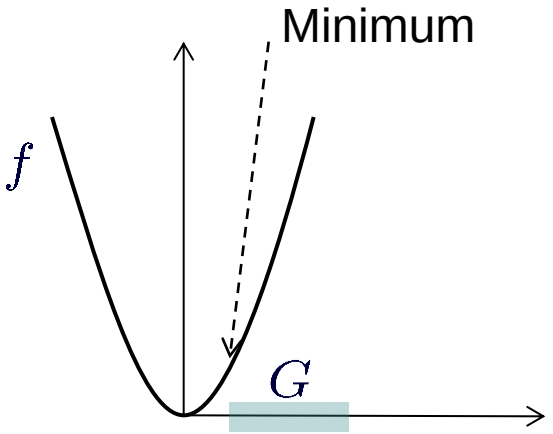
Optimierung

Vorlesung 2

Gradientenverfahren

Zielfunktion und zulässige Menge

- Ein Optimierungsproblem besteht aus einer **zulässigen Menge** G und einer **Zielfunktion** $f : G \to \mathbb{R}$



- Beispiel:

  $f(x) = x^2$

  $G = \{x \in \mathbb{R} | 2 \leq x \leq 5\}$

- **Minimum**: $\min_x f(x) = 4$

- **Minimierer**: $\operatorname{argmin}_x f(x) = 2$

- Die nächsten Vorlesungen: $G = \mathbb{R}^n$
  - Kontinuierliche Variablen (beliebiger Dimensionalität)
  - Keine Nebenbedingungen

Differenzierbare Funktionen

- Nachfolgend gehen wir davon aus, dass die Funktion f mindestens einmal differenzierbar ist: $f \in \mathcal{C}^1$

- Gradient von f :

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n} \right)^\top$$

- **Notwendige Bedingung** für ein Minimum von f :

$$\nabla f(\boldsymbol{x}) = \mathbf{o}$$

- Ist f konvex, so ist dies auch eine **hinreichende Bedingung**.

- Allgemein stellt die Bedingung $\nabla f(\boldsymbol{x}) = \mathbf{o}$ ein **nichtlineares Gleichungssystem** dar, das numerisch gelöst werden muss.

Gradientenverfahren

- Iteratives Verfahren

$$\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \tau^k \boldsymbol{d}^k$$

mit einem Startpunkt $\boldsymbol{x}^0$, einer Änderungsrichtung $\boldsymbol{d}^k$ und einer Schrittweite τ^k (Bem: Vektoren werden fett gedruckt)

- Beim Gradientenverfahren entspricht die Änderungsrichtung dem negativen Gradienten der Zielfunktion f an der aktuellen Stelle $\boldsymbol{x}^k$

$$\boldsymbol{d}^k := -\nabla f(\boldsymbol{x}^k)$$

Optimalität ergibt sich aus der Taylor-Entwicklung.

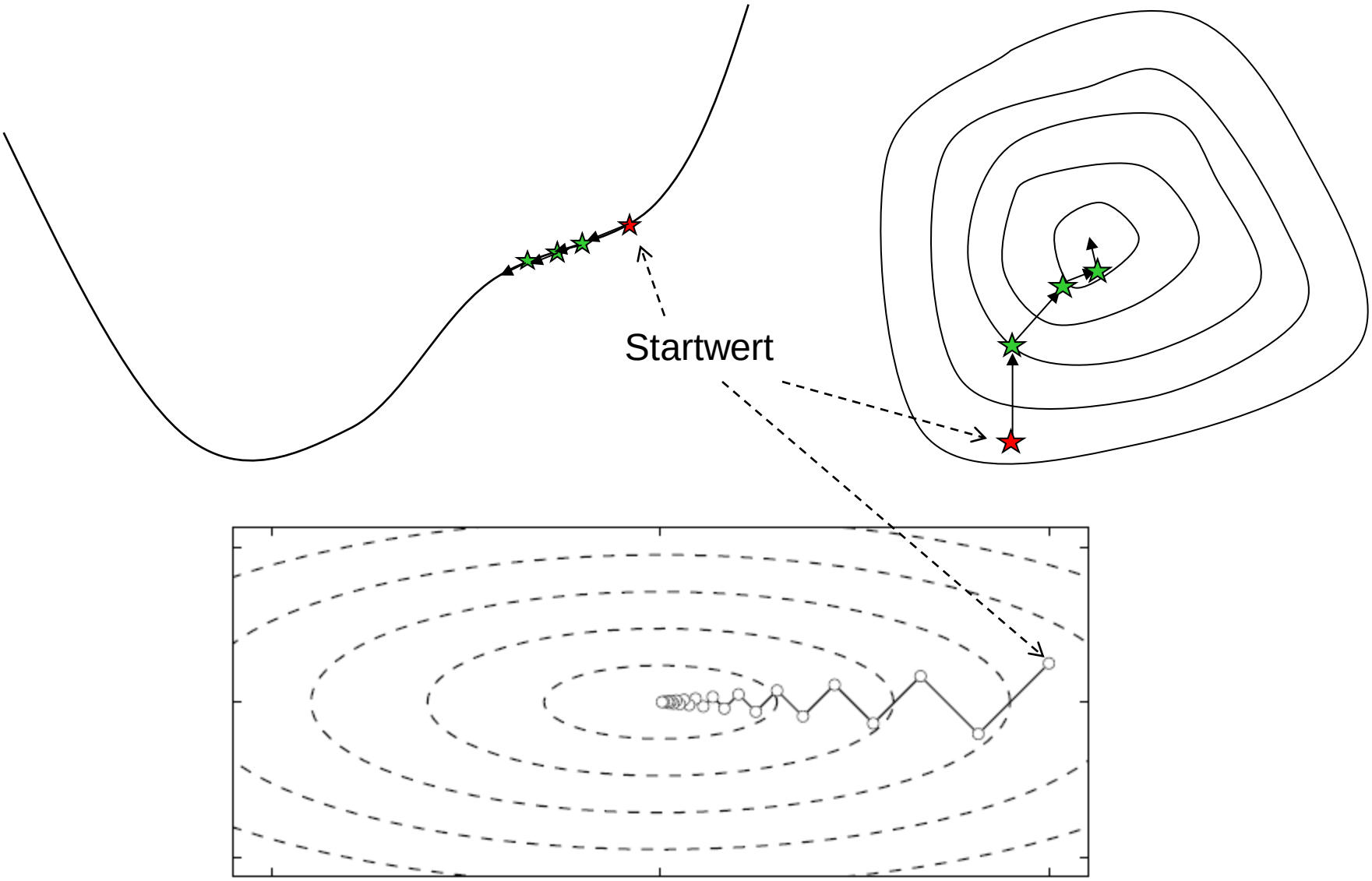
- Die Schrittweite τ^k wird optimal bestimmt, so dass

$$f(\boldsymbol{x}^k + \tau^k \boldsymbol{d}^k) \leq f(\boldsymbol{x}^k + \alpha \boldsymbol{d}^k) \quad \forall \alpha \geq 0$$

Mehr zur Schrittweitenbestimmung später…

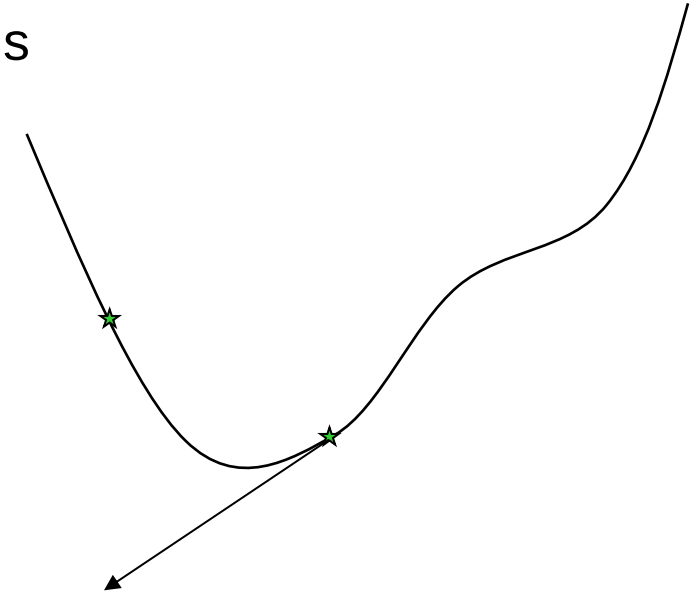
Taylor-Entwicklung

Gradientenverfahren



Schrittweitenbestimmung

- Die Wahl der Schrittweite ist entscheidend für **Konvergenz** und Effizienz des Verfahrens



- Bei zu großen Schrittweiten schießt das Verfahren über das Ziel hinaus (u.U. keine **Konvergenz**)

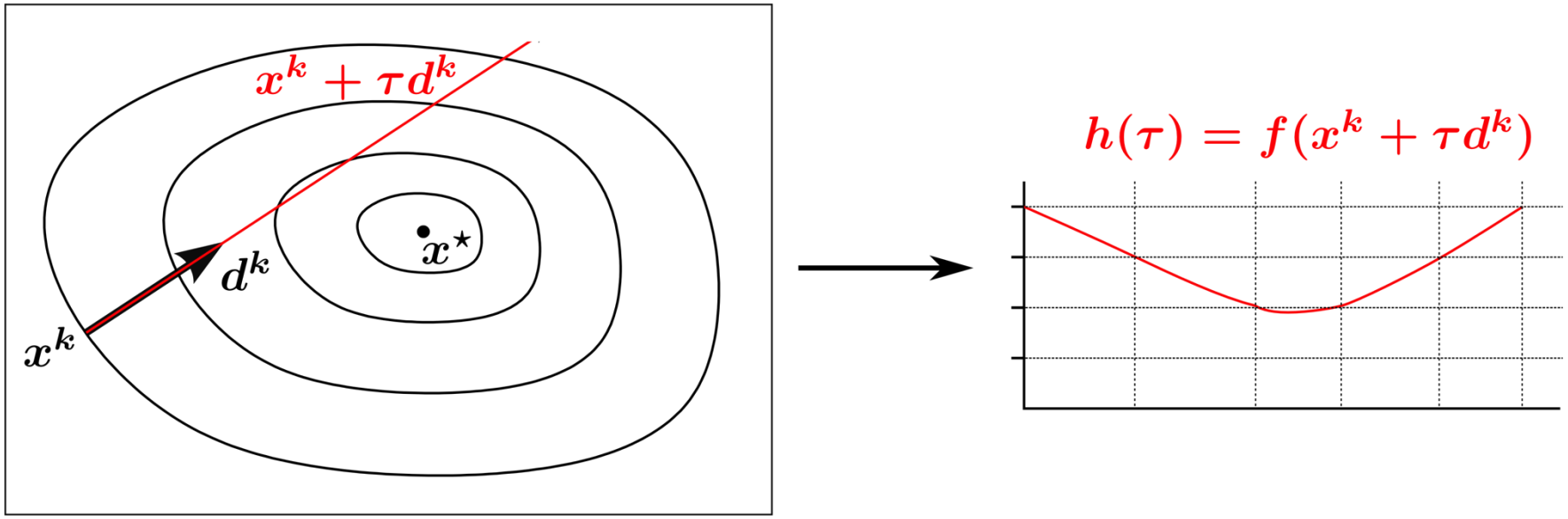
- Bei kleinen Schrittweiten benötigt das Verfahren viele Iterationen zur Lösung
  → hoher Rechenaufwand

- Richtung bereits durch d^k vorgegeben
  → eindimensionales Optimierungsproblem über τ^k
  → **Line search**

- In einigen Fällen ist eine feste Schrittweite τ, die Konvergenz garantiert, effizienter als die Bestimmung einer optimalen Schrittweite.

Richtungsableitung

- Zur Schrittweitenbestimmung: Situation im Schritt k.



- Im folgenden: Betrachte Richtungsableitung im Punkt x^k in Richtung d^k:

$$\lim_{\epsilon \to 0} \frac{f(\boldsymbol{x}^k + \epsilon \boldsymbol{d}^k) - f(\boldsymbol{x}^k)}{\epsilon} = \nabla f(\boldsymbol{x}^k)^T \boldsymbol{d}^k$$

- Im folgenden: Zusammenhang Richtungsableitung und $h(\tau)$

Line search – eindimensionale Optimierung

- Optimierungsaufgabe mit einer eindimensionalen Funktion $h : \mathbb{R}_+ \to \mathbb{R}$

  $\tau^* = \operatorname{argmin}_\tau h(\tau)$

- Falls h differenzierbar ist, ergibt sich die notwendige Bedingung

  $h'(\tau) = 0$

- Normalerweise nichtlinear und dann nur mit großem Aufwand exakt lösbar

- Exakte Lösung oft nicht von Vorteil, da aktuelle Abstiegsrichtung auch bei optimaler Schrittweite ohnehin nicht direkt zum Ziel führt

  → Approximative Schrittweitensuche unter Einhaltung gewisser Qualitätsbedingungen

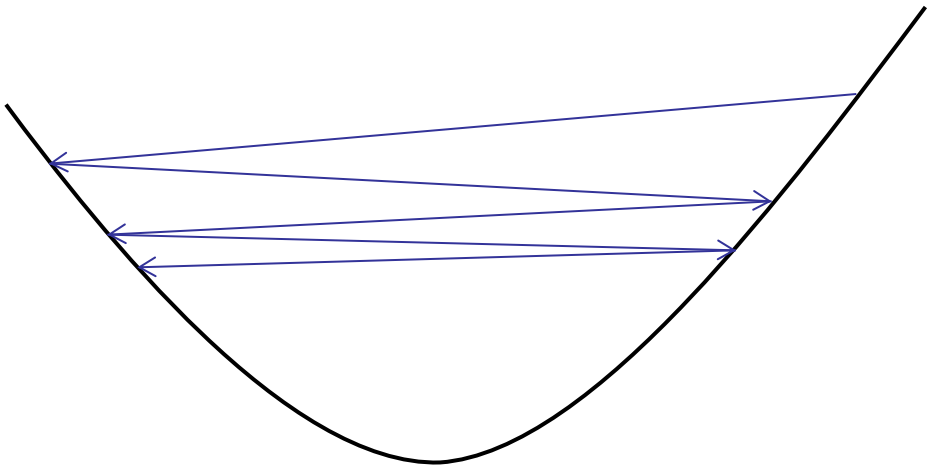
  Diese Bedingungen garantieren Konvergenz und Effizienz der verschiedenen Verfahren

Hinreichender Abstieg

- Einfache Reduktion des Funktionswerts

  $f(\boldsymbol{x}^k + \tau \boldsymbol{d}^k) - f(\boldsymbol{x}^k) < 0$

  ist nicht ausreichend.



- Obwohl jede Iteration den Funktionswert reduziert, wird die Reduktion möglicherweise immer kleiner und erreicht das Optimum nie

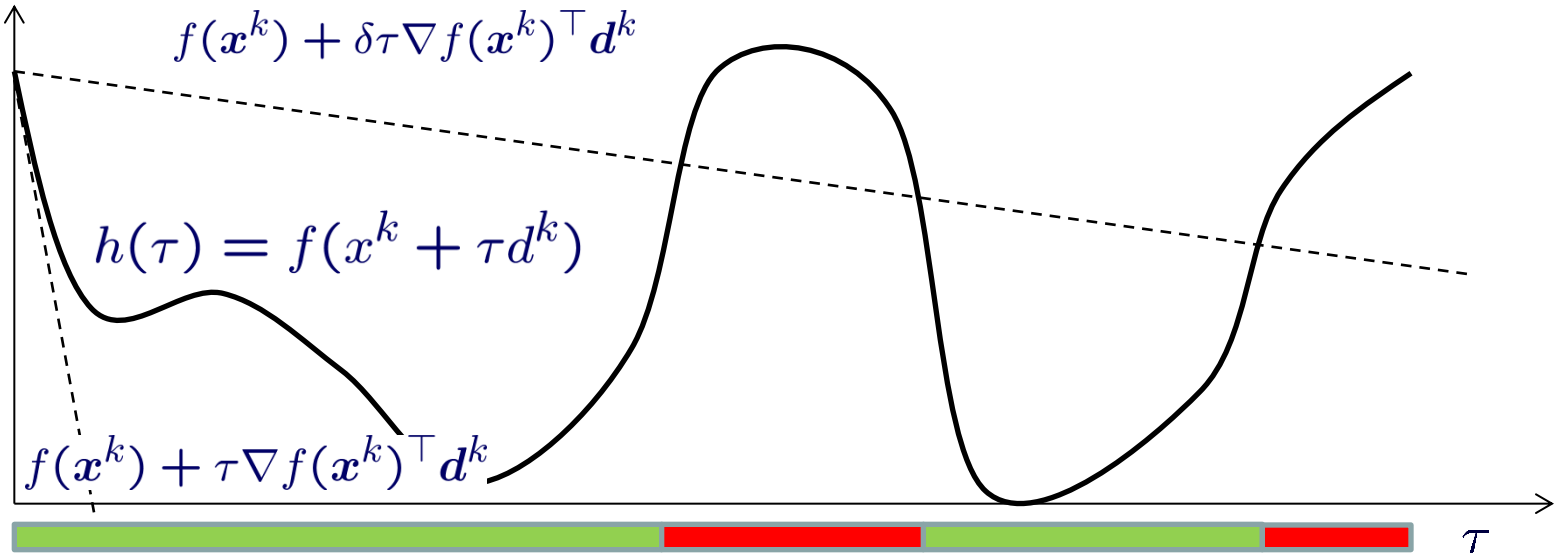
Wolfe Bedingungen

1. Hinreichender Abstieg (Armijo-Bedingung):

$$f(\boldsymbol{x}^k+\tau\boldsymbol{d}^k) \leq f(\boldsymbol{x}^k)+\delta\tau\nabla f(\boldsymbol{x}^k)^\top \boldsymbol{d}^k \qquad \delta \in (0,1)$$

Stellt sicher, dass die Zielfunktion f durch die Schrittweite τ in Abhängigkeit der Steilheit des Gradienten “hinreichend” reduziert wird. Üblicherweise $\delta = 10^{-4}$

Je steiler der Gradient und je größer der Schritt umso mehr muss sich der Funktionswert reduzieren



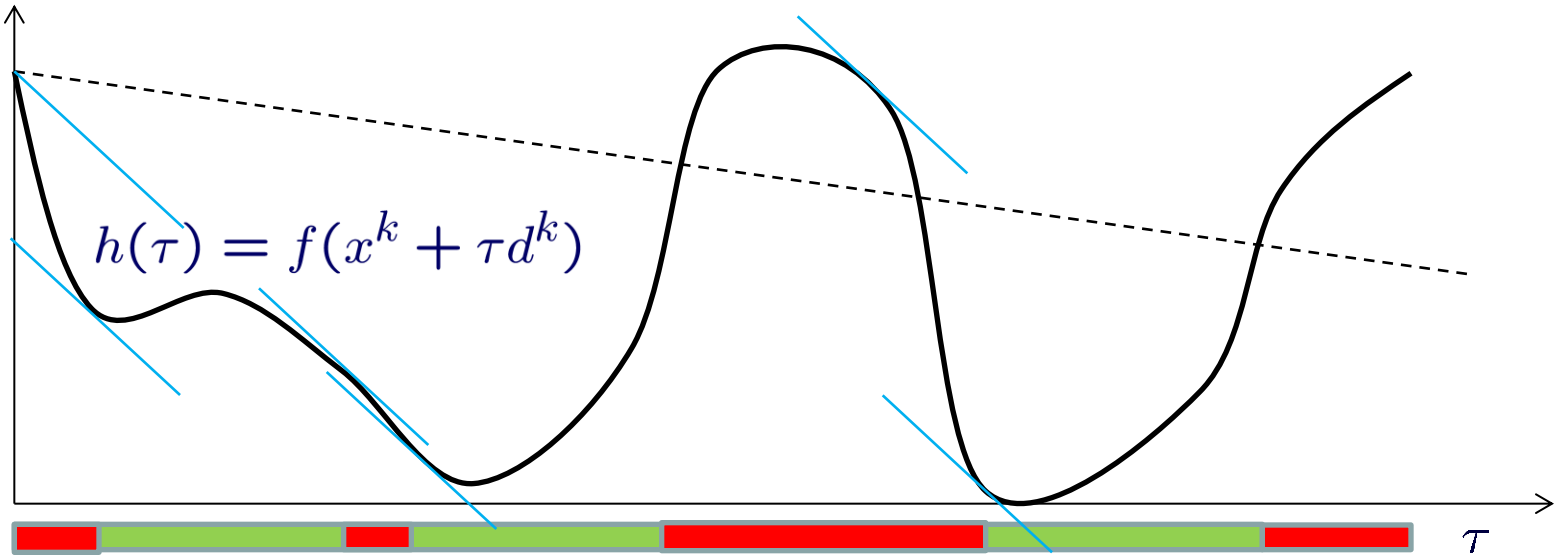
Wolfe Bedingungen

2. Krümmungsbedingung (schwache Wolfe Bedingung)

$$\nabla f(\boldsymbol{x}^k + \tau \boldsymbol{d}^k)^\top \boldsymbol{d}^k \geq \eta \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{d}^k \qquad \eta \in (\delta, 1)$$

Gradient an der Zielposition ist weniger steil als an der Startposition oder sogar positiv.

Verhindert ineffizient kleine Schritte in Bereiche hinein, in denen die aktuelle Abstiegsrichtung immer noch sehr gut ist.

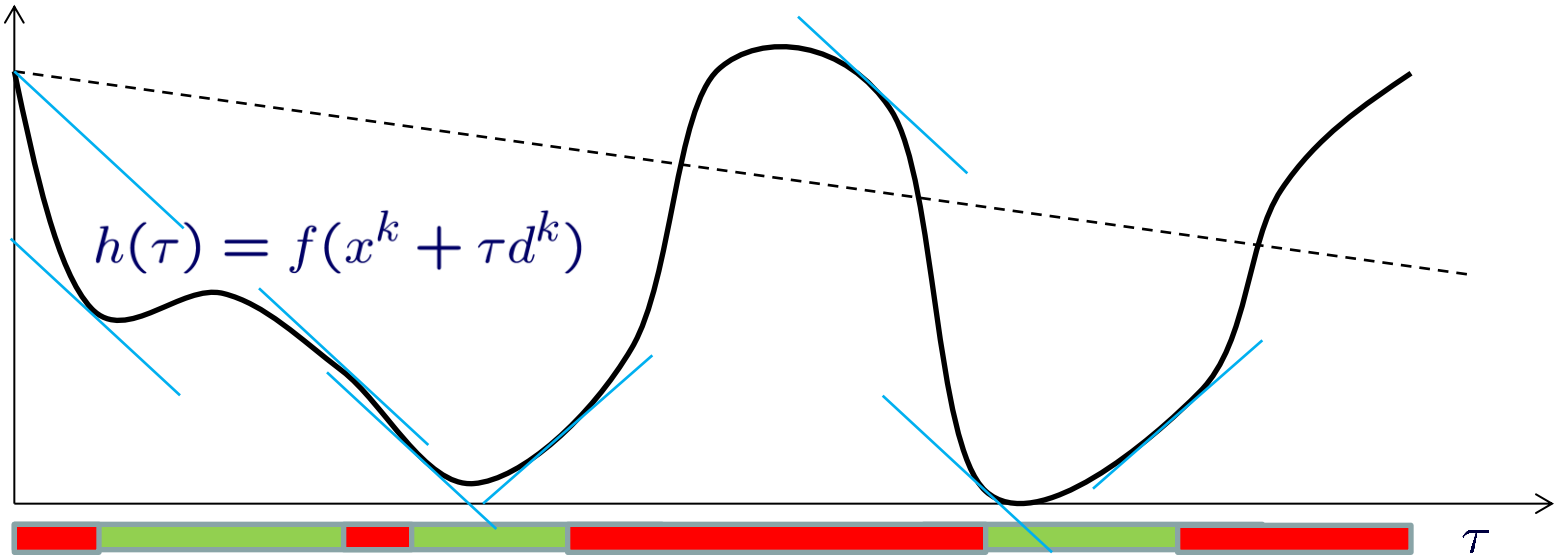


Wolfe Bedingungen

2. Krümmungsbedingung (starke Wolfe Bedingung)

$$|\nabla f(\boldsymbol{x}^k + \tau \boldsymbol{d}^k)^\top \boldsymbol{d}^k| \leq \eta |\nabla f(\boldsymbol{x}^k)^\top \boldsymbol{d}^k| \qquad \eta \in (\delta, 1)$$

Mit dieser strikteren Bedingung ist die Zielposition meist in der Nähe eines lokalen Optimums von $h(\tau)$



Existenz geeigneter Schrittweiten

Annahmen:

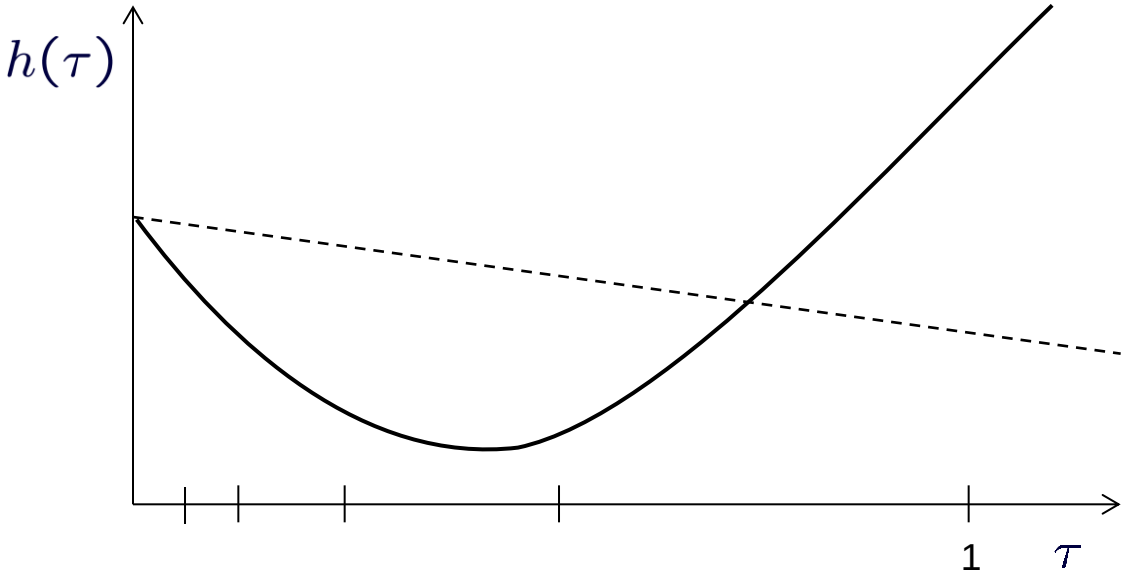
- $f(\boldsymbol{x})$ ist kontinuierlich differenzierbar
- $\boldsymbol{d}^k$ ist eine Abstiegsrichtung an der Stelle $\boldsymbol{x}^k$
- $f(\boldsymbol{x})$ ist entlang des Strahls $[\boldsymbol{x}^k + \tau \boldsymbol{d}^k | \tau > 0]$ von unten beschränkt
- $0 < \delta < \eta < 1$

Sind diese Annahmen erfüllt, existieren Intervalle von Schrittweiten, welche die starken Wolfe Bedingung erfüllen.

(Beweis in Nocedal-Wright pp. 35)

Einfaches Line Search Verfahren: Backtracking line search

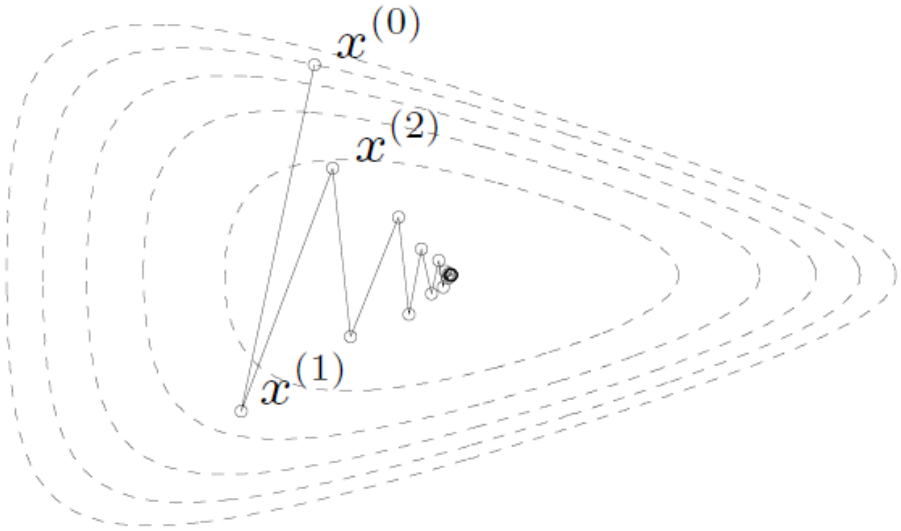
- Ziel: Approximative Minimierung von $h(\tau) = f(\boldsymbol{x}^k + \tau \boldsymbol{d}^k), \quad \tau \in (0, 1]$



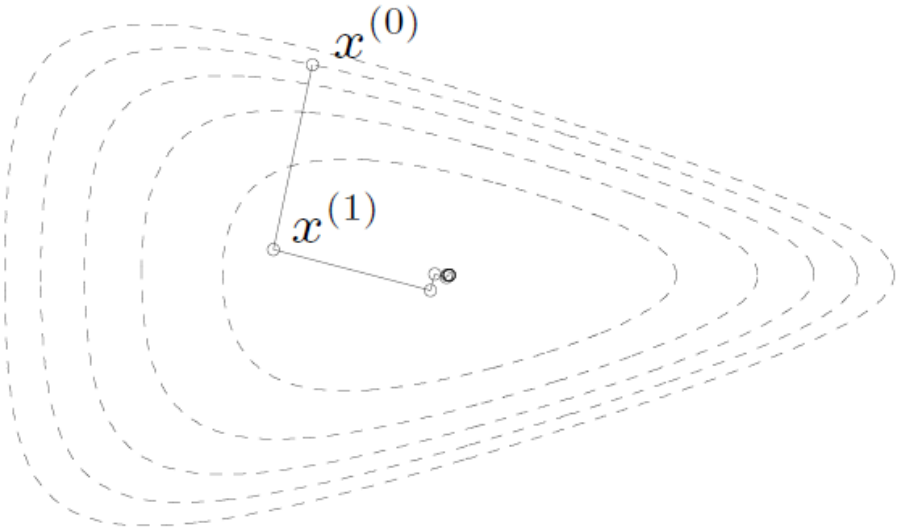
- Starte mit $\tau^0 = 1$ und reduziere $\tau^{k+1} = \beta \tau^k, \ \beta \in (0, 1)$ solange bis die Armijo-Bedingung erfüllt ist.

- Lässt den Gradienten ungenutzt

Relevanz der Schrittweite



Backtracking



Optimale Schrittweite

Interpolationsverfahren

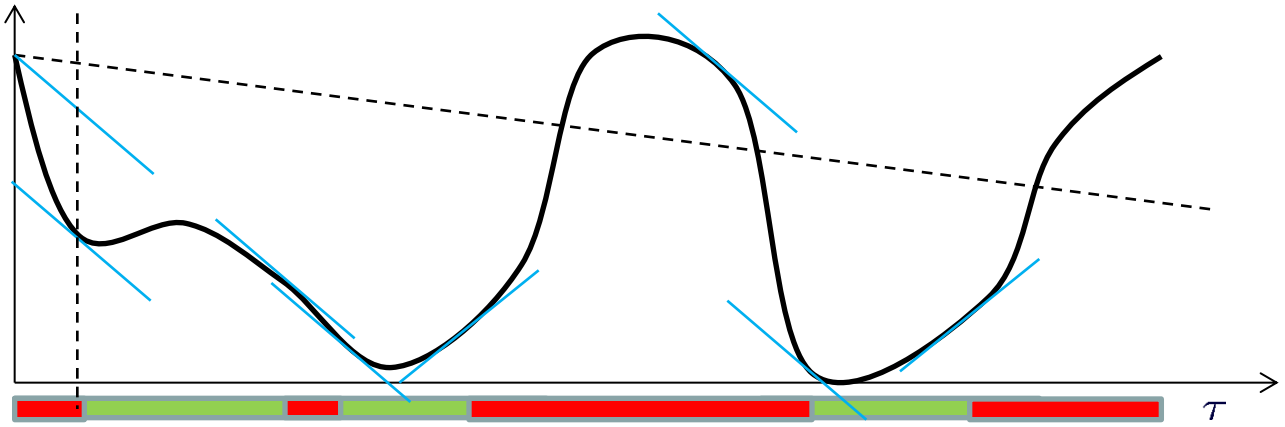
- Starte mit τ_0 (z.B. $\tau_0 = 1$)
- Erfüllt τ_0 die Armijo-Bedingung, sind wir fertig
- Andernfalls gibt es im Intervall $(0, \tau_0)$ eine Schrittweite, welche die Armijo-Bedingung erfüllt
- Quadratische Approximation von $h(\tau)$ mithilfe von $h(0), h'(0), h(\tau_0),$

$$h_q(\tau) = \left(\frac{h(\tau_0) - h(0) - \tau_0 h'(0)}{\tau_0^2} \right) \tau^2 + h'(0)\tau + h(0)$$

- Minimum bei $\tau_1 = \dfrac{h'(0)\tau_0^2}{2(h(\tau_0) - h(0) - h'(0)\tau_0)}$
- Solange τ_i die Armijo-Bedingung noch nicht erfüllt, kubische Interpolation mit $h(0), h'(0), h(\tau_0), h(\tau_1),$

Line Search Verfahren für die Wolfe Bedingungen

- Bisher nur die Armijo-Bedingung sichergestellt
- Wolfe-Bedingungen aufwendiger zu erfüllen
- Verfahren bestehen aus zwei Komponenten:
  - **Bracketing** erweitert das Suchintervall bis darin geeignete Schrittweiten garantiert werden können
  - **Zooming** reduziert das Suchintervall bis eine geeignete Schrittweite gefunden wird (basierend auf Interpolationsverfahren)



- Details in Nocedal-Wright pp. 60; Public Domain Softwarepakete

Konvergenz

- Das Gradientenabstiegsverfahren mit einer Schrittweite, welche die Armijo-Bedingung erfüllt, konvergiert zu einem lokalen Minimum.

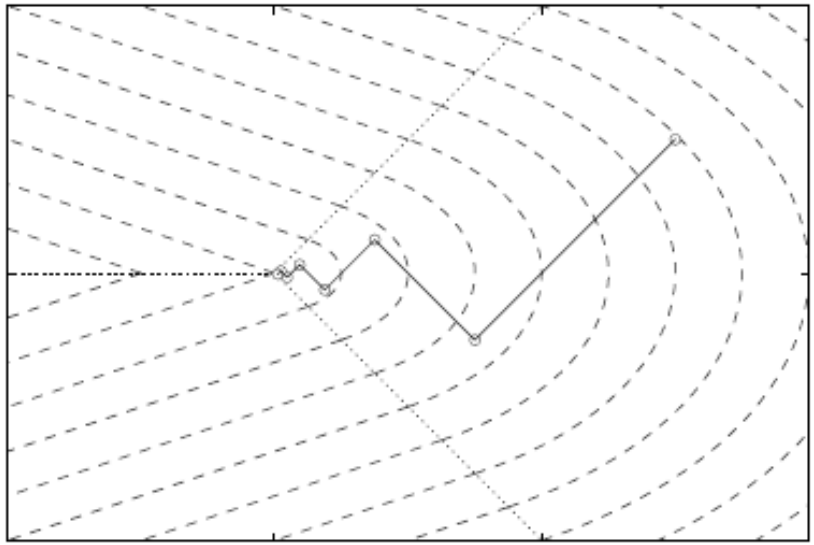
- Grund: Die Armijo-Bedingung

$$f(\boldsymbol{x}^k + \tau \boldsymbol{d}^k) - f(\boldsymbol{x}^k) \leq \delta \tau \nabla f(\boldsymbol{x}^k)^\top \boldsymbol{d}^k, \quad \delta \in (0, 1)$$

  stellt sicher, dass der Funktionswert in jeder Iteration hinreichend sinkt, solange die notwendige Bedingung nicht erfüllt ist, also $|\nabla f(\boldsymbol{x}^k)| > 0$

- Erinnerung: Wir gehen davon aus, dass f einmal stetig differenzierbar ist.

  Notwendig für die Bestimmung des Gradienten und für Konvergenz zu einem lokalen Minimum.



Nicht differenzierbares Beispiel
Fixpunkt ist kein lokales Minimum

Zickzack-Verhalten des Gradientenabstiegverfahrens

- Auf dem Weg zum Minimum werden immer wieder dieselben Richtungen gewählt
  → ineffizient

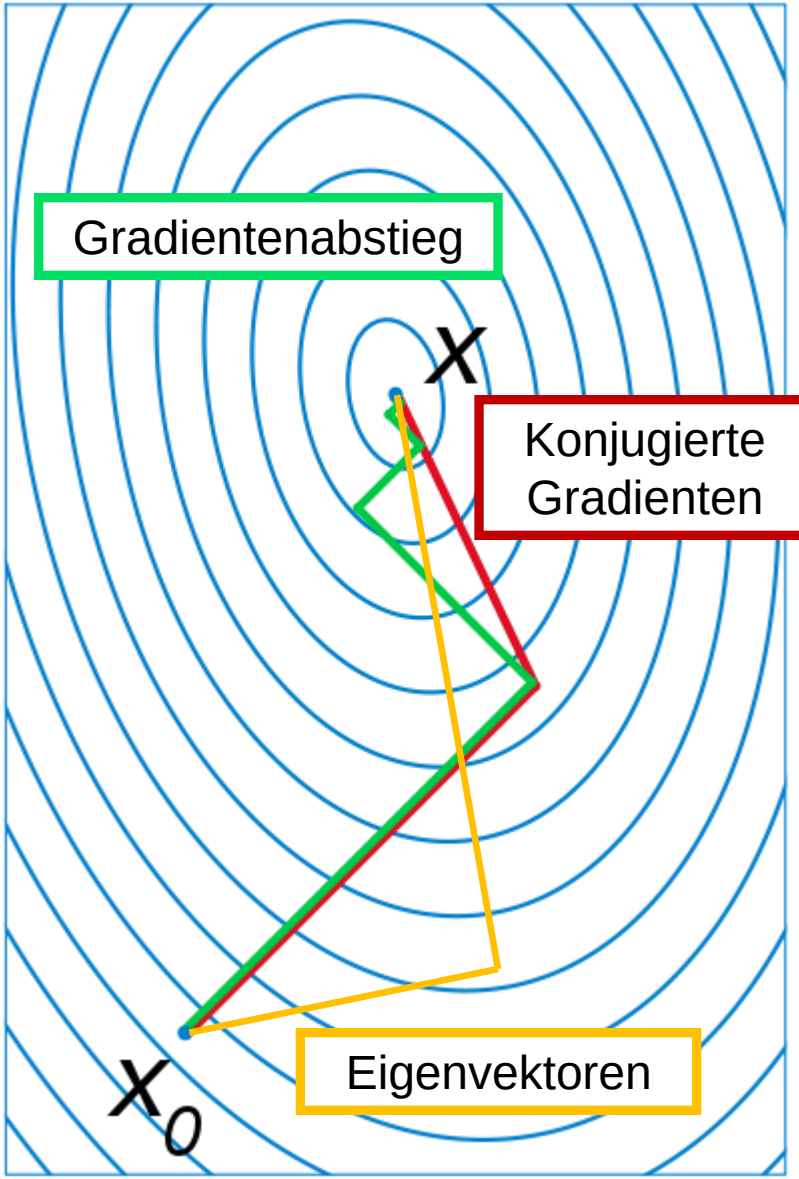
- Mit einer Reihe von orthogonalen Richtungen wären wir wesentlich schneller beim Minimum.

- Im Falle einer quadratischen Zielfunktion

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^\top A\boldsymbol{x} + \boldsymbol{b}^\top \boldsymbol{x}$$

  sogar in n Schritten

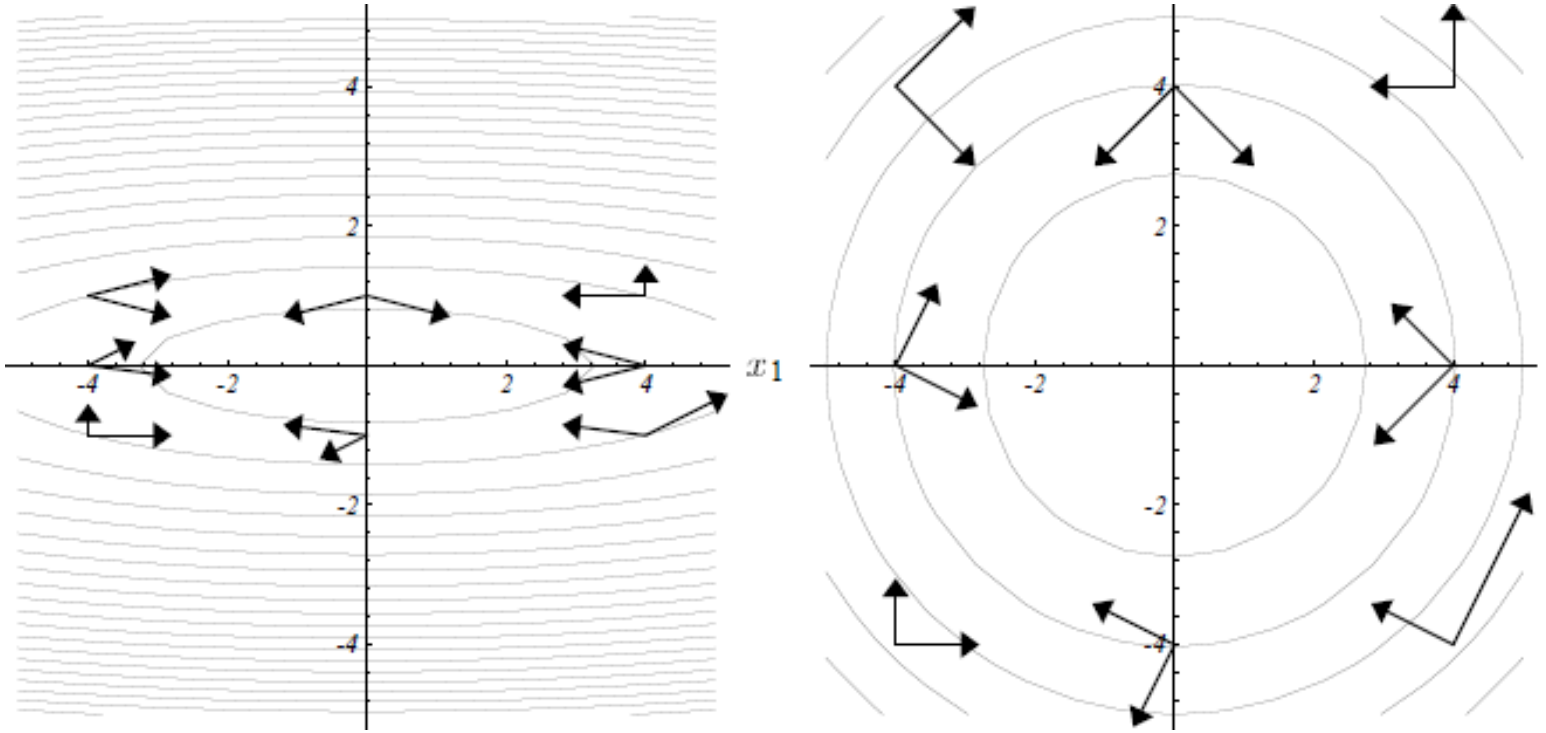
- Hierfür bräuchten wir aber die Eigenvektoren von A
  → auch ineffizient



CG-Verfahren

- Statt Orthogonalität im Euklidischen Raum, Orthogonalität mit der durch A definierten Metrik:

$$\boldsymbol{x}^\top A\boldsymbol{x} \neq 0, \quad \boldsymbol{x}^\top A\boldsymbol{y} = 0$$



A-orthogonale Vektoren für zwei unterschiedliche quadratische Zielfunktionen

Quelle: J. Shewchuk

CG Verfahren für quadratische Funktionen

- Wähle als erste Richtung den Gradienten an der Startposition x^0 :

$$d^0 = -\nabla f(x^0) = b - Ax^0$$

- Wähle die optimale Schrittweite:

$$\tau^k = \frac{|\nabla f(x^k)|^2}{d^{k\top} A d^k}$$

analytische Lösung im Fall quadratischer Funktionen, da dann $h'(\tau) = 0$ eine lineare Gleichung ist

- Neue Lösung:

$$x^{k+1} = x^k + \tau^k d^k$$

- Wähle neue Richtung

$$d^{k+1} = -\nabla f(x^{k+1}) + \beta^k d^k$$

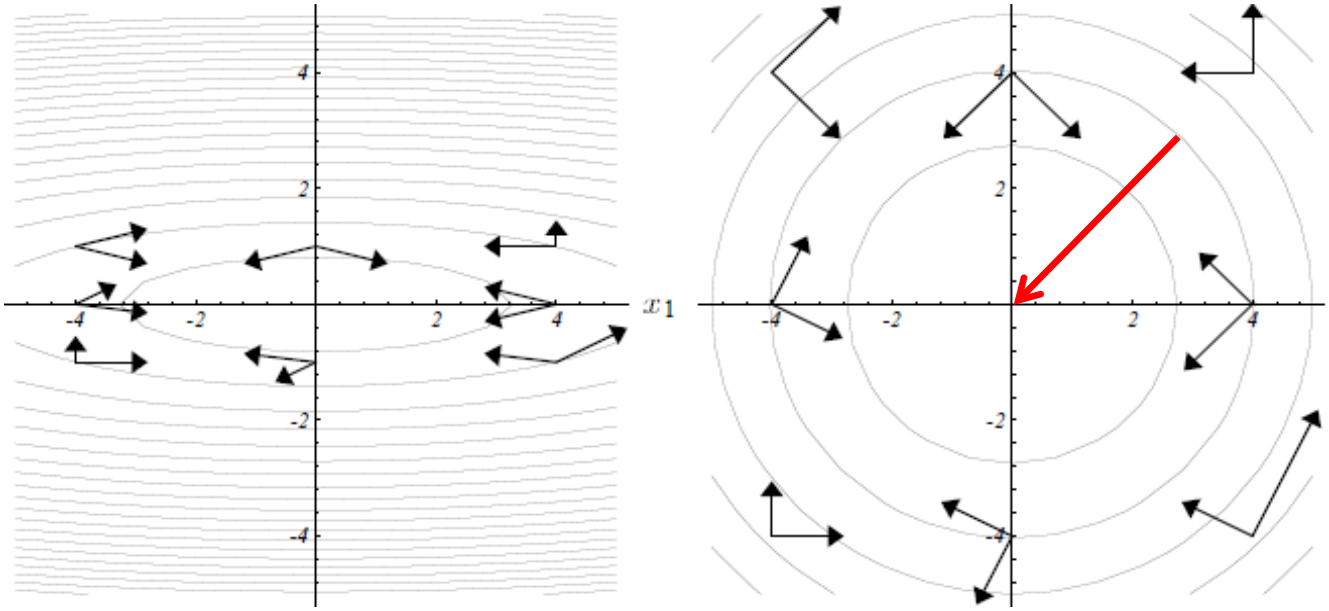
mit $\beta^k = \frac{|\nabla f(x^{k+1})|^2}{|\nabla f(x^k)|^2}$ (Lösung für die d^{k+1} und d^k orthogonal sind)

CG Verfahren zum Lösen linearer Gleichungssysteme

- Der Minimierer von $f(x) = \frac{1}{2}x^\top Ax - b^\top x$ entspricht der Lösung des linearen Gleichungssystems $Ax = b$
- Notwendige Bedingung für ein Minimum von $f(x) = \frac{1}{2}x^\top Ax - b^\top x$

  $\nabla f(x) = Ax - b = 0$
- Das CG Verfahren wird daher meist zum Lösen linearer Gleichungssysteme eingesetzt.
- Die Effizienz hängt von der **Konditionszahl** von $A \in \mathbb{R}^{n \times n}$ ab.
- Die Konditionszahl einer Matrix ist der größte Eigenwert geteilt durch den kleinsten.
- Bei exakter Zahlendarstellung garantiert das CG Verfahren in n Schritten die exakte Lösung, bei kleiner Konditionszahl bereits wesentlich schneller.

Konvergenz abhängig von Konditionszahl



Links: höhere Konditionszahl. CG Verfahren konvergiert nach 2 Schritten.
Rechts: Konditionszahl = 1. CG Verfahren konvergiert in einem Schritt.
Quelle: J. Shewchuk

- Das CG Verfahren wird daher oft mit einem **Präkonditionierer** kombiniert, der vorab die Konditionszahl der Matrix reduziert.

Verallgemeinerung für nicht-quadratische Funktionen

- Wähle als erste Richtung den Gradienten an der Startposition x^0 :

  $p^0 = -\nabla f(x^0)$

- Wähle eine Schrittweite τ^k
  - mithilfe von Line Search (keine analytische Lösung verfügbar)

- Neue Lösung:

  $x^{k+1} = x^k + \tau^k p^k$

- Wähle neue Richtung

  $p^{k+1} = -\nabla f(x^{k+1}) + \beta^k p^k$

  mit $\beta^k = \frac{|\nabla f(x^{k+1})|^2}{|\nabla f(x^k)|^2}$ (Fletcher/Reeves)

- Bessere Varianten (z.B. Polak/Ribiére) verfügbar

Zusammenfassung

- Gradientenabstieg ist ein einfaches Verfahren zur lokalen Minimierung allgemeiner differenzierbarer Funktionen.
- Die Schrittweite muss bestimmte Bedingungen erfüllen um Konvergenz und Effizienz zu garantieren (Wolfe Bedingungen)
- Eine effizientere Variante ist das CG Verfahren. Insbesondere quadratische Funktionen lassen sich damit effizient minimieren.

Übungsaufgabe

1. Machen Sie sich mit Python vertraut.

2. Minimieren Sie die Funktion

$$f(x) = \frac{1}{2}x^4 + 2x^3 - 3x - 4$$

   – Visualisieren Sie die Funktion. Ist sie konvex?
   – Berechnen Sie den Gradienten.
   – Optimieren Sie die Funktion mit Gradientenabstieg. Verwenden Sie bitte Ihre eigene Implementierung. Probieren Sie verschiedene Startpunkte aus. Implementieren Sie Backtracking Line Search zur Bestimmung der Schrittweite.
   – Visualisieren Sie, was während der Optimierung geschieht. Für diese 1D-Funktion geht diese sehr schön, bei hochdimensionalen Problemen aus der Praxis nicht mehr.

Übungsaufgabe

3. Minimieren Sie die zweidimensionale Funktion

$$f(x_1, x_2) = 2x_1^2 + 2x_2^2 + 3x_1x_2 - x_1 + x_2$$

   - Visualisieren Sie die Isolinien. Ist die Funktion konvex?
   - Berechnen Sie den Gradienten.
   - Optimieren Sie die Funktion mit Gradientenabstieg. Verwenden Sie bitte wieder Ihre eigene Implementierung.
   - Optimieren Sie die Funktion analytisch. Stimmt Ihre numerische Lösung mit dem Minimum überein?