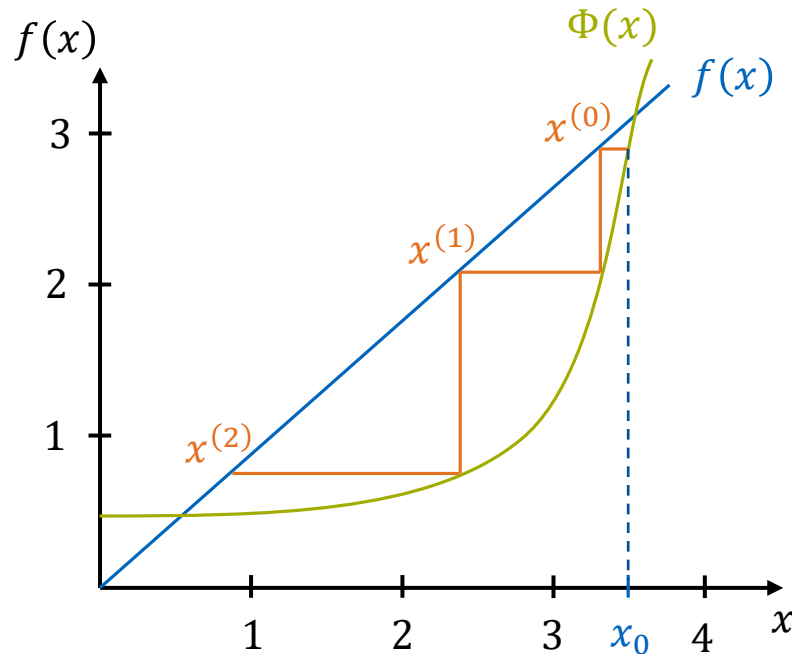
Man beginnt bei x_0 auf $\Phi(x)$ und zieht eine gerade Linie nach links oder rechts bis man auf $f(x)$ trifft.

Dann zieht man eine gerade Linie nach oben oder unten zurück auf $\Phi(x)$ und wieder eine Linie nach rechts oder links zu $f(x)$.

→ ein Iterationsschritt

Fixpunktiteration

Graphisch sind von der Iterationsfunktion $\Phi(x)$ die Fixpunkte der Funktion $f(x)$ genau auf den **Schnittpunkten** von $\Phi(x)$ und $f(x)$



Man bildet eine „Treppe“, mit der man zwischen $f(x)$ und $\Phi(x)$ hin und herspringt.

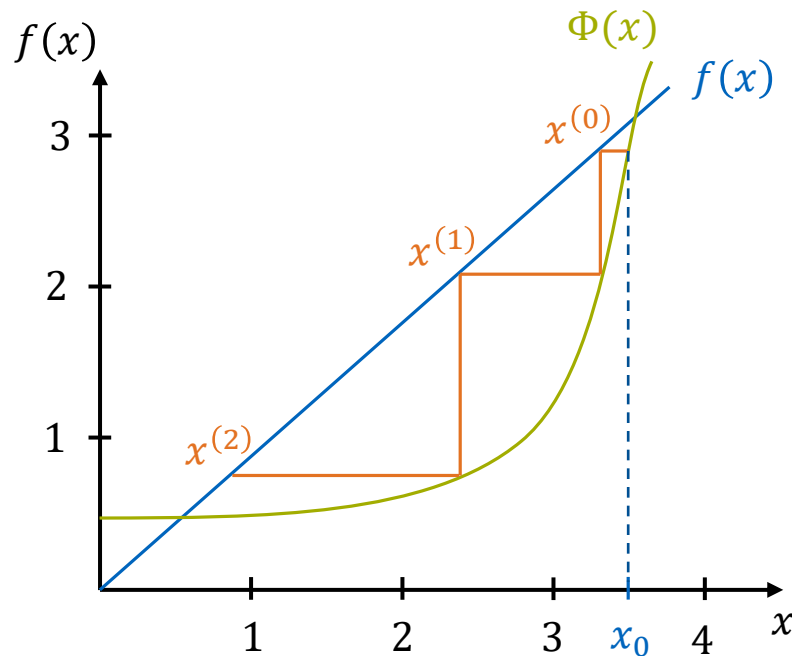
Man beginnt bei x_0 auf $\Phi(x)$ und zieht eine gerade Linie nach links oder rechts bis man auf $f(x)$ trifft.

Dann zieht man eine gerade Linie nach oben oder unten zurück auf $\Phi(x)$ und wieder eine Linie nach rechts oder links zu $f(x)$.

→ ein Iterationsschritt

Fixpunktiteration

Graphisch sind von der Iterationsfunktion $\Phi(x)$ die Fixpunkte der Funktion $f(x)$ genau auf den **Schnittpunkten** von $\Phi(x)$ und $f(x)$

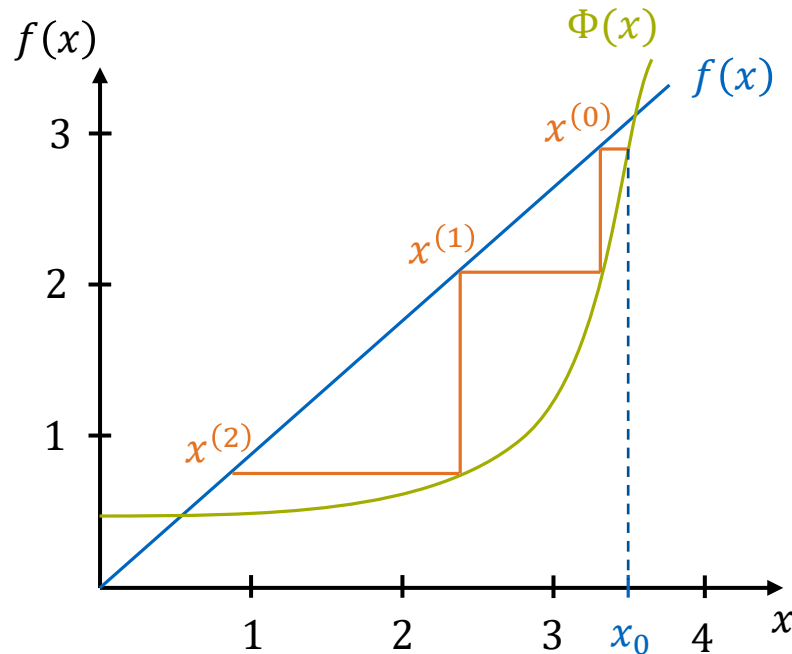


Es gibt verschiedene Arten von Fixpunkten (von $\Phi(x)$ abhängig)

- $|\Phi'(x^*)| < 1 \Rightarrow$ anziehender Fixpunkt (konvergiert)
- $|\Phi'(x^*)| > 1 \Rightarrow$ abstoßender Fixpunkt (divergiert)
- $|\Phi'(x^*)| = 1 \Rightarrow$ indifferent

Fixpunktiteration

Graphisch sind von der Iterationsfunktion $\Phi(x)$ die Fixpunkte der Funktion $f(x)$ genau auf den **Schnittpunkten** von $\Phi(x)$ und $f(x)$



Es gibt verschiedene Arten von Fixpunkten (von $\Phi(x)$ abhängig)

- $|\Phi'(x^*)| < 1 \Rightarrow$ anziehender Fixpunkt (konvergiert)
- $|\Phi'(x^*)| > 1 \Rightarrow$ abstoßender Fixpunkt (divergiert)
- $|\Phi'(x^*)| = 1 \Rightarrow$ indifferent

Generell: Ist ein Fixpunkt anziehend, ist die Iterationsvorschrift gut.

Splitting-Verfahren

Mit dem Splitting-Verfahren können wir **Gleichungssysteme iterativ lösen**.

Splitting-Verfahren

Mit dem Splitting-Verfahren können wir **Gleichungssysteme iterativ lösen**.

Die Idee ist, dass wir wieder Fixpunktiteration benutzen mit der Iterationsvorschrift

$$\Phi(x) = x + M^{-1}(b - Ax)$$

und dem Fixpunkt x^* , der die Lösung des Gleichungssystems $Ax^* = b$ darstellt.

Splitting-Verfahren

Mit dem Splitting-Verfahren können wir **Gleichungssysteme iterativ lösen**.

Die Idee ist, dass wir wieder Fixpunktiteration benutzen mit der Iterationsvorschrift

$$\Phi(x) = x + M^{-1}(b - Ax)$$

und dem Fixpunkt x^* , der die Lösung des Gleichungssystems $Ax^* = b$ darstellt.

Wir wählen M als Approximation zu A , um das Gleichungssystem zu lösen.

Splitting-Verfahren

Mit dem Splitting-Verfahren können wir **Gleichungssysteme iterativ lösen**.

Die Idee ist, dass wir wieder Fixpunktiteration benutzen mit der Iterationsvorschrift

$$\Phi(x) = x + M^{-1}(b - Ax)$$

und dem Fixpunkt x^* , der die Lösung des Gleichungssystems $Ax^* = b$ darstellt.

Wir wählen M als Approximation zu A , um das Gleichungssystem zu lösen.

Je nach dem welche Matrix M man wählt, hat man unterschiedliche Eigenschaften:

- **M ähnlich zu A :** bessere Konvergenz, sehr teuer (Inverse einer vollen Matrix $\mathcal{O}(n^3)$)
- **M unähnlich zu A :** schlechtere Konvergenz, nicht so teuer

Jacobi- und Gauss-Seidel-Verfahren

Jacobi-Verfahren:

$x^{(k+1)} = x^{(k)} + \text{diag}(A)^{-1}(b - Ax^{(k)})$ in 3 Schritte aufgeteilt:

$$1) \quad r^{(k)} = b - Ax^{(k)}$$

$$2) \quad y_i^{(k)} = \frac{1}{a_{ii}} \cdot r_i^{(k)}$$

$$3) \quad x^{(k+1)} = y^{(k)} + x^{(k)}$$

Jacobi- und Gauss-Seidel-Verfahren

Gauss-Seidel-Verfahren: „in-place“ Prinzip. Wir benutzen im selben Iterationsschritt alle Komponenten vor i , die schon berechnet wurden.

$$1) \quad r_i^{(k)} = b_i - \sum_{j=1}^{i-1} (a_{ij} x_j^{(k+1)}) - \sum_{j=i}^n (a_{ij} x_j^{(k)})$$

$$2) \quad y_i^{(k)} = \frac{1}{a_{ii}} \cdot r_i^{(k)}$$

$$3) \quad x_i^{(k+1)} = y_i^{(k)} + x_i^{(k)}$$

Jacobi- und Gauss-Seidel-Verfahren

Jacobi-Verfahren:

$x^{(k+1)} = x^{(k)} + \text{diag}(A)^{-1}(b - Ax^{(k)})$ in 3 Schritte aufgeteilt:

$$1) \quad r^{(k)} = b - Ax^{(k)}$$

$$2) \quad y_i^{(k)} = \frac{1}{a_{ii}} \cdot r_i^{(k)}$$

$$3) \quad x^{(k+1)} = y^{(k)} + x^{(k)}$$

Gauss-Seidel-Verfahren:

„in-place“ Prinzip. Wir benutzen im selben Iterationsschritt alle Komponenten vor i , die schon berechnet wurden.

$$1) \quad r_i^{(k)} = b_i - \sum_{j=1}^{i-1} (a_{ij}x_j^{(k+1)}) - \sum_{j=i}^n (a_{ij}x_j^{(k)})$$

$$2) \quad y_i^{(k)} = \frac{1}{a_{ii}} \cdot r_i^{(k)}$$

$$3) \quad x_i^{(k+1)} = y_i^{(k)} + x_i^{(k)}$$

Verfahren des steilsten Abstiegs

Das Verfahren des steilsten Abstiegs ist eine Alternative zu den vorherigen Methoden, falls Matrix A **symmetrisch** und **positiv definit** ist (d.h. alle Eigenwerte > 0).

Verfahren des steilsten Abstiegs

Das Verfahren des steilsten Abstiegs ist eine Alternative zu den vorherigen Methoden, falls Matrix A **symmetrisch** und **positiv definit** ist (d.h. alle Eigenwerte > 0).

Wir benutzen hier einen Minimierungsansatz, indem wir

- zuerst die **Hyperfläche** definieren: $f(x) = x^T A x - b^T x + c$
 - diese dann zum **Gradienten** ableiten: $\nabla f(x) = A x - b$
 - den dann $= 0$ setzen und als **Lösung** erhalten: $A x = b$
- Minimum der Hyperfläche ist Lösung des Gleichungssystems

Verfahren des steilsten Abstiegs

In anderen Worten:

Man bestimmt zu jedem Zeitpunkt den Gradienten im aktuellen Punkt ($= \nabla f(x) = Ax - b$) und geht dann eine bestimmte Länge in negative Gradientenrichtung (steilster Abstieg $= r^{(i)}$).

Schrittweite $\alpha^{(i)}$ ist genau der Weg vom aktuellen Punkt bis zum Minimum in dieser Richtung.

Verfahren des steilsten Abstiegs

In anderen Worten:

Man bestimmt zu jedem Zeitpunkt den Gradienten im aktuellen Punkt ($= \nabla f(x) = Ax - b$) und geht dann eine bestimmte Länge in negative Gradientenrichtung (steilster Abstieg $= r^{(i)}$).

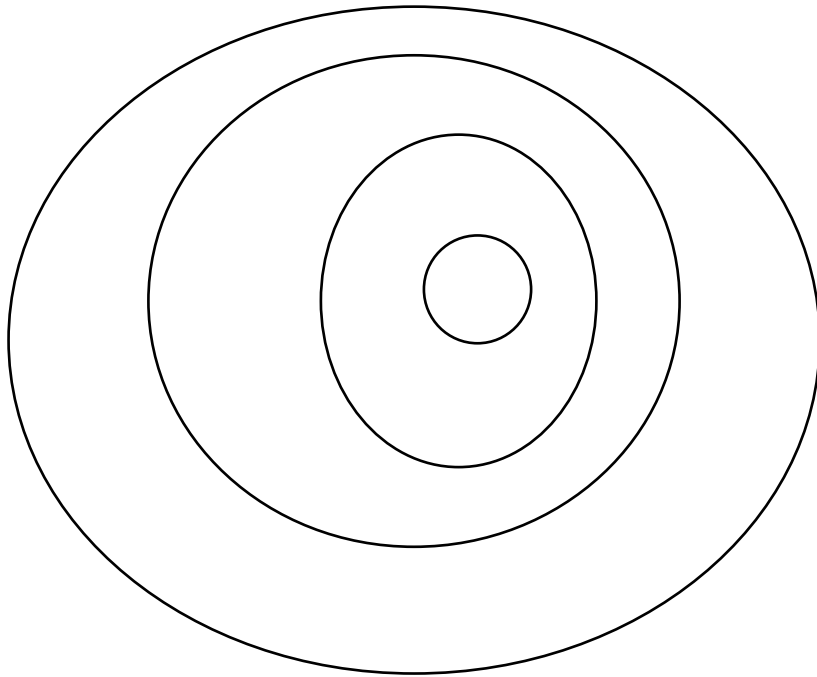
Schrittweite $\alpha^{(i)}$ ist genau der Weg vom aktuellen Punkt bis zum Minimum in dieser Richtung.

Jeder „Zeitpunkt“ k ist ein Iterationsschritt und lässt sich in 3 Unterschritte aufteilen:

- | | | |
|----|--|---|
| 1) | $r^{(k)} = b - Ax^{(k)}$ | aktuelles Residuum (Schrittrichtung) |
| 2) | $\alpha^{(k)} = \frac{r^{(k)T} r^{(k)}}{r^{(k)T} A r^{(k)}}$ | Schrittweite bis zum Minimum in Schrittrichtung |
| 3) | $x^{(k+1)} = x^{(k)} + \alpha^{(k)} r^{(k)}$ | aktuelle Position |

Verfahren des steilsten Abstiegs

Graphisch wird das häufig so dargestellt, dass man die Höhenlinien der Hyperfläche (also alle Punkte mit denselben y -Werten) als Kreise einzeichnet.



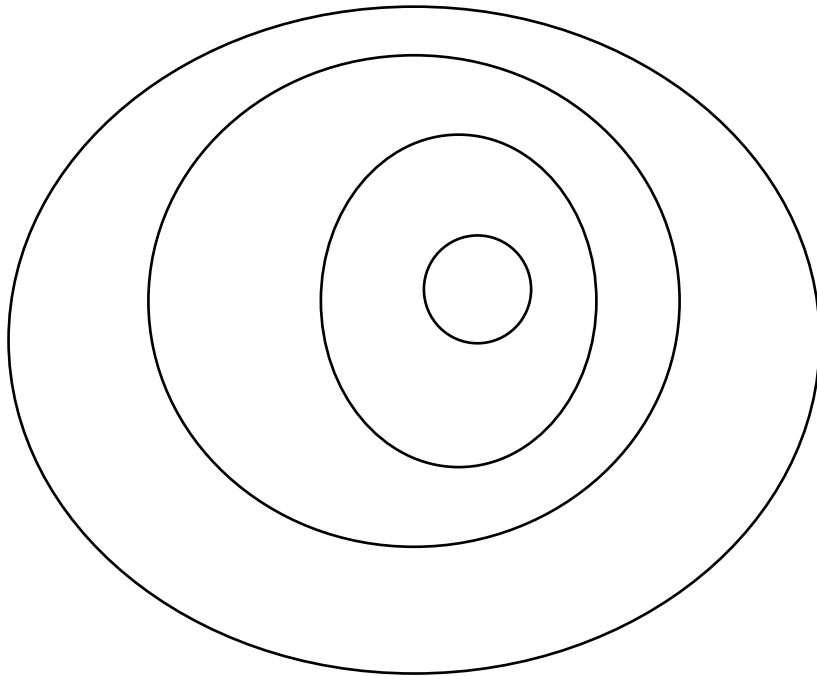
Je gleichmäßiger die Kreise ineinander liegen, desto schneller konvergiert das Verfahren.

→ konvergiert sofort bei konzentrischen Kreisen

In der Praxis sind diese Kreise aber weite Ellipsen, weshalb das Verfahren meist sehr langsam konvergiert.

Verfahren des steilsten Abstiegs

Graphisch wird das häufig so dargestellt, dass man die Höhenlinien der Hyperfläche (also alle Punkte mit denselben y-Werten) als Kreise einzeichnet.



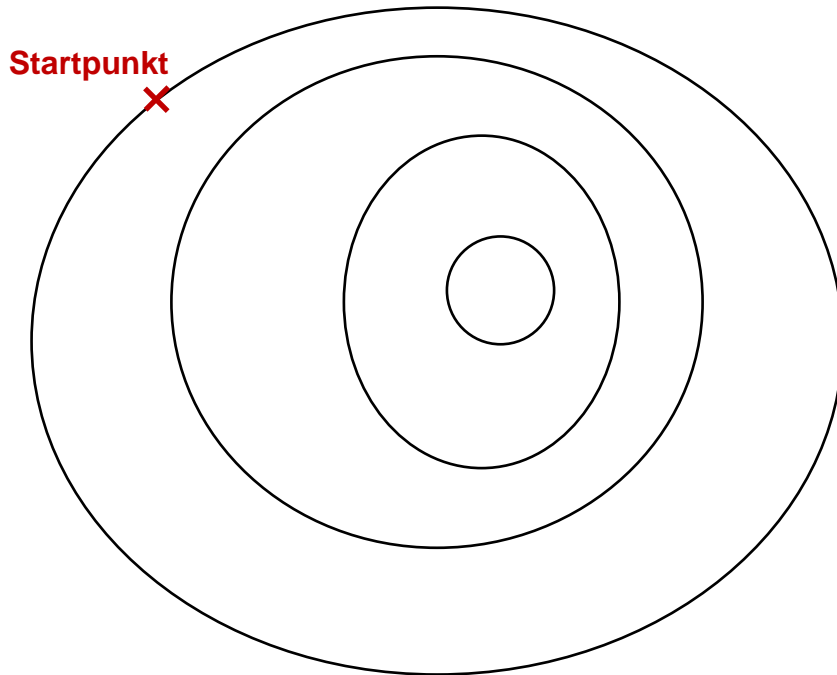
Bei jedem Iterationsschritt ist die Schritttrichtung senkrecht zur momentanen Höhenlinie, da dies der jeweils steilste Abstieg in dem Punkt ist.

α ist die Schrittweite vom aktuellen Punkt bis zum Minimum im „Splice“ der Hyperfläche in eben genannter Schritttrichtung.

Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):

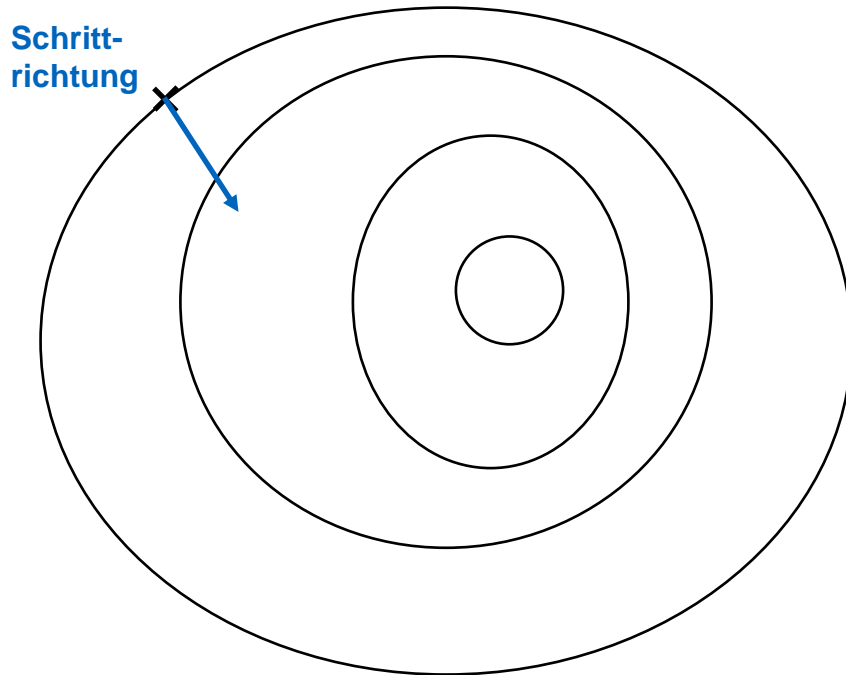


Beginnend von einem **Startpunkt** aus
beginnen wir dem ersten Iterationsschritt.

Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):

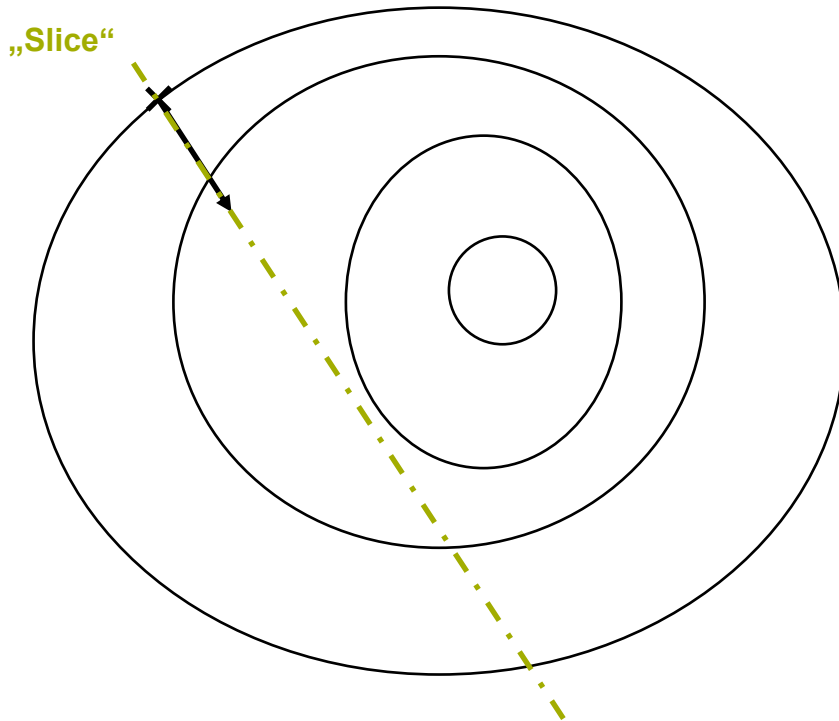


Horizontal von der Höhenlinie, auf der sich der Startpunkt befindet, zeichnen wir die **Schrittrichtung** ein.

Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):

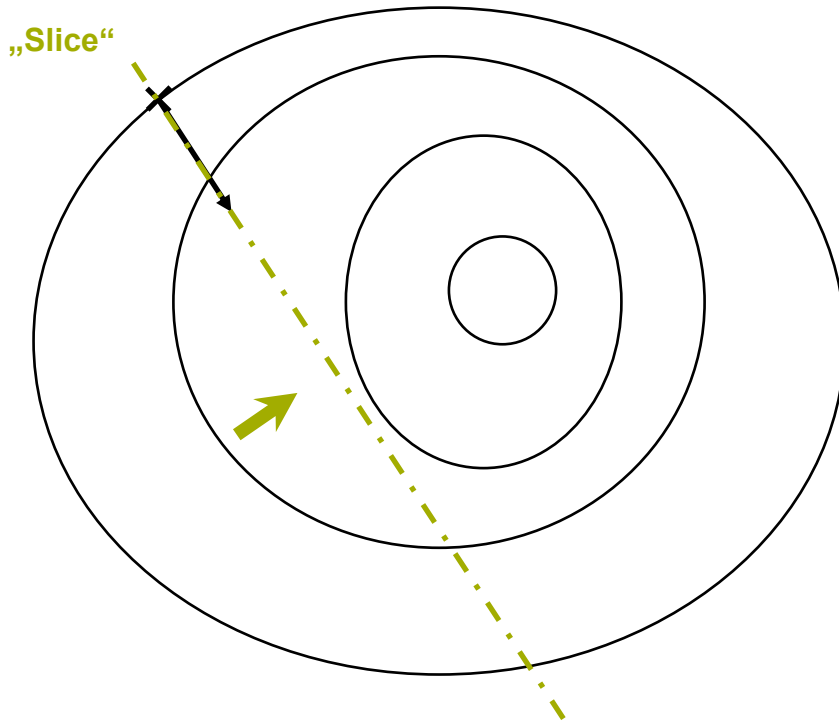


In der gleichen Richtung zeichnen wir jetzt ein „**Slice**“ ein, das unseren Teilschnitt der Hyperfläche bestimmt.

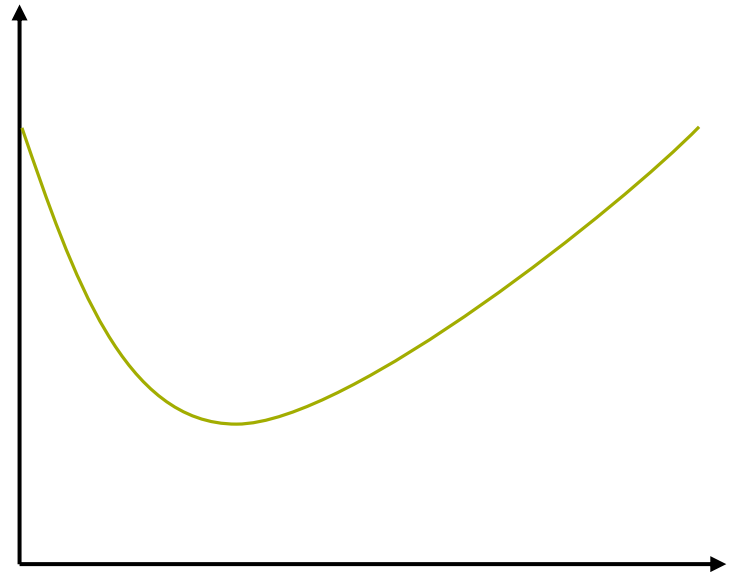
Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):



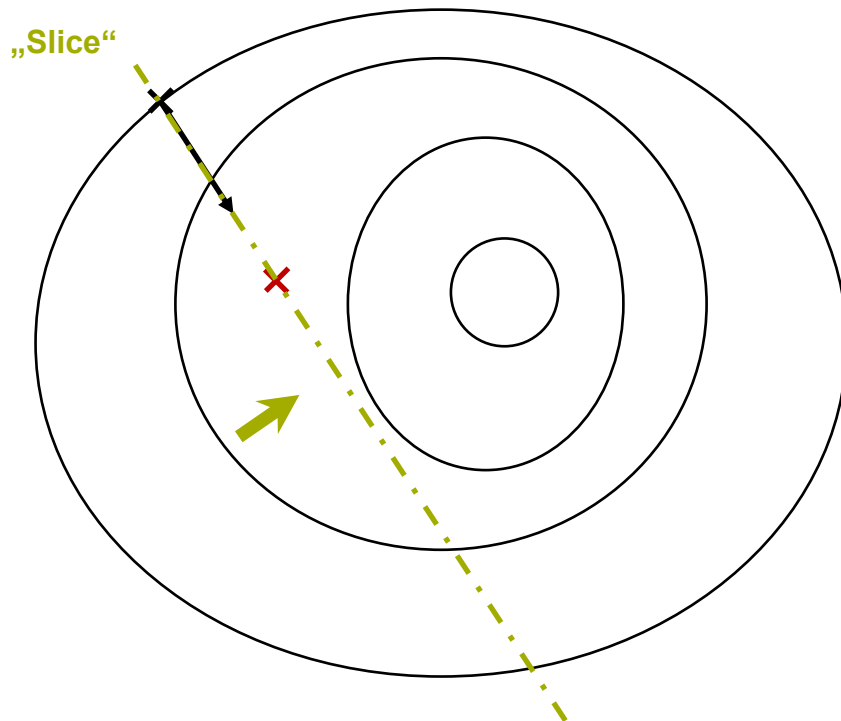
„Slice“ (Seitenansicht, mit Pfeil markiert):



Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):



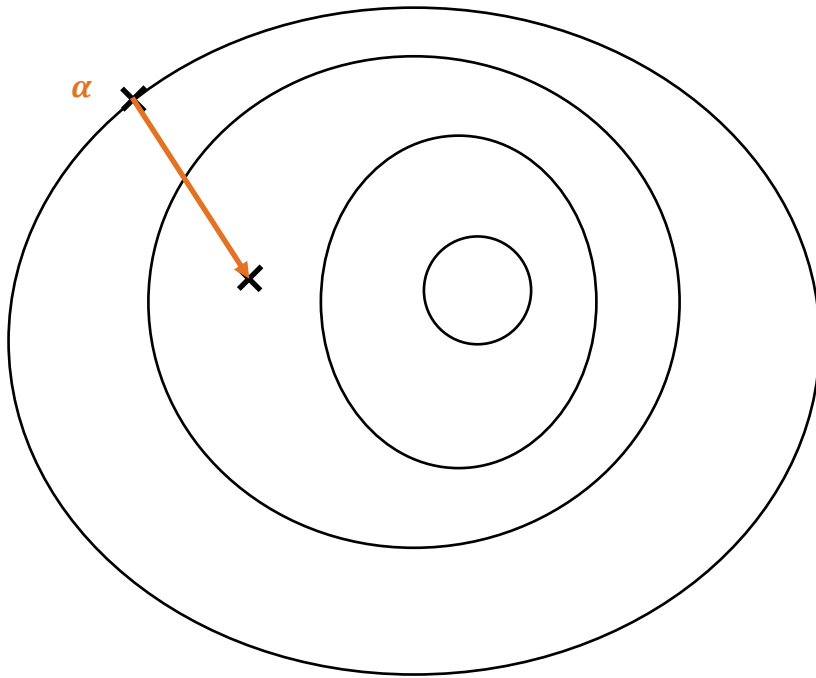
„Slice“ (Seitenansicht, mit Pfeil markiert):



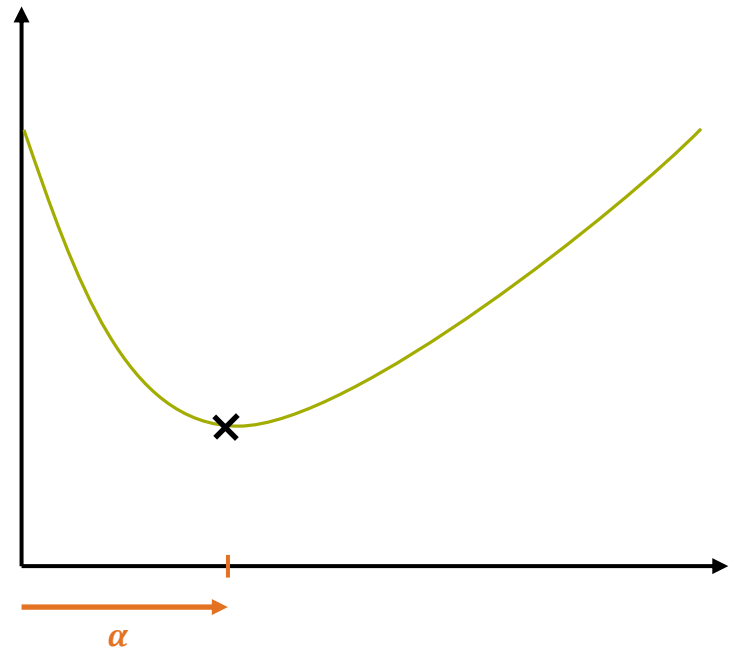
Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):



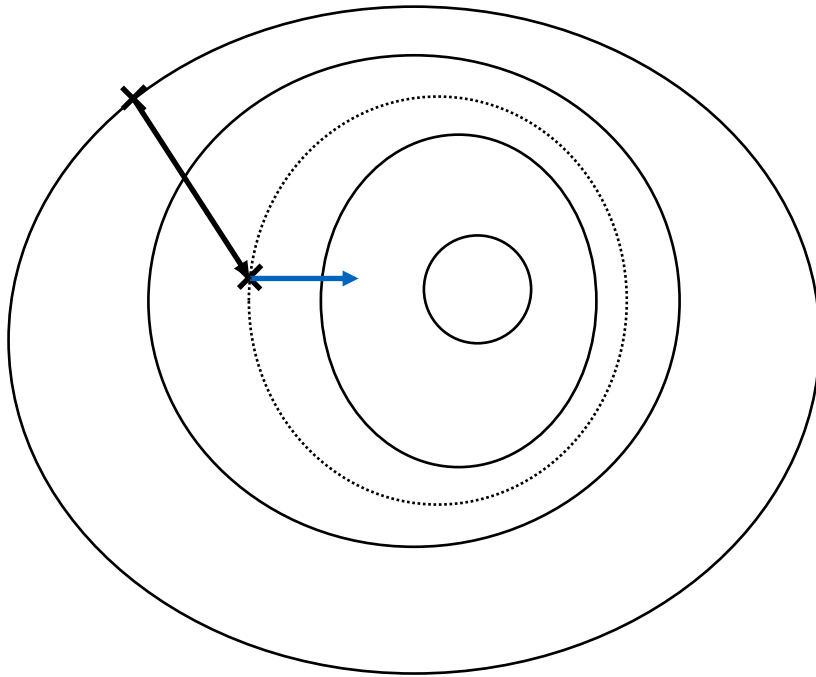
Slice von eben:



Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):



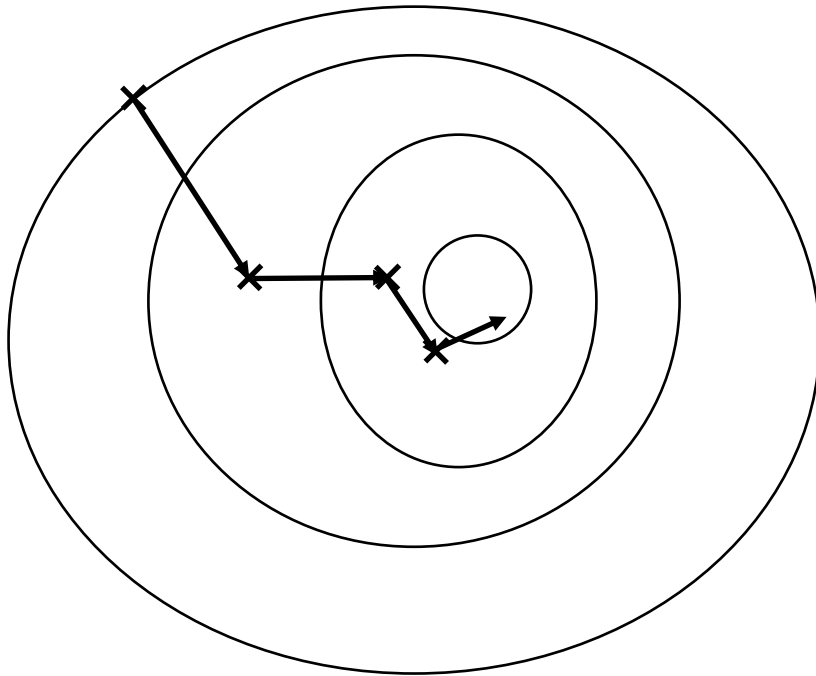
Nun ist der erste Iterationsschritt abgeschlossen.

Wir können beginnend mit der aktuellen **Schrittrichtung** auf der aktuellen Höhenlinie einen weiteren Iterationsschritt durchführen.

Verfahren des steilsten Abstiegs

Wie sehen diese Iterationsschritte jetzt graphisch aus?

Höhenlinien (Draufsicht):



Wir können die Iteration beliebig oft durchführen und nähern uns dem Minimum der Hyperfläche in einem „Zick-Zack“ Muster langsam an.

Newton-Verfahren

Mithilfe des Newton-Verfahrens können wir **Nullstellen** von $f(x)$ iterativ **approximieren**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Wie funktioniert das informell?

Newton-Verfahren

Mithilfe des Newton-Verfahrens können wir **Nullstellen** von $f(x)$ iterativ **approximieren**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Wie funktioniert das informell?

Wir legen eine Tangente auf $f(x)$ in x_0 und bestimmen die Nullstelle \widetilde{x}_0 dieser Tangente

$$\rightarrow x_1 = \widetilde{x}_0$$

Newton-Verfahren

Mithilfe des Newton-Verfahrens können wir **Nullstellen** von $f(x)$ iterativ **approximieren**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Wie funktioniert das informell?

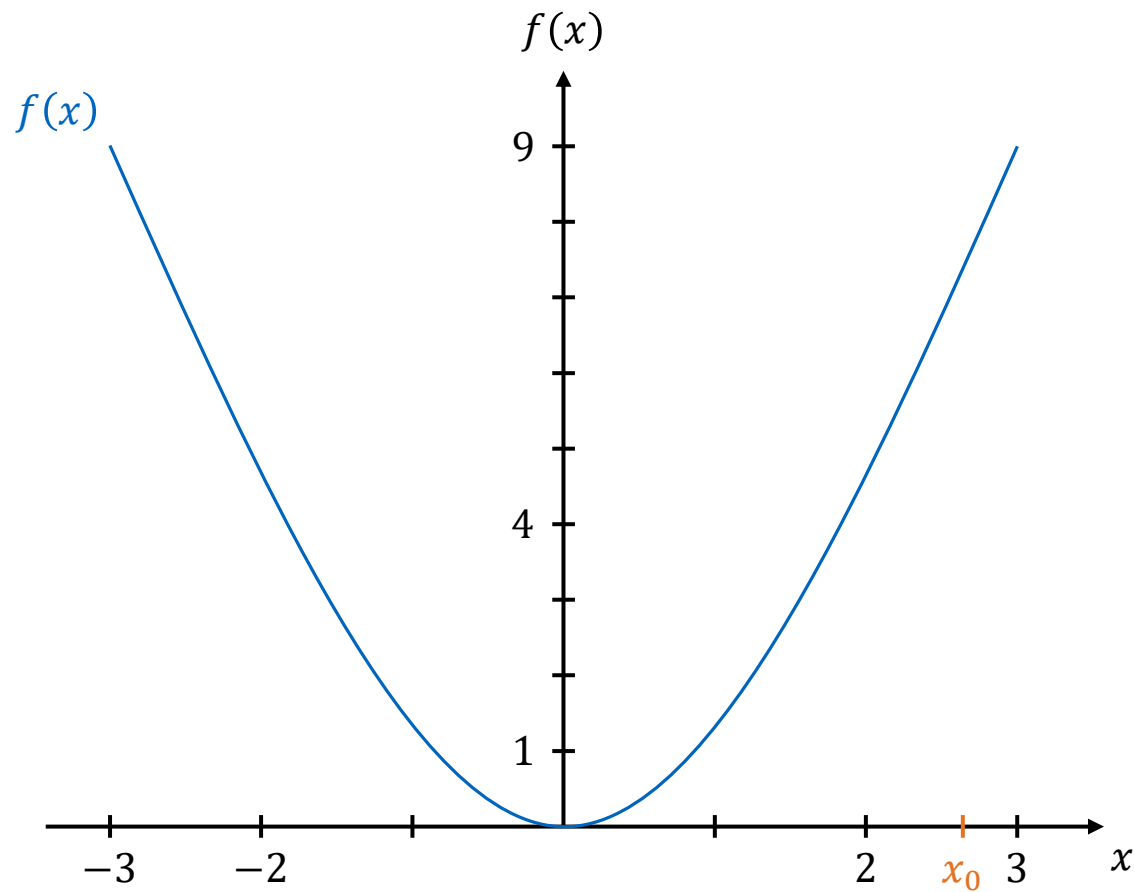
Wir legen eine Tangente auf $f(x)$ in x_0 und bestimmen die Nullstelle \widetilde{x}_0 dieser Tangente

$$\rightarrow x_1 = \widetilde{x}_0$$

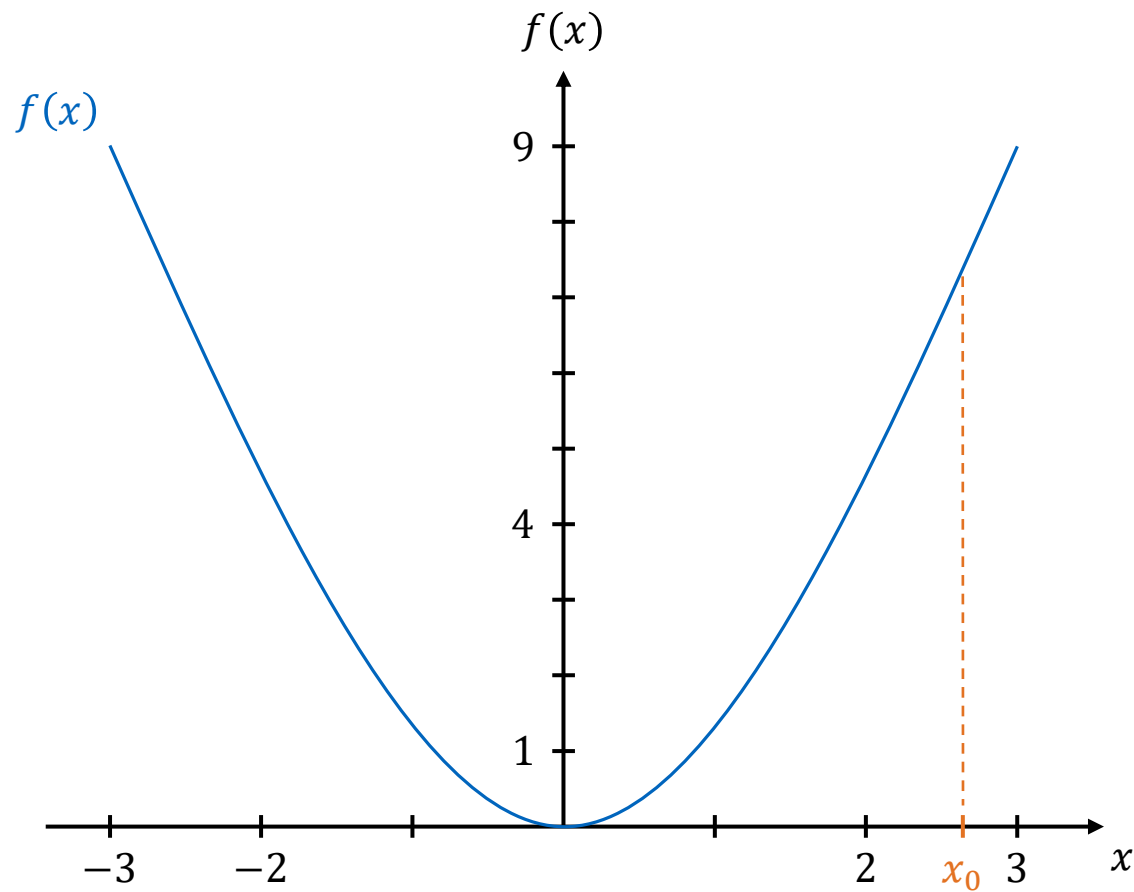
Je öfter wir das wiederholen, desto näher kommen wir der Nullstelle

$$\rightarrow x_{k+1} = \widetilde{x}_k$$

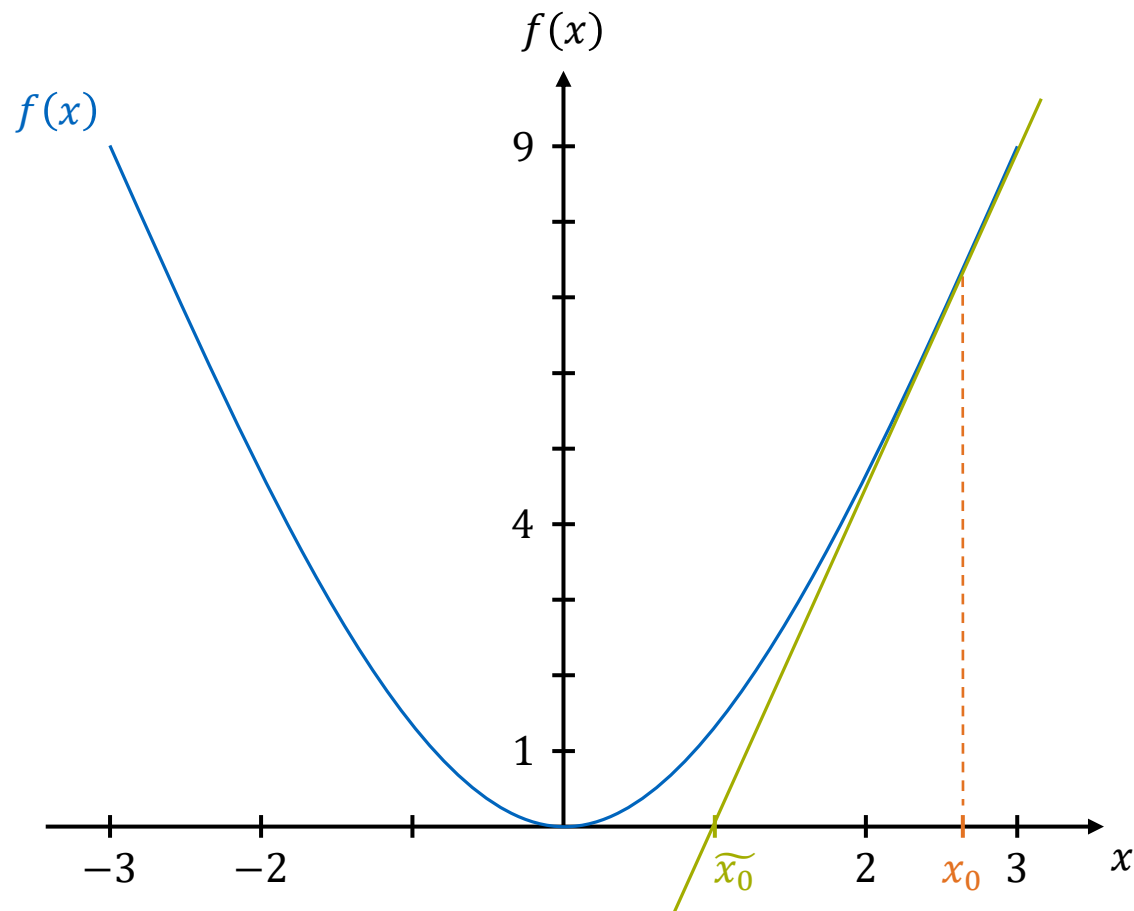
Newton-Verfahren



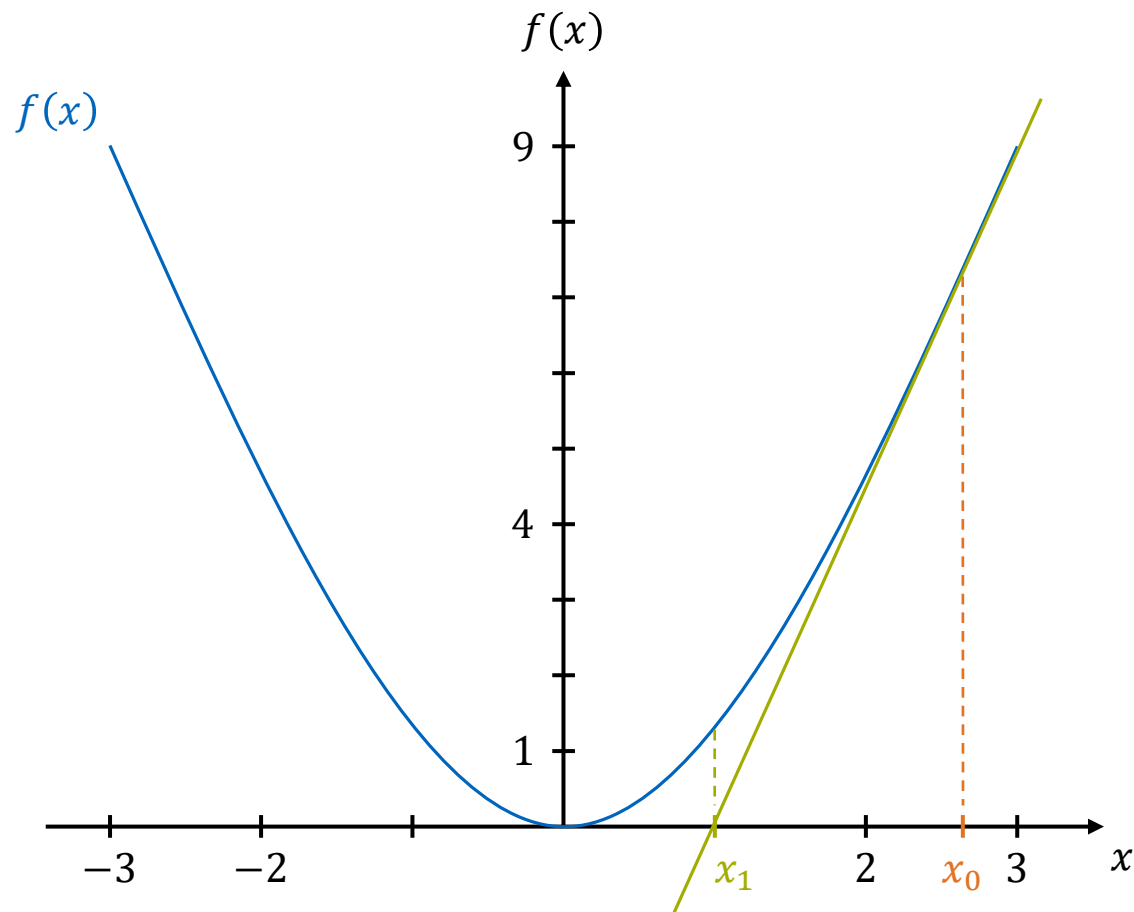
Newton-Verfahren



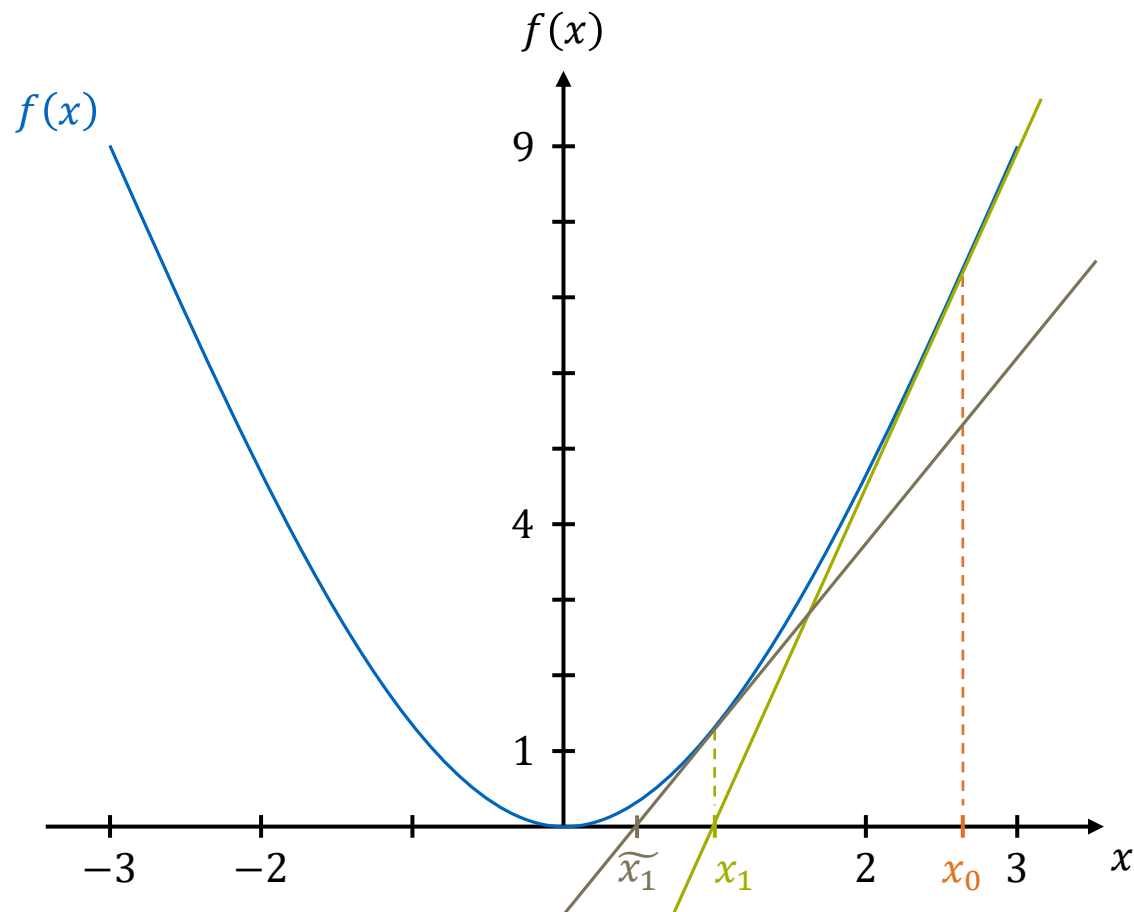
Newton-Verfahren



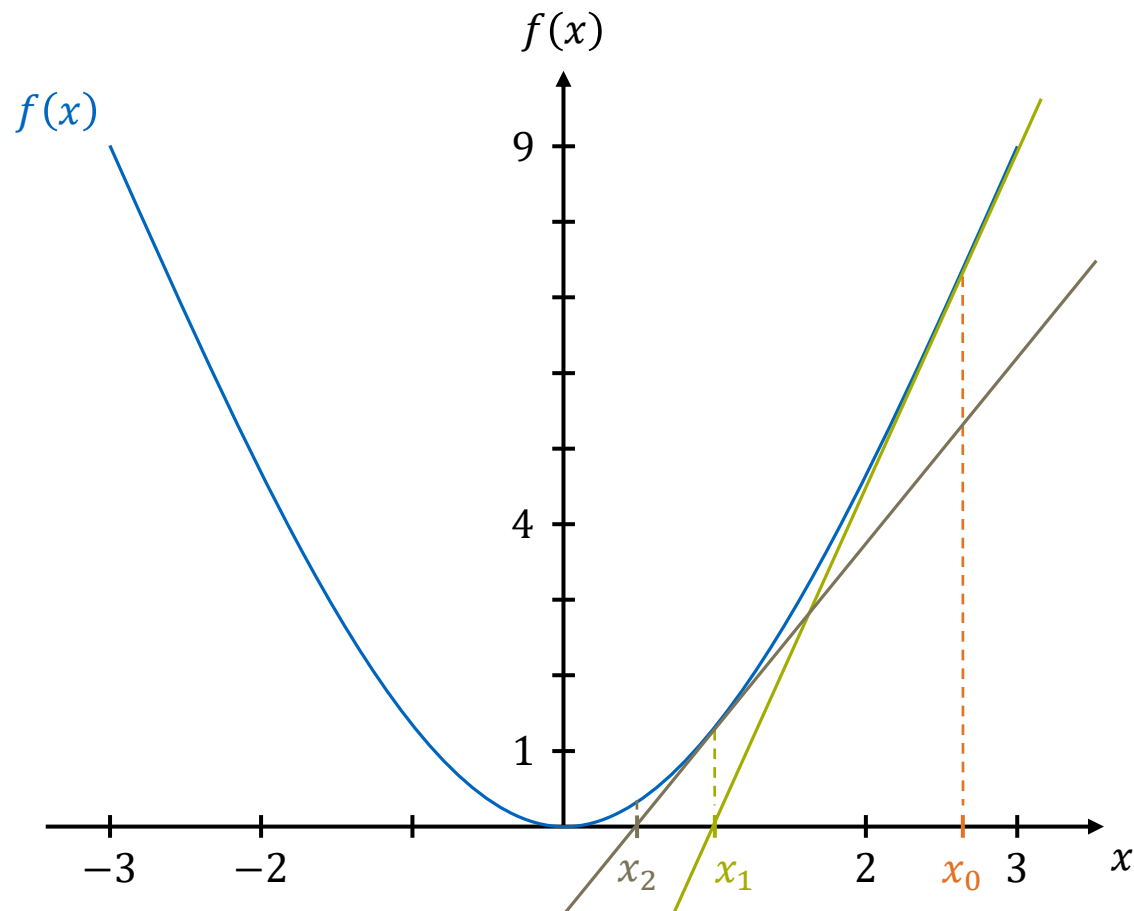
Newton-Verfahren



Newton-Verfahren



Newton-Verfahren



Newton-Verfahren

Mithilfe des Newton-Verfahrens können wir **Nullstellen** von $f(x)$ iterativ **approximieren**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Achtung:

Newton-Verfahren hat nur lokale Konvergenz!

→ Startpunkt „muss bereits in der Nähe der Nullstelle liegen“

Newton-Verfahren

Mithilfe des Newton-Verfahrens können wir **Nullstellen** von $f(x)$ iterativ **approximieren**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Was hat das jetzt mit Fixpunkten zu tun?

Newton-Verfahren

Mithilfe des Newton-Verfahrens können wir **Nullstellen** von $f(x)$ iterativ **approximieren**:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Was hat das jetzt mit Fixpunkten zu tun?

Wenn man eine Nullstelle x^* in das Newton-Verfahren eingibt, kommt auch x^* wieder heraus.

→ Nullstellen sind Fixpunkte der Iterationsvorschrift $\Phi(x)$ des Newton-Verfahrens