6 Freie Nichtlineare Optimierung

Verfahren zur Minimierung glatter Funktionen ohne Nebenbedingungen,

$$\min_{x \in \mathbb{R}^n} f(x), \quad f : \mathbb{R}^n \to \mathbb{R} \quad \text{hinreichend glatt}$$

Hinreichend glatt bedeutet, dass f so oft stetig differenzierbar sein soll, wie für das jeweilige Verfahren erforderlich.

Ziele für die Verfahren:
- Finde ein lokales Minimum (sogar weniger: finde ein $\bar{x}$, das die notwendigen Opt.-Bed. 1. Ordnung erfüllt, s. dort)
- Schnelle Konvergenz in der Nähe lokaler Optima
- Der Rechenaufwand soll möglichst klein bleiben
- Numerische Stabilität und hohe Genauigkeit

Anwendungen:
- Nichltineare kleinste Quadrate Probleme
- Als Löser für Optimierungsprobleme mit Nebenbedingungen [s. Barriere-, Straf- und augmentierte Lagrange-Verfahren]

In welcher Form soll f für die Verfahren zugänglich sein?

Inhalt

6.1 Orakel allgemein und Orakel 0. Ordnung

In vielen Anwendungen ist die Funktion f nicht analytisch verfügbar, sondern ergibt sich z.B. aus der Lösung eines Systems partieller Differentialgleichungen oder einer Simulation.

Daher setzen allgemeine Optimierungsverfahren nur eine Unterroutine voraus, die das Verfahren nach dem Wert der Funktion und eventuell auch nach Ableitungsinformation in dem jeweils betrachteten Punkt befragen kann $\rightarrow$ „Orakel“.

Ist die Funktion analytisch gegeben, erzeugen Modellierungssprachen wie AMPL, GAMS, ... automatisch entsprechende Orakel/Unterroutinen, die Wert und Ableitungsinformation liefern.

6.1 Orakel allgemein und Orakel 0. Ordnung

In vielen Anwendungen ist die Funktion f nicht analytisch verfügbar, sondern ergibt sich z.B. aus der Lösung eines Systems partieller Differentialgleichungen oder einer Simulation.

Daher setzen allgemeine Optimierungsverfahren nur eine Unterroutine voraus, die das Verfahren nach dem Wert der Funktion und eventuell auch nach Ableitungsinformation in dem jeweils betrachteten Punkt befragen kann $\rightarrow$ „Orakel“.

Ist die Funktion analytisch gegeben, erzeugen Modellierungssprachen wie AMPL, GAMS, ... automatisch entsprechende Orakel/Unterroutinen, die Wert und Ableitungsinformation liefern.

Ein **Orakel 0. Ordnung** berechnet für gegebenes $x \in \mathbb{R}^n$ nur den Funktionswert $f(x)$, aber keine Ableitungsinformation.
Verfahren für glatte Funktionen benötigen Ableitungsinformation und approximieren diese numerisch durch vielfache Funktionsaufrufe (s. später).

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

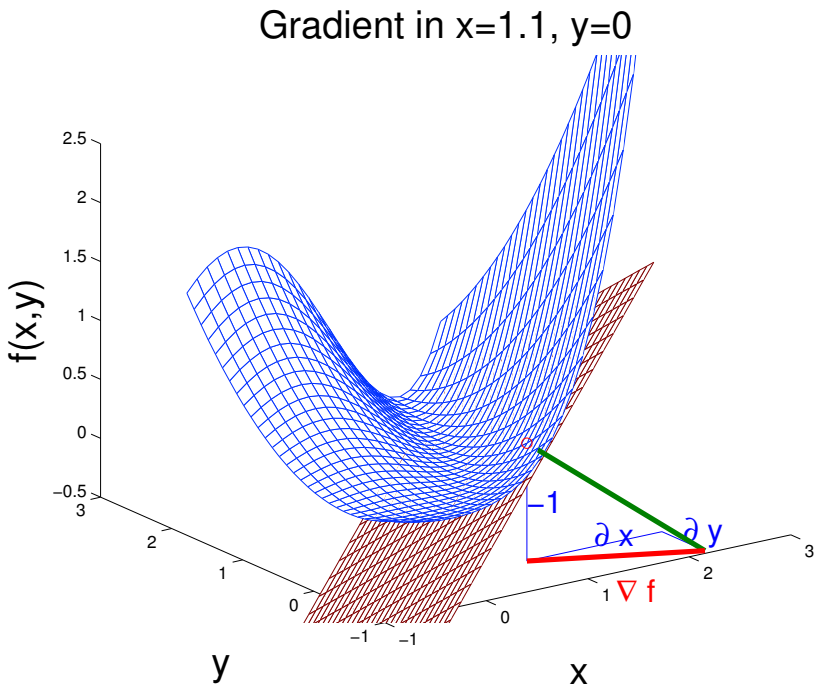
Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



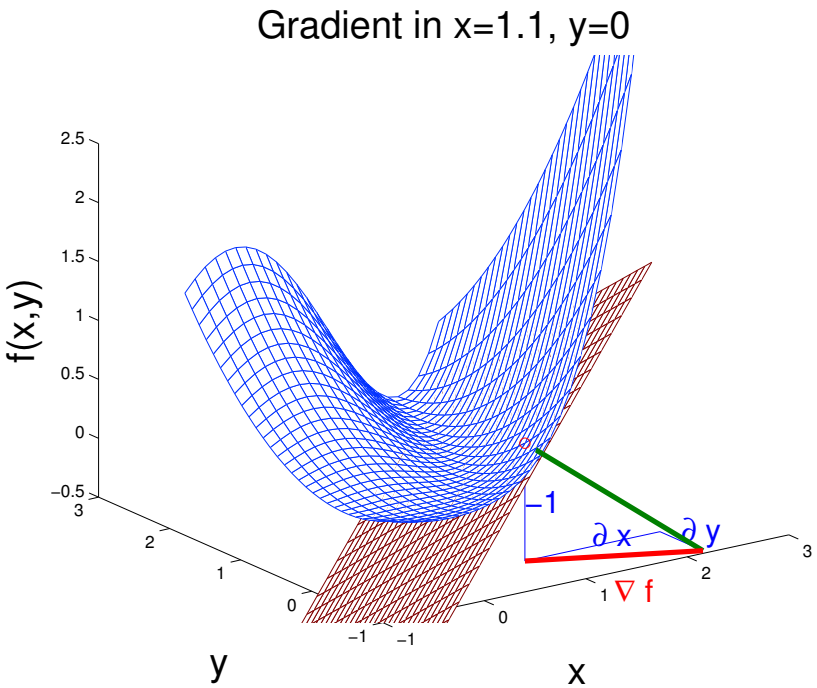
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



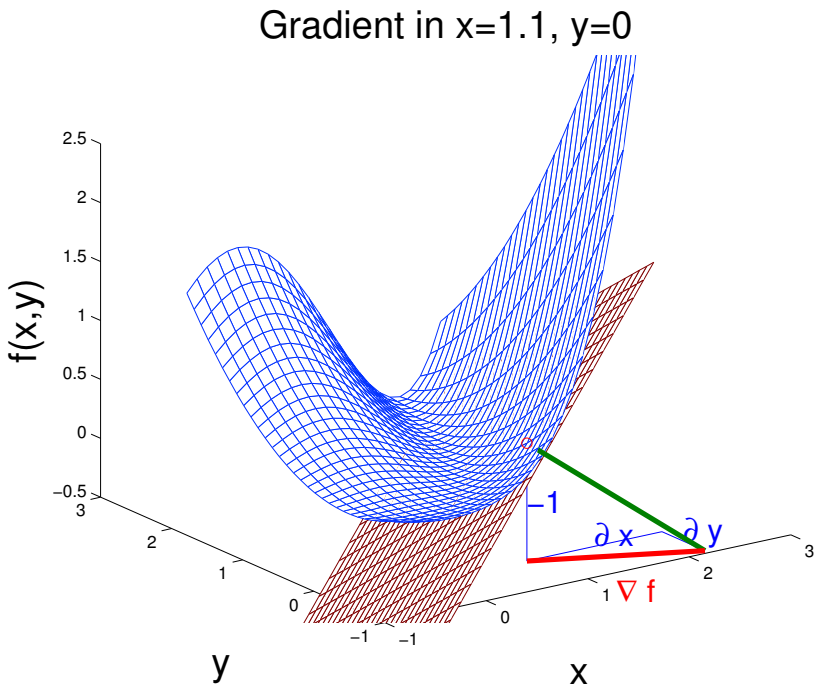
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$

- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.

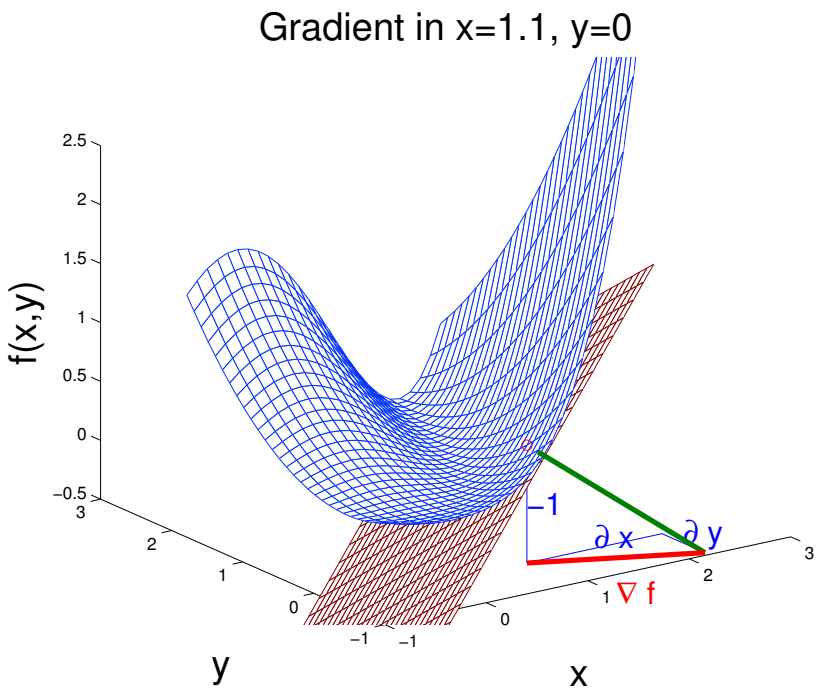


Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



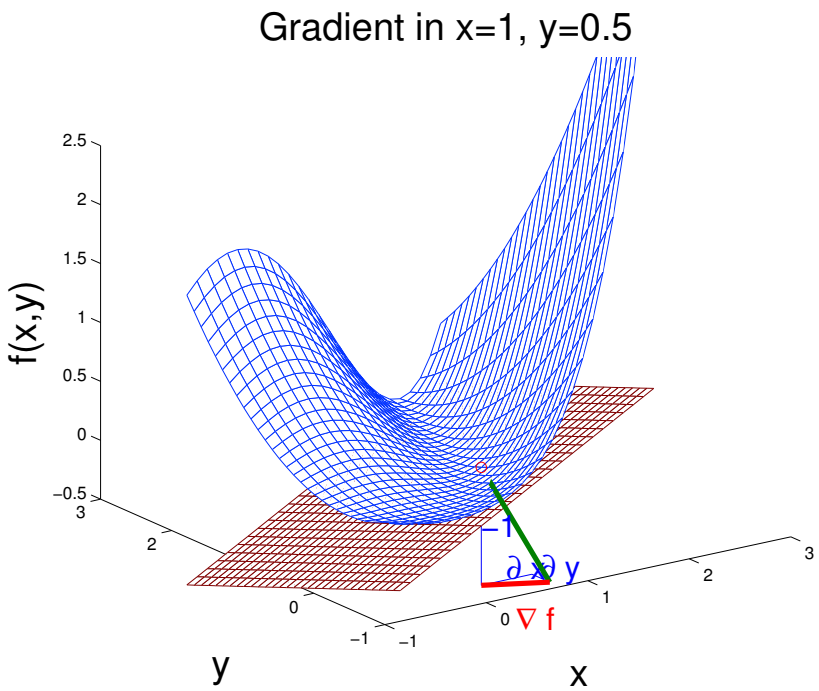
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



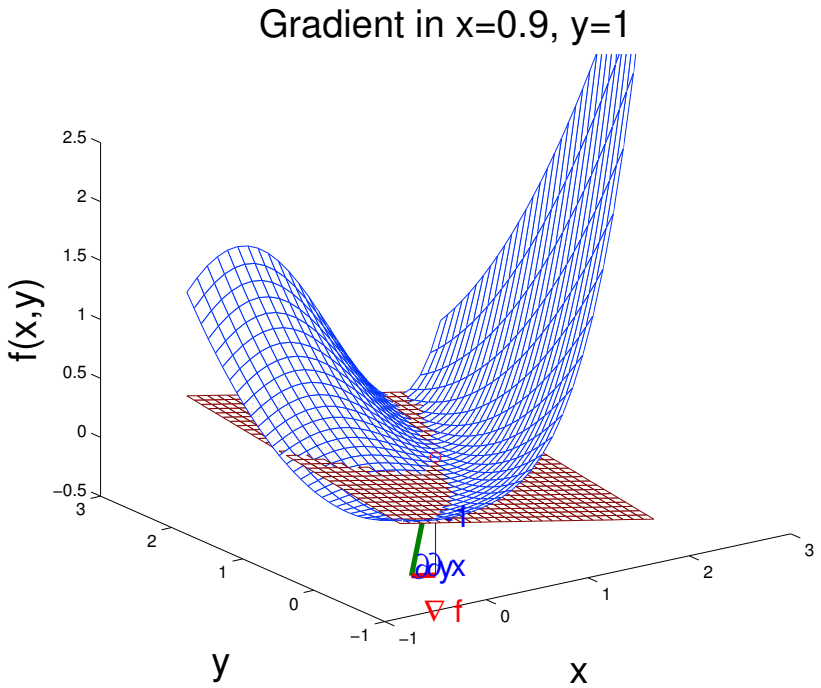
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



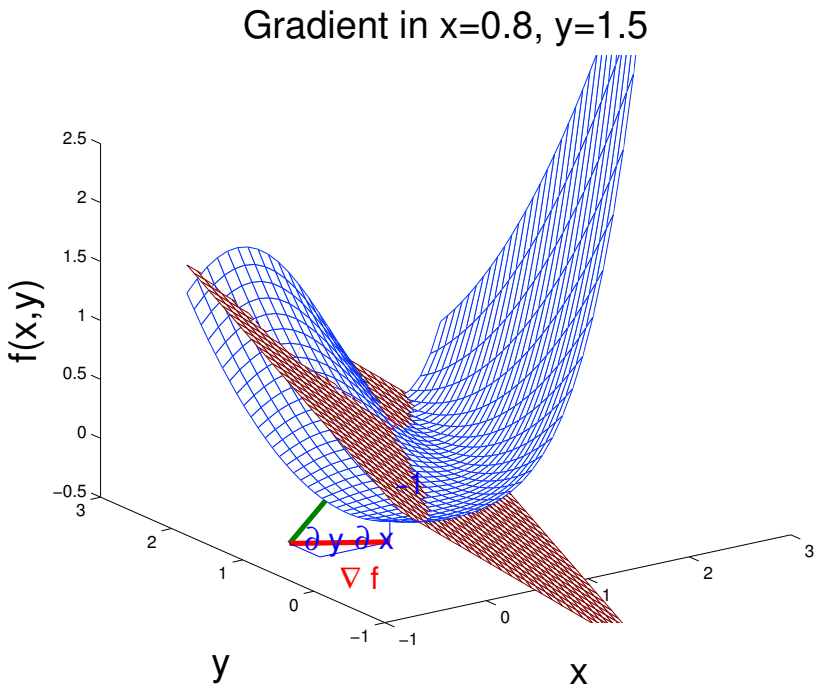
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



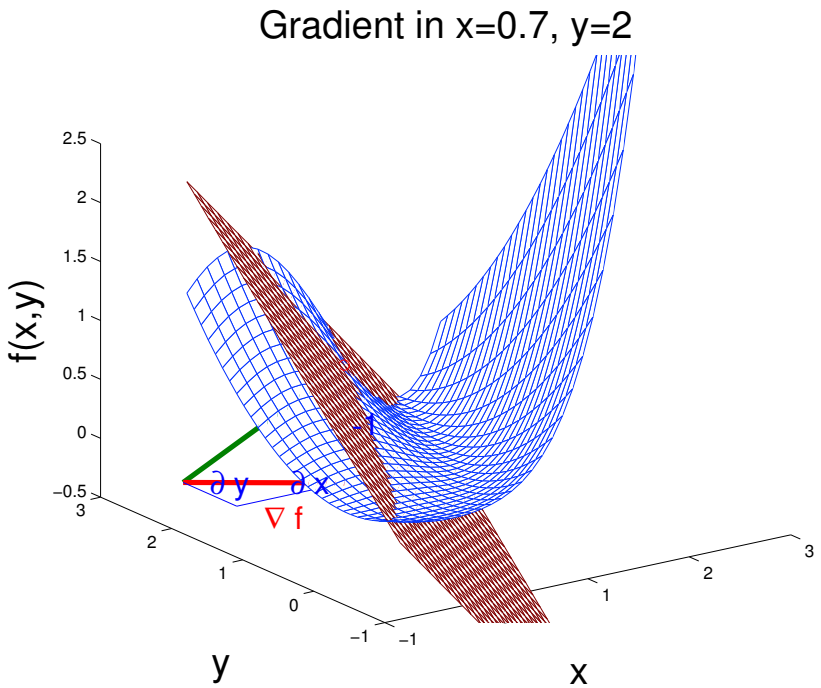
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



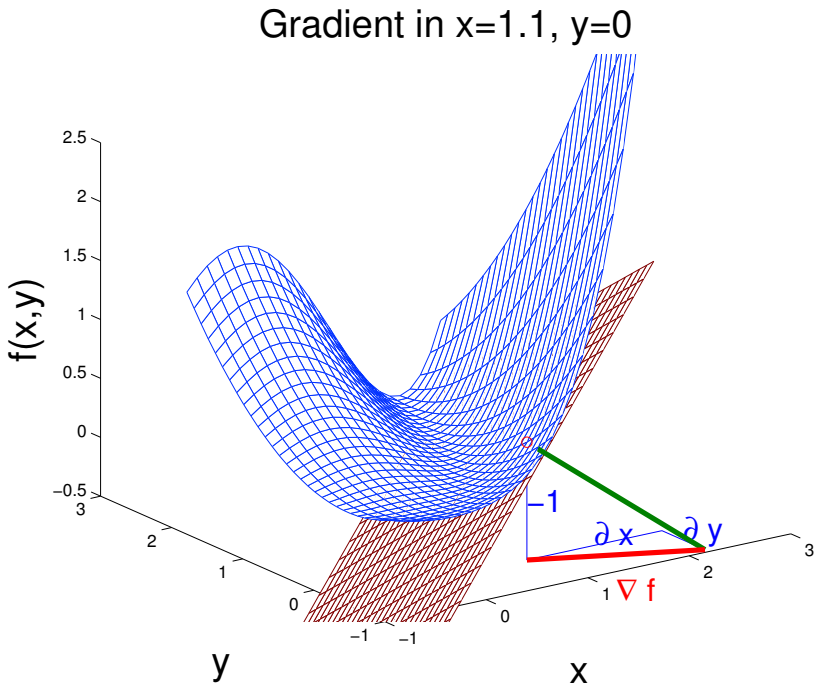
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Die Tangentialebene bildet das lineare Modell von f um $\bar{x}$,

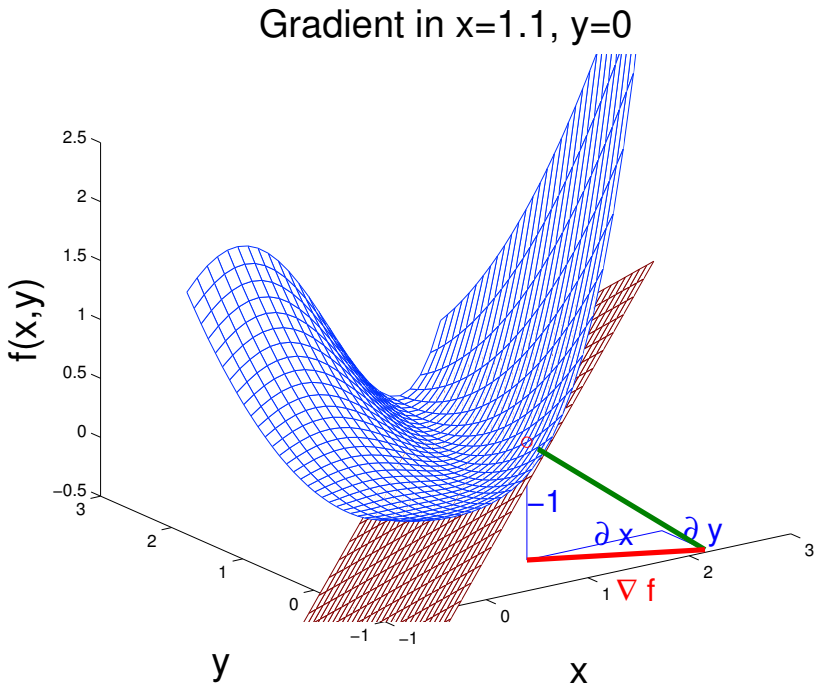
$$\bar{f}_{\bar{x}}(x) := f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}).$$

Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

Für $\bar{x} \in \mathbb{R}^n$ wird Funktionswert $f(\bar{x})$ und Gradient $\nabla f(\bar{x}) \in \mathbb{R}^n$ berechnet.

Der Gradient:

$$\nabla f(x) := \begin{bmatrix} \frac{\partial f}{\partial x_1}(x) \\ \vdots \\ \frac{\partial f}{\partial x_n}(x) \end{bmatrix}$$



- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ ist der Normalvektor zur Tangentialebene an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.
- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x.
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.

Die Tangentialebene bildet das lineare Modell von f um $\bar{x}$,

$$\bar{f}_{\bar{x}}(x) := f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}).$$

Für x nahe bei $\bar{x}$ ist es eine gute Näherung an f: für glattes f erfüllt ∇f

$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x})}{\|x - \bar{x}\|} = \lim_{h \to 0} \frac{f(\bar{x} + h) - f(\bar{x}) - \nabla f(\bar{x})^T h}{\|h\|} = 0.$$

Gradient und Richtungsableitung, lineares Modell

Das lineare Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in $\bar{x}$: die Richtungsableitung von f in $\bar{x}$ in Richtung h,

$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

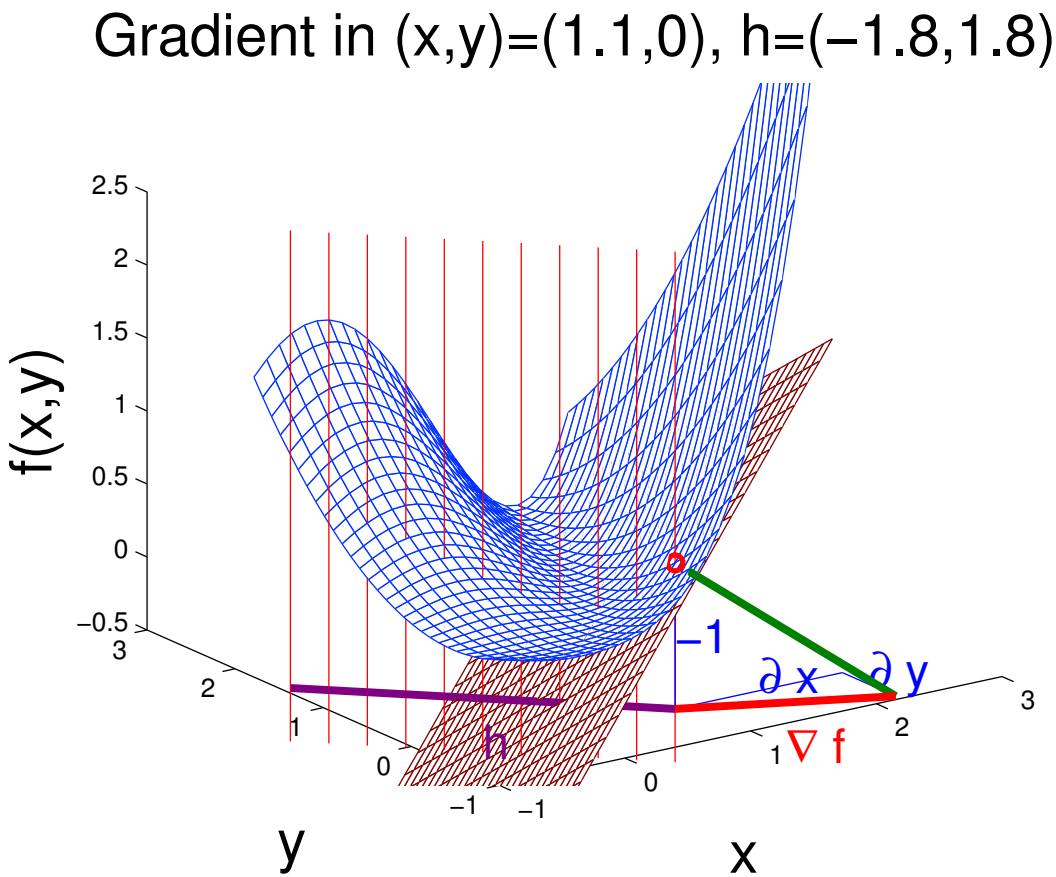
[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]

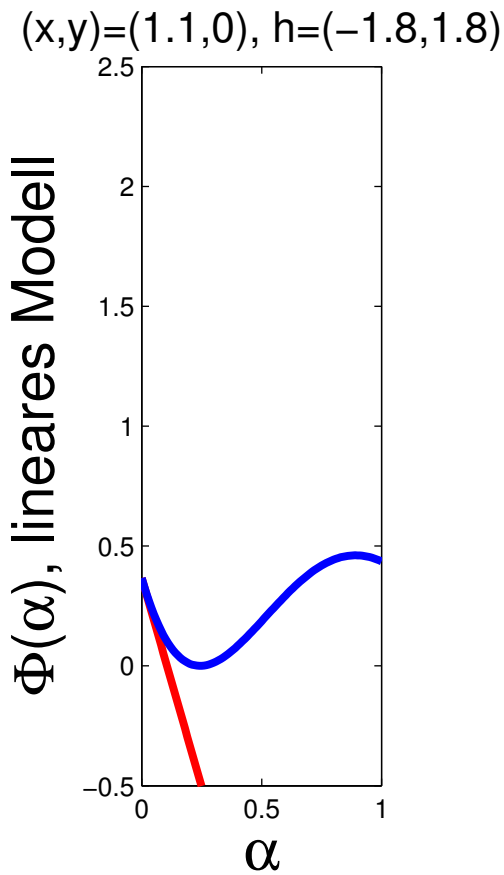
Gradient und Richtungsableitung, lineares Modell

Das lineare Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in $\bar{x}$: die Richtungsableitung von f in $\bar{x}$ in Richtung h,

$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



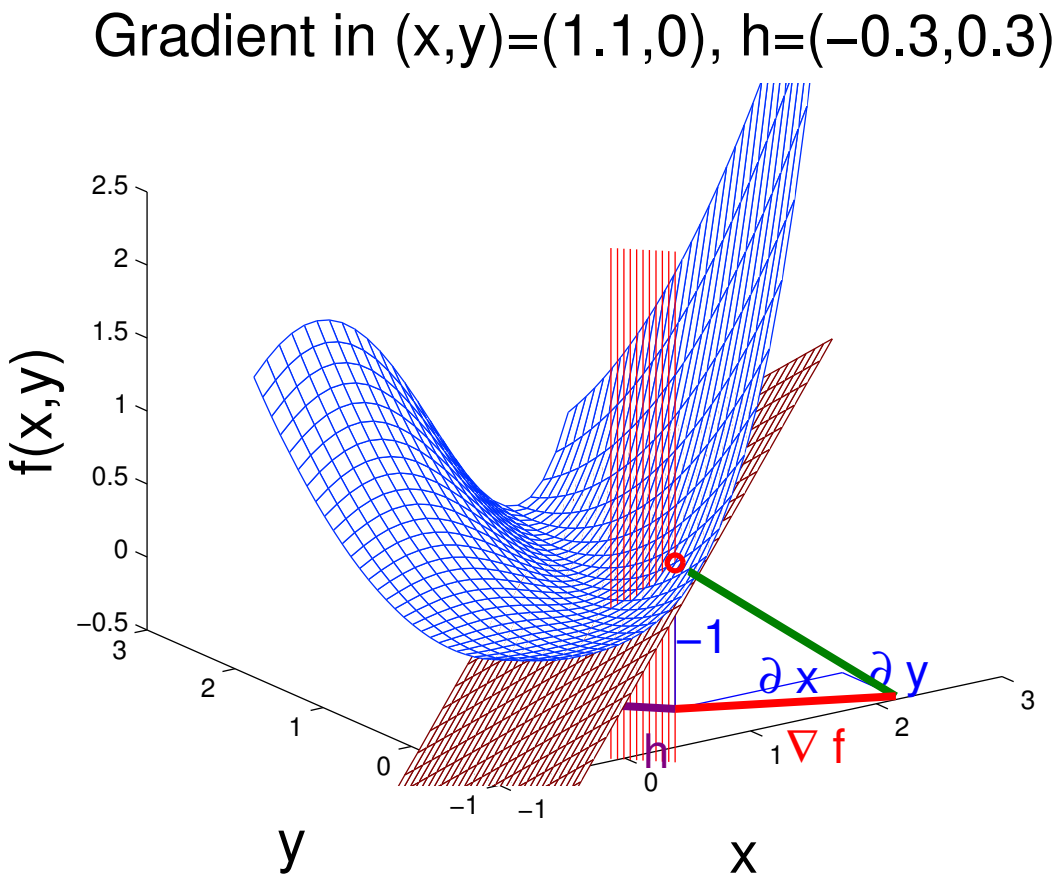


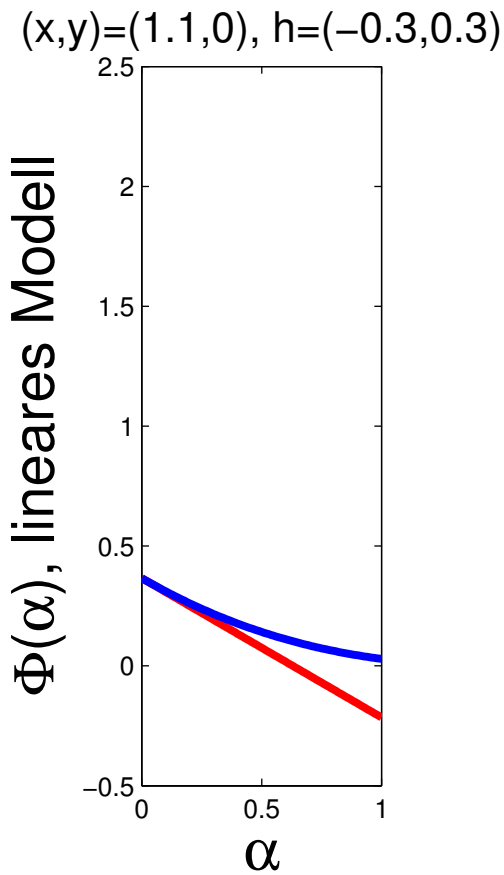
Gradient und Richtungsableitung, lineares Modell

Das lineare Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in $\bar{x}$: die Richtungsableitung von f in $\bar{x}$ in Richtung h,

$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



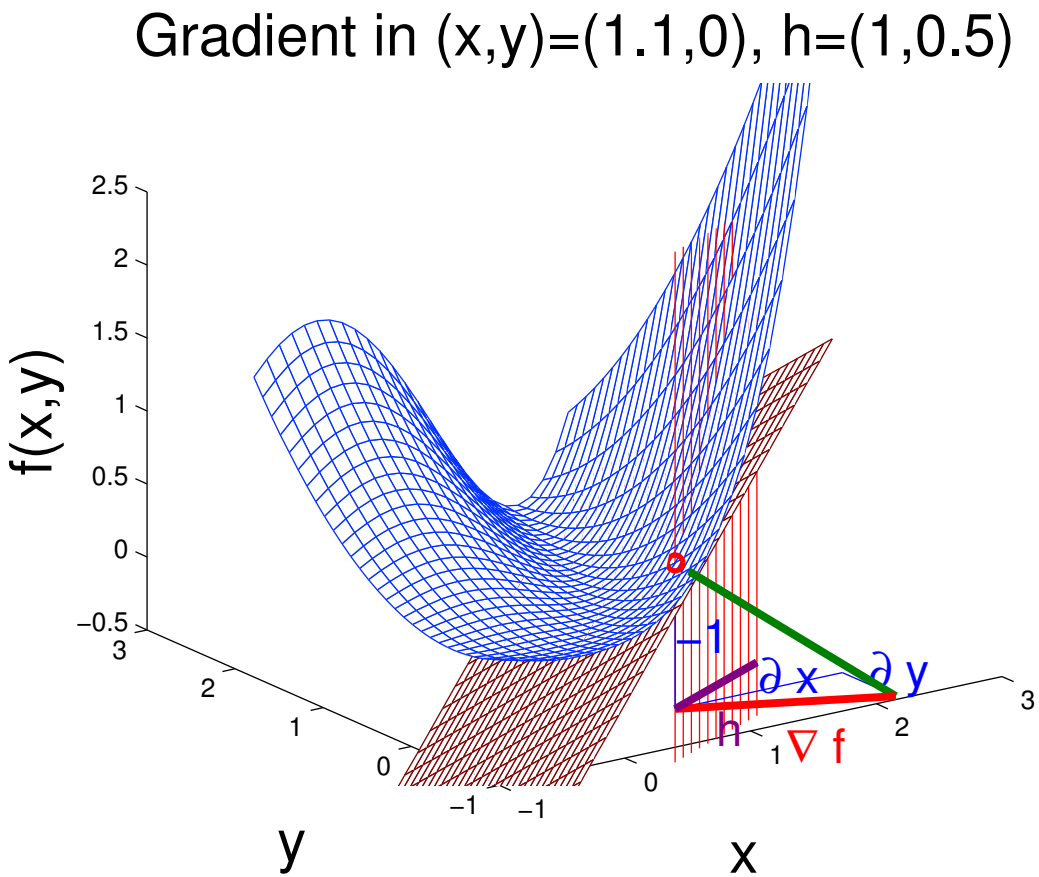


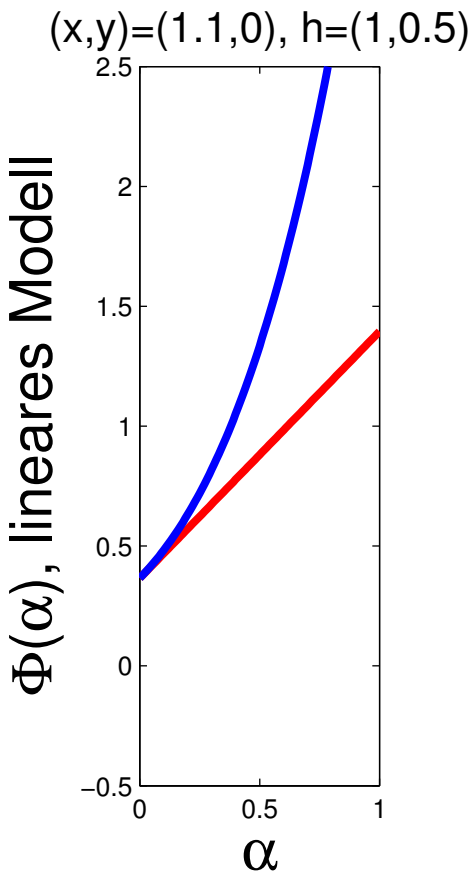
Gradient und Richtungsableitung, lineares Modell

Das lineare Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in $\bar{x}$: die Richtungsableitung von f in $\bar{x}$ in Richtung h,

$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



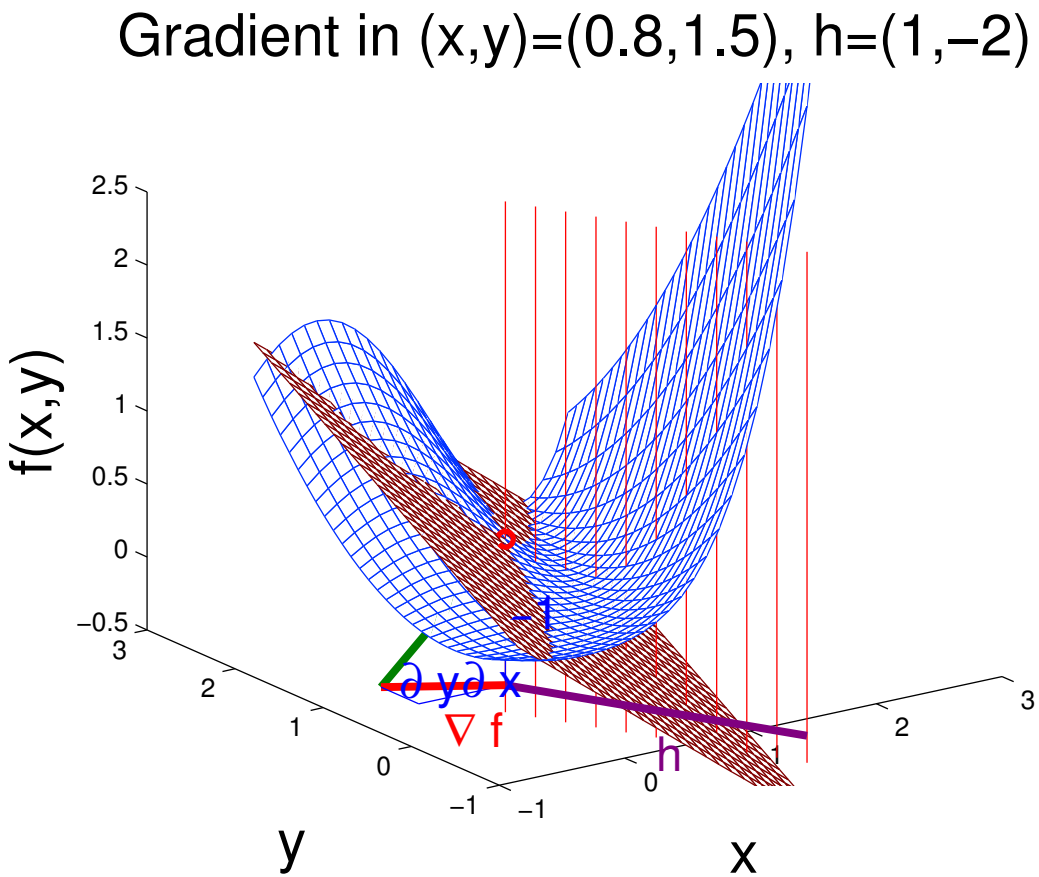


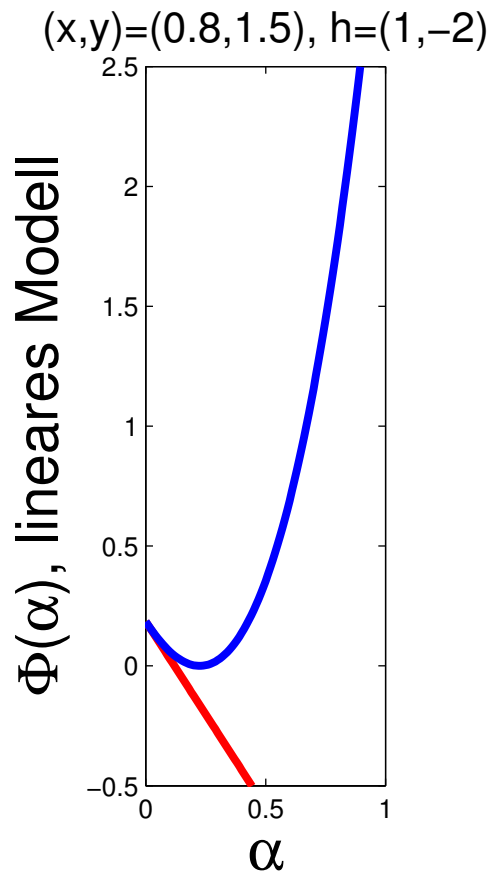
Gradient und Richtungsableitung, lineares Modell

Das lineare Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in $\bar{x}$: die Richtungsableitung von f in $\bar{x}$ in Richtung h,

$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



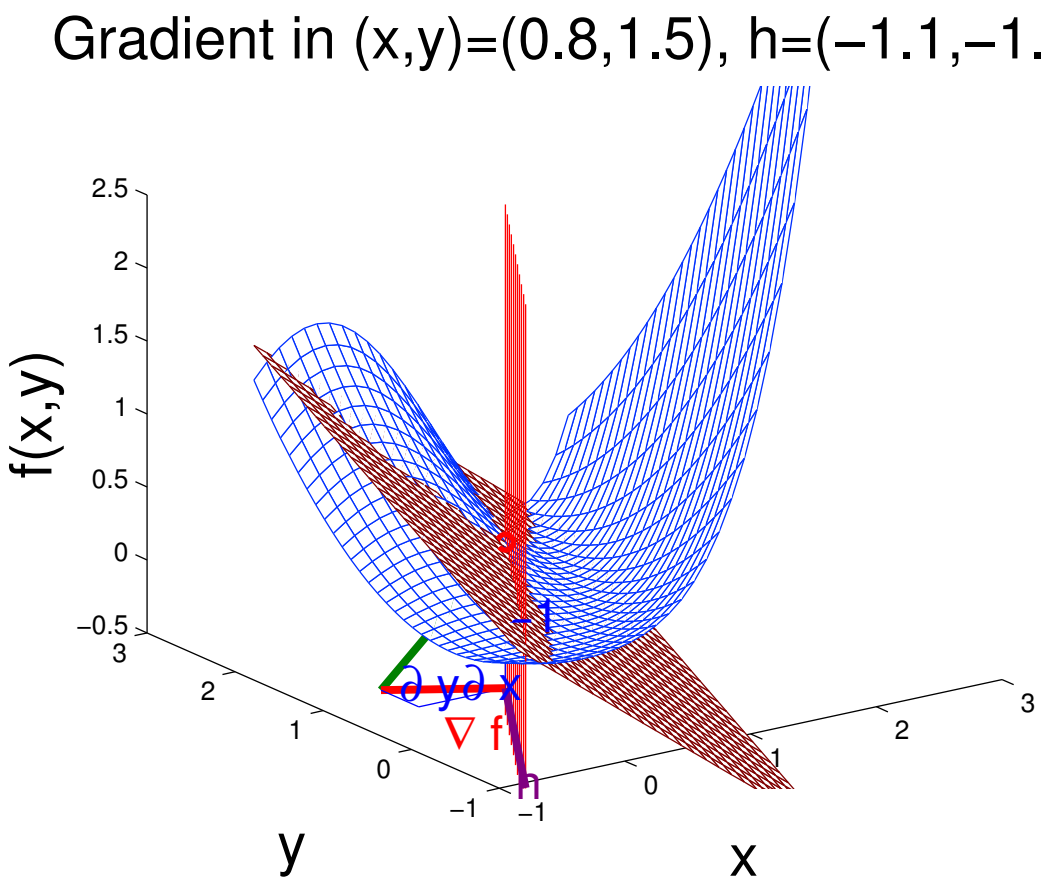


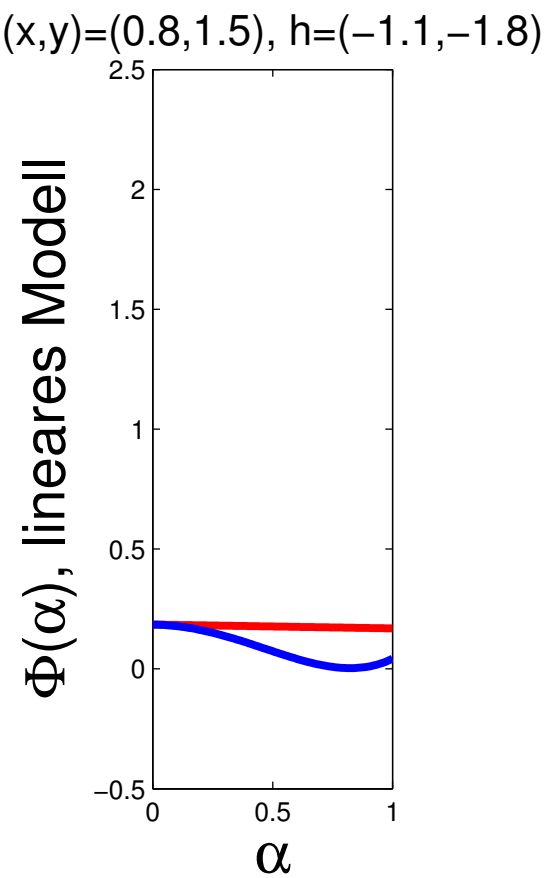
Gradient und Richtungsableitung, lineares Modell

Das lineare Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in $\bar{x}$: die Richtungsableitung von f in $\bar{x}$ in Richtung h,

$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x}+\alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x}) \qquad [D_{\lambda h} f(\bar{x}) = \lambda D_h f(\bar{x})]$$

[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x+\alpha h)$]





Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

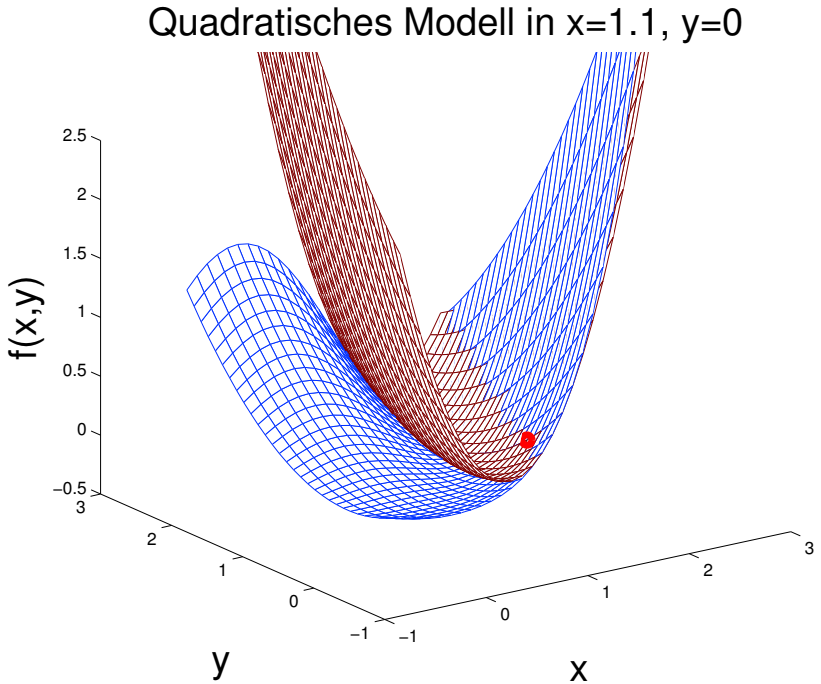
$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

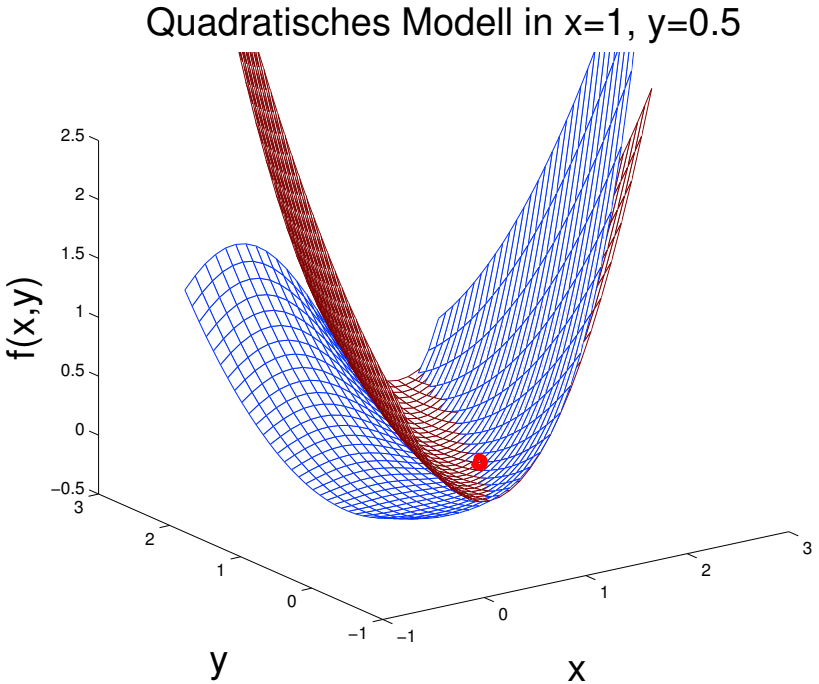


- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

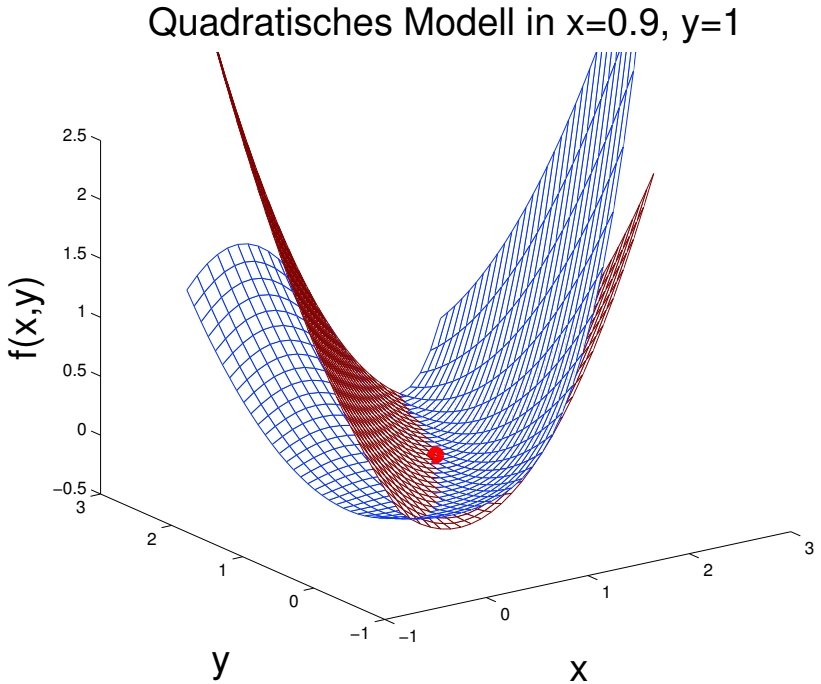


- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

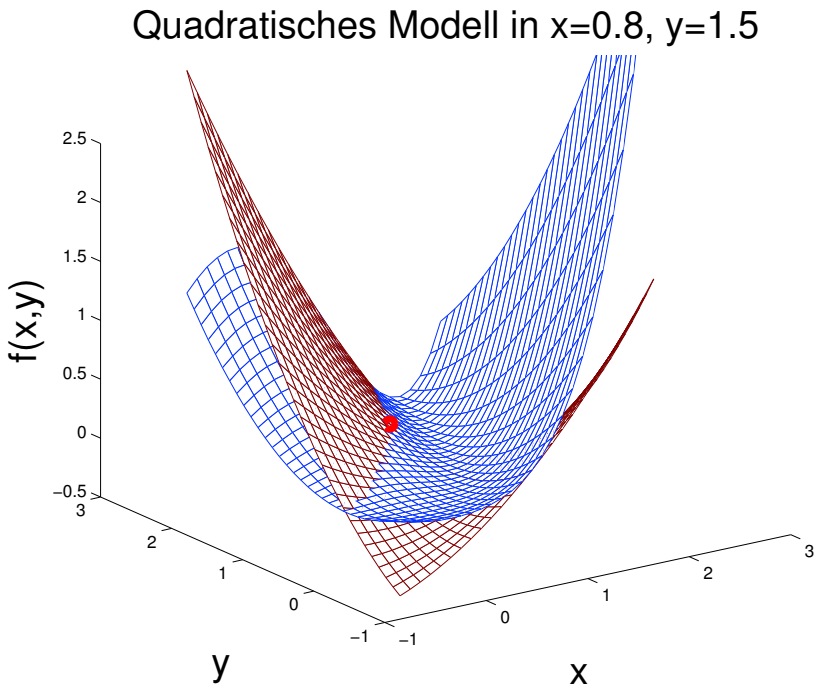


- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

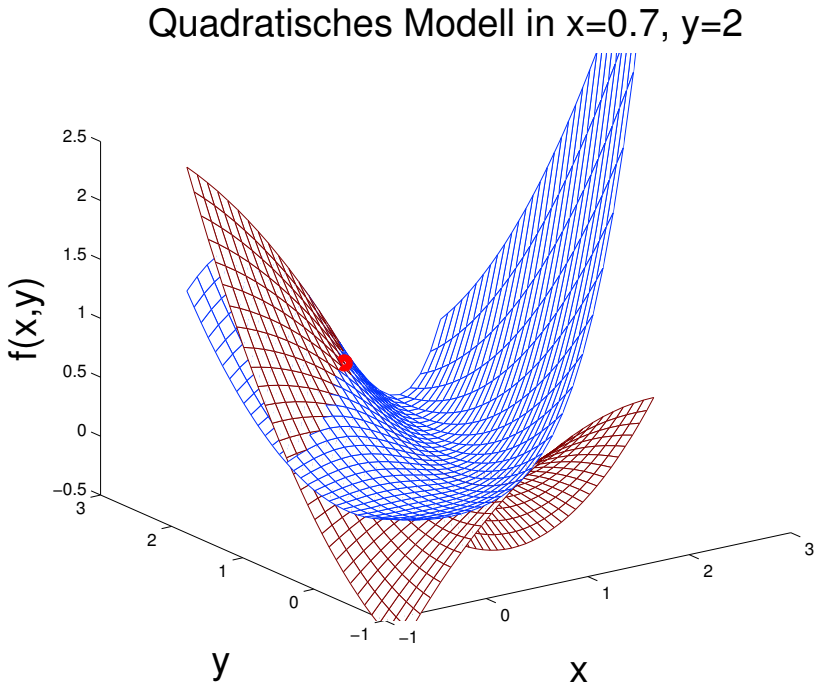


- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$

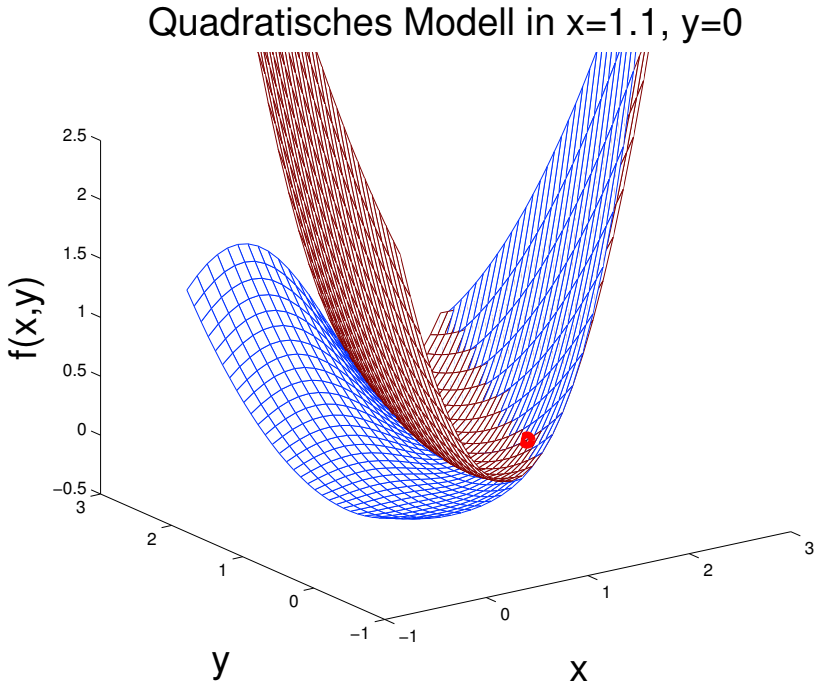


- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$



- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

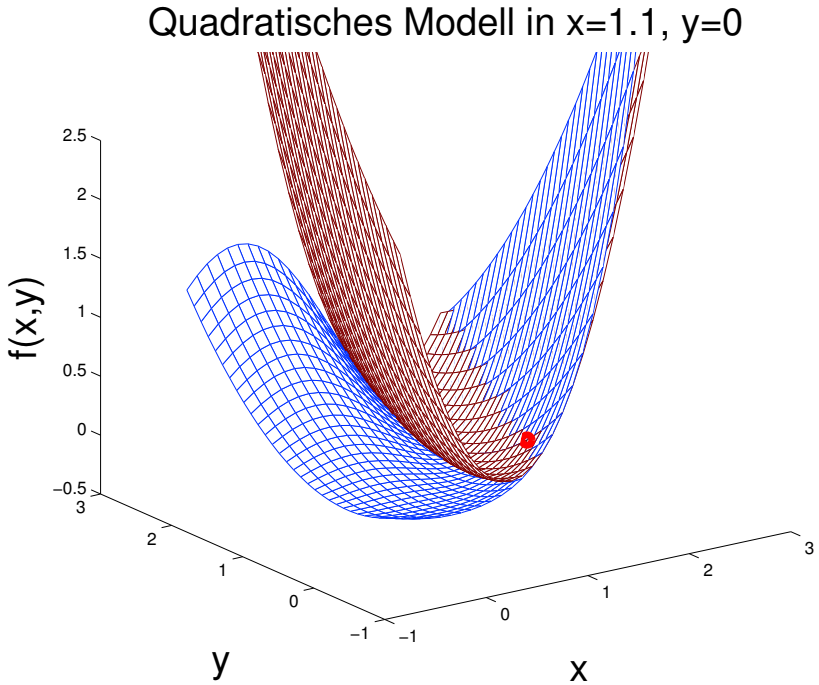
$f(\bar{x})$, $\nabla f(\bar{x})$ und $\nabla^2 f(\bar{x})$ bilden das quadratische Modell von f um $\bar{x}$,

$$\breve{f}_{\bar{x}}(x) := f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla^2 f(\bar{x})(x - \bar{x}).$$

Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und Hessematrix $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

$$\nabla^2 f(x) := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_1}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n}(x) & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n}(x) \end{bmatrix}$$



- $\nabla^2 f(x)$ ist symmetrisch, falls f zweimal stetig differenzierbar ist.
- Das quadrat. Modell aus $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$ schmiegt sich in $\begin{bmatrix} x \\ f(x) \end{bmatrix}$ an den Graphen $\left\{ \begin{bmatrix} x \\ f(x) \end{bmatrix} : x \in \mathbb{R}^n \right\}$ von f an.
- $\nabla^2 f(x)$ ist die Krümmung von f in x.

$f(\bar{x})$, $\nabla f(\bar{x})$ und $\nabla^2 f(\bar{x})$ bilden das quadratische Modell von f um $\bar{x}$,

$$\breve{f}_{\bar{x}}(x) := f(\bar{x}) + \nabla f(\bar{x})^T (x - \bar{x}) + \frac{1}{2}(x - \bar{x})^T \nabla^2 f(\bar{x})(x - \bar{x}).$$

Für x nahe bei $\bar{x}$ ist es eine sehr gute Näherung an f: für glattes f erfüllt $\nabla^2 f$

$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x}) - \frac{1}{2}(x - \bar{x})^T \nabla^2 f(\bar{x})(x - \bar{x})}{\|x - \bar{x}\|^2} = 0.$$

Quadratisches Modell in Richtung h

Das quadratische Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ die gleiche Steigung und Krümmung wie f in $\bar{x}$.

$$\breve{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

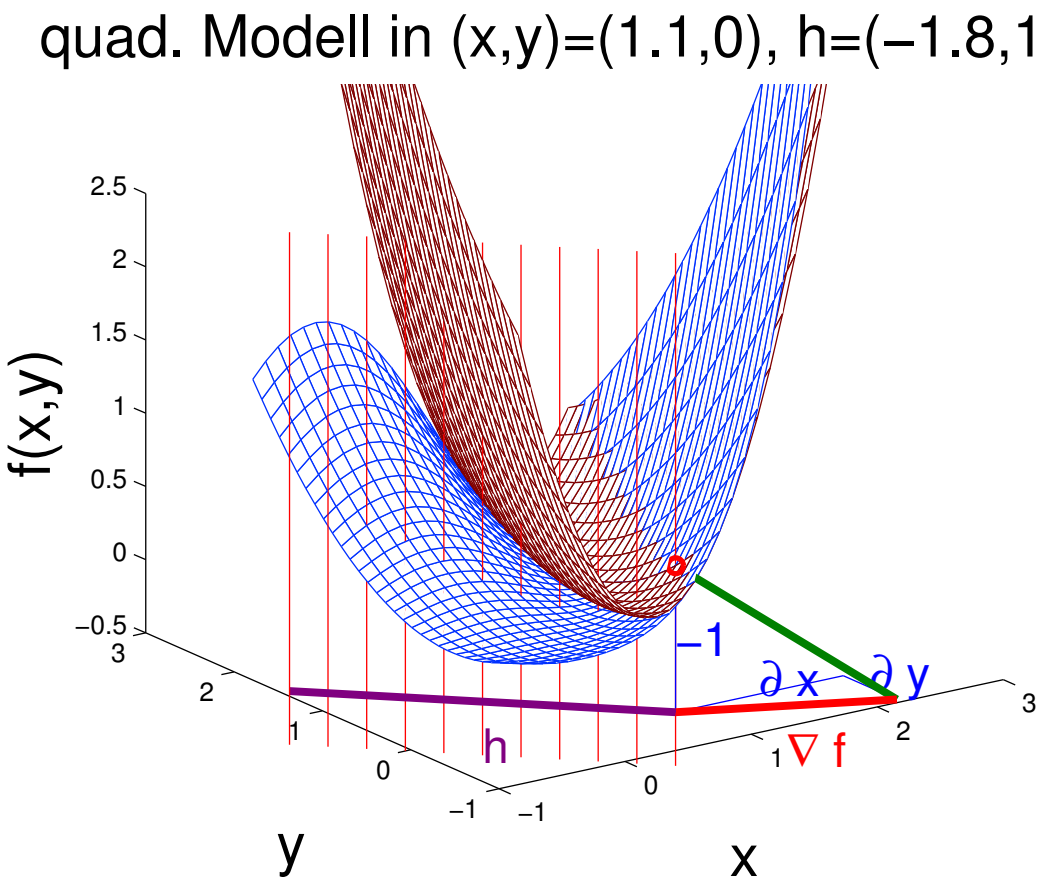
[Taylor-Entw. 2. Ord. der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]

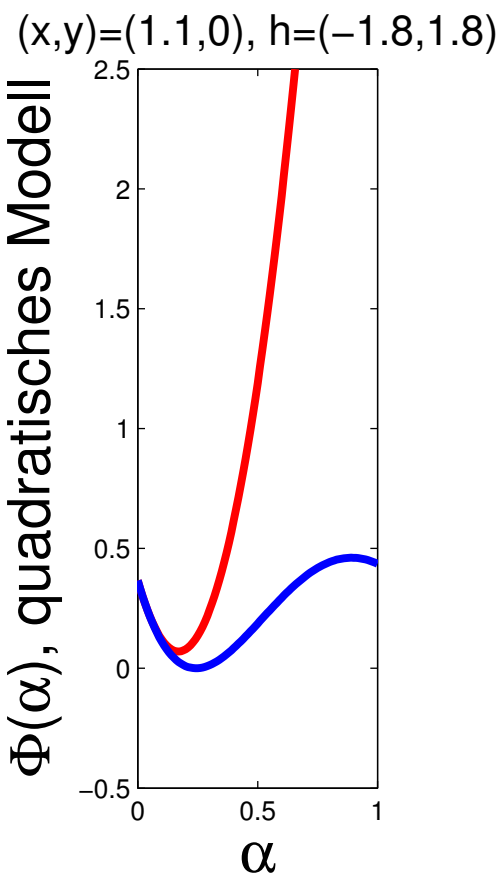
Quadratisches Modell in Richtung h

Das quadratische Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ die gleiche Steigung und Krümmung wie f in $\bar{x}$.

$$\breve{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

[Taylor-Entw. 2. Ord. der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



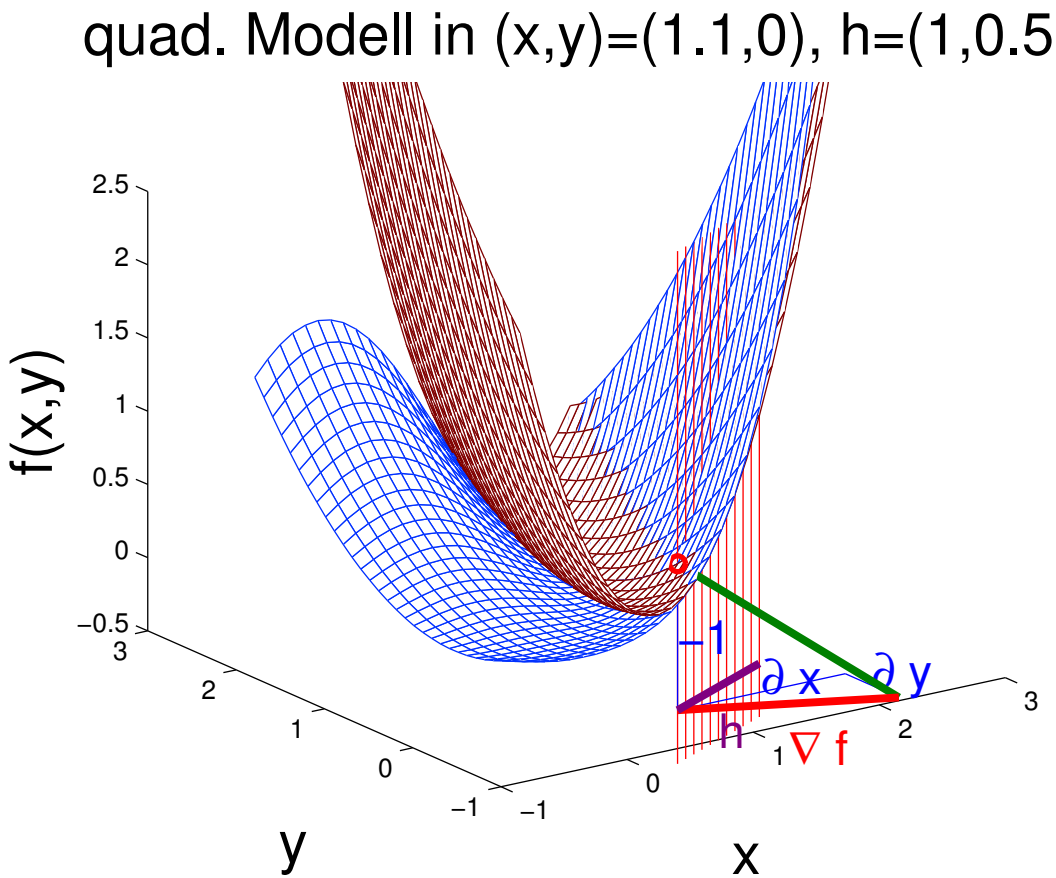


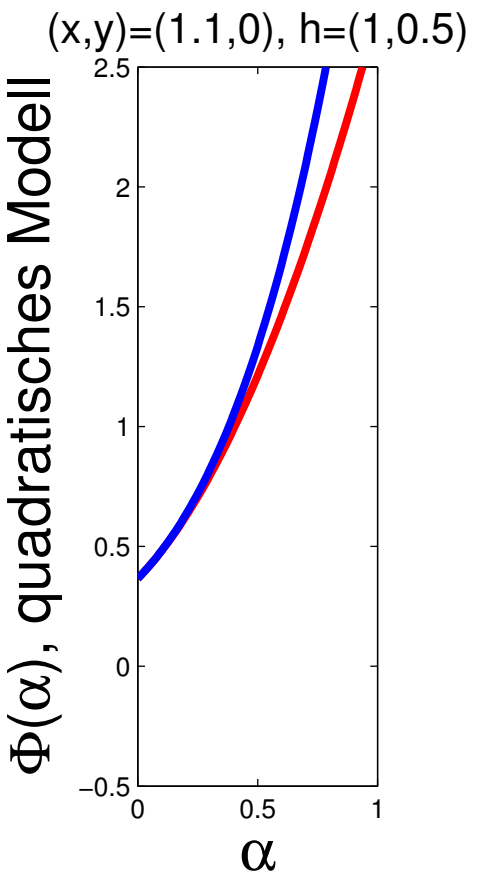
Quadratisches Modell in Richtung h

Das quadratische Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ die gleiche Steigung und Krümmung wie f in $\bar{x}$.

$$\breve{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

[Taylor-Entw. 2. Ord. der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



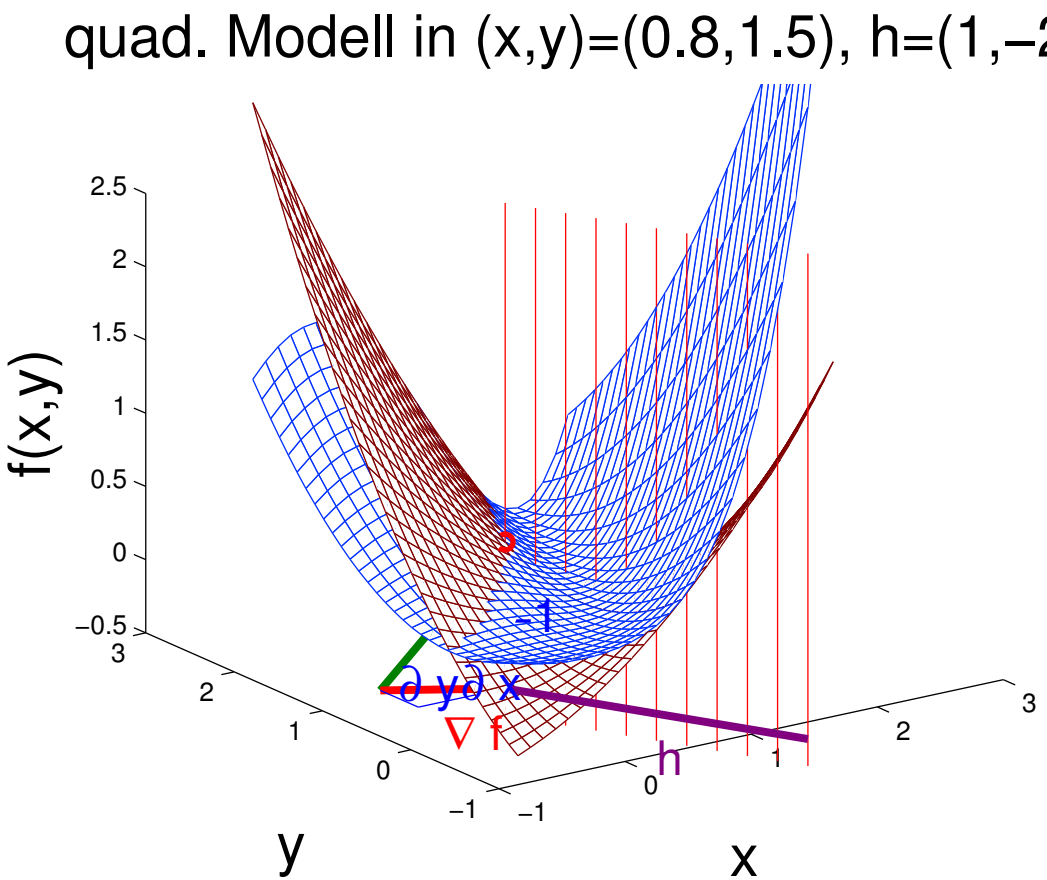


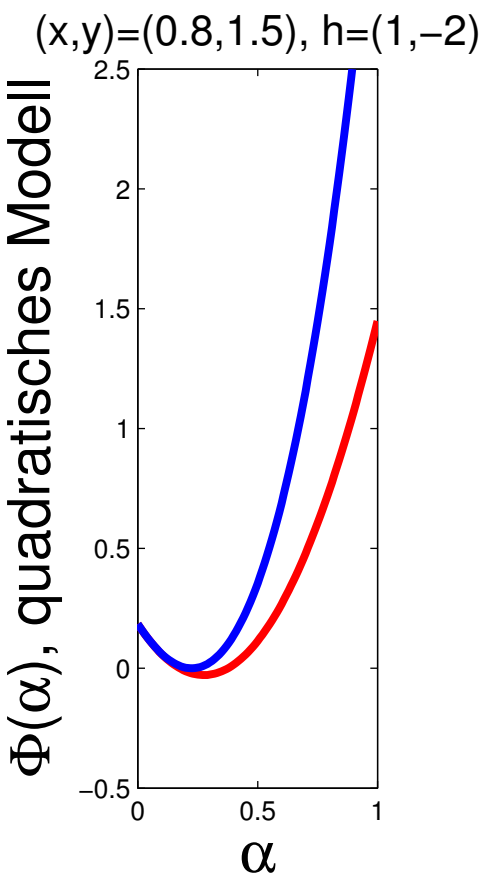
Quadratisches Modell in Richtung h

Das quadratische Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ die gleiche Steigung und Krümmung wie f in $\bar{x}$.

$$\breve{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

[Taylor-Entw. 2. Ord. der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



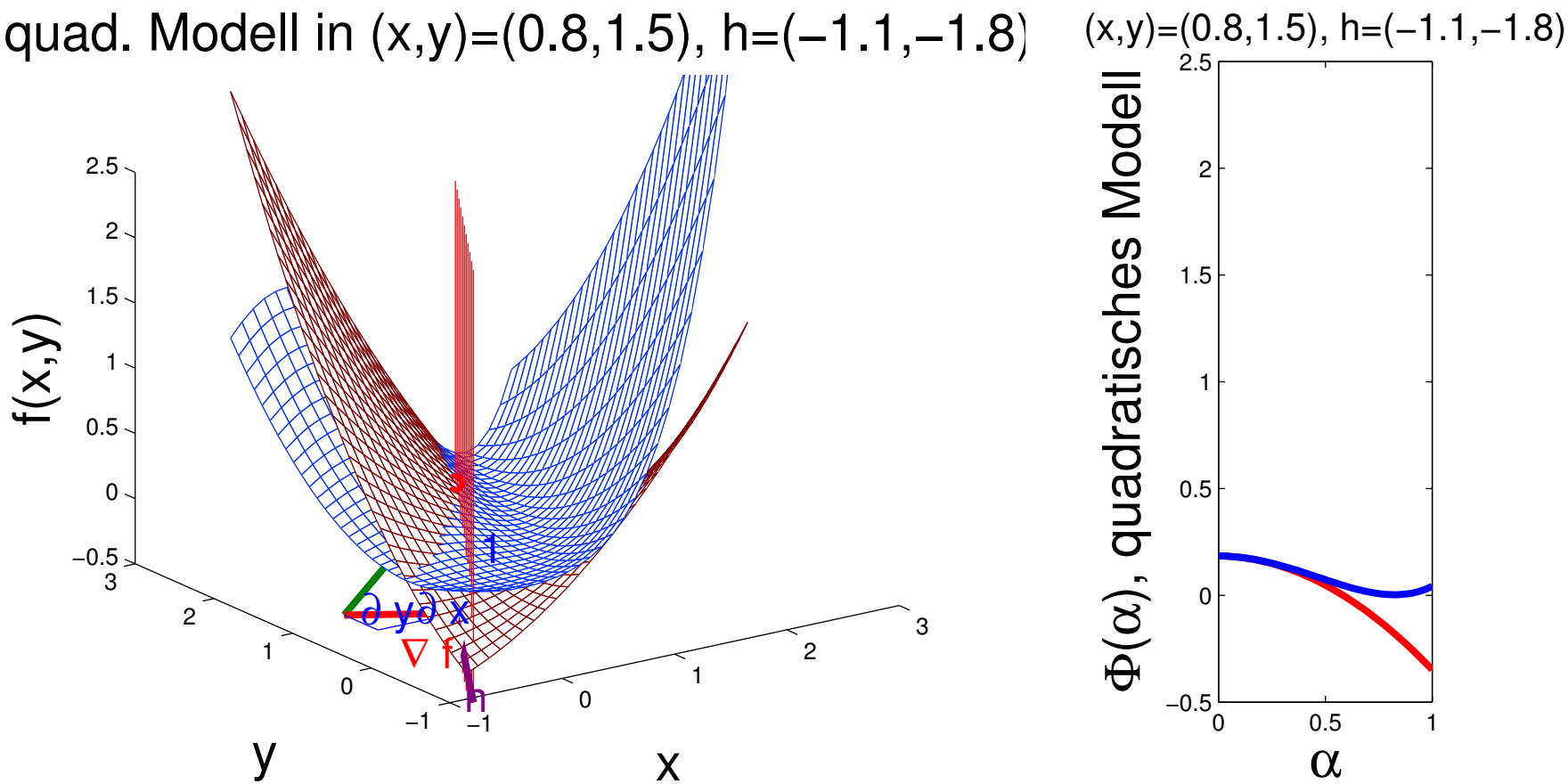


Quadratisches Modell in Richtung h

Das quadratische Modell von f in $\bar{x}$ hat in jede Richtung $h \in \mathbb{R}^n$ die gleiche Steigung und Krümmung wie f in $\bar{x}$.

$$\breve{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

[Taylor-Entw. 2. Ord. der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]



Der Satz von Taylor/Mittelwertsatz

Satz (Taylor/Mittelwertsatz)

Sei f oft genug stetig differenzierbar und $\bar{x}, h \in \mathbb{R}^n$, dann

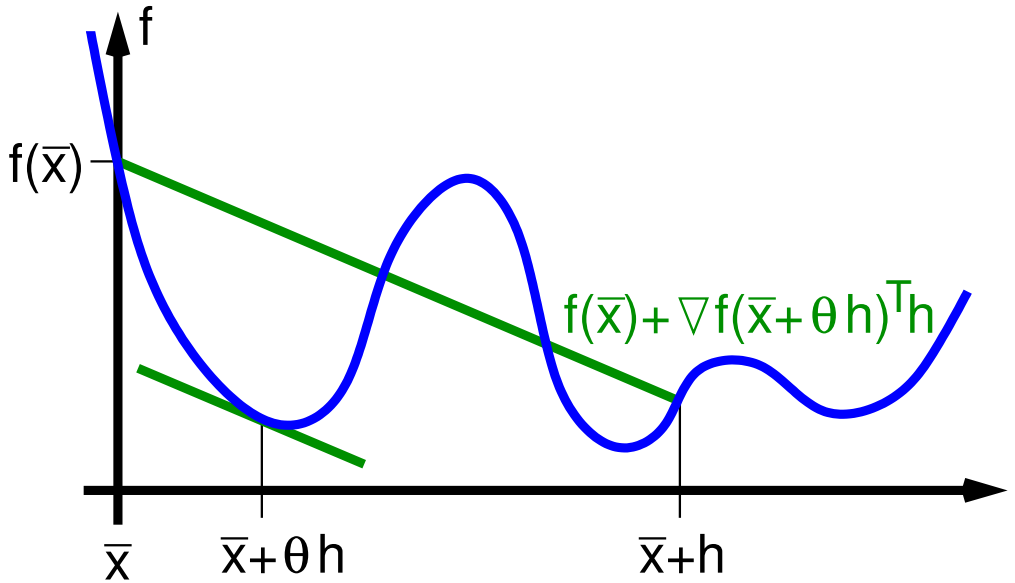
$$\exists \theta_1 \in (0,1): \quad f(\bar{x}+h) = f(\bar{x}) + \nabla f(\bar{x}+\theta_1 h)^T h,$$

$$\exists \theta_2 \in (0,1): \quad f(\bar{x}+h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \tfrac{1}{2} h^T \nabla^2 f(\bar{x}+\theta_2 h) h,$$

$$\exists \theta_3 \in (0,1): \quad f(\bar{x}+h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \tfrac{1}{2} h^T \nabla^2 f(\bar{x}) h + \tfrac{1}{6} \nabla^3 f(\bar{x}+\theta_3 h)[h,h,h]$$

[∇^3 steht für die 3. Ableitung] „Taylor-Entwicklung von f um $\bar{x}$“

Illustration des ersten Falls des Mittelwertsatzes:



Lipschitz-Stetigkeit, „Klein-**o**-Notation“

Eine Funktion $G : \mathbb{R}^n \to \mathbb{R}^m$ heißt Lipschitz-stetig auf einer Menge $S \subseteq \mathbb{R}^n$, falls es eine Konstante $L > 0$ gibt mit $\|G(x) - G(y)\| \leq L\|x - y\| \ \forall x, y \in S$.

[$\Rightarrow$ Die Werte können sich nur bei größerem Abstand stark ändern!]

Lipschitz-Stetigkeit, „Klein-**o**-Notation“

Eine Funktion $G : \mathbb{R}^n \to \mathbb{R}^m$ heißt Lipschitz-stetig auf einer Menge $S \subseteq \mathbb{R}^n$, falls es eine Konstante $L > 0$ gibt mit $\|G(x) - G(y)\| \leq L\|x - y\| \ \forall x, y \in S$.
[$\Rightarrow$ Die Werte können sich nur bei größerem Abstand stark ändern!]

Jede auf $\mathbb{R}^n$ stetig differenzierbare Funktion ist für jedes $\bar{x} \in \mathbb{R}^n$ und $\rho > 0$ auf der ρ-Kugel um $\bar{x}$, $B_\rho(\bar{x}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \rho\}$ Lipschitz-stetig.

- Ist ∇f um $\bar{x}$ für großes ρ Lipschitz-stetig mit kleinem L, dann ist das lineare Modell auf B_ρ eine gute Näherung an f.
- Ist $\nabla^2 f$ um $\bar{x}$ für großes ρ Lipschitz-stetig mit kleinem L, dann ist das quadratische Modell auf B_ρ eine gute Näherung an f.

Lipschitz-Stetigkeit, „Klein-**o**-Notation“

Eine Funktion $G : \mathbb{R}^n \to \mathbb{R}^m$ heißt Lipschitz-stetig auf einer Menge $S \subseteq \mathbb{R}^n$, falls es eine Konstante $L > 0$ gibt mit $\|G(x) - G(y)\| \leq L\|x - y\| \ \forall x, y \in S$.
[$\Rightarrow$ Die Werte können sich nur bei größerem Abstand stark ändern!]

Jede auf $\mathbb{R}^n$ stetig differenzierbare Funktion ist für jedes $\bar{x} \in \mathbb{R}^n$ und $\rho > 0$ auf der ρ-Kugel um $\bar{x}$, $B_\rho(\bar{x}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \rho\}$ Lipschitz-stetig.

- Ist ∇f um $\bar{x}$ für großes ρ Lipschitz-stetig mit kleinem L, dann ist das lineare Modell auf B_ρ eine gute Näherung an f.
- Ist $\nabla^2 f$ um $\bar{x}$ für großes ρ Lipschitz-stetig mit kleinem L, dann ist das quadratische Modell auf B_ρ eine gute Näherung an f.

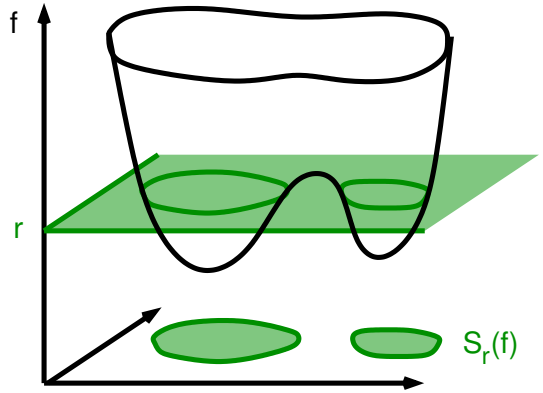
Aus dem Satz von Taylor und der Lipschitz-Stetigkeit von $\nabla^k f$ folgt
$f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \mathbf{o}(\|h\|)$,
$f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \frac{1}{2} h^T \nabla^2 f(\bar{x}) h + \mathbf{o}(\|h\|^2)$

Das Landau-Symbol $\mathbf{o}(g(y))$ steht immer als Ersatz für eine nicht weiter interessierende Funktion $g'(y)$ mit der Eigenschaft $\lim_{y \to 0} \frac{g'(y)}{g(y)} \to 0$, also ein g', das schneller klein wird als g.
Bei Folgen $g(y^{(k)})$ steht es für ein $g'(y^{(k)})$ mit $\lim_{k \to \infty} \frac{g'(y^{(k)})}{g(y^{(k)})} \to 0$.

Niveaumengen und Niveaulinien

Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

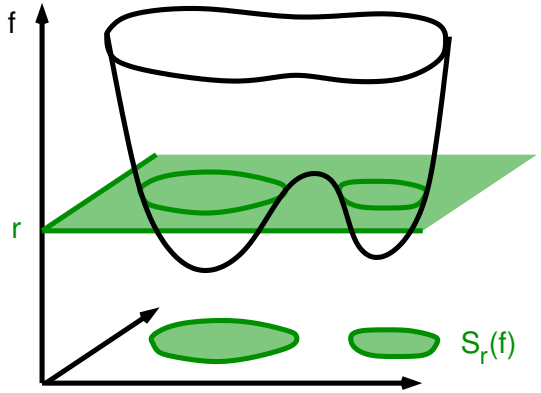
$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$



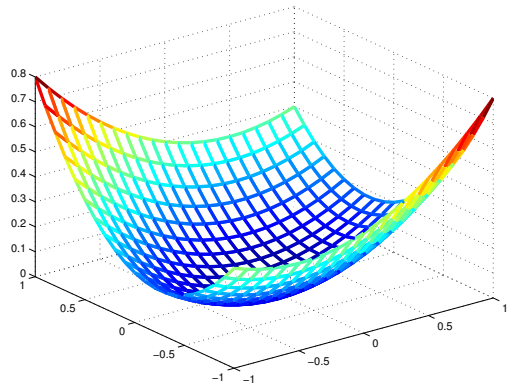
Niveaumengen und Niveaulinien

Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

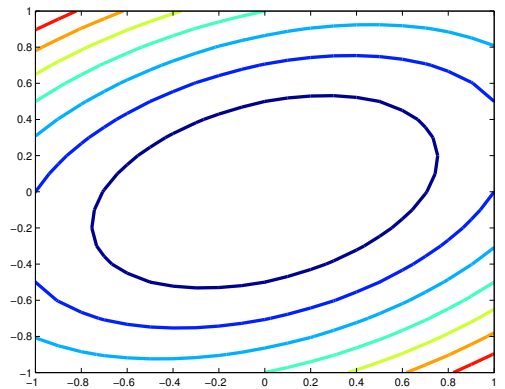
$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$



Funktionsdarstellungen über Niveaulinien [„Linien“]
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (*contour plots*)
helfen, Verfahren zu illustrieren.
[Höhenlinien in Landkarten, Wetterkarten]



Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$ gilt $0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|)$.

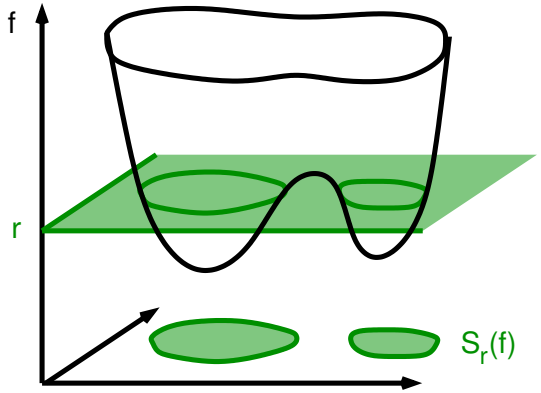


Bsp: quadratische Funktion

$$\tfrac{1}{2} x^T Q x + q^T x + d$$

mit Q positiv definit.

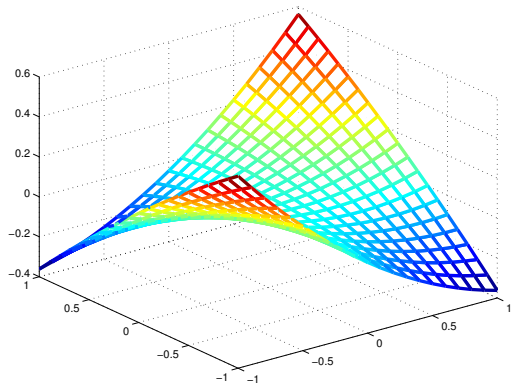
Niveaumengen und Niveaulinien



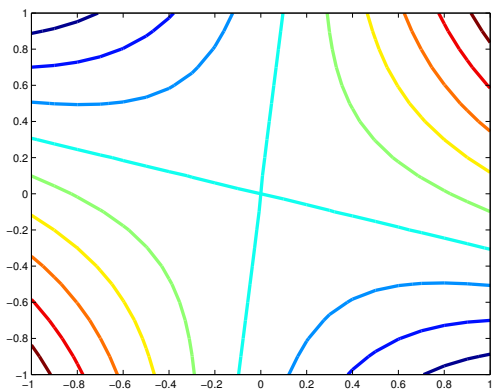
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$

Funktionsdarstellungen über Niveaulinien [„Linien“]
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (*contour plots*)
helfen, Verfahren zu illustrieren.
[Höhenlinien in Landkarten, Wetterkarten]



Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$ gilt $0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|)$.

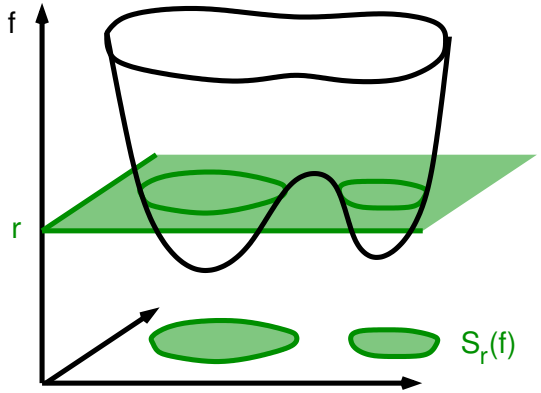


Bsp: quadratische Funktion

$$\tfrac{1}{2} x^T Q x + q^T x + d$$

mit Q indefinit.

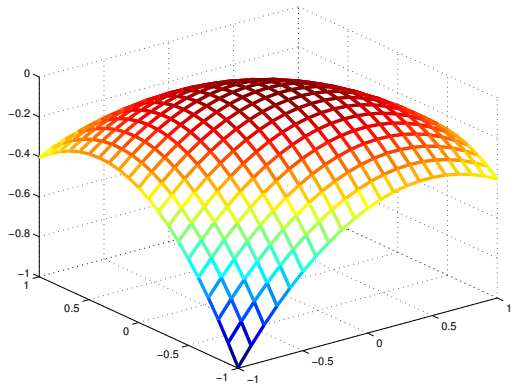
Niveaumengen und Niveaulinien



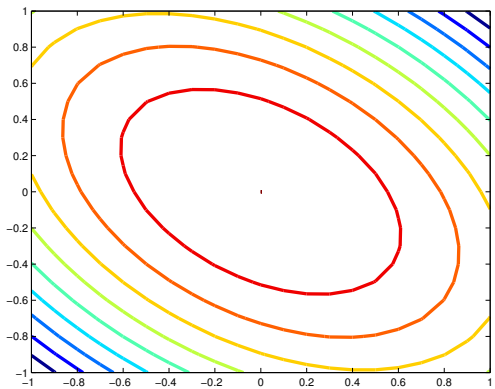
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$

Funktionsdarstellungen über Niveaulinien [„Linien“]
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (*contour plots*)
helfen, Verfahren zu illustrieren.
[Höhenlinien in Landkarten, Wetterkarten]



Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$ gilt $0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|)$.

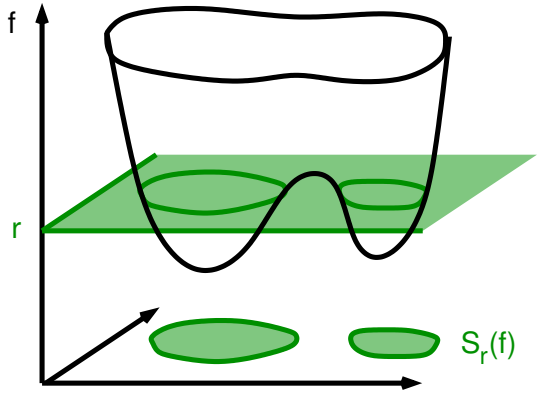


Bsp: quadratische Funktion

$$\tfrac{1}{2} x^T Q x + q^T x + d$$

mit Q negativ definit.

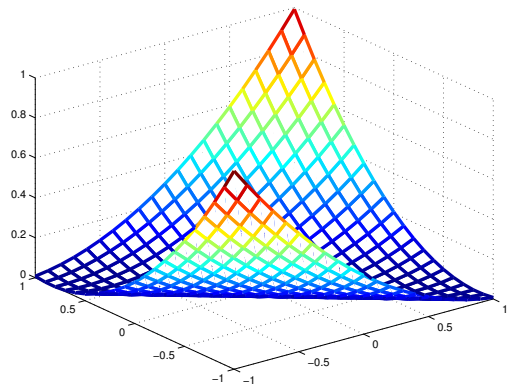
Niveaumengen und Niveaulinien



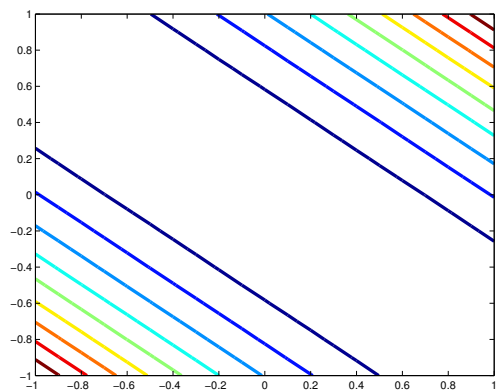
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$

Funktionsdarstellungen über Niveaulinien [„Linien“]
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (*contour plots*)
helfen, Verfahren zu illustrieren.
[Höhenlinien in Landkarten, Wetterkarten]



Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$ gilt $0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|)$.

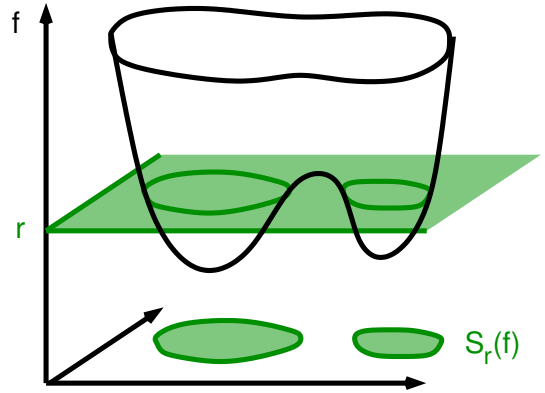


Bsp: quadratische Funktion

$$\tfrac{1}{2} x^T Q x + q^T x + d$$

mit Q positiv semidefinit.

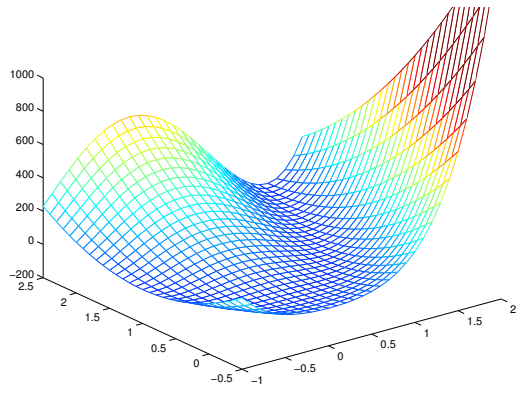
Niveaumengen und Niveaulinien



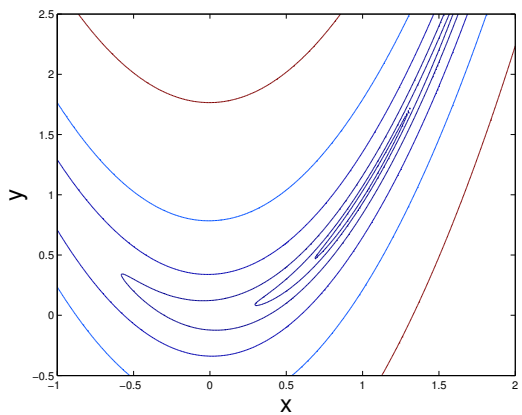
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$

Funktionsdarstellungen über Niveaulinien [„Linien“]
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (*contour plots*)
helfen, Verfahren zu illustrieren.
[Höhenlinien in Landkarten, Wetterkarten]



Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$ gilt $0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|)$.



Bsp: Rosenbrock-Funktion (*banana shape*)

$$f(x,y) = \tfrac{1}{4}[(y-x^2)^2 + \tfrac{1}{100}(1-x)^2]$$

Minimum wird in (1,1) angenommen.

Inhalt

6.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Minimum einer glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, so gilt

$$\nabla f(\bar{x}) = 0.$$

Sonst wäre (setze $h = -\nabla f(\bar{x})$ in Taylor) für α klein genug

$$f(\bar{x} - \alpha \nabla f(\bar{x})) = f(\bar{x}) - \alpha \underbrace{\nabla f(\bar{x})^T \nabla f(\bar{x})}_{= \|\nabla f(\bar{x})\|^2 > 0} + \mathbf{o}(\alpha) < f(\bar{x}).$$

6.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Minimum einer glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, so gilt

$$\nabla f(\bar{x}) = 0.$$

Sonst wäre (setze $h = -\nabla f(\bar{x})$ in Taylor) für α klein genug

$$f(\bar{x} - \alpha \nabla f(\bar{x})) = f(\bar{x}) - \alpha \underbrace{\nabla f(\bar{x})^T \nabla f(\bar{x})}_{= \|\nabla f(\bar{x})\|^2 > 0} + \mathbf{o}(\alpha) < f(\bar{x}).$$

Ein Punkt $\bar{x}$ mit $\nabla f(\bar{x}) = 0$ heißt stationärer Punkt von f.

Stationarität ist notwendig, aber i.A. nicht hinreichend für Minimalität!
Bsp: $x = 0$ ist stationärer Punkt von $f(x) = x^3$ oder $f(x) = -x^2$.

6.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Minimum einer glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, so gilt

$$\nabla f(\bar{x}) = 0.$$

Sonst wäre (setze $h = -\nabla f(\bar{x})$ in Taylor) für α klein genug

$$f(\bar{x} - \alpha \nabla f(\bar{x})) = f(\bar{x}) - \alpha \underbrace{\nabla f(\bar{x})^T \nabla f(\bar{x})}_{= \|\nabla f(\bar{x})\|^2 > 0} + \mathbf{o}(\alpha) < f(\bar{x}).$$

Ein Punkt $\bar{x}$ mit $\nabla f(\bar{x}) = 0$ heißt stationärer Punkt von f.

Stationarität ist notwendig, aber i.A. nicht hinreichend für Minimalität!
Bsp: $x = 0$ ist stationärer Punkt von $f(x) = x^3$ oder $f(x) = -x^2$.

Ausnahme: Für konvexes f ist jeder stationäre Punkt globales Minimum!

6.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Minimum einer glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, so gilt

$$\nabla f(\bar{x}) = 0.$$

Sonst wäre (setze $h = -\nabla f(\bar{x})$ in Taylor) für α klein genug

$$f(\bar{x} - \alpha \nabla f(\bar{x})) = f(\bar{x}) - \alpha \underbrace{\nabla f(\bar{x})^T \nabla f(\bar{x})}_{= \|\nabla f(\bar{x})\|^2 > 0} + \mathbf{o}(\alpha) < f(\bar{x}).$$

Ein Punkt $\bar{x}$ mit $\nabla f(\bar{x}) = 0$ heißt stationärer Punkt von f.

Stationarität ist notwendig, aber i.A. nicht hinreichend für Minimalität!
Bsp: $x = 0$ ist stationärer Punkt von $f(x) = x^3$ oder $f(x) = -x^2$.

Ausnahme: Für konvexes f ist jeder stationäre Punkt globales Minimum!

Bsp: $f(x) = \frac{1}{2} x^T Q x + q^T x + c$ mit $Q \succ 0$ ist (streng) konvex.
Mit $\nabla f(x) = Qx + q$ bestimmt $\nabla f(x^*) = 0$ das Minimum x^* eindeutig,

$$x^* = -Q^{-1} q.$$

6.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Minimum einer glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, so gilt

$$\nabla f(\bar{x}) = 0.$$

Sonst wäre (setze $h = -\nabla f(\bar{x})$ in Taylor) für α klein genug

$$f(\bar{x} - \alpha \nabla f(\bar{x})) = f(\bar{x}) - \alpha \underbrace{\nabla f(\bar{x})^T \nabla f(\bar{x})}_{= \|\nabla f(\bar{x})\|^2 > 0} + \mathbf{o}(\alpha) < f(\bar{x}).$$

Ein Punkt $\bar{x}$ mit $\nabla f(\bar{x}) = 0$ heißt stationärer Punkt von f.

Stationarität ist notwendig, aber i.A. nicht hinreichend für Minimalität!
Bsp: $x = 0$ ist stationärer Punkt von $f(x) = x^3$ oder $f(x) = -x^2$.

Ausnahme: Für konvexes f ist jeder stationäre Punkt globales Minimum!

Geometrisch bedeutet $\nabla f(x) = 0$, dass die Tangentialebene an f in x „waagrecht“ liegt.
Ist sie nicht waagrecht, kann man sicher noch ein wenig hinunterrutschen.

6.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Minimum einer glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, so gilt

$$\nabla f(\bar{x}) = 0.$$

Sonst wäre (setze $h = -\nabla f(\bar{x})$ in Taylor) für α klein genug

$$f(\bar{x} - \alpha \nabla f(\bar{x})) = f(\bar{x}) - \alpha \underbrace{\nabla f(\bar{x})^T \nabla f(\bar{x})}_{= \|\nabla f(\bar{x})\|^2 > 0} + \mathbf{o}(\alpha) < f(\bar{x}).$$

Ein Punkt $\bar{x}$ mit $\nabla f(\bar{x}) = 0$ heißt stationärer Punkt von f.

Stationarität ist notwendig, aber i.A. nicht hinreichend für Minimalität!
Bsp: $x = 0$ ist stationärer Punkt von $f(x) = x^3$ oder $f(x) = -x^2$.

Ausnahme: Für konvexes f ist jeder stationäre Punkt globales Minimum!

Konsequenz für Optimierungsverfahren:
Ist $\nabla f(x) \neq 0$, so kann man die Funktion in Richtung $-\nabla f(x)$ immer verbessern (u.U. nur für sehr kleine Schrittweite).
Die Schrittrichtung $h = -\nabla f(x)$ heißt steilster Abstieg (*steepest descent*).

Notwendige Optimalitätsbedingung 2. Ordnung

Satz (Notwendige Optimalitätsbedingung 2. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Min. einer hinr. glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, gilt

$$\nabla f(\bar{x}) = 0 \quad \textit{und} \quad \nabla^2 f(\bar{x}) \succeq 0.$$

Sonst gibt es ein $h \in \mathbb{R}^n$ mit $h^T \nabla^2 f(\bar{x}) h < 0$, Taylor ergibt für kleine α

$$f(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \underbrace{\nabla f(\bar{x})^T h}_{=0 \ (\nabla f(\bar{x})=0)} + \frac{\alpha^2}{2} \underbrace{h^T \nabla^2 f(\bar{x}) h}_{<0} + \mathbf{o}(\alpha^2) < f(\bar{x}).$$

Notwendige Optimalitätsbedingung 2. Ordnung

Satz (Notwendige Optimalitätsbedingung 2. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Min. einer hinr. glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, gilt

$$\nabla f(\bar{x}) = 0 \quad \textit{und} \quad \nabla^2 f(\bar{x}) \succeq 0.$$

Sonst gibt es ein $h \in \mathbb{R}^n$ mit $h^T \nabla^2 f(\bar{x}) h < 0$, Taylor ergibt für kleine α

$$f(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \underbrace{\nabla f(\bar{x})^T h}_{=0 \ (\nabla f(\bar{x})=0)} + \tfrac{\alpha^2}{2} \underbrace{h^T \nabla^2 f(\bar{x}) h}_{<0} + \mathbf{o}(\alpha^2) < f(\bar{x}).$$

Die Bedingung ist wieder nur notwendig und i.A. nicht hinreichend.

Bsp: $f(x, y) = x^2 - y^4$, $\nabla^2 f(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & -12y^2 \end{bmatrix}$ für $(x, y) = (0, 0)$.

Notwendige Optimalitätsbedingung 2. Ordnung

Satz (Notwendige Optimalitätsbedingung 2. Ordnung)

Ist $\bar{x} \in \mathbb{R}^n$ ein lokales Min. einer hinr. glatten Funktion $f : \mathbb{R}^n \to \mathbb{R}$, gilt

$$\nabla f(\bar{x}) = 0 \quad \textit{und} \quad \nabla^2 f(\bar{x}) \succeq 0.$$

Sonst gibt es ein $h \in \mathbb{R}^n$ mit $h^T \nabla^2 f(\bar{x}) h < 0$, Taylor ergibt für kleine α

$$f(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \underbrace{\nabla f(\bar{x})^T h}_{=0 \ (\nabla f(\bar{x})=0)} + \tfrac{\alpha^2}{2} \underbrace{h^T \nabla^2 f(\bar{x}) h}_{<0} + \mathbf{o}(\alpha^2) < f(\bar{x}).$$

Die Bedingung ist wieder nur notwendig und i.A. nicht hinreichend.

Bsp: $f(x, y) = x^2 - y^4$, $\nabla^2 f(x, y) = \begin{bmatrix} 2 & 0 \\ 0 & -12y^2 \end{bmatrix}$ für $(x, y) = (0, 0)$.

Konsequenz für Optimierungsverfahren:
Ist zwar $\nabla f(\bar{x}) = 0$ aber $\lambda_{\min}(\nabla^2 f(\bar{x})) < 0$, so kann f in Richtung eines Eigenvektors zu $\lambda_{\min}$ verbessert werden.

Hinreichende Optimalitätsbedingung 2. Ordnung

Satz (Hinreichende Optimalitätsbedingung 2. Ordnung)

Gilt für ein $\bar{x} \in \mathbb{R}^n$ *sowohl* $\nabla f(\bar{x}) = 0$ *als auch* $\nabla^2 f(\bar{x}) \succ 0$,
so ist $\bar{x}$ *ein lokales Minimum von* f.

Denn für beliebiges $h \in \mathbb{R}^n \setminus \{0\}$ und α klein genug gilt mit Taylor

$$f(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \underbrace{\nabla f(\bar{x})^T h}_{=0 \ (\nabla f(\bar{x})=0)} + \frac{\alpha^2}{2} \underbrace{h^T \nabla^2 f(\bar{x}) h}_{>0} + \mathbf{o}(\alpha^2) > f(\bar{x}).$$

Hinreichende Optimalitätsbedingung 2. Ordnung

Satz (Hinreichende Optimalitätsbedingung 2. Ordnung)

Gilt für ein $\bar{x} \in \mathbb{R}^n$ *sowohl* $\nabla f(\bar{x}) = 0$ *als auch* $\nabla^2 f(\bar{x}) \succ 0$,
so ist $\bar{x}$ *ein lokales Minimum von* f.

Denn für beliebiges $h \in \mathbb{R}^n \setminus \{0\}$ und α klein genug gilt mit Taylor

$$f(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \underbrace{\nabla f(\bar{x})^T h}_{=0 \ (\nabla f(\bar{x})=0)} + \frac{\alpha^2}{2} \underbrace{h^T \nabla^2 f(\bar{x}) h}_{>0} + \mathbf{o}(\alpha^2) > f(\bar{x}).$$

Die Bedingung ist hinreichend, aber nicht notwendig: $f(x) = x^4$ in $x = 0$.
In der Praxis ist sie erstaunlich oft erfüllt.

Hinreichende Optimalitätsbedingung 2. Ordnung

Satz (Hinreichende Optimalitätsbedingung 2. Ordnung)

Gilt für ein $\bar{x} \in \mathbb{R}^n$ *sowohl* $\nabla f(\bar{x}) = 0$ *als auch* $\nabla^2 f(\bar{x}) \succ 0$,
so ist $\bar{x}$ *ein lokales Minimum von* f.

Denn für beliebiges $h \in \mathbb{R}^n \setminus \{0\}$ und α klein genug gilt mit Taylor

$$f(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \underbrace{\nabla f(\bar{x})^T h}_{=0 \ (\nabla f(\bar{x})=0)} + \frac{\alpha^2}{2} \underbrace{h^T \nabla^2 f(\bar{x}) h}_{>0} + \mathbf{o}(\alpha^2) > f(\bar{x}).$$

Die Bedingung ist hinreichend, aber nicht notwendig: $f(x) = x^4$ in $x = 0$.
In der Praxis ist sie erstaunlich oft erfüllt.

Konsequenz für Optimierungsverfahren:
In der Nähe eines lokalen Minimums sieht die Funktion wie eine konvexe quadratische Funktion aus, das quadratische Modell ist dort eine gute Approximation!

Bemerkungen

- In Optimalitätsbedingungen für Maximierungsprobleme muss man nur $\nabla^2 f \succeq 0 (\succ 0)$ durch $\nabla^2 f \preceq 0 (\prec 0)$ ersetzen, der Rest bleibt gleich.

Bemerkungen

- In Optimalitätsbedingungen für Maximierungsprobleme muss man nur $\nabla^2 f \succeq 0 (\succ 0)$ durch $\nabla^2 f \preceq 0 (\prec 0)$ ersetzen, der Rest bleibt gleich.
- Stationäre Punkte sind entweder lokale Minima, lokale Maxima oder Sattelpunkte (manche Richtungen führen aufwärts, manche abwärts).

Bemerkungen

- In Optimalitätsbedingungen für Maximierungsprobleme muss man nur $\nabla^2 f \succeq 0 (\succ 0)$ durch $\nabla^2 f \preceq 0 (\prec 0)$ ersetzen, der Rest bleibt gleich.
- Stationäre Punkte sind entweder lokale Minima, lokale Maxima oder Sattelpunkte (manche Richtungen führen aufwärts, manche abwärts).
- Alle Optimierungsverfahren versuchen eine Folge zu erzeugen, die gegen einen stationären Punkt konvergiert. In der Nähe eines stationären Punktes soll die Konvergenz möglichst quadratisch sein, wie beim Newton-Verfahren.

Inhalt

6.3 Das Newton-Verfahren

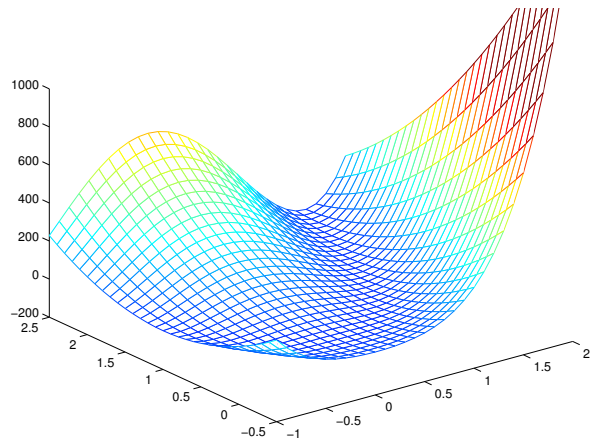
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

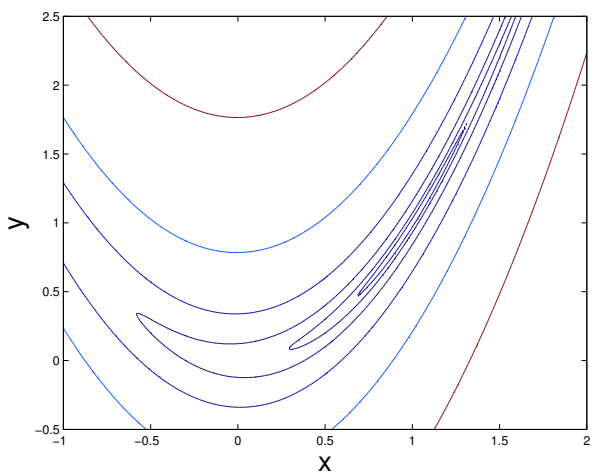
Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$





6.3 Das Newton-Verfahren

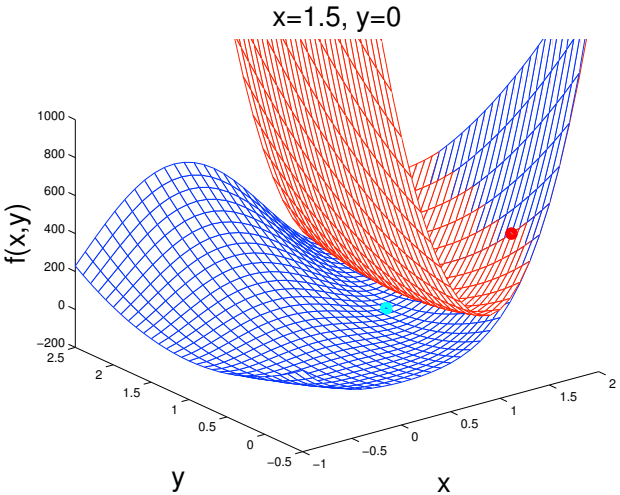
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

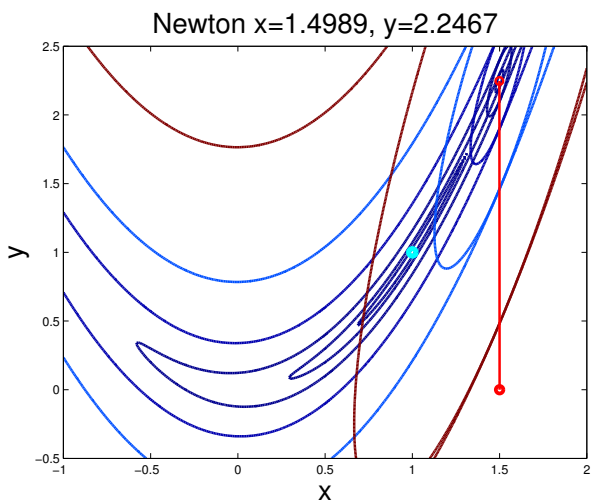
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



Das Newton-Verfahren wählt als nächsten Punkt $x^{(k+1)} = x^{(k)} + h$ den stationären Punkt des quadratischen Modells (= das Minimum falls $\nabla^2 f(x^{(k)}) \succ 0$),

$$h_N^{(k)} := -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \quad [= -Q^{-1} q]$$

$h_N^{(k)}$ ist der Newton-Schritt und ist für $\nabla^2 f(x^{(k)})$ regulär definiert.



6.3 Das Newton-Verfahren

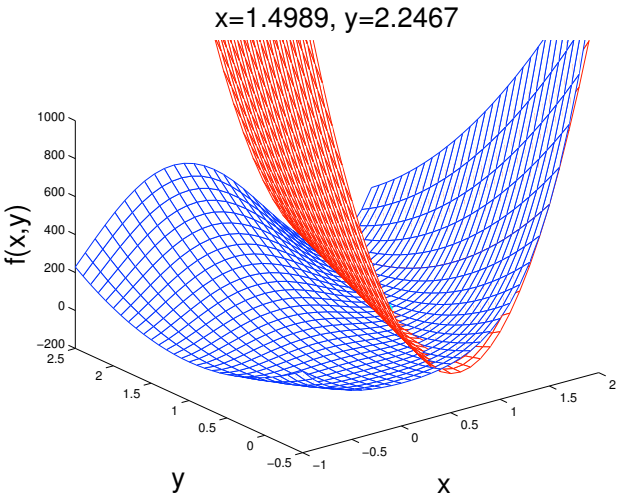
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

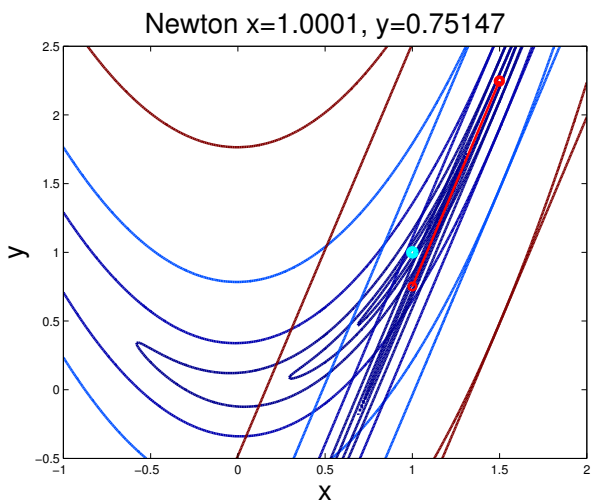
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



Das Newton-Verfahren wählt als nächsten Punkt $x^{(k+1)} = x^{(k)} + h$ den stationären Punkt des quadratischen Modells (= das Minimum falls $\nabla^2 f(x^{(k)}) \succ 0$),

$$h_N^{(k)} := -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \quad [= -Q^{-1} q]$$

$h_N^{(k)}$ ist der Newton-Schritt und ist für $\nabla^2 f(x^{(k)})$ regulär definiert.



6.3 Das Newton-Verfahren

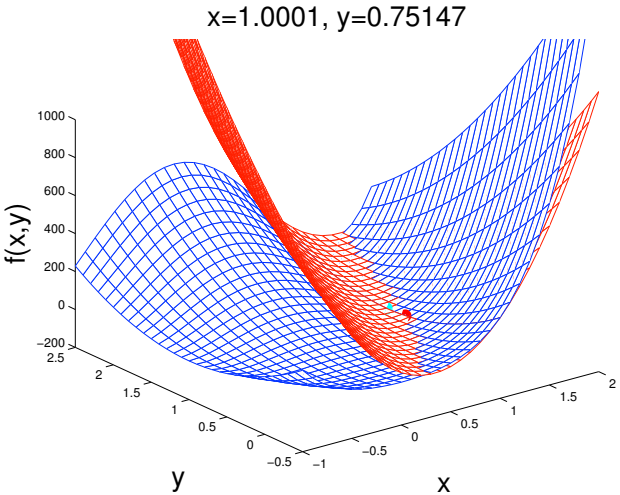
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

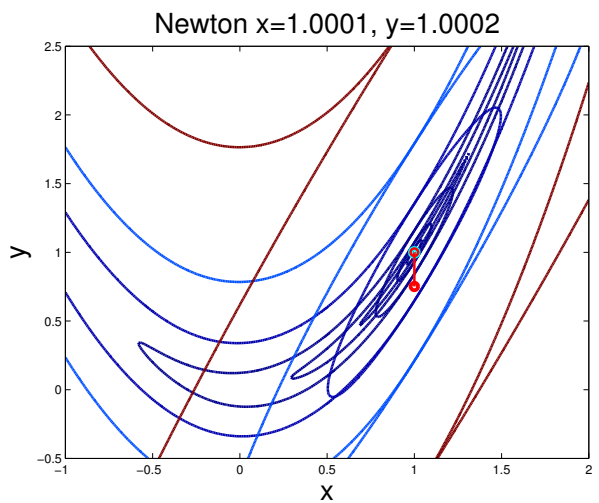
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



Das Newton-Verfahren wählt als nächsten Punkt $x^{(k+1)} = x^{(k)} + h$ den stationären Punkt des quadratischen Modells (= das Minimum falls $\nabla^2 f(x^{(k)}) \succ 0$),

$$h_N^{(k)} := -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \quad [= -Q^{-1} q]$$

$h_N^{(k)}$ ist der Newton-Schritt und ist für $\nabla^2 f(x^{(k)})$ regulär definiert.



6.3 Das Newton-Verfahren

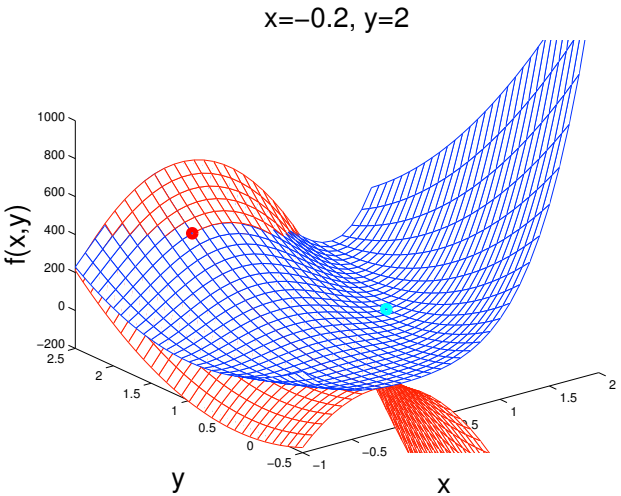
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

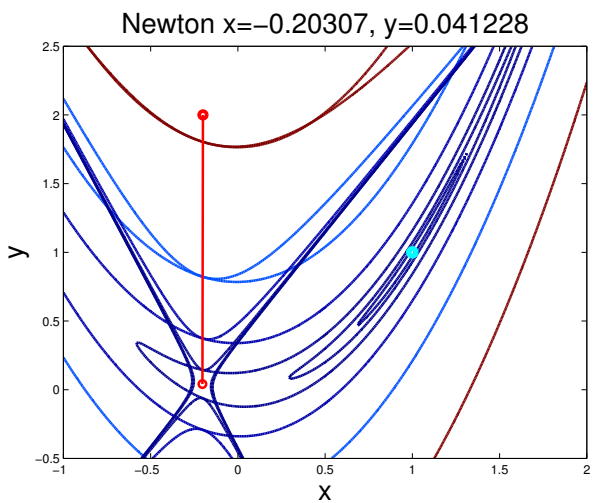
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



Das Newton-Verfahren wählt als nächsten Punkt $x^{(k+1)} = x^{(k)} + h$ den stationären Punkt des quadratischen Modells (= das Minimum falls $\nabla^2 f(x^{(k)}) \succ 0$),

$$h_N^{(k)} := -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \quad [= -Q^{-1} q]$$

$h_N^{(k)}$ ist der Newton-Schritt und ist für $\nabla^2 f(x^{(k)})$ regulär definiert.



6.3 Das Newton-Verfahren

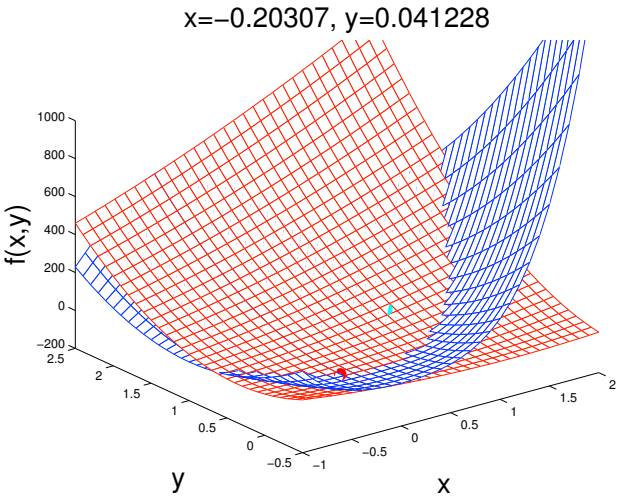
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

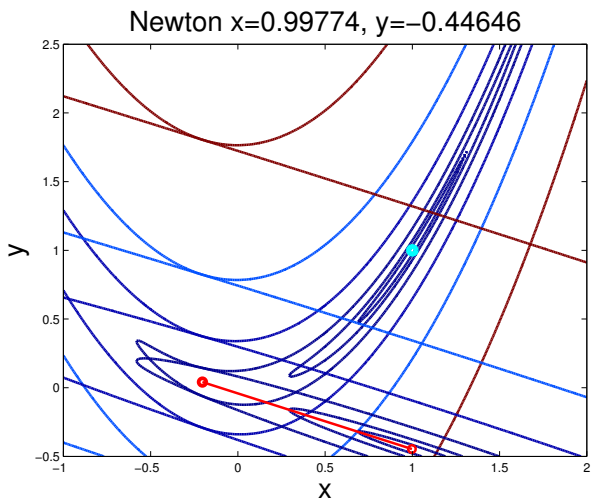
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



Das Newton-Verfahren wählt als nächsten Punkt $x^{(k+1)} = x^{(k)} + h$ den stationären Punkt des quadratischen Modells (= das Minimum falls $\nabla^2 f(x^{(k)}) \succ 0$),

$$h_N^{(k)} := -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \quad [= -Q^{-1} q]$$

$h_N^{(k)}$ ist der Newton-Schritt und ist für $\nabla^2 f(x^{(k)})$ regulär definiert.



6.3 Das Newton-Verfahren

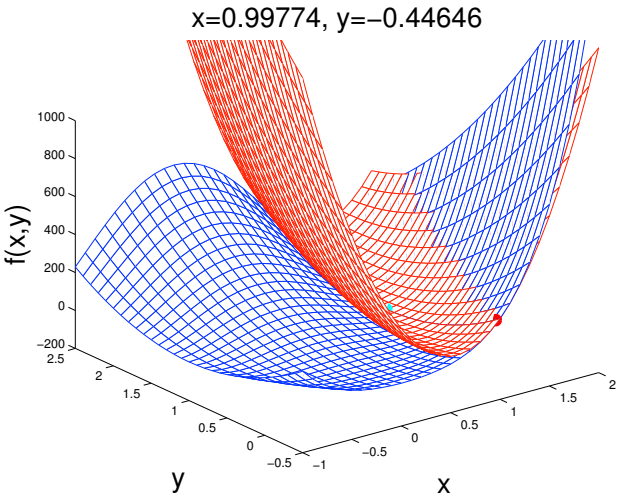
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

Für jedes $x^{(k)}$ nahe bei x^* ist dann das quadratische Modell streng konvex,

$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \tfrac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

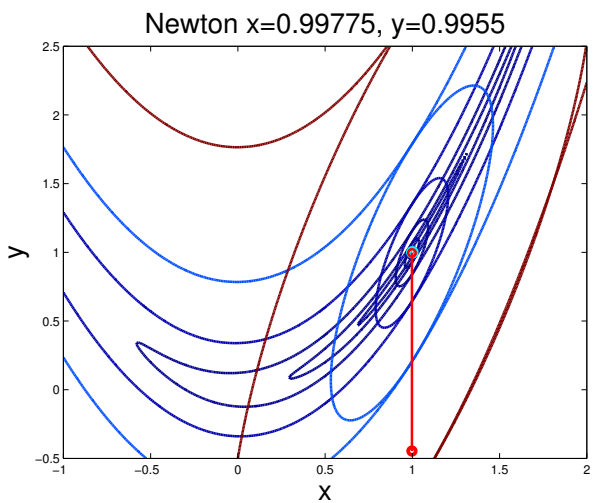
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



Das Newton-Verfahren wählt als nächsten Punkt $x^{(k+1)} = x^{(k)} + h$ den stationären Punkt des quadratischen Modells (= das Minimum falls $\nabla^2 f(x^{(k)}) \succ 0$),

$$h_N^{(k)} := -\nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)}). \quad [= -Q^{-1} q]$$

$h_N^{(k)}$ ist der Newton-Schritt und ist für $\nabla^2 f(x^{(k)})$ regulär definiert.



Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

Sei f zweimal stetig differenzierbar, x^* ein lokales Minimum, das die hinreichenden Optimalitätsbedingungen erfüllt, $\nabla^2 f$ sei Lipschitz-stetig in einer Umgebung von x^*. Für jeden nahe genug an x^* gelegenen Startpunkt $x^{(0)}$ gilt für die Folge

$$x^{(k+1)} := x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

1. *Die $x^{(k)}$ konvergieren quadratisch gegen x^*, d.h.,*

$$\exists K \in \mathbb{N}, c > 0 : \|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2 \text{ für } k > K.$$

2. *Die Gradienten-Normen $\|\nabla f(x^{(k)})\|$ konvergieren quadratisch gegen 0.*

Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

Sei f zweimal stetig differenzierbar, x^* ein lokales Minimum, das die hinreichenden Optimalitätsbedingungen erfüllt, $\nabla^2 f$ sei Lipschitz-stetig in einer Umgebung von x^*. Für jeden nahe genug an x^* gelegenen Startpunkt $x^{(0)}$ gilt für die Folge

$$x^{(k+1)} := x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

1. Die $x^{(k)}$ konvergieren quadratisch gegen x^*, d.h.,

$$\exists K \in \mathbb{N}, c > 0 : \|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2 \text{ für } k > K.$$

2. Die Gradienten-Normen $\|\nabla f(x^{(k)})\|$ konvergieren quadratisch gegen 0.

- Der Satz gilt nur lokal und gibt keine Konvergenzgarantie für weit entfernte Startpunkte (das kann selbst für konvexes f fehlschlagen).

Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

Sei f zweimal stetig differenzierbar, x^* ein lokales Minimum, das die hinreichenden Optimalitätsbedingungen erfüllt, $\nabla^2 f$ sei Lipschitz-stetig in einer Umgebung von x^*. Für jeden nahe genug an x^* gelegenen Startpunkt $x^{(0)}$ gilt für die Folge

$$x^{(k+1)} := x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

1. Die $x^{(k)}$ konvergieren quadratisch gegen x^*, d.h.,

$$\exists K \in \mathbb{N}, c > 0 : \|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2 \text{ für } k > K.$$

2. Die Gradienten-Normen $\|\nabla f(x^{(k)})\|$ konvergieren quadratisch gegen 0.

- Der Satz gilt nur lokal und gibt keine Konvergenzgarantie für weit entfernte Startpunkte (das kann selbst für konvexes f fehlschlagen).
- Die Funktionswerte $f(x^{(k)})$ müssen keineswegs monoton fallen.

Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

Sei f zweimal stetig differenzierbar, x^* ein lokales Minimum, das die hinreichenden Optimalitätsbedingungen erfüllt, $\nabla^2 f$ sei Lipschitz-stetig in einer Umgebung von x^*. Für jeden nahe genug an x^* gelegenen Startpunkt $x^{(0)}$ gilt für die Folge

$$x^{(k+1)} := x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

1. Die $x^{(k)}$ konvergieren quadratisch gegen x^*, d.h.,

$$\exists K \in \mathbb{N}, c > 0 : \|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2 \text{ für } k > K.$$

2. Die Gradienten-Normen $\|\nabla f(x^{(k)})\|$ konvergieren quadratisch gegen 0.

- Der Satz gilt nur lokal und gibt keine Konvergenzgarantie für weit entfernte Startpunkte (das kann selbst für konvexes f fehlschlagen).
- Die Funktionswerte $f(x^{(k)})$ müssen keineswegs monoton fallen.
- Startet die Folge in der Nähe eines anderen stationären Punktes (Maximum oder Sattelpunkt), konvergiert die Folge zu diesem.

Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

Sei f zweimal stetig differenzierbar, x^* ein lokales Minimum, das die hinreichenden Optimalitätsbedingungen erfüllt, $\nabla^2 f$ sei Lipschitz-stetig in einer Umgebung von x^*. Für jeden nahe genug an x^* gelegenen Startpunkt $x^{(0)}$ gilt für die Folge

$$x^{(k+1)} := x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

1. Die $x^{(k)}$ konvergieren quadratisch gegen x^*, d.h.,

$$\exists K \in \mathbb{N}, c > 0 : \|x^{(k+1)} - x^*\| \leq c\|x^{(k)} - x^*\|^2 \text{ für } k > K.$$

2. Die Gradienten-Normen $\|\nabla f(x^{(k)})\|$ konvergieren quadratisch gegen 0.

- Der Satz gilt nur lokal und gibt keine Konvergenzgarantie für weit entfernte Startpunkte (das kann selbst für konvexes f fehlschlagen).
- Die Funktionswerte $f(x^{(k)})$ müssen keineswegs monoton fallen.
- Startet die Folge in der Nähe eines anderen stationären Punktes (Maximum oder Sattelpunkt), konvergiert die Folge zu diesem.
- Kommt man ins Gebiet quadratischer Konvergenz, verdoppelt sich pro Iteration die Anzahl korrekt berechneter Stellen.

Bemerkungen

- Um von lokalen zu „globalen“ Minimierungsverfahren zu kommen, wird in Globalisierungsstrategien $f(x^{(k+1)}) < f(x^{(k)})$ gefordert, siehe z.B. Line-Search- und Trust-Region-Verfahren.

Bemerkungen

- Um von lokalen zu „globalen“ Minimierungsverfahren zu kommen, wird in Globalisierungsstrategien $f(x^{(k+1)}) < f(x^{(k)})$ gefordert, siehe z.B. Line-Search- und Trust-Region-Verfahren.
- Die Bestimmung von $\nabla^2 f$ ist oft zu aufwendig bzgl. Rechenzeit und Speicherbedarf. Meist setzen Verfahren daher nur Orakel 1. Ordnung voraus und approximieren $\nabla^2 f$ lokal zur Konvergenzverbesserung. Damit sind aber die Bedingungen 2. Ordnung nicht gut überprüfbar.

Bemerkungen

- Um von lokalen zu „globalen“ Minimierungsverfahren zu kommen, wird in Globalisierungsstrategien $f(x^{(k+1)}) < f(x^{(k)})$ gefordert, siehe z.B. Line-Search- und Trust-Region-Verfahren.
- Die Bestimmung von $\nabla^2 f$ ist oft zu aufwendig bzgl. Rechenzeit und Speicherbedarf. Meist setzen Verfahren daher nur Orakel 1. Ordnung voraus und approximieren $\nabla^2 f$ lokal zur Konvergenzverbesserung. Damit sind aber die Bedingungen 2. Ordnung nicht gut überprüfbar.
- Ein nichtlineares Optimierungsverfahren heißt global konvergent, wenn es für jede nach unten beschränkte Funktion und jeden Startpunkt $x^{(0)}$ eine Punktfolge $x^{(k)}$ mit $\|\nabla f(x^{(k)})\| \to 0$ erzeugt. [Das kann auch $\|x^{(k)}\| \to \infty$ bedeuten, etwa für $f(x) = \frac{1}{x}$.]

Bemerkungen

- Um von lokalen zu „globalen“ Minimierungsverfahren zu kommen, wird in Globalisierungsstrategien $f(x^{(k+1)}) < f(x^{(k)})$ gefordert, siehe z.B. Line-Search- und Trust-Region-Verfahren.
- Die Bestimmung von $\nabla^2 f$ ist oft zu aufwendig bzgl. Rechenzeit und Speicherbedarf. Meist setzen Verfahren daher nur Orakel 1. Ordnung voraus und approximieren $\nabla^2 f$ lokal zur Konvergenzverbesserung. Damit sind aber die Bedingungen 2. Ordnung nicht gut überprüfbar.
- Ein nichtlineares Optimierungsverfahren heißt global konvergent, wenn es für jede nach unten beschränkte Funktion und jeden Startpunkt $x^{(0)}$ eine Punktfolge $x^{(k)}$ mit $\|\nabla f(x^{(k)})\| \to 0$ erzeugt. [Das kann auch $\|x^{(k)}\| \to \infty$ bedeuten, etwa für $f(x) = \frac{1}{x}$.]
- Durch die Bedingung $f(x^{(k+1)}) < f(x^{(k)})$ hoffen die Verfahren im Konvergenzfall ein Minimum gefunden zu haben, manchmal ist es jedoch ein Sattelpunkt. Für den Anwender ist das meist leicht zu erkennen, für das Verfahren nicht $\to$ besseren Startpunkt wählen.

Bemerkungen

- Um von lokalen zu „globalen“ Minimierungsverfahren zu kommen, wird in Globalisierungsstrategien $f(x^{(k+1)}) < f(x^{(k)})$ gefordert, siehe z.B. Line-Search- und Trust-Region-Verfahren.
- Die Bestimmung von $\nabla^2 f$ ist oft zu aufwendig bzgl. Rechenzeit und Speicherbedarf. Meist setzen Verfahren daher nur Orakel 1. Ordnung voraus und approximieren $\nabla^2 f$ lokal zur Konvergenzverbesserung. Damit sind aber die Bedingungen 2. Ordnung nicht gut überprüfbar.
- Ein nichtlineares Optimierungsverfahren heißt global konvergent, wenn es für jede nach unten beschränkte Funktion und jeden Startpunkt $x^{(0)}$ eine Punktfolge $x^{(k)}$ mit $\|\nabla f(x^{(k)})\| \to 0$ erzeugt. [Das kann auch $\|x^{(k)}\| \to \infty$ bedeuten, etwa für $f(x) = \frac{1}{x}$.]
- Durch die Bedingung $f(x^{(k+1)}) < f(x^{(k)})$ hoffen die Verfahren im Konvergenzfall ein Minimum gefunden zu haben, manchmal ist es jedoch ein Sattelpunkt. Für den Anwender ist das meist leicht zu erkennen, für das Verfahren nicht $\to$ besseren Startpunkt wählen.
- Wir nutzen die Kurzschreibweise f_k für $f(x^{(k)})$, ∇f_k für $\nabla f(x^{(k)})$ und $\nabla^2 f_k$ für $\nabla^2 f(x^{(k)})$.

Inhalt

6.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

1. Rufe das Orakel für $x^{(k)}$ auf $\rightarrow f_k$, ∇f_k, (vielleicht auch $\nabla^2 f_k$).

6.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

1. Rufe das Orakel für $x^{(k)}$ auf $\rightarrow f_k$, ∇f_k, (vielleicht auch $\nabla^2 f_k$).
2. Ist $\|\nabla f_k\|$ klein genug, STOP.

6.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

1. Rufe das Orakel für $x^{(k)}$ auf $\rightarrow f_k$, ∇f_k, (vielleicht auch $\nabla^2 f_k$).
2. Ist $\|\nabla f_k\|$ klein genug, STOP.
3. Abstiegsrichtung: Wähle $h^{(k)} \in \mathbb{R}^n$ mit $\nabla f_k^T h^{(k)} < 0$.

6.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

1. Rufe das Orakel für $x^{(k)}$ auf $\rightarrow f_k$, ∇f_k, (vielleicht auch $\nabla^2 f_k$).
2. Ist $\|\nabla f_k\|$ klein genug, STOP.
3. Abstiegsrichtung: Wähle $h^{(k)} \in \mathbb{R}^n$ mit $\nabla f_k^T h^{(k)} < 0$.
4. Line-Search: Finde eine Schrittweite $\alpha_k \geq 0$ mit
   $f(x^{(k)} + \alpha_k h^{(k)})$ „ausreichend“ kleiner als f_k

6.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

1. Rufe das Orakel für $x^{(k)}$ auf $\rightarrow f_k$, ∇f_k, (vielleicht auch $\nabla^2 f_k$).
2. Ist $\|\nabla f_k\|$ klein genug, STOP.
3. Abstiegsrichtung: Wähle $h^{(k)} \in \mathbb{R}^n$ mit $\nabla f_k^T h^{(k)} < 0$.
4. Line-Search: Finde eine Schrittweite $\alpha_k \geq 0$ mit
   $f(x^{(k)} + \alpha_k h^{(k)})$ „ausreichend“ kleiner als f_k
5. Setze $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$, $k \leftarrow k+1$, gehe zu 1.

6.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

1. Rufe das Orakel für $x^{(k)}$ auf $\rightarrow f_k$, ∇f_k, (vielleicht auch $\nabla^2 f_k$).
2. Ist $\|\nabla f_k\|$ klein genug, STOP.
3. Abstiegsrichtung: Wähle $h^{(k)} \in \mathbb{R}^n$ mit $\nabla f_k^T h^{(k)} < 0$.
4. Line-Search: Finde eine Schrittweite $\alpha_k \geq 0$ mit
   $f(x^{(k)} + \alpha_k h^{(k)})$ „ausreichend“ kleiner als f_k
5. Setze $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$, $k \leftarrow k+1$, gehe zu 1.

Zwei Hauptaufgaben:

- Bestimmung einer Abstiegsrichtung
- Bestimmung einer Schrittweite (Line-Search)

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Die meisten Algorithmen bestimmen h für ein $B \succ 0$ in der Form

$$h := -B^{-1}\nabla f(\bar{x}), \quad \text{denn } \nabla f(\bar{x})^T h = -\underbrace{\nabla f(\bar{x})^T B^{-1} \nabla f(\bar{x})}_{>0} < 0.$$

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Die meisten Algorithmen bestimmen h für ein $B \succ 0$ in der Form

$$h := -B^{-1}\nabla f(\bar{x}), \quad \text{denn } \nabla f(\bar{x})^T h = -\underbrace{\nabla f(\bar{x})^T B^{-1} \nabla f(\bar{x})}_{>0} < 0.$$

Beispiele (s. später zu Vor- und Nachteilen):

- $B = I$: steilster Abstieg $h = -\nabla f(\bar{x})$ (steepest descent).

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Die meisten Algorithmen bestimmen h für ein $B \succ 0$ in der Form

$$h := -B^{-1}\nabla f(\bar{x}), \quad \text{denn } \nabla f(\bar{x})^T h = -\underbrace{\nabla f(\bar{x})^T B^{-1} \nabla f(\bar{x})}_{>0} < 0.$$

Beispiele (s. später zu Vor- und Nachteilen):

- $B = I$: steilster Abstieg $h = -\nabla f(\bar{x})$ (steepest descent).
- $B = \nabla^2 f(\bar{x})$ Newton-Richtung (Abstiegsrichtung für $\nabla^2 f(\bar{x}) \succ 0$)

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Die meisten Algorithmen bestimmen h für ein $B \succ 0$ in der Form

$$h := -B^{-1}\nabla f(\bar{x}), \quad \text{denn } \nabla f(\bar{x})^T h = -\underbrace{\nabla f(\bar{x})^T B^{-1} \nabla f(\bar{x})}_{>0} < 0.$$

Beispiele (s. später zu Vor- und Nachteilen):

- $B = I$: steilster Abstieg $h = -\nabla f(\bar{x})$ (steepest descent).
- $B = \nabla^2 f(\bar{x})$ Newton-Richtung (Abstiegsrichtung für $\nabla^2 f(\bar{x}) \succ 0$)
- $B = [\nabla^2 f(\bar{x}) + \lambda I] \succ 0$ modifizierte Newton-Richtung

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Die meisten Algorithmen bestimmen h für ein $B \succ 0$ in der Form

$$h := -B^{-1}\nabla f(\bar{x}), \quad \text{denn } \nabla f(\bar{x})^T h = -\underbrace{\nabla f(\bar{x})^T B^{-1} \nabla f(\bar{x})}_{>0} < 0.$$

Beispiele (s. später zu Vor- und Nachteilen):

- $B = I$: steilster Abstieg $h = -\nabla f(\bar{x})$ (steepest descent).
- $B = \nabla^2 f(\bar{x})$ Newton-Richtung (Abstiegsrichtung für $\nabla^2 f(\bar{x}) \succ 0$)
- $B = [\nabla^2 f(\bar{x}) + \lambda I] \succ 0$ modifizierte Newton-Richtung
- $B \succ 0$ als Approximation von $\nabla^2 f(\bar{x})$ Quasi-Newton-Richtung

Abstiegsrichtung (für $\bar{x}$ mit $\nabla f(\bar{x}) \neq 0$)

Eine Richtung $h \in \mathbb{R}^n$ heißt Abstiegsrichtung für f in $\bar{x}$, falls $\nabla f(\bar{x})^T h < 0$.

Die meisten Algorithmen bestimmen h für ein $B \succ 0$ in der Form

$$h := -B^{-1}\nabla f(\bar{x}), \quad \text{denn } \nabla f(\bar{x})^T h = -\underbrace{\nabla f(\bar{x})^T B^{-1} \nabla f(\bar{x})}_{>0} < 0.$$

Beispiele (s. später zu Vor- und Nachteilen):

- $B = I$: steilster Abstieg $h = -\nabla f(\bar{x})$ (steepest descent).
- $B = \nabla^2 f(\bar{x})$ Newton-Richtung (Abstiegsrichtung für $\nabla^2 f(\bar{x}) \succ 0$)
- $B = [\nabla^2 f(\bar{x}) + \lambda I] \succ 0$ modifizierte Newton-Richtung
- $B \succ 0$ als Approximation von $\nabla^2 f(\bar{x})$ Quasi-Newton-Richtung

---

Für globale Konvergenz der Line-Search Verfahren ist nur wichtig, dass die Richtungen nicht orthogonal zur steilsten Abstiegsrichtung werden:

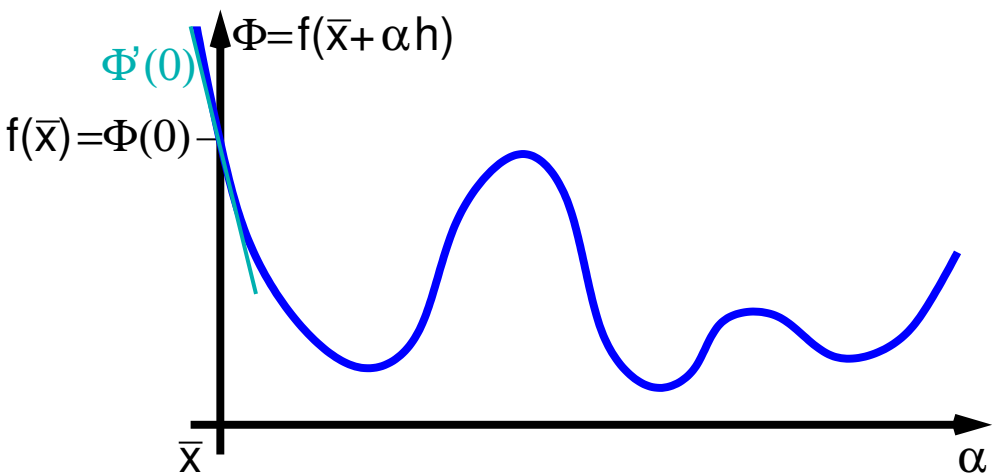
$$\exists \delta > 0 : \frac{-\nabla f_k}{\|\nabla f_k\|}^T \frac{h^{(k)}}{\|h^{(k)}\|} = \cos \angle(-\nabla f_k, h^{(k)}) \geq \delta > 0 \text{ für } k > 0.$$

Das ist erfüllt, falls $\frac{\lambda_{\max}(B_k)}{\lambda_{\min}(B_k)} < \kappa$ für ein $\kappa > 0$ und gilt z.B. für $B = I$ oder Newton-Richtung in der Nähe von x^* unter den Vor. des Newton-Satzes.

Line-Search für Abstiegsrichtung h

Bestimme Schrittweite $\bar{\alpha} \geq 0$ als **Näherung** zu $\min_{\alpha \geq 0} \Phi(\alpha) := f(\bar{x} + \alpha h)$.

Berechnung von $\bar{\alpha} \in \operatorname{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (exakter Line-Search) wäre sinnlos aufwendig, da die Richtung h meist weit am Optimum vorbeiführt.

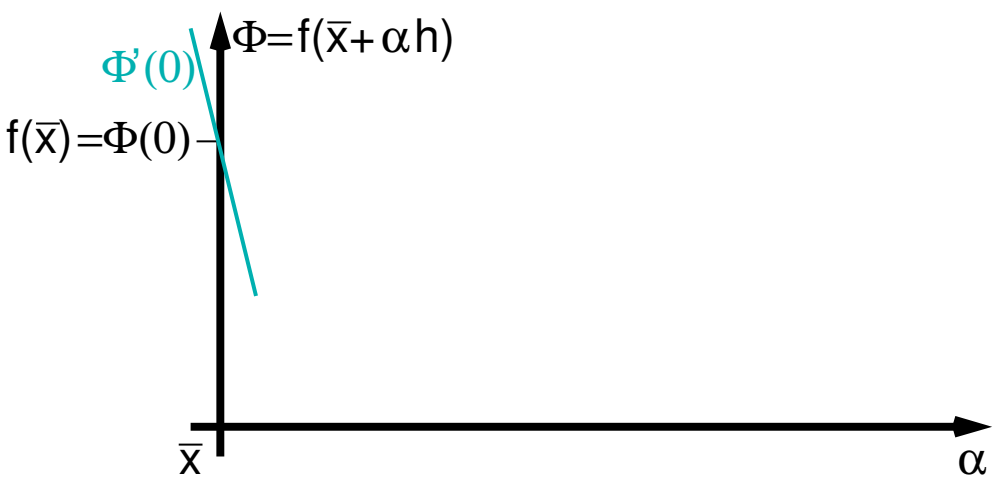


Line-Search für Abstiegsrichtung h

Bestimme Schrittweite $\bar{\alpha} \geq 0$ als **Näherung** zu $\min_{\alpha \geq 0} \Phi(\alpha) := f(\bar{x} + \alpha h)$.

Berechnung von $\bar{\alpha} \in \operatorname{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (exakter Line-Search) wäre sinnlos aufwendig, da die Richtung h meist weit am Optimum vorbeiführt.

Anfangs sehr wenig Information: $\Phi(0) = f(\bar{x})$, Ableitung $\Phi'(0) = \nabla f(\bar{x})^T h$



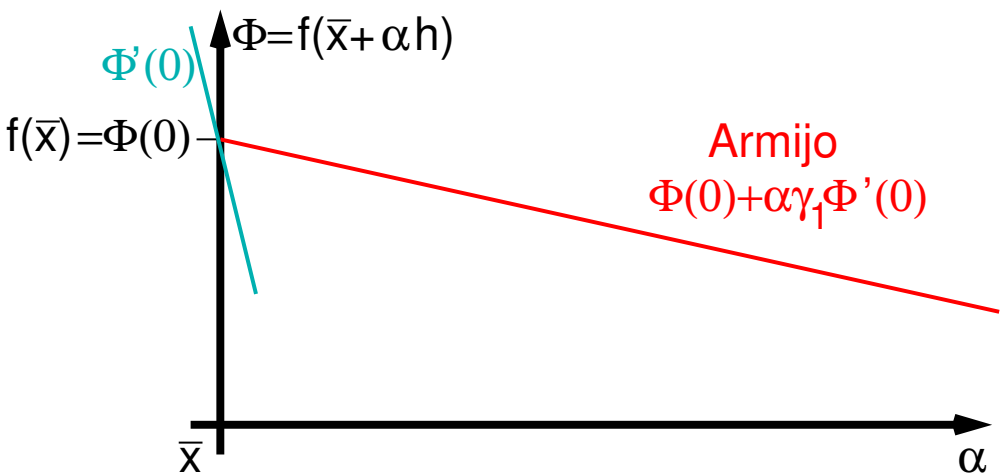
Line-Search für Abstiegsrichtung h

Bestimme Schrittweite $\bar{\alpha} \geq 0$ als **Näherung** zu $\min_{\alpha \geq 0} \Phi(\alpha) := f(\bar{x} + \alpha h)$.

Berechnung von $\bar{\alpha} \in \operatorname{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (exakter Line-Search) wäre sinnlos aufwendig, da die Richtung h meist weit am Optimum vorbeiführt.

Anfangs sehr wenig Information: $\Phi(0) = f(\bar{x})$, Ableitung $\Phi'(0) = \nabla f(\bar{x})^T h$
Ein $\bar{\alpha}$ mit ausreichendem Abstieg (sufficient decrease) erfüllt:

1. Mindestanteil $0 < \gamma_1 < 1$ an dem durch $\Phi'(0)$ „versprochenen“Abstieg:
   $\Phi(\bar{\alpha}) \leq \Phi(0) + \bar{\alpha}\gamma_1\Phi'(0)$ (Armijo-Bedingung) [für kleine α erfüllt]



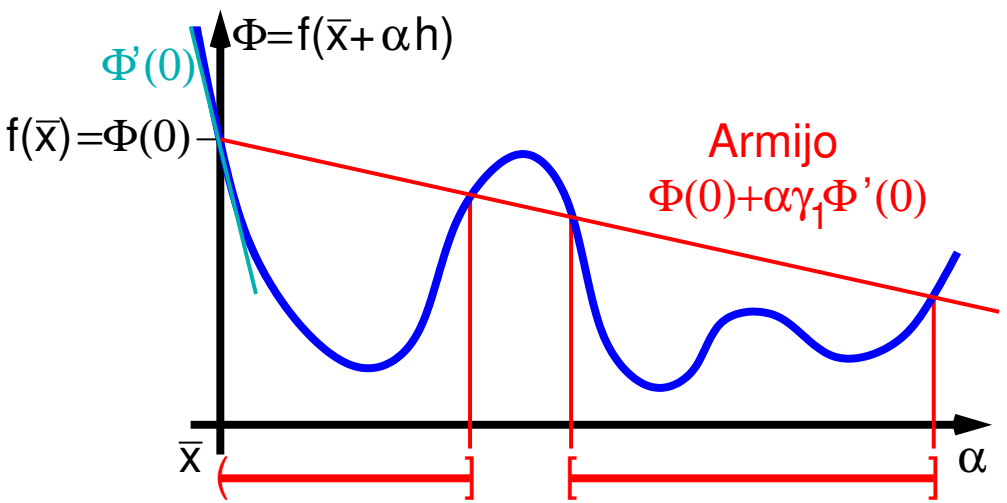
Line-Search für Abstiegsrichtung h

Bestimme Schrittweite $\bar{\alpha} \geq 0$ als **Näherung** zu $\min_{\alpha \geq 0} \Phi(\alpha) := f(\bar{x} + \alpha h)$.

Berechnung von $\bar{\alpha} \in \operatorname{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (exakter Line-Search) wäre sinnlos aufwendig, da die Richtung h meist weit am Optimum vorbeiführt.

Anfangs sehr wenig Information: $\Phi(0) = f(\bar{x})$, Ableitung $\Phi'(0) = \nabla f(\bar{x})^T h$
Ein $\bar{\alpha}$ mit ausreichendem Abstieg (sufficient decrease) erfüllt:

1. Mindestanteil $0 < \gamma_1 < 1$ an dem durch $\Phi'(0)$ „versprochenen“Abstieg:
   $\Phi(\bar{\alpha}) \leq \Phi(0) + \bar{\alpha}\gamma_1\Phi'(0)$ (Armijo-Bedingung) [für kleine α erfüllt]



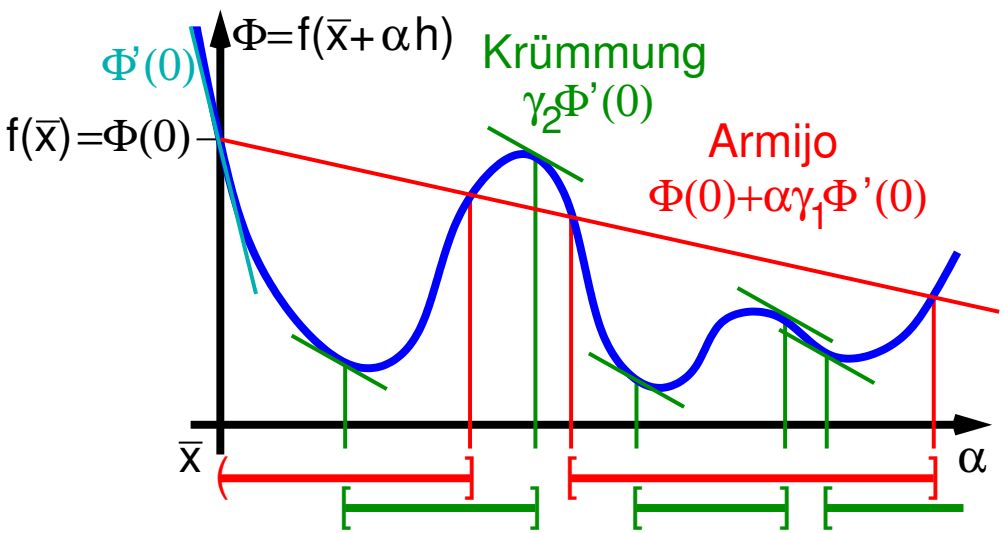
Line-Search für Abstiegsrichtung h

Bestimme Schrittweite $\bar{\alpha} \geq 0$ als **Näherung** zu $\min_{\alpha \geq 0} \Phi(\alpha) := f(\bar{x} + \alpha h)$.

Berechnung von $\bar{\alpha} \in \operatorname{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (exakter Line-Search) wäre sinnlos aufwendig, da die Richtung h meist weit am Optimum vorbeiführt.

Anfangs sehr wenig Information: $\Phi(0) = f(\bar{x})$, Ableitung $\Phi'(0) = \nabla f(\bar{x})^T h$
Ein $\bar{\alpha}$ mit ausreichendem Abstieg (sufficient decrease) erfüllt:

1. Mindestanteil $0 < \gamma_1 < 1$ an dem durch $\Phi'(0)$ „versprochenen“Abstieg:
   $\Phi(\bar{\alpha}) \leq \Phi(0) + \bar{\alpha}\gamma_1\Phi'(0)$ (Armijo-Bedingung) [für kleine α erfüllt]
2. An der Stelle $\bar{\alpha}$ ist der Abstieg Φ' schlecht $(0 < \gamma_1 < \gamma_2 < 1)$:
   $\Phi'(\bar{\alpha}) \geq \gamma_2\Phi'(0)$ (Krümmungs-Bedingung) [$\nabla f^T h$ stark geändert]



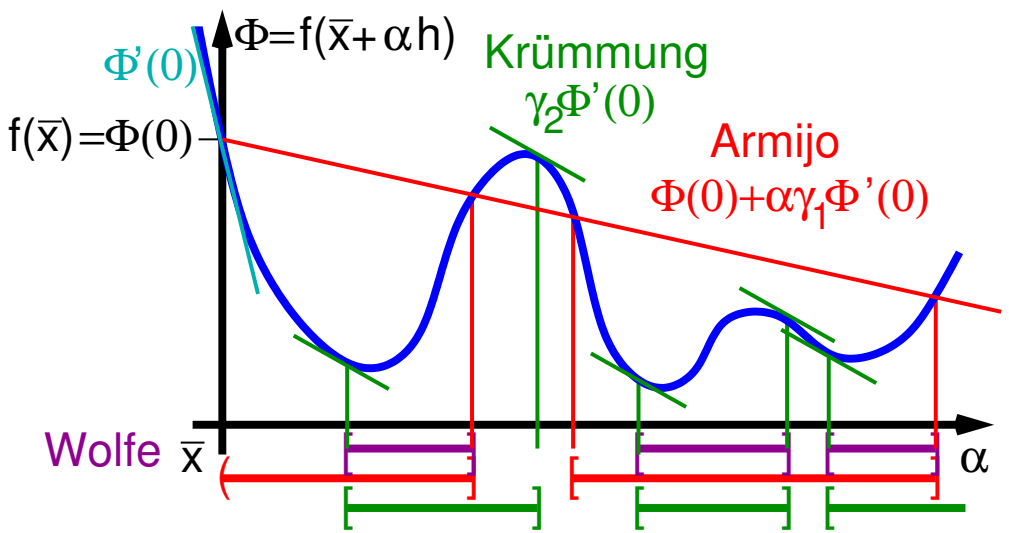
Line-Search für Abstiegsrichtung h

Bestimme Schrittweite $\bar{\alpha} \geq 0$ als **Näherung** zu $\min_{\alpha \geq 0} \Phi(\alpha) := f(\bar{x} + \alpha h)$.

Berechnung von $\bar{\alpha} \in \operatorname{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (exakter Line-Search) wäre sinnlos aufwendig, da die Richtung h meist weit am Optimum vorbeiführt.

Anfangs sehr wenig Information: $\Phi(0) = f(\bar{x})$, Ableitung $\Phi'(0) = \nabla f(\bar{x})^T h$
Ein $\bar{\alpha}$ mit ausreichendem Abstieg (sufficient decrease) erfüllt:

1. Mindestanteil $0 < \gamma_1 < 1$ an dem durch $\Phi'(0)$ „versprochenen“Abstieg:
   $\Phi(\bar{\alpha}) \leq \Phi(0) + \bar{\alpha}\gamma_1\Phi'(0)$ (Armijo-Bedingung) [für kleine α erfüllt]
2. An der Stelle $\bar{\alpha}$ ist der Abstieg Φ' schlecht $(0 < \gamma_1 < \gamma_2 < 1)$:
   $\Phi'(\bar{\alpha}) \geq \gamma_2\Phi'(0)$ (Krümmungs-Bedingung) [$\nabla f^T h$ stark geändert]



Armijo- und Krümmungs-Bedingung gemeinsam heißen Wolfe-Bedingungen und Schrittweiten, die diese erfüllen, garantieren ausreichenden Abstieg.
$[\gamma_1 = 10^{-4}, \gamma_2 \in \{0.1, 0.9\}]$

Wolfe-Bedingungen und globale Konvergenz

Für $0 < \gamma_1 < \gamma_2 < 1$ erfüllt Schrittweite α_k die Wolfe-Bedingungen, wenn

$$\begin{array}{rcll} f(x^{(k)} + \alpha_k h^{(k)}) & \leq & f_k + \alpha_k \gamma_1 \nabla f_k^T h^{(k)} & \text{(Armijo)} \\ \nabla f(x^{(k)} + \alpha_k h^{(k)})^T h^{(k)} & \geq & \gamma_2 \nabla f_k^T h^{(k)} & \text{(Krümmung)} \end{array}$$

Armijo sichert Abstieg, Krümmung eine Mindestschrittweite, falls ∇f Lipschitz-stetig ist. Beides ist mit einem Orakel 1. Ordnung überprüfbar. Solche Schrittweiten gibt es immer, wenn f nach unten beschränkt ist.

Wolfe-Bedingungen und globale Konvergenz

Für $0 < \gamma_1 < \gamma_2 < 1$ erfüllt Schrittweite α_k die Wolfe-Bedingungen, wenn

$$\begin{array}{rcll} f(x^{(k)} + \alpha_k h^{(k)}) & \leq & f_k + \alpha_k \gamma_1 \nabla f_k^T h^{(k)} & \text{(Armijo)} \\ \nabla f(x^{(k)} + \alpha_k h^{(k)})^T h^{(k)} & \geq & \gamma_2 \nabla f_k^T h^{(k)} & \text{(Krümmung)} \end{array}$$

Armijo sichert Abstieg, Krümmung eine Mindestschrittweite, falls ∇f Lipschitz-stetig ist. Beides ist mit einem Orakel 1. Ordnung überprüfbar. Solche Schrittweiten gibt es immer, wenn f nach unten beschränkt ist.

Satz (Globale Konvergenz von Line-Search-Verfahren)

Sei f nach unten beschränkt. Für den Startpunkt $x^{(0)}$ sei ∇f auf der Niveaumenge $\{x \in \mathbb{R}^n : f(x) < f_0\}$ Lipschitz-stetig. Garantiert ein Line-Search-Verfahren $-\frac{\nabla f_k^T h^{(k)}}{\|\nabla f_k\| \|h^{(k)}\|} \geq \delta$ für ein $\delta > 0$ sowie die Wolfe-Bedingungen für die Schrittweiten α_k, dann gilt $\|\nabla f_k\| \to 0$.

Wolfe-Bedingungen und globale Konvergenz

Für $0 < \gamma_1 < \gamma_2 < 1$ erfüllt Schrittweite α_k die Wolfe-Bedingungen, wenn

$$\begin{array}{rcll} f(x^{(k)} + \alpha_k h^{(k)}) & \leq & f_k + \alpha_k \gamma_1 \nabla f_k^T h^{(k)} & \text{(Armijo)} \\ \nabla f(x^{(k)} + \alpha_k h^{(k)})^T h^{(k)} & \geq & \gamma_2 \nabla f_k^T h^{(k)} & \text{(Krümmung)} \end{array}$$

Armijo sichert Abstieg, Krümmung eine Mindestschrittweite, falls ∇f Lipschitz-stetig ist. Beides ist mit einem Orakel 1. Ordnung überprüfbar. Solche Schrittweiten gibt es immer, wenn f nach unten beschränkt ist.

Satz (Globale Konvergenz von Line-Search-Verfahren)

Sei f nach unten beschränkt. Für den Startpunkt $x^{(0)}$ sei ∇f auf der Niveaumenge $\{x \in \mathbb{R}^n : f(x) < f_0\}$ Lipschitz-stetig. Garantiert ein Line-Search-Verfahren $-\frac{\nabla f_k^T h^{(k)}}{\|\nabla f_k\| \|h^{(k)}\|} \geq \delta$ für ein $\delta > 0$ sowie die Wolfe-Bedingungen für die Schrittweiten α_k, dann gilt $\|\nabla f_k\| \to 0$.

Vorsicht: Man hofft auf Konvergenz gegen ein Minimum, aber sowohl $\|x^{(k)}\| \to \infty$ als auch Konvergenz gegen einen Sattelpunkt sind nicht ausgeschlossen!

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.
- Die vorhergehende Schrittweite dient meist als Startwert, beim allerersten Mal nutzt man gerne $\alpha = \frac{1}{\|h\|}$.

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.
- Die vorhergehende Schrittweite dient meist als Startwert, beim allerersten Mal nutzt man gerne $\alpha = \frac{1}{\|h\|}$.
- Der nächsten Kandidat wird z.B. über kubische Interpolation, die neue und alte Funktionswerte und Ableitungen nutzt, bestimmt.

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.
- Die vorhergehende Schrittweite dient meist als Startwert, beim allerersten Mal nutzt man gerne $\alpha = \frac{1}{\|h\|}$.
- Der nächsten Kandidat wird z.B. über kubische Interpolation, die neue und alte Funktionswerte und Ableitungen nutzt, bestimmt.
- Jeder Fehl-Versuch erlaubt, das Suchintervall zu verkleinern.

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.
- Die vorhergehende Schrittweite dient meist als Startwert, beim allerersten Mal nutzt man gerne $\alpha = \frac{1}{\|h\|}$.
- Der nächsten Kandidat wird z.B. über kubische Interpolation, die neue und alte Funktionswerte und Ableitungen nutzt, bestimmt.
- Jeder Fehl-Versuch erlaubt, das Suchintervall zu verkleinern.
- Eine solide und effiziente Implementation, die auch mit numerischen Schwierigkeiten umgehen kann, ist sehr schwer und aufwendig.

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.
- Die vorhergehende Schrittweite dient meist als Startwert, beim allerersten Mal nutzt man gerne $\alpha = \frac{1}{\|h\|}$.
- Der nächsten Kandidat wird z.B. über kubische Interpolation, die neue und alte Funktionswerte und Ableitungen nutzt, bestimmt.
- Jeder Fehl-Versuch erlaubt, das Suchintervall zu verkleinern.
- Eine solide und effiziente Implementation, die auch mit numerischen Schwierigkeiten umgehen kann, ist sehr schwer und aufwendig.
- Entscheidend für den Erfolg ist vor allem die Schrittrichtung!

Bestimmung der Schrittweite in der Praxis

- Ziel ist, mit möglichst wenig Funktionsauswertungen einen Wolfepunkt zu finden.
- Die vorhergehende Schrittweite dient meist als Startwert, beim allerersten Mal nutzt man gerne $\alpha = \frac{1}{\|h\|}$.
- Der nächsten Kandidat wird z.B. über kubische Interpolation, die neue und alte Funktionswerte und Ableitungen nutzt, bestimmt.
- Jeder Fehl-Versuch erlaubt, das Suchintervall zu verkleinern.
- Eine solide und effiziente Implementation, die auch mit numerischen Schwierigkeiten umgehen kann, ist sehr schwer und aufwendig.
- Entscheidend für den Erfolg ist vor allem die Schrittrichtung!

AUSNAHME: Für Newton-ähnliche Richtungen wird immer Schrittweite 1 zuerst probiert und nur auf Armijo getestet. Solange Armijo nicht erfüllt ist, reduziert man die Schrittweite (durch Interpolation oder einfaches Backtracking, d.h., Multiplikation der Schrittweite mit einem Faktor $0 < \sigma < 1$).

Inhalt

6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

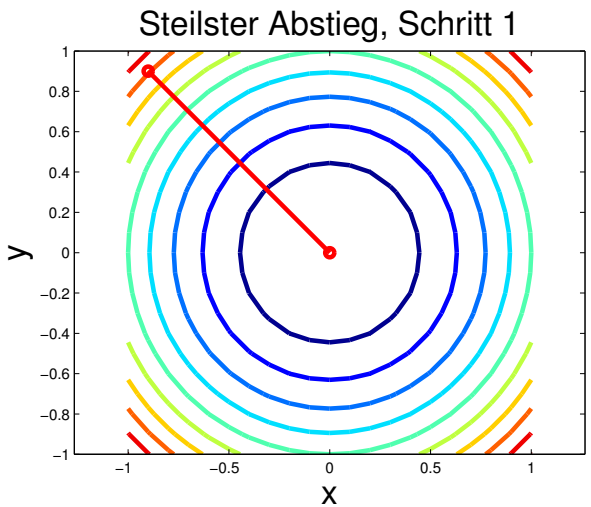
Ideal skaliert ($Q = I$):

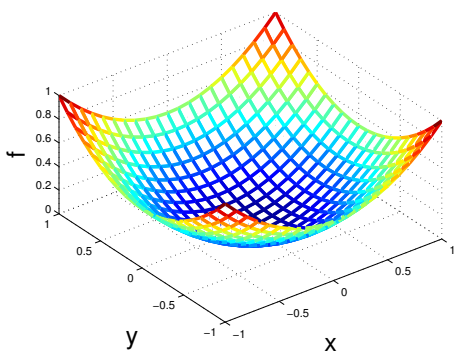
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0, c = 0$$

Startpunkt $\begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.9 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T Qh} = 1$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

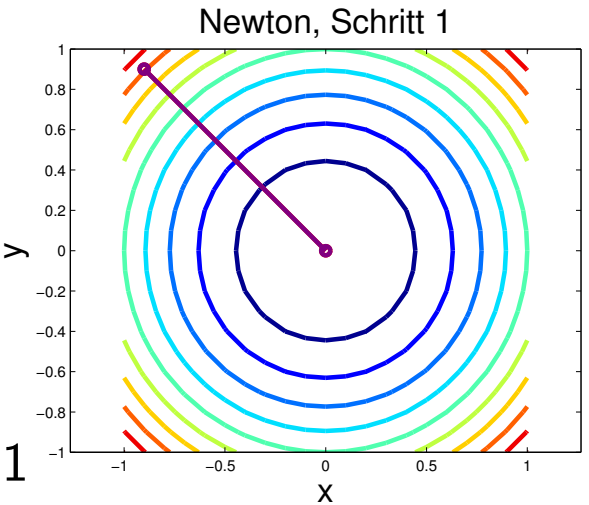
Ideal skaliert ($Q = I$):

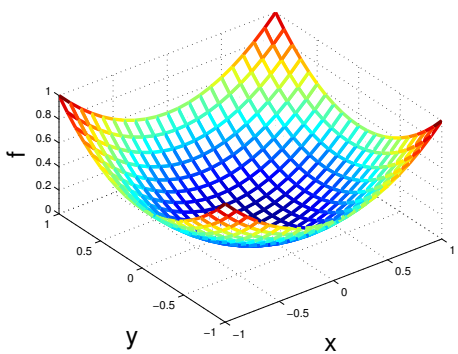
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0, c = 0$$

Startpunkt $\begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.9 \\ 0.9 \end{bmatrix}$

Newton ($B = Q$):

$$h = -Q^{-1}\nabla f,\ \bar{\alpha} = \frac{\nabla f^T Q^{-1}\nabla f}{\nabla f^T Q^{-1}\nabla f} = 1$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

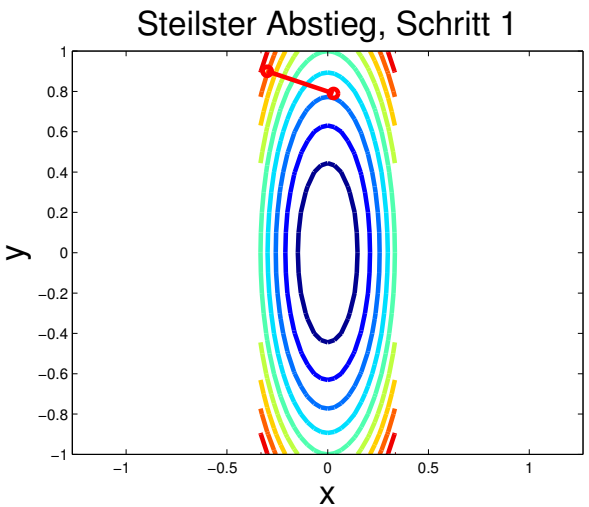
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

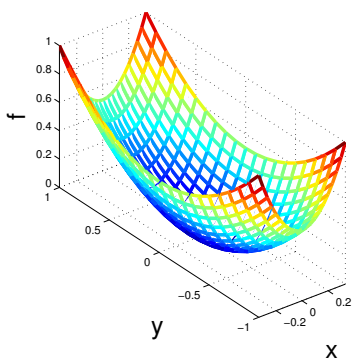
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

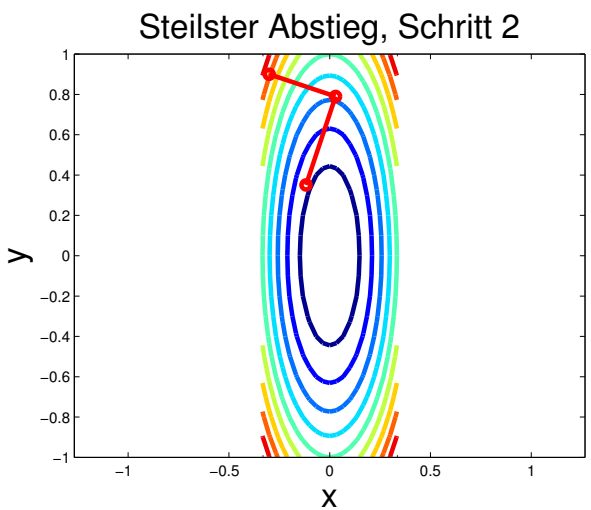
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

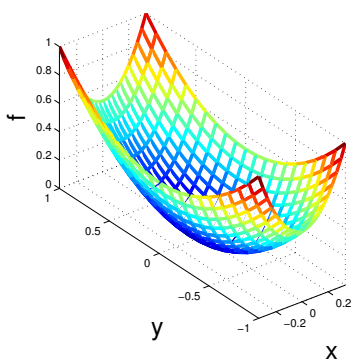
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

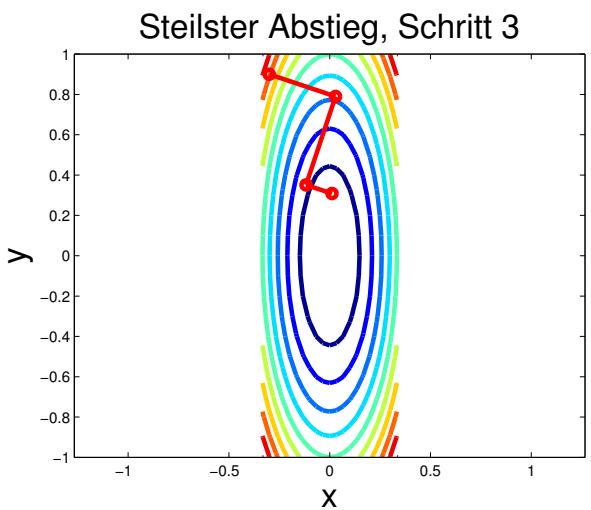
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

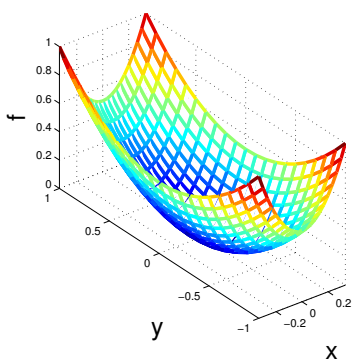
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

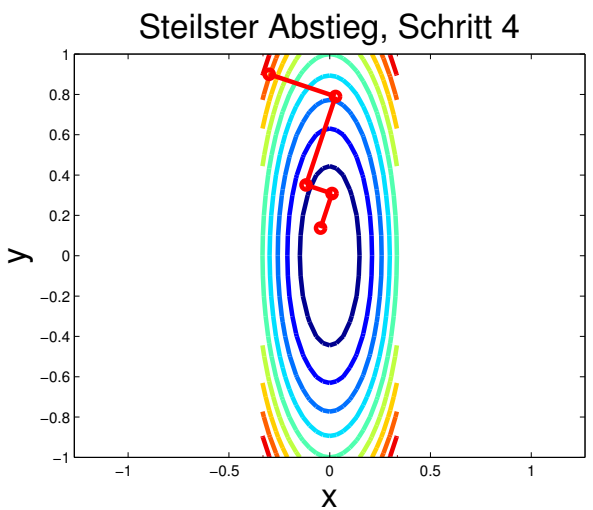
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

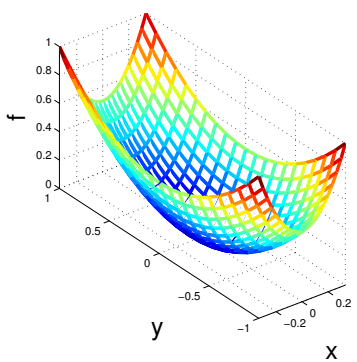
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

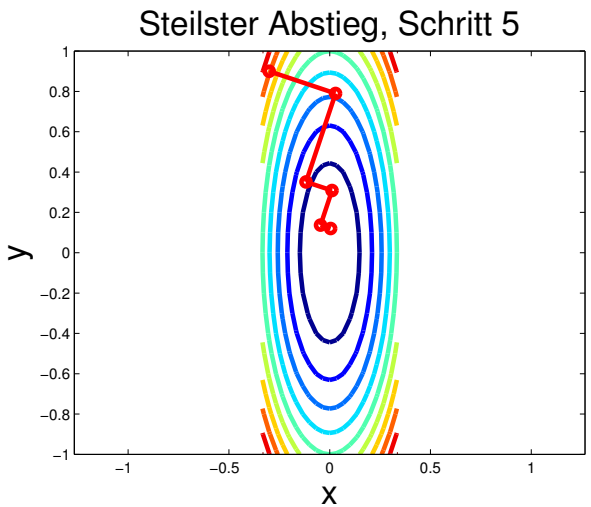
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

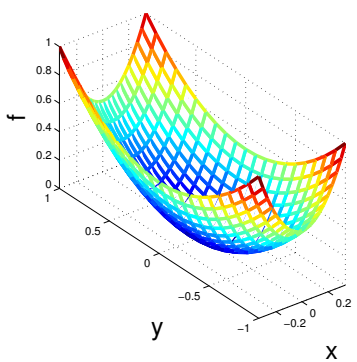
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

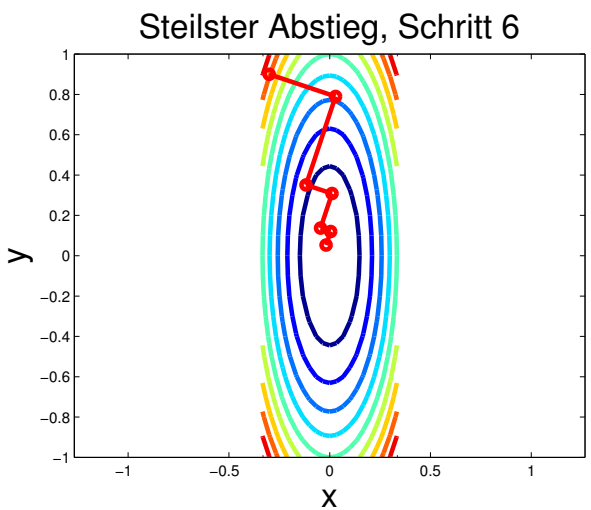
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

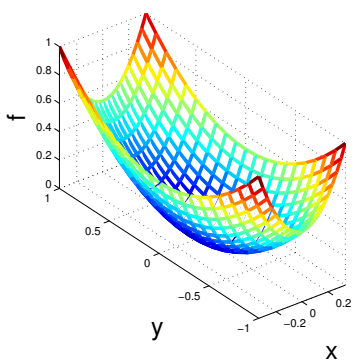
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

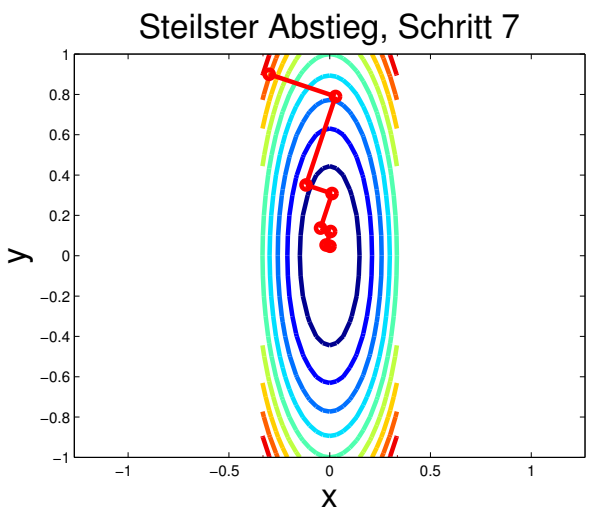
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

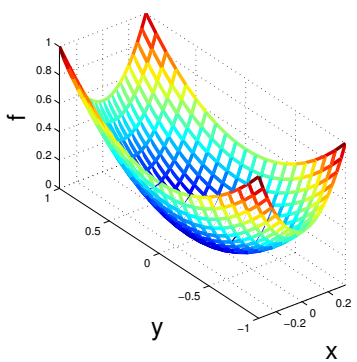
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

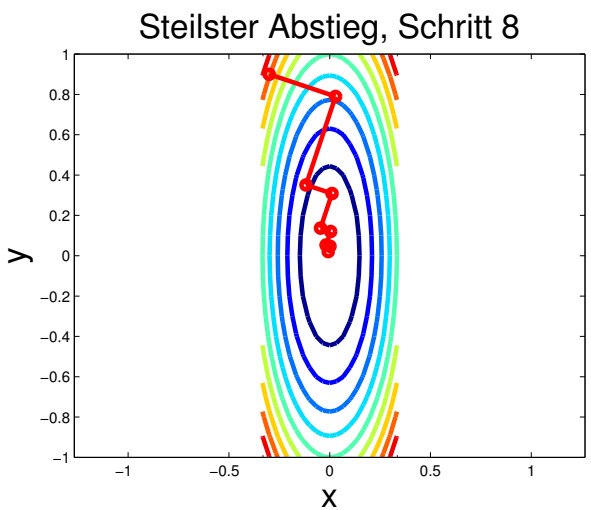
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

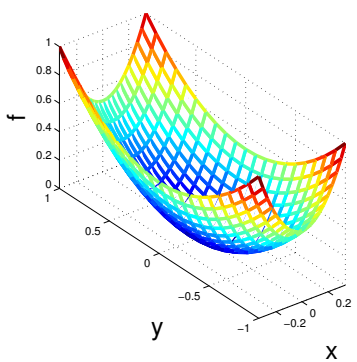
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Steilster Abstieg ($B = I$):

$$h = -\nabla f,\ \ \bar{\alpha} = \frac{\|h\|^2}{h^T\tilde{Q}h}$$





6.5 Skalierung

Bei Verfahren 1. Ordnung (nur Gradient) hat die Skalierung der Variablen (z.B. ob Daten in Metern oder Millimetern gegeben sind) großen Einfluss auf das Konvergenzverhalten und ist kaum vernünftig anpassbar.
Das Newtonverfahren (nutzt 2. Ordnung) ist jedoch skalierungsunabhängig!

Bsp: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$, $\nabla f(x) = Qx + q$, $\nabla^2 f(x) = Q$, $Q \succ 0$
exakter Line-Search für Abstiegsrichtung $h = -B^{-1}\nabla f(\bar{x})$ in $\bar{x}$ ($B \succ 0$):

$$\Phi(\alpha) = \frac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x} + c + \alpha\nabla f(\bar{x})^T h + \frac{\alpha^2}{2}h^T Qh$$

$$\Phi'(\alpha) = \nabla f(\bar{x})^T h + \alpha h^T Qh = 0 \quad \Rightarrow \quad \bar{\alpha} = \frac{-\nabla f^T h}{h^T Qh}$$

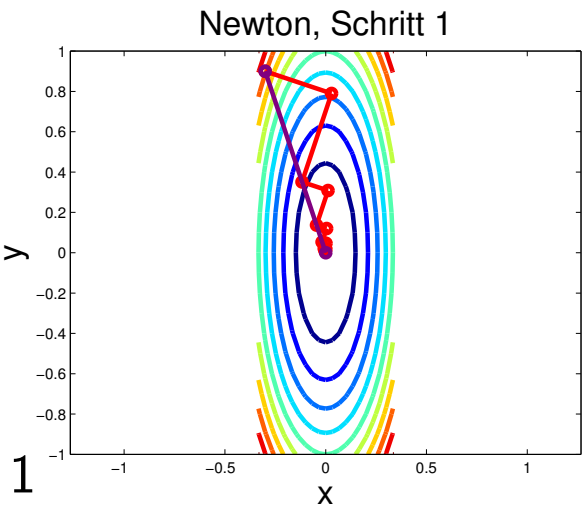
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

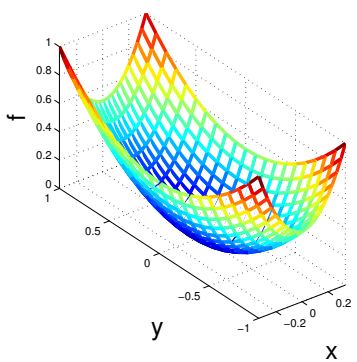
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

Startpunkt $\begin{bmatrix} \tilde{x}_1^{(0)} \\ x_2^{(0)} \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.9 \end{bmatrix}$

Newton ($B = \tilde{Q}$):

$$h = -\tilde{Q}^{-1}\nabla f,\ \bar{\alpha} = \frac{\nabla f^T \tilde{Q}^{-1}\nabla f}{\nabla f^T \tilde{Q}^{-1}\nabla f} = 1$$





Steilster Abstieg konvergiert SEHR langsam

Ein Verfahren konvergiert linear gegen x^*, wenn es eine Konstante $0 < \gamma < 1$ gibt mit $\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|$.

Satz

Für $f(x) = \frac{1}{2} x^T Q x + q^T x + c$ mit $Q \succ 0$ und exaktem Line-Search konvergiert das steilste Abstiegsverfahren im Allgemeinen linear mit Konstante $\gamma = \frac{\lambda_{\max}(Q) - \lambda_{\min}(Q)}{\lambda_{\max}(Q) + \lambda_{\min}(Q)}$.

Für $\lambda_{\max} \gg \lambda_{\min}$ (die Niveaumengen sind ganz schmale Ellipsen) ist $\gamma \approx 1$ und man sieht kaum Verbesserung.

Steilster Abstieg konvergiert SEHR langsam

Ein Verfahren konvergiert linear gegen x^*, wenn es eine Konstante $0 < \gamma < 1$ gibt mit $\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|$.

Satz

Für $f(x) = \frac{1}{2} x^T Q x + q^T x + c$ mit $Q \succ 0$ und exaktem Line-Search konvergiert das steilste Abstiegsverfahren im Allgemeinen linear mit Konstante $\gamma = \frac{\lambda_{\max}(Q) - \lambda_{\min}(Q)}{\lambda_{\max}(Q) + \lambda_{\min}(Q)}$.

Für $\lambda_{\max} \gg \lambda_{\min}$ (die Niveaumengen sind ganz schmale Ellipsen) ist $\gamma \approx 1$ und man sieht kaum Verbesserung.

In der Nähe eines Optimums mit hinreichenden Optimalitätsbedingungen ist die Funktion annähernd quadratisch streng konvex

$\rightarrow$ steilster Abstieg konvergiert am Ende fast immer schlecht

$\rightarrow$ Newton konvergiert am Ende fast immer hervorragend

Steilster Abstieg konvergiert SEHR langsam

Ein Verfahren konvergiert linear gegen x^*, wenn es eine Konstante $0 < \gamma < 1$ gibt mit $\|x_{k+1} - x^*\| \leq \gamma \|x_k - x^*\|$.

Satz

Für $f(x) = \frac{1}{2} x^T Q x + q^T x + c$ mit $Q \succ 0$ und exaktem Line-Search konvergiert das steilste Abstiegsverfahren im Allgemeinen linear mit Konstante $\gamma = \frac{\lambda_{\max}(Q) - \lambda_{\min}(Q)}{\lambda_{\max}(Q) + \lambda_{\min}(Q)}$.

Für $\lambda_{\max} \gg \lambda_{\min}$ (die Niveaumengen sind ganz schmale Ellipsen) ist $\gamma \approx 1$ und man sieht kaum Verbesserung.

In der Nähe eines Optimums mit hinreichenden Optimalitätsbedingungen ist die Funktion annähernd quadratisch streng konvex

$\rightarrow$ steilster Abstieg konvergiert am Ende fast immer schlecht

$\rightarrow$ Newton konvergiert am Ende fast immer hervorragend

---

Da für große n jede Iteration des Newton-Verfahrens wegen der Berechnung von $\nabla^2 f$ und $-(\nabla^2 f)^{-1} \nabla f$ sehr aufwendig ist, versucht man $(\nabla^2 f)^{-1}$ sukzessive aus den Werten von ∇f zu approximieren.

Inhalt

6.6 (Quasi-Newton-Verfahren)

Ein Verfahren konvergiert superlinear gegen x^*, falls $\lim_{k\to\infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|} = 0$ (wird kleiner als jede Konstante der linearen Konvergenz).

Satz

Konvergiert $x^{(k+1)} = x^{(k)} + h^{(k)}$ gegen ein x^*, das die hinreichenden Opt.-Bed. erfüllt, so ist die Konvergenz superlinear genau dann, wenn die Schrittrichtung $h^{(k)}$ sich schneller der Newton-Richtung $h_N^{(k)}$ annähert als sie klein wird, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

6.6 (Quasi-Newton-Verfahren)

Ein Verfahren konvergiert superlinear gegen x^*, falls $\lim_{k\to\infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|} = 0$ (wird kleiner als jede Konstante der linearen Konvergenz).

Satz

Konvergiert $x^{(k+1)} = x^{(k)} + h^{(k)}$ gegen ein x^*, das die hinreichenden Opt.-Bed. erfüllt, so ist die Konvergenz superlinear genau dann, wenn die Schrittrichtung $h^{(k)}$ sich schneller der Newton-Richtung $h_N^{(k)}$ annähert als sie klein wird, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

$\Rightarrow$ Superlineare Konvergenz erfordert Approximation der Newton-Richtung. Für $h = -B^{-1}\nabla f$ sollte B also die Hessematrix nachbauen. Diese erfüllt

$$\nabla f(x+h) = \nabla f(x) + \nabla^2 f(x)h + \mathbf{o}(h). \quad \text{[Taylor]}$$

6.6 (Quasi-Newton-Verfahren)

Ein Verfahren konvergiert superlinear gegen x^*, falls $\lim_{k\to\infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|} = 0$ (wird kleiner als jede Konstante der linearen Konvergenz).

Satz

Konvergiert $x^{(k+1)} = x^{(k)} + h^{(k)}$ gegen ein x^*, das die hinreichenden Opt.-Bed. erfüllt, so ist die Konvergenz superlinear genau dann, wenn die Schrittrichtung $h^{(k)}$ sich schneller der Newton-Richtung $h_N^{(k)}$ annähert als sie klein wird, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

$\Rightarrow$ Superlineare Konvergenz erfordert Approximation der Newton-Richtung.
Für $h = -B^{-1}\nabla f$ sollte B also die Hessematrix nachbauen. Diese erfüllt

$$\nabla f(x+h) = \nabla f(x) + \nabla^2 f(x)h + \mathbf{o}(h). \quad \text{[Taylor]}$$

Forderungen an B_{k+1} für Schrittrichtung: $h^{(k+1)} = -B_{k+1}^{-1}\nabla f_{k+1}$
$\to$ $B_{k+1} \succ 0$, damit $h^{(k+1)}$ Abstiegsrichtung ist

6.6 (Quasi-Newton-Verfahren)

Ein Verfahren konvergiert superlinear gegen x^*, falls $\lim_{k\to\infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|} = 0$ (wird kleiner als jede Konstante der linearen Konvergenz).

Satz

Konvergiert $x^{(k+1)} = x^{(k)} + h^{(k)}$ gegen ein x^*, das die hinreichenden Opt.-Bed. erfüllt, so ist die Konvergenz superlinear genau dann, wenn die Schrittrichtung $h^{(k)}$ sich schneller der Newton-Richtung $h_N^{(k)}$ annähert als sie klein wird, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

$\Rightarrow$ Superlineare Konvergenz erfordert Approximation der Newton-Richtung.
Für $h = -B^{-1}\nabla f$ sollte B also die Hessematrix nachbauen. Diese erfüllt

$$\nabla f(x+h) = \nabla f(x) + \nabla^2 f(x)h + \mathbf{o}(h). \quad \text{[Taylor]}$$

Forderungen an B_{k+1} für Schrittrichtung: $h^{(k+1)} = -B_{k+1}^{-1}\nabla f_{k+1}$

$\to$ $B_{k+1} \succ 0$, damit $h^{(k+1)}$ Abstiegsrichtung ist

$\to$ Das quadratische Modell $m_{k+1}(h) := f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$ sollte in x_k den Gradienten ∇f_k gut approximieren:

$$\nabla f_k = \nabla_h m_{k+1}(x^{(k)} - x^{(k+1)}) = \nabla f_{k+1} + B_{k+1}(x^{(k)} - x^{(k+1)})$$

6.6 (Quasi-Newton-Verfahren)

Ein Verfahren konvergiert superlinear gegen x^*, falls $\lim_{k\to\infty} \frac{\|x^{(k+1)}-x^*\|}{\|x^{(k)}-x^*\|} = 0$ (wird kleiner als jede Konstante der linearen Konvergenz).

Satz

Konvergiert $x^{(k+1)} = x^{(k)} + h^{(k)}$ gegen ein x^*, das die hinreichenden Opt.-Bed. erfüllt, so ist die Konvergenz superlinear genau dann, wenn die Schrittrichtung $h^{(k)}$ sich schneller der Newton-Richtung $h_N^{(k)}$ annähert als sie klein wird, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

$\Rightarrow$ Superlineare Konvergenz erfordert Approximation der Newton-Richtung.
Für $h = -B^{-1}\nabla f$ sollte B also die Hessematrix nachbauen. Diese erfüllt

$$\nabla f(x+h) = \nabla f(x) + \nabla^2 f(x)h + \mathbf{o}(h). \quad \text{[Taylor]}$$

Forderungen an B_{k+1} für Schrittrichtung: $h^{(k+1)} = -B_{k+1}^{-1}\nabla f_{k+1}$

$\rightarrow$ $B_{k+1} \succ 0$, damit $h^{(k+1)}$ Abstiegsrichtung ist

$\rightarrow$ Das quadratische Modell $m_{k+1}(h) := f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$ sollte in x_k den Gradienten ∇f_k gut approximieren:

$$\nabla f_k = \nabla_h m_{k+1}(x^{(k)} - x^{(k+1)}) = \nabla f_{k+1} + B_{k+1}(x^{(k)} - x^{(k+1)})$$

Sekanten-Gleichung: $B_{k+1}(x^{(k+1)} - x^{(k)}) = \nabla f_{k+1} - \nabla f_k$

(Quasi-Newton mit BFGS-Update)

Wegen Sekanten-Glg., $B_{k+1} \succ 0$ und $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$ muss

$$0 < \alpha_k^2 (h^{(k)})^T B_{k+1} h^{(k)} = \alpha (\nabla f_{k+1} - \nabla f_k)^T h^{(k)}$$

gelten. Das garantiert die Krümmung in den Wolfe-Bedingungen:

$$\alpha_k [\nabla f_{k+1}^T h^{(k)} - \nabla f_k^T h^{(k)}] > \alpha_k \underbrace{(c_2 - 1)}_{<0} \underbrace{\nabla f_k^T h^{(k)}}_{<0} > 0$$

(Quasi-Newton mit BFGS-Update)

Wegen Sekanten-Glg., $B_{k+1} \succ 0$ und $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$ muss

$$0 < \alpha_k^2 (h^{(k)})^T B_{k+1} h^{(k)} = \alpha (\nabla f_{k+1} - \nabla f_k)^T h^{(k)}$$

gelten. Das garantiert die Krümmung in den Wolfe-Bedingungen:

$$\alpha_k [\nabla f_{k+1}^T h^{(k)} - \nabla f_k^T h^{(k)}] > \alpha_k \underbrace{(c_2 - 1)}_{<0} \underbrace{\nabla f_k^T h^{(k)}}_{<0} > 0$$

Für $h^{(k+1)} = -B_{k+1}^{-1} \nabla f_{k+1}$ ist die Inverse $H_{k+1} := B_{k+1}^{-1}$ günstiger.

Die Matrix $H_{k+1} \succ 0$ mit $H_{k+1}(\nabla f_{k+1} - \nabla f_k) = x^{(k+1)} - x^{(k)}$, die sich gegenüber H_k in geeigneter Norm am wenigsten ändert, erhält man durch die Rang-2-Korrektur von Broyden, Fletcher, Goldfarb und Shanno:

$$H_{k+1} = (I - \frac{1}{s_k^T y_k} s_k y_k^T) H_k (I - \frac{1}{s_k^T y_k} y_k s_k^T) + \frac{1}{s_k^T y_k} s_k s_k^T$$

wobei $s_k := x^{(k+1)} - x^{(k)}$, $y_k := \nabla f_{k+1} - \nabla f_k$.

(Quasi-Newton mit BFGS-Update)

Wegen Sekanten-Glg., $B_{k+1} \succ 0$ und $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$ muss

$$0 < \alpha_k^2 (h^{(k)})^T B_{k+1} h^{(k)} = \alpha (\nabla f_{k+1} - \nabla f_k)^T h^{(k)} \quad [= y_k^T s_k]$$

gelten. Das garantiert die Krümmung in den Wolfe-Bedingungen:

$$\alpha_k [\nabla f_{k+1}^T h^{(k)} - \nabla f_k^T h^{(k)}] > \alpha_k \underbrace{(c_2 - 1)}_{<0} \underbrace{\nabla f_k^T h^{(k)}}_{<0} > 0$$

Für $h^{(k+1)} = -B_{k+1}^{-1} \nabla f_{k+1}$ ist die Inverse $H_{k+1} := B_{k+1}^{-1}$ günstiger.
Die Matrix $H_{k+1} \succ 0$ mit $H_{k+1}(\nabla f_{k+1} - \nabla f_k) = x^{(k+1)} - x^{(k)}$, die sich gegenüber H_k in geeigneter Norm am wenigsten ändert, erhält man durch die Rang-2-Korrektur von Broyden, Fletcher, Goldfarb und Shanno:

$$H_{k+1} = (I - \frac{1}{s_k^T y_k} s_k y_k^T) H_k (I - \frac{1}{s_k^T y_k} y_k s_k^T) + \frac{1}{s_k^T y_k} s_k s_k^T$$

wobei $s_k := x^{(k+1)} - x^{(k)}$, $y_k := \nabla f_{k+1} - \nabla f_k$. [$H_k \succ 0$ und Wolfe $\Rightarrow$ $H_{k+1} \succ 0$, denn $s_k^T H_{k+1} s_k \geq \frac{1}{s_k^T y_k} (s_k^T s_k)^2 > 0$, $y_k^T H_{k+1} y_k = s_k^T y_k > 0$.]

(Quasi-Newton mit BFGS-Update)

Wegen Sekanten-Glg., $B_{k+1} \succ 0$ und $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$ muss

$$0 < \alpha_k^2 (h^{(k)})^T B_{k+1} h^{(k)} = \alpha (\nabla f_{k+1} - \nabla f_k)^T h^{(k)} \quad [= y_k^T s_k]$$

gelten. Das garantiert die Krümmung in den Wolfe-Bedingungen:

$$\alpha_k [\nabla f_{k+1}^T h^{(k)} - \nabla f_k^T h^{(k)}] > \alpha_k \underbrace{(c_2 - 1)}_{<0} \underbrace{\nabla f_k^T h^{(k)}}_{<0} > 0$$

Für $h^{(k+1)} = -B_{k+1}^{-1} \nabla f_{k+1}$ ist die Inverse $H_{k+1} := B_{k+1}^{-1}$ günstiger.
Die Matrix $H_{k+1} \succ 0$ mit $H_{k+1}(\nabla f_{k+1} - \nabla f_k) = x^{(k+1)} - x^{(k)}$, die sich gegenüber H_k in geeigneter Norm am wenigsten ändert, erhält man durch die Rang-2-Korrektur von Broyden, Fletcher, Goldfarb und Shanno:

$$H_{k+1} = (I - \frac{1}{s_k^T y_k} s_k y_k^T) H_k (I - \frac{1}{s_k^T y_k} y_k s_k^T) + \frac{1}{s_k^T y_k} s_k s_k^T$$

wobei $s_k := x^{(k+1)} - x^{(k)}$, $y_k := \nabla f_{k+1} - \nabla f_k$. [$H_k \succ 0$ und Wolfe $\Rightarrow$ $H_{k+1} \succ 0$, denn $s_k^T H_{k+1} s_k \geq \frac{1}{s_k^T y_k} (s_k^T s_k)^2 > 0$, $y_k^T H_{k+1} y_k = s_k^T y_k > 0$.]

Man startet mit $H_0 = \nabla^2 f_0^{-1}$ (falls $\succ 0$) oder $H_0 = \frac{1}{\|\nabla f_0\|} I$.
Für große n bildet man H_k nicht explizit, sondern speichert nur die letzten $\bar{k}$ Paare (s_k, y_k) für ein festes $\bar{k} \in \mathbb{N}$ (limited memory BFGS).

Man kann zeigen: Für eine streng konvexe quadratische Funktion wird H_k eine immer bessere Approximation von $\nabla^2 f^{-1}$ und Line-Search Verfahren mit BFGS-Richtung und Wolfe-Bedingungen konvergieren superlinear.

Man kann zeigen: Für eine streng konvexe quadratische Funktion wird H_k eine immer bessere Approximation von $\nabla^2 f^{-1}$ und Line-Search Verfahren mit BFGS-Richtung und Wolfe-Bedingungen konvergieren superlinear.

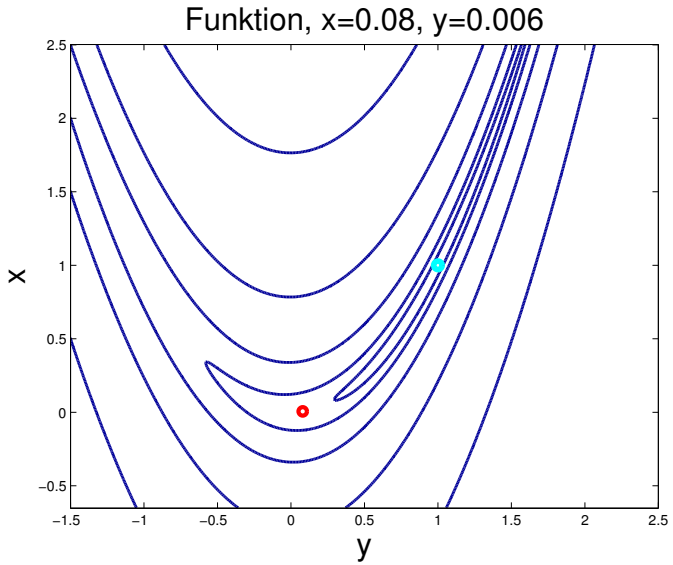
In der Nähe eines x^*, das die hinreichenden Optimalitätsbedingungen erfüllt, ist ein hinreichend glattes f annähernd streng konvex und quadratisch.

Konsequenz: Obwohl sowohl BFGS als auch Steilster Abstieg nur ein Orakel 1. Ordnung benötigen, konvergiert BFGS viel besser als Steilster Abstieg bei etwa vergleichbarem Aufwand pro Iteration.

Inhalt

6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

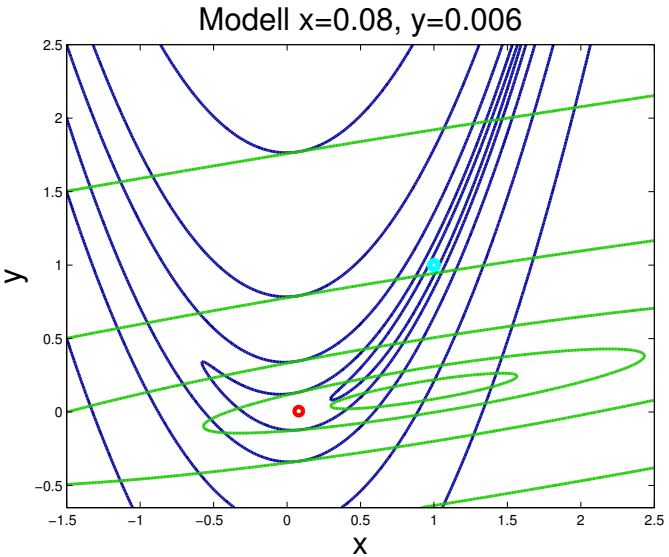


6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

---

0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]

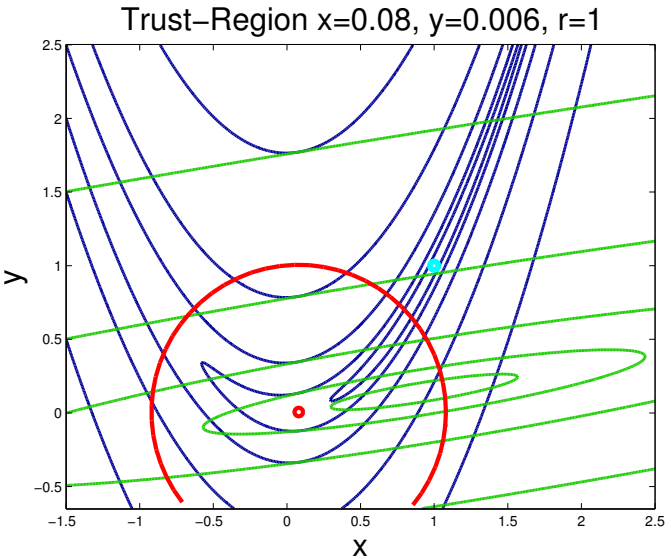


6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

---

0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

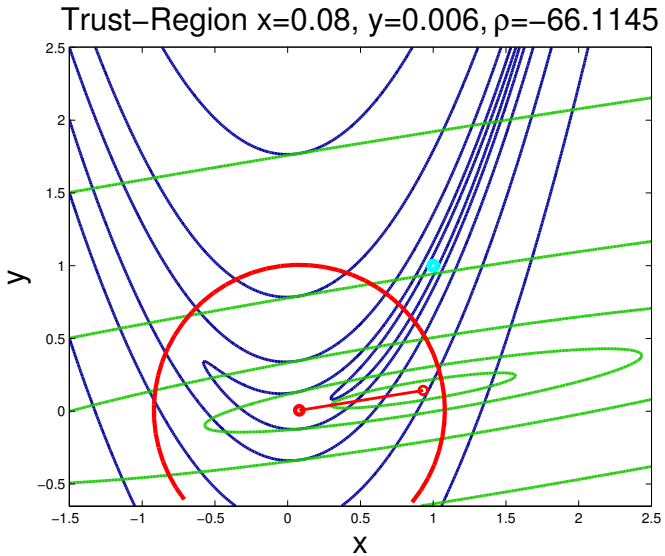


6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

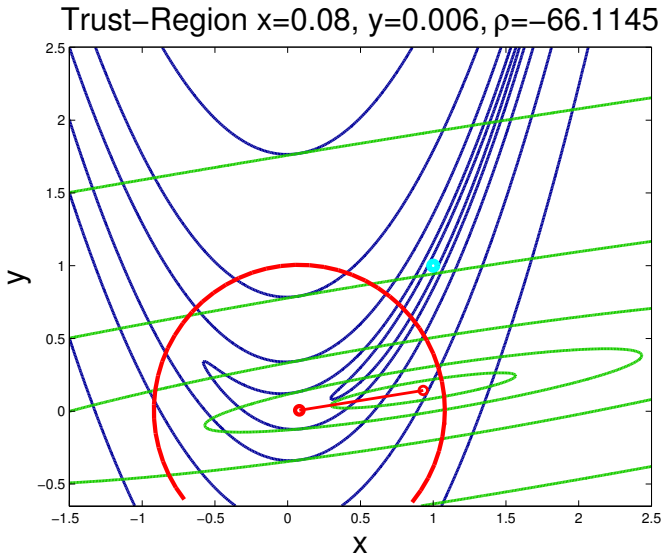
---

0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \leq \Delta_k} m_k(h)$
[nur annähern]



6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

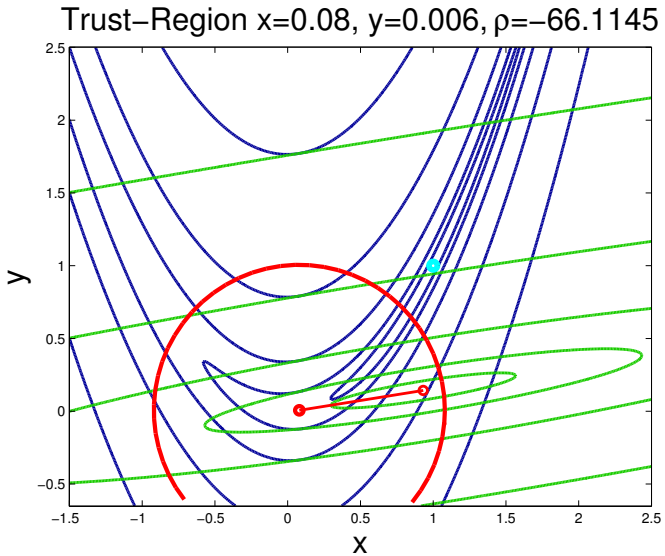


0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]
2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,

$$\text{setze } \Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$$

6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

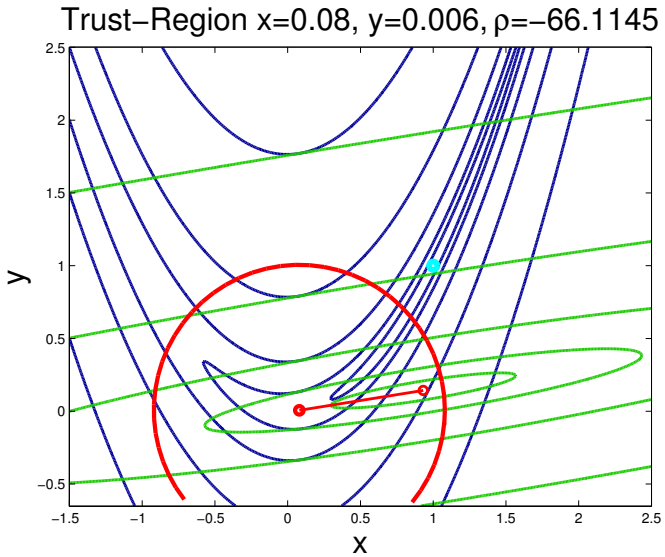


0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]
2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,

$$\text{setze } \Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$$

6.7 Trust-Region-Verfahren

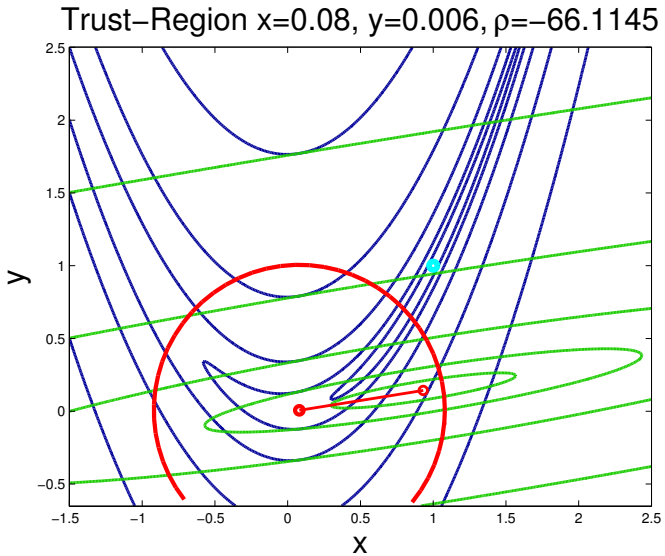
In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:



0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]
2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \\ \Delta_k & \text{sonst.} \end{cases}$
[$m_k(h^{(k)})$ zu schlecht]
[Δ zu klein]
[Wert ok]
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, \frac{1}{4})$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

6.7 Trust-Region-Verfahren

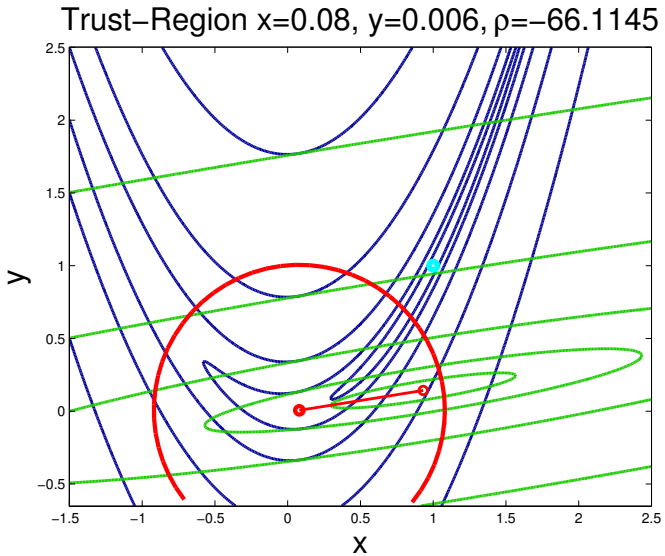
In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:



0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]
2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \\ \Delta_k & \text{sonst.} \end{cases}$
[$m_k(h^{(k)})$ zu schlecht]
[Δ zu klein]
[Wert ok]
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, \frac{1}{4})$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

6.7 Trust-Region-Verfahren

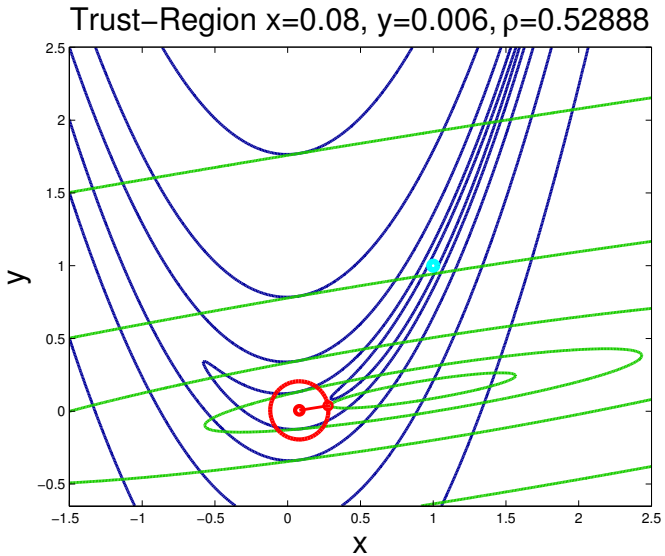
In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:



0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]
2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, \frac{1}{4})$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.
4. $k \leftarrow k + 1$. Falls $\|\nabla f_k\| > \varepsilon$, GOTO 1, sonst STOP.

6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:



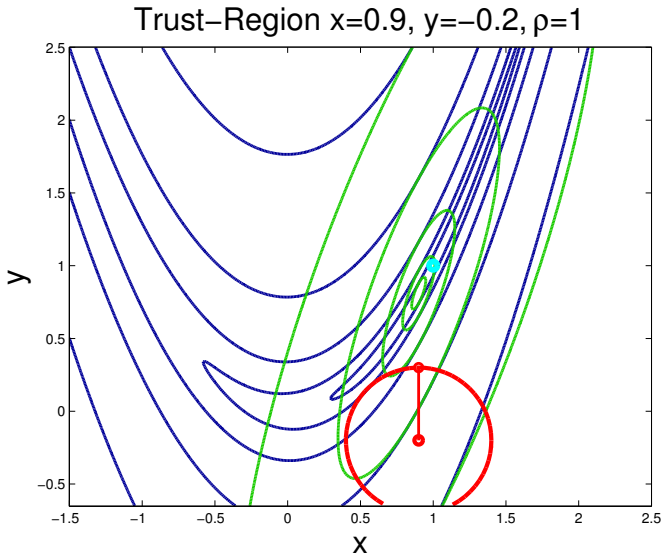
0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]
2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, \frac{1}{4})$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.
4. $k \leftarrow k+1$. Falls $\|\nabla f_k\| > \varepsilon$, GOTO 1, sonst STOP.

6.7 Trust-Region-Verfahren

In Line-Search-Verf. wird getrennt Schrittrichtung und -länge ermittelt, in Trust-Region-Verfahren beides zugleich nach folgendem Schema:

---

0. Wähle ein Model m_0 von f um $x^{(0)}$,
$$m_0(h) = f_0 + \nabla f_0^T h + \tfrac{1}{2} h^T B_0 h$$
[B_0 z.B. aus Newton oder Quasi-Newton]
und einen Trust-Region-Radius Δ_0 mit
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \le \Delta_0$$



1. Löse das Trust-Region-Unterproblem:
$h^{(k)} \approx \text{Argmin}_{\|h\| \le \Delta_k} m_k(h)$
[nur annähern]

2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,

setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$

3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, \frac{1}{4})$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

4. $k \leftarrow k+1$. Falls $\|\nabla f_k\| > \varepsilon$, GOTO 1, sonst STOP.

Konvergenz von Trust-Region-Verfahren

Weil m_k in 0 auch den Gradienten ∇f_k hat, gilt $\rho_k > \eta$ für Δ klein genug.

Konvergenz von Trust-Region-Verfahren

Weil m_k in 0 auch den Gradienten ∇f_k hat, gilt $\rho_k > \eta$ für Δ klein genug.
Globale Konvergenz ist (unter milden technischen Voraussetzungen) garantiert, wenn für eine Konstante $c_1 \in (0,1)$ und für jedes k die Näherungslösung $h^{(k)}$ die Trust-Region-Bedingung

$$\text{(TRC)} \qquad m_k(0) - m_k(h^{(k)}) \geq c_1 \|\nabla f_k\| \min\left\{\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right\}$$

erfüllt. [$\hat{=}$ red. Abstieg mal Schrittlänge, stellt *sufficient decrease* sicher]

Konvergenz von Trust-Region-Verfahren

Weil m_k in 0 auch den Gradienten ∇f_k hat, gilt $\rho_k > \eta$ für Δ klein genug.
Globale Konvergenz ist (unter milden technischen Voraussetzungen) garantiert, wenn für eine Konstante $c_1 \in (0,1)$ und für jedes k die Näherungslösung $h^{(k)}$ die Trust-Region-Bedingung

$$\text{(TRC)} \qquad m_k(0) - m_k(h^{(k)}) \geq c_1 \|\nabla f_k\| \min\left\{\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right\}$$

erfüllt. [$\hat{=}$ red. Abstieg mal Schrittlänge, stellt *sufficient decrease* sicher]

Diese Bedingung erfüllt der **Cauchy-Punkt** $h_C^{(k)}$ für $c_1 = \frac{1}{2}$. Er minimiert das Modell in Richtung des steilsten Abstiegs: Löse für $h = -\frac{\nabla f_k}{\|\nabla f_k\|}$

$$\min_{\alpha \in [0,\Delta_k]} m_k(\alpha h) = f_k + \alpha \nabla f_k^T h + \frac{\alpha^2}{2} h^T B_k h = f_k - \alpha \|\nabla f_k\| + \frac{\alpha^2}{2\|\nabla f_k\|^2} \nabla f_k^T B_k \nabla f_k$$

Konvergenz von Trust-Region-Verfahren

Weil m_k in 0 auch den Gradienten ∇f_k hat, gilt $\rho_k > \eta$ für Δ klein genug.
Globale Konvergenz ist (unter milden technischen Voraussetzungen) garantiert, wenn für eine Konstante $c_1 \in (0,1)$ und für jedes k die Näherungslösung $h^{(k)}$ die Trust-Region-Bedingung

$$\text{(TRC)} \qquad m_k(0) - m_k(h^{(k)}) \geq c_1 \|\nabla f_k\| \min\left\{\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right\}$$

erfüllt. [$\hat{=}$ red. Abstieg mal Schrittlänge, stellt *sufficient decrease* sicher]

Diese Bedingung erfüllt der **Cauchy-Punkt** $h_C^{(k)}$ für $c_1 = \frac{1}{2}$. Er minimiert das Modell in Richtung des steilsten Abstiegs: Löse für $h = -\frac{\nabla f_k}{\|\nabla f_k\|}$

$$\min_{\alpha \in [0,\Delta_k]} m_k(\alpha h) = f_k + \alpha \nabla f_k^T h + \frac{\alpha^2}{2} h^T B_k h = f_k - \alpha \|\nabla f_k\| + \frac{\alpha^2}{2\|\nabla f_k\|^2} \nabla f_k^T B_k \nabla f_k$$

$$\Rightarrow \alpha_k^C = \begin{cases} \Delta_k & \text{falls } \nabla f_k^T B_k \nabla f_k \leq 0, \\ \min\{\Delta_k, \frac{\|\nabla f_k\|^3}{\nabla f_k^T B_k \nabla f_k}\} & \text{sonst.} \end{cases} \qquad \left[-\|\nabla f_k\| + \alpha \frac{\nabla f_k^T B_k \nabla f_k}{\|\nabla f_k\|^2} = 0\right]$$

$$\boxed{h_C^{(k)} = -\alpha_k^C \frac{\nabla f_k}{\|\nabla f_k\|}}$$

Konvergenz von Trust-Region-Verfahren

Weil m_k in 0 auch den Gradienten ∇f_k hat, gilt $\rho_k > \eta$ für Δ klein genug.
Globale Konvergenz ist (unter milden technischen Voraussetzungen) garantiert, wenn für eine Konstante $c_1 \in (0,1)$ und für jedes k die Näherungslösung $h^{(k)}$ die Trust-Region-Bedingung

$$\text{(TRC)} \qquad m_k(0) - m_k(h^{(k)}) \geq c_1 \|\nabla f_k\| \min\left\{\Delta_k, \frac{\|\nabla f_k\|}{\|B_k\|}\right\}$$

erfüllt. [$\hat{=}$ red. Abstieg mal Schrittlänge, stellt *sufficient decrease* sicher]

Diese Bedingung erfüllt der **Cauchy-Punkt** $h_C^{(k)}$ für $c_1 = \frac{1}{2}$. Er minimiert das Modell in Richtung des steilsten Abstiegs: Löse für $h = -\frac{\nabla f_k}{\|\nabla f_k\|}$

$$\min_{\alpha \in [0,\Delta_k]} m_k(\alpha h) = f_k + \alpha \nabla f_k^T h + \frac{\alpha^2}{2} h^T B_k h = f_k - \alpha \|\nabla f_k\| + \frac{\alpha^2}{2\|\nabla f_k\|^2} \nabla f_k^T B_k \nabla f_k$$

$$\Rightarrow \alpha_k^C = \begin{cases} \Delta_k & \text{falls } \nabla f_k^T B_k \nabla f_k \leq 0, \\ \min\{\Delta_k, \frac{\|\nabla f_k\|^3}{\nabla f_k^T B_k \nabla f_k}\} & \text{sonst.} \end{cases} \qquad \left[-\|\nabla f_k\| + \alpha \frac{\nabla f_k^T B_k \nabla f_k}{\|\nabla f_k\|^2} = 0\right]$$

$$\boxed{h_C^{(k)} = -\alpha_k^C \frac{\nabla f_k}{\|\nabla f_k\|}}$$

Jede Verbesserung gegenüber $h_C^{(k)}$ erfüllt ebenso (TRC) für $c_1 = \frac{1}{2}$ und ist global konvergent.

Wie berechnet man ∇f ($\nabla^2 f$) für das Orakel 1. (2.) Ordnung?

Wie berechnet man ∇f ($\nabla^2 f$) für das Orakel 1. (2.) Ordnung?
Falls f analytisch vorliegt:
Selber differenzieren oder Matlab, Maple, ... nutzen.

Wie berechnet man ∇f ($\nabla^2 f$) für das Orakel 1. (2.) Ordnung?
Falls f analytisch vorliegt:
Selber differenzieren oder Matlab, Maple, ... nutzen.

Falls f nur als Unterprogramm gegeben ist oder symbolisches Differenzieren fehlschlägt:
- Numerisches Differenzieren
- Automatisches Differenzieren (falls Source-Code verfügbar)

(Für beides gibt es kommerzielle und frei verfügbare Software)

Inhalt

6.8 Numerisches Differenzieren

Für die Berechnung von $\nabla f(\bar{x})$ werden die partiellen Ableitungen durch endliche Differenzen angenähert [Vorsicht: Stellenauslöschung!]:

$$\frac{\partial f}{\partial x_i}(\bar{x}) \approx \frac{f(\bar{x} + \varepsilon e_i) - f(\bar{x} - \varepsilon e_i)}{2\varepsilon}$$

6.8 Numerisches Differenzieren

Für die Berechnung von $\nabla f(\bar{x})$ werden die partiellen Ableitungen durch endliche Differenzen angenähert [Vorsicht: Stellenauslöschung!]:

$$\frac{\partial f}{\partial x_i}(\bar{x}) \approx \frac{f(\bar{x} + \varepsilon e_i) - f(\bar{x} - \varepsilon e_i)}{2\varepsilon}$$

Die Numerik lehrt: Ist u das Maschinenepsilon (größte Fließkomma-Zahl mit float$(1 + u)$=float(1)), liefert $\varepsilon = u^{\frac{2}{3}}$ das beste Ergebnis mit einem Fehler von Konstante mal ε^2.

6.8 Numerisches Differenzieren

Für die Berechnung von $\nabla f(\bar{x})$ werden die partiellen Ableitungen durch endliche Differenzen angenähert [Vorsicht: Stellenauslöschung!]:

$$\frac{\partial f}{\partial x_i}(\bar{x}) \approx \frac{f(\bar{x} + \varepsilon e_i) - f(\bar{x} - \varepsilon e_i)}{2\varepsilon}$$

Die Numerik lehrt: Ist u das Maschinenepsilon (größte Fließkomma-Zahl mit float$(1 + u)$=float(1)), liefert $\varepsilon = u^{\frac{2}{3}}$ das beste Ergebnis mit einem Fehler von Konstante mal ε^2.

Pro Koordinate sind so zwei Funktionsberechnungen erforderlich, aber der Fehler der einseitigen Formel $\frac{f(\bar{x}+\varepsilon e_i)-f(\bar{x})}{\varepsilon}$ wird für $\varepsilon = \sqrt{u}$ minimiert und wäre nur durch Konstante mal ε beschränkt, ist also meist zu groß.

6.8 Numerisches Differenzieren

Für die Berechnung von $\nabla f(\bar{x})$ werden die partiellen Ableitungen durch endliche Differenzen angenähert [Vorsicht: Stellenauslöschung!]:

$$\frac{\partial f}{\partial x_i}(\bar{x}) \approx \frac{f(\bar{x} + \varepsilon e_i) - f(\bar{x} - \varepsilon e_i)}{2\varepsilon}$$

Die Numerik lehrt: Ist u das Maschinenepsilon (größte Fließkomma-Zahl mit float$(1+u)$=float(1)), liefert $\varepsilon = u^{\frac{2}{3}}$ das beste Ergebnis mit einem Fehler von Konstante mal ε^2.

Pro Koordinate sind so zwei Funktionsberechnungen erforderlich, aber der Fehler der einseitigen Formel $\frac{f(\bar{x}+\varepsilon e_i)-f(\bar{x})}{\varepsilon}$ wird für $\varepsilon = \sqrt{u}$ minimiert und wäre nur durch Konstante mal ε beschränkt, ist also meist zu groß.

Bei der Berechung von $\nabla^2 f$ oder der Jacobimatrix J_g einer Funktion $g : \mathbb{R}^n \to \mathbb{R}^m$ rentiert es sich zu untersuchen, ob eine eventuelle Dünnbesetztheit der Matrizen genutzt werden kann, um die Anzahl der Funktionsauswertungen zu reduzieren. Dies wird hier nicht weiter untersucht.

Inhalt

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$
Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$x_4 := x_1 x_2$

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$

Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$x_4 := x_1 x_2 \quad \to \nabla_x x_4 = \frac{\partial x_1 x_2}{\partial x_1} \nabla_x x_1 + \frac{\partial x_1 x_2}{\partial x_2} \nabla_x x_2 = x_2 \nabla_x x_1 + x_1 \nabla_x x_2 = \begin{bmatrix} x_2 \\ x_1 \\ 0 \end{bmatrix}$

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6/x_3$

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$

Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$x_4 := x_1 x_2 \quad \to \nabla_x x_4 = \frac{\partial x_1 x_2}{\partial x_1} \nabla_x x_1 + \frac{\partial x_1 x_2}{\partial x_2} \nabla_x x_2 = x_2 \nabla_x x_1 + x_1 \nabla_x x_2 = \begin{bmatrix} x_2 \\ x_1 \\ 0 \end{bmatrix}$

$x_5 := e^{x_4} \quad \to \nabla_x x_5 = \frac{\partial e^{x_4}}{\partial x_4} \nabla_x x_4 = e^{x_4} \nabla_x x_4$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$

Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$x_4 := x_1 x_2 \quad \to \nabla_x x_4 = \frac{\partial x_1 x_2}{\partial x_1} \nabla_x x_1 + \frac{\partial x_1 x_2}{\partial x_2} \nabla_x x_2 = x_2 \nabla_x x_1 + x_1 \nabla_x x_2 = \begin{bmatrix} x_2 \\ x_1 \\ 0 \end{bmatrix}$

$x_5 := e^{x_4} \quad \to \nabla_x x_5 = \frac{\partial e^{x_4}}{\partial x_4} \nabla_x x_4 = e^{x_4} \nabla_x x_4$

$x_6 := x_4 + x_5 \to \nabla_x x_6 = \frac{\partial x_4 + x_5}{\partial x_4} \nabla_x x_4 + \frac{\partial x_4 + x_5}{\partial x_5} \nabla_x x_5 = \nabla_x x_4 + \nabla_x x_5$

$x_7 := x_6/x_3$

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$

Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$$x_4 := x_1 x_2 \quad \to \nabla_x x_4 = \frac{\partial x_1 x_2}{\partial x_1} \nabla_x x_1 + \frac{\partial x_1 x_2}{\partial x_2} \nabla_x x_2 = x_2 \nabla_x x_1 + x_1 \nabla_x x_2 = \begin{bmatrix} x_2 \\ x_1 \\ 0 \end{bmatrix}$$

$$x_5 := e^{x_4} \quad \to \nabla_x x_5 = \frac{\partial e^{x_4}}{\partial x_4} \nabla_x x_4 = e^{x_4} \nabla_x x_4$$

$$x_6 := x_4 + x_5 \to \nabla_x x_6 = \frac{\partial x_4 + x_5}{\partial x_4} \nabla_x x_4 + \frac{\partial x_4 + x_5}{\partial x_5} \nabla_x x_5 = \nabla_x x_4 + \nabla_x x_5$$

$$x_7 := x_6/x_3 \to \nabla_x x_7 = \frac{\partial x_6/x_3}{\partial x_6} \nabla_x x_6 + \frac{\partial x_6/x_3}{\partial x_3} \nabla_x x_3 = x_3^{-1} \nabla_x x_6 - x_6 x_3^{-2} \nabla_x x_3$$

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$

Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$$x_4 := x_1 x_2 \quad \to \nabla_x x_4 = \frac{\partial x_1 x_2}{\partial x_1} \nabla_x x_1 + \frac{\partial x_1 x_2}{\partial x_2} \nabla_x x_2 = x_2 \nabla_x x_1 + x_1 \nabla_x x_2 = \begin{bmatrix} x_2 \\ x_1 \\ 0 \end{bmatrix}$$

$$x_5 := e^{x_4} \quad \to \nabla_x x_5 = \frac{\partial e^{x_4}}{\partial x_4} \nabla_x x_4 = e^{x_4} \nabla_x x_4$$

$$x_6 := x_4 + x_5 \to \nabla_x x_6 = \frac{\partial x_4 + x_5}{\partial x_4} \nabla_x x_4 + \frac{\partial x_4 + x_5}{\partial x_5} \nabla_x x_5 = \nabla_x x_4 + \nabla_x x_5$$

$$x_7 := x_6/x_3 \to \nabla_x x_7 = \frac{\partial x_6/x_3}{\partial x_6} \nabla_x x_6 + \frac{\partial x_6/x_3}{\partial x_3} \nabla_x x_3 = x_3^{-1} \nabla_x x_6 - x_6 x_3^{-2} \nabla_x x_3$$

Beachte:
Ein Compiler zerlegt die Berechnung von f ebenso in elementare Schritte mit Zwischenvariablen x_i, am Ende ist $f = x_7$ und $\nabla_x f = \nabla_x x_7$.

6.9 Automatisches Differenzieren

Die Kettenregel für $g(y(x))$ mit $g : \mathbb{R}^m \to \mathbb{R}^k$, $y : \mathbb{R}^n \to \mathbb{R}^m$,

$$\nabla_x g(y(x)) = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)$$

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$

Beginne mit $\nabla_x x_1 = e_1$, $\nabla_x x_2 = e_2$, $\nabla_x x_3 = e_3$,

$$x_4 := x_1 x_2 \quad \to \nabla_x x_4 = \frac{\partial x_1 x_2}{\partial x_1} \nabla_x x_1 + \frac{\partial x_1 x_2}{\partial x_2} \nabla_x x_2 = x_2 \nabla_x x_1 + x_1 \nabla_x x_2 = \begin{bmatrix} x_2 \\ x_1 \\ 0 \end{bmatrix}$$

$$x_5 := e^{x_4} \quad \to \nabla_x x_5 = \frac{\partial e^{x_4}}{\partial x_4} \nabla_x x_4 = e^{x_4} \nabla_x x_4$$

$$x_6 := x_4 + x_5 \to \nabla_x x_6 = \frac{\partial x_4 + x_5}{\partial x_4} \nabla_x x_4 + \frac{\partial x_4 + x_5}{\partial x_5} \nabla_x x_5 = \nabla_x x_4 + \nabla_x x_5$$

$$x_7 := x_6/x_3 \to \nabla_x x_7 = \frac{\partial x_6/x_3}{\partial x_6} \nabla_x x_6 + \frac{\partial x_6/x_3}{\partial x_3} \nabla_x x_3 = x_3^{-1} \nabla_x x_6 - x_6 x_3^{-2} \nabla_x x_3$$

Beachte:
Ein Compiler zerlegt die Berechnung von f ebenso in elementare Schritte mit Zwischenvariablen x_i, am Ende ist $f = x_7$ und $\nabla_x f = \nabla_x x_7$.
Für jeden elementaren Schritt eines Programms können die partiellen Ableitungen explizit angegeben werden! Die Gradienten aufzusummieren ist aber ineffizient und besser organisierbar.

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, D_h ... Richtungsableitung für h

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, D_h ... Richtungsableitung für h

Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,

$x_4 := x_1 x_2$

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, D_h ... Richtungsableitung für h

Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,

$x_4 := x_1 x_2 \quad \rightarrow D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, D_h ... Richtungsableitung für h

Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,

$x_4 := x_1 x_2 \quad \to D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$

$x_5 := e^{x_4} \quad \to D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $\quad D_h$ ... Richtungsableitung für h

Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,

$x_4 := x_1 x_2 \quad \to D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$

$x_5 := e^{x_4} \quad \to D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$

$x_6 := x_4 + x_5 \to D_h x_6 = \ldots = D_h x_4 + D_h x_5$

$x_7 := x_6 / x_3$

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $\quad D_h$ ... Richtungsableitung für h

Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,

$x_4 := x_1 x_2 \quad \to D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$

$x_5 := e^{x_4} \quad \to D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$

$x_6 := x_4 + x_5 \to D_h x_6 = \ldots = D_h x_4 + D_h x_5$

$x_7 := x_6 / x_3 \to D_h x_7 = \ldots = x_3^{-1} D_h x_6 - x_6 x_3^{-2} D_h x_3$

(Automatische Vorwärts-Berechnung von $\nabla f(x)^T h$)

Gleichzeitig mit f kann für ein gegebenes $h \in \mathbb{R}^n$ die Richtungsableitung $D_h f(x) := \nabla f(x)^T h$ berechnet werden. Für D_h lautet die Kettenregel

$$D_h g(y(x)) = \nabla_x g(y(x))^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) \nabla_x y_i(x)^T h = \sum_{i=1}^{m} \frac{\partial g}{\partial y_i}(y) D_h y_i(x)$$

Am Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $\quad D_h$ ... Richtungsableitung für h

Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,

$x_4 := x_1 x_2 \quad \to D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$

$x_5 := e^{x_4} \quad \to D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$

$x_6 := x_4 + x_5 \to D_h x_6 = \ldots = D_h x_4 + D_h x_5$

$x_7 := x_6/x_3 \to D_h x_7 = \ldots = x_3^{-1} D_h x_6 - x_6 x_3^{-2} D_h x_3$

$D_h f(x) = \nabla f(x)^T h$ kann also zugleich mit f und mit relativ geringem Zusatzaufwand und Speicher berechnet werden!

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$		$\frac{\partial f}{\partial x_i}$
x_1	$=$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} =$		
x_2	$=$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} =$		
x_3	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$		
$x_4 := x_1 x_2$	$=$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} =$	, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$		
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$		
$x_7 := x_6/x_3$	$=$	$\emptyset$			

Vorwärts-Berechnung:

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$		$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} =$		
x_2	$=$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} =$		
x_3	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$		
$x_4 := x_1 x_2$	$=$	$\{5,6\}$	$\frac{\partial x_5}{\partial x_4} =$	, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$		
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$		
$x_7 := x_6/x_3$	$=$	$\emptyset$			

Vorwärts-Berechnung: x_1

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$		$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} =$		
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} =$		
x_3	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$		
$x_4 := x_1 x_2$	$=$	$\{5,6\}$	$\frac{\partial x_5}{\partial x_4} =$	, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$		
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$		
$x_7 := x_6/x_3$	$=$	$\emptyset$			

Vorwärts-Berechnung: x_2

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$		$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} =$		
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} =$		
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$		
$x_4 := x_1 x_2$	$=$	$\{5,6\}$	$\frac{\partial x_5}{\partial x_4} =$	, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$		
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$		
$x_7 := x_6/x_3$	$=$	$\emptyset$			

Vorwärts-Berechnung: x_3

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$	
$x_4 := x_1 x_2$	$=0$	$\{5,6\}$	$\frac{\partial x_5}{\partial x_4} =$, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$	
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$	
$x_7 := x_6/x_3$	$=$	$\emptyset$		

Vorwärts-Berechnung: x_4

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$	
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$	
$x_7 := x_6/x_3$	$=$	$\emptyset$		

Vorwärts-Berechnung: x_5

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$	
$x_7 := x_6/x_3$	$=$	$\emptyset$		

Vorwärts-Berechnung: x_6

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5,6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	

Vorwärts-Berechnung: x_7

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_7}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_6}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_5}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_4}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_3}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	$\frac{\partial f}{\partial x_2} = \frac{2}{3} \cdot 2 = \frac{4}{3}$
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_2}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	$\frac{\partial f}{\partial x_1} = \frac{2}{3} \cdot 0 = 0$
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	$\frac{\partial f}{\partial x_2} = \frac{2}{3} \cdot 2 = \frac{4}{3}$
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_1}$

(Automatische Rückwärts-Berechnung von $\nabla f(x)$)

Taucht x_i nur in den Variablen $j \in N_i$ explizit auf, sagt die Kettenregel

$$\frac{\partial f}{\partial x_i} = \sum_{j \in N_i} \frac{\partial f}{\partial x_j} \frac{\partial x_j}{\partial x_i}$$

Die Werte $\frac{\partial x_j}{\partial x_i}$ werden in der Vorwärts-Berechnung ermittelt. Sind einmal alle Werte $\frac{\partial f}{\partial x_j}$ für $j \in N_i$ bekannt, ergibt sich $\frac{\partial f}{\partial x_i}$ aus obiger Formel.

Bsp: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$ für $(x_1, x_2, x_3) = (2, 0, 3)$

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	$\frac{\partial f}{\partial x_1} = \frac{2}{3} \cdot 0 = 0$
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	$\frac{\partial f}{\partial x_2} = \frac{2}{3} \cdot 2 = \frac{4}{3}$
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\nabla f = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3})^T = (0, \frac{4}{3}, \frac{1}{9})^T$

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).
- $\nabla f^T h$ ($\nabla^2 fh$) in Rechengenauigkeit! (im Gegensatz zu num. Diff.)

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).
- $\nabla f^T h$ ($\nabla^2 fh$) in Rechengenauigkeit! (im Gegensatz zu num. Diff.)
- Grenzen: AD-Software kommt z.B. nicht immer mit Verzweigungen/bedingten Berechnungen im Source-code zurecht.

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).
- $\nabla f^T h$ ($\nabla^2 fh$) in Rechengenauigkeit! (im Gegensatz zu num. Diff.)
- Grenzen: AD-Software kommt z.B. nicht immer mit Verzweigungen/bedingten Berechnungen im Source-code zurecht.

Verfahren, die $\nabla^2 f(x)$ nicht explizit benötigen, sondern nur eine Routine, die $\nabla^2 f(x)$ mal Vektor berechnet, heißen „Hessian-free methods“.

Inhalt

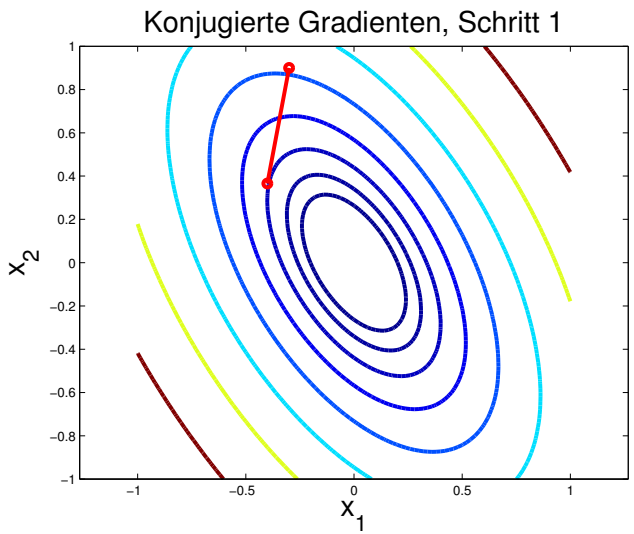
6.10 Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

6.10 Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

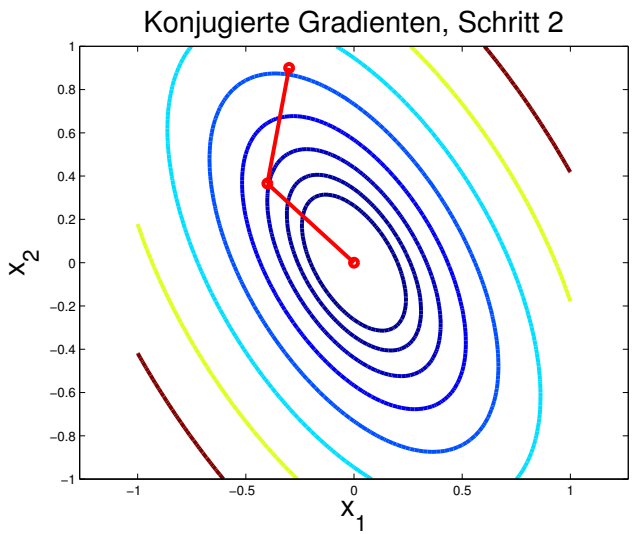
Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.



6.10 Das lineare Konjugierte-Gradienten-Verfahren

Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.



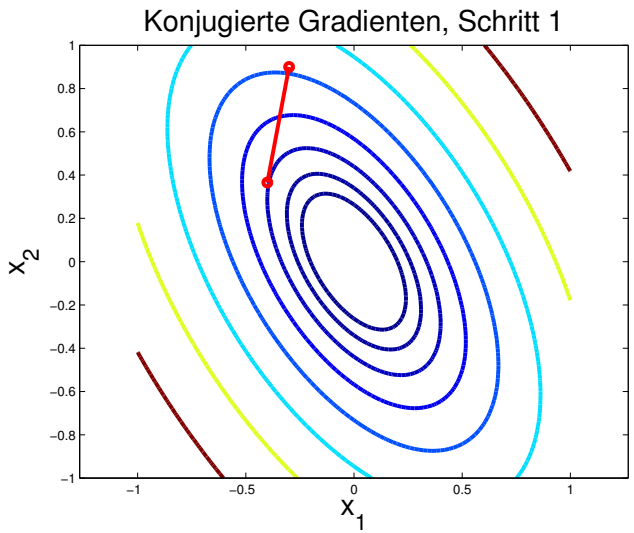
6.10 Das lineare Konjugierte-Gradienten-Verfahren

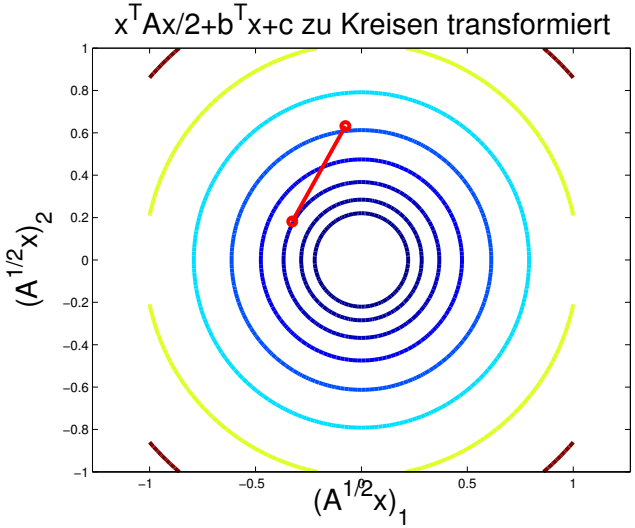
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen konjugiert bzgl. $A \succ 0$, falls $d^T Ah = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.
Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.





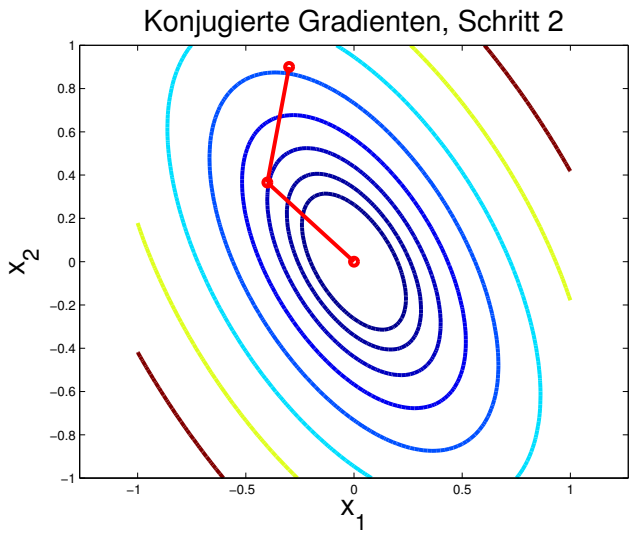
6.10 Das lineare Konjugierte-Gradienten-Verfahren

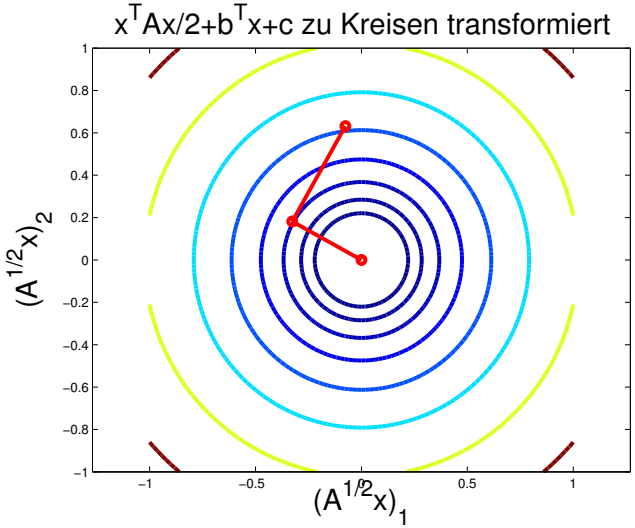
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen konjugiert bzgl. $A \succ 0$, falls $d^T Ah = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.
Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.





Konsequenz:
Das Optimum ist nach höchstens n Schritten gefunden.
[Nur theoretisch!]

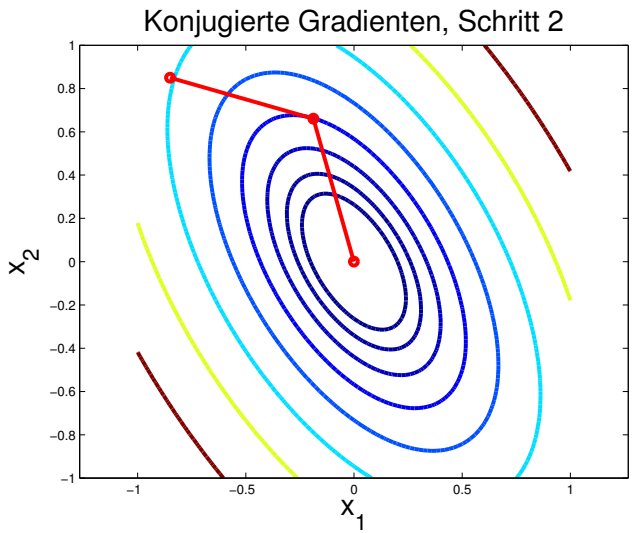
6.10 Das lineare Konjugierte-Gradienten-Verfahren

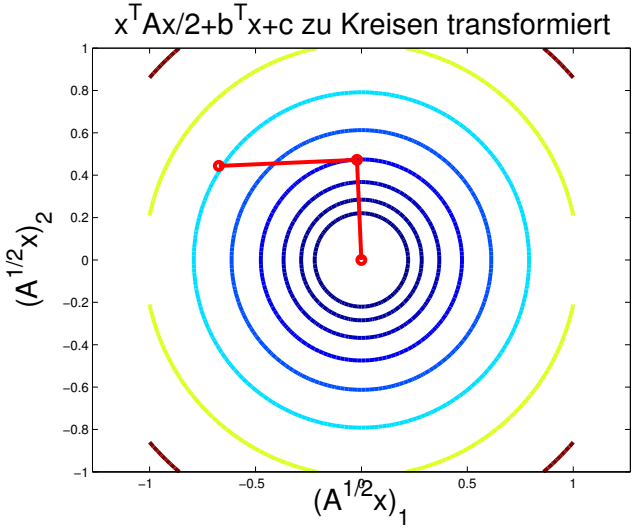
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen konjugiert bzgl. $A \succ 0$, falls $d^T Ah = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.
Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.





Konsequenz:
Das Optimum ist nach höchstens n Schritten gefunden.
[Nur theoretisch!]

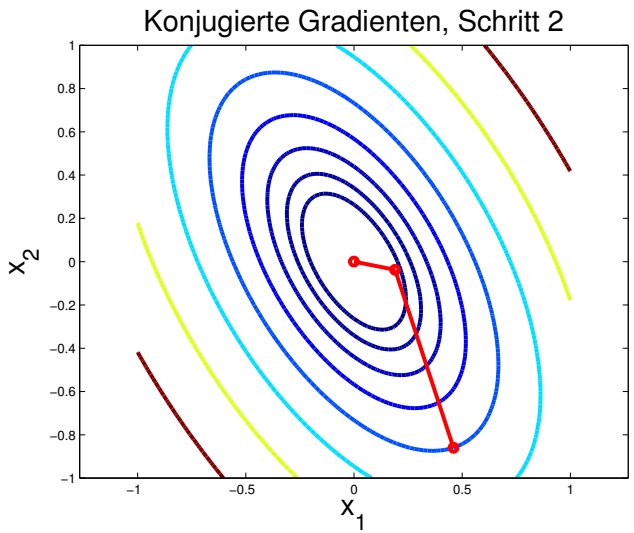
6.10 Das lineare Konjugierte-Gradienten-Verfahren

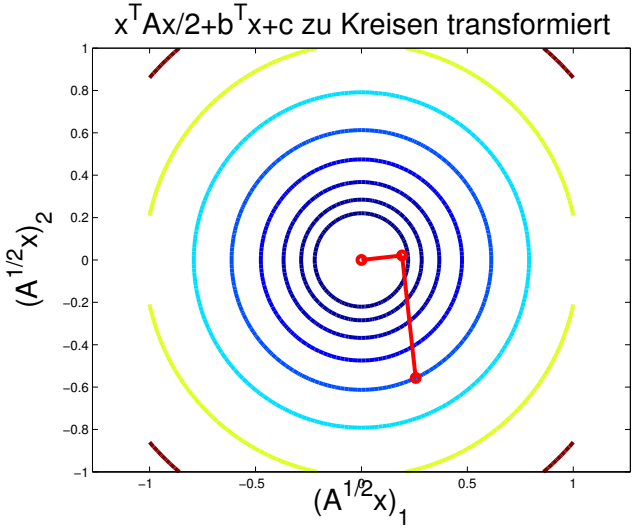
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen konjugiert bzgl. $A \succ 0$, falls $d^T Ah = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.
Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.





Konsequenz:
Das Optimum ist nach höchstens n Schritten gefunden.
[Nur theoretisch!]

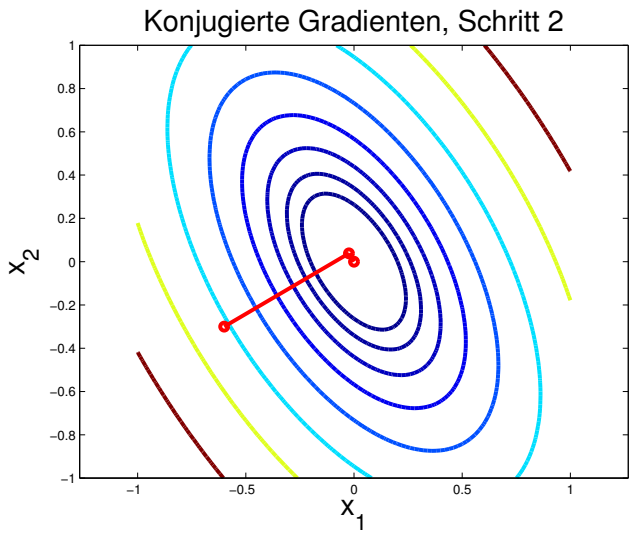
6.10 Das lineare Konjugierte-Gradienten-Verfahren

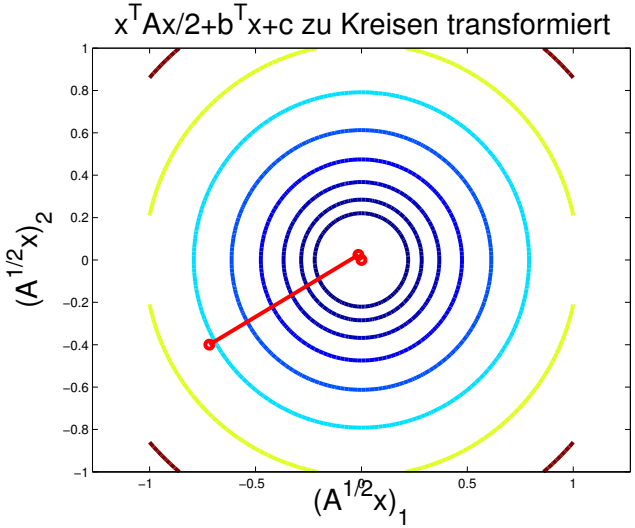
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.

Zwei Richtungen $d, h \in \mathbb{R}^n$ heißen konjugiert bzgl. $A \succ 0$, falls $d^T Ah = 0$.

Geometrisch: Niveaumengen sind Ellipsen, also affine Bilder von Kreisen.
Konjugierte Richtungen sind darin die Bilder orthogonaler Richtungen.





Konsequenz:
Das Optimum ist nach höchstens n Schritten gefunden.
[Nur theoretisch!]

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \text{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \text{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)}$ $\quad [\in \operatorname{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$

   Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.

Aus exaktem Line-Search folgt $(r^{(k+1)})^T h^{(k)} = 0$

und für jedes $0 \leq i < k$, wegen $x^{(k+1)} = x^{(i+1)} + \sum_{j=i+1}^{k} \alpha_j h^{(j)}$,

$$(h^{(i)})^T (r^{(k+1)}) = \underbrace{(h^{(i)})^T [Ax^{(i+1)} - b]}_{=(h^{(i)})^T r^{(i+1)} = 0} + \sum_{j=i+1}^{k} \alpha_i \underbrace{(h^{(i)})^T A h^{(j)}}_{=0} = 0$$

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)}$ $[\in \operatorname{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$
   Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$ $[\in \operatorname{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$
   $\beta_{k+1} := \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T A h^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.

Wähle β_{k+1} so, dass $h^{(k+1)}$ konjugiert zu $h^{(k)}$: $0 = (h^{(k+1)})^T A h^{(k)}$, also
$0 = -(r^{(k+1)})^T A h^{(k)} + \beta_{k+1} (h^{(k)})^T A h^{(k)} \Rightarrow \beta_k = \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$.

Für jedes $0 \leq i \leq k$ gilt (mit $\beta_0 = 0$) wegen $r^{(i)} = \beta_i h^{(i-1)} - h^{(i)}$
$(r^{(i)})^T r^{(k+1)} = \beta_i (h^{(i-1)})^T r^{(k+1)} - (h^{(i)})^T r^{(k+1)} = 0$
und jedes $0 \leq i < k$ wegen $A h^{(i)} = (r^{(i+1)} - r^{(i)})/\alpha_i$
$(h^{(k+1)})^T A h^{(i)} = -(r^{(k+1)})^T A h^{(i)} + \beta_{k+1} \underbrace{(h^{(k)})^T A h^{(i)}}_{=0} = -\frac{(r^{(k+1)})^T (r^{(i+1)} - r^{(i)})}{\alpha_i} = 0$

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \text{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)}$ $[\in \text{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$
   Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$ $[\in \text{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$
   $\beta_{k+1} := \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T A h^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.
6. $k \leftarrow k + 1$, gehe zu 1.

(Die Schritte des linearen CG-Verfahrens)

0. $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b$ $[= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
1. Falls $\|r^{(k)}\| < \varepsilon$ STOP.
2. exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)}$ $[\in \operatorname{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$
   Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$ $[\in \operatorname{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$
   $\beta_{k+1} := \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T A h^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.
6. $k \leftarrow k + 1$, gehe zu 1.

Satz (Lineares CG-Verfahren)

Ist in Iteration k der Gradient $r_k \neq 0$, so gilt
$\operatorname{span}\{r^{(0)}, \ldots, A^k r^{(0)}\} = \operatorname{span}\{r^{(0)}, \ldots, r^{(k)}\} = \operatorname{span}\{h^{(0)}, \ldots, h^{(k)}\}$
und $(r^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T A h^{(k)} = 0$ *für* $0 \leq i < k$.
Insbesondere ist $x^{(k)} = \operatorname{argmin}\{f(x) : x \in x^{(0)} + \operatorname{span}\{r^{(0)}, \ldots, A^{k-1} r^{(0)}\}\}$
und nach höchstens n Iterationen endet das Verfahren mit $r^{(k)} = 0$.

(Der lineare CG-Algorithmus in Standardform)

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

Für α: $(r^{(k)})^T h^{(k)} = (r^{(k)})^T(-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$

Für β: $(r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T(r^{(k+1)} - r^{(k)})/\alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k + 1$, GOTO 1.

(Der lineare CG-Algorithmus in Standardform)

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

Für α: $(r^{(k)})^T h^{(k)} = (r^{(k)})^T(-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$

Für β: $(r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T(r^{(k+1)} - r^{(k)})/\alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k+1$, GOTO 1.

- Pro Iteration wird nur eine Matrix-Vektor-Multiplikation benötigt.
  [Besonders effizient für dünn besetzte Matrix A oder Hessian-free]

(Der lineare CG-Algorithmus in Standardform)

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

Für α: $(r^{(k)})^T h^{(k)} = (r^{(k)})^T(-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$

Für β: $(r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T(r^{(k+1)} - r^{(k)})/\alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k+1$, GOTO 1.

- Pro Iteration wird nur eine Matrix-Vektor-Multiplikation benötigt.
  [Besonders effizient für dünn besetzte Matrix A oder Hessian-free]
- Stellenauslöschung verursacht numerische Probleme, $k \leq n$ reicht nicht.

$\rightarrow$ In der Praxis ist nur mit Präkonditionierung gute Genauigkeit erzielbar.

[Präkond. $\approx$ Koordinatentrafo Richtung Kreis, stark problemabhängig!]

(Das nichtlineare Konjugierte-Gradienten-Verfahren)

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

(Das nichtlineare Konjugierte-Gradienten-Verfahren)

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

(Das nichtlineare Konjugierte-Gradienten-Verfahren)

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung

(Das nichtlineare Konjugierte-Gradienten-Verfahren)

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz

(Das nichtlineare Konjugierte-Gradienten-Verfahren)

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz
- Gilbert-Nocedal: $\beta_{k+1}^{GN} = \max\{\beta_{k+1}^{PR}, 0\}$ praktisch gut, mit Konvergenz

(Das nichtlineare Konjugierte-Gradienten-Verfahren)

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz
- Gilbert-Nocedal: $\beta_{k+1}^{GN} = \max\{\beta_{k+1}^{PR}, 0\}$ praktisch gut, mit Konvergenz
- Regelm. Neustart oder $\beta_k = 0$ ist jeweils ein steilster Abstiegsschritt

$\rightarrow$ global konvergent

Inhalt

6.11 Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

6.11 Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Iterative Verfahren brechen ab, wenn das Residuum

$$r^{(k)} = \nabla^2 f_k h^{(k)} + \nabla f_k$$

klein genug ist. Inexakte Newton-Verfahren fordern

$$\text{(INC)} \qquad \|r^{(k)}\| \leq \eta_k \|\nabla f_k\|$$

und geben dafür eine forcing sequence $\eta_k \in (0, 1)$ vor.

6.11 Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Iterative Verfahren brechen ab, wenn das Residuum

$$r^{(k)} = \nabla^2 f_k h^{(k)} + \nabla f_k$$

klein genug ist. Inexakte Newton-Verfahren fordern

$$\text{(INC)} \qquad \|r^{(k)}\| \leq \eta_k \|\nabla f_k\|$$

und geben dafür eine forcing sequence $\eta_k \in (0, 1)$ vor.

Satz (Inexakte Newton-Verfahren)

Sei f hinr. glatt und x^* erfülle die hinr. Opt.-Bed. Ist $x^{(0)}$ nahe genug bei x^*, so gilt für jede Punktfolge $x^{(k+1)} = x^{(k)} + h^{(k)}$ mit $h^{(k)}$ erfüllt (INC):
Falls $\eta_k \leq \eta < 1$, konvergiert $x^{(k)}$ linear gegen x^*.
Falls $\eta_k \to 0$, konvergiert $x^{(k)}$ superlinear gegen x^*.
Falls $\eta_k \leq \gamma \|\nabla f_k\|$ mit festem $\gamma > 0$, konvergieren die $x^{(k)}$ quadratisch.

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$,[„Newton“-Schritt] sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not> 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$,[„Newton“-Schritt] sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.
3. $k \leftarrow k + 1$, GOTO 1.

- Das CG-Verfahren garantiert: $h^{(k)}$ ist Abstiegsrichtung.
  $p^{(i)}$ minimiert $m(p) = \frac{1}{2} p^T \nabla^2 f_k p + \nabla f_k^T p$ über $\text{span}\{d^{(0)}, \ldots, d^{(i-1)}\}$.
  Weil $p^{(i)} = \sum_{j=0}^{i-1} \xi_j d^{(j)}$, $(d^{(j)})^T \nabla^2 f_k d^{(j)} > 0$ und $(d^{(j)})^T \nabla^2 f_k d^{(l)} = 0$ $(j \neq l)$
  folgt $0 = m(0) \geq m(p^{(i)}) = \nabla f_k^T p^{(i)} + \frac{1}{2} \sum_{j=0}^{i-1} \xi_j^2 (d^{(j)})^T \nabla^2 f_k d^{(j)}$

(Das Line-Search-Newton-CG-Verfahren)

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$,[„Newton“-Schritt] sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.
3. $k \leftarrow k + 1$, GOTO 1.

- Das CG-Verfahren garantiert: $h^{(k)}$ ist Abstiegsrichtung.
- CG benötigt nur Hessematrix mal Vektor $\rightarrow$ *Hessian-free*
- Immer zuerst Schrittlänge 1 für superlineare Konvergenz versuchen!
- CG braucht problemspezifische Präkonditionierer, gefährdet gute Konv.

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h $[g = \nabla f$, Newton: $B = \nabla^2 f]$

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h $[g = \nabla f$, Newton: $B = \nabla^2 f]$

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h $[g = \nabla f$, Newton: $B = \nabla^2 f]$

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$
8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$
8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$
9. $i \leftarrow i + 1$, GOTO 1.

(Das Steihaug CG-Verf. für das Trust-Region-Problem)

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$
8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$
9. $i \leftarrow i + 1$, GOTO 1.

- $p^{(1)}$ ist Cauchy-Punkt, CG-Verfahren verbessert monoton $\Rightarrow$ global konv.
- Benötigt auch nur Hessematrix mal Vektor $\rightarrow$ Hessian-free
- Es gilt $\|p^{(0)}\| < \cdots < \|p^{(i)}\| < \cdots \leq \Delta \Rightarrow$ lokal inexakter Newton

Inhalt

6.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

6.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.

6.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.
Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte $\tilde{y}$ mit Wahrscheinlichkeit

$$p(\tilde{y}; x, t) := \prod_{j=1}^{m} \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{m}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{m}[\tilde{y}_j - \Phi(t_j; x)]^2\right)$$

auftreten.

6.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.
Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte $\tilde{y}$ mit Wahrscheinlichkeit

$$p(\tilde{y}; x, t) := \prod_{j=1}^{m} \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{m}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{m}[\tilde{y}_j - \Phi(t_j; x)]^2\right)$$

auftreten. Betrachtet man $p(\tilde{y}; x, t)$ als Likelihood-Funktion dafür, dass x die korrekten Parameter sind, ist die beste Parameterwahl für $\tilde{y}$ und t in diesem Sinn ein Maximum-Likelihood-Schätzer $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

6.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.
Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte $\tilde{y}$ mit Wahrscheinlichkeit

$$p(\tilde{y}; x, t) := \prod_{j=1}^{m} \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{m}{2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{j=1}^{m}[\tilde{y}_j - \Phi(t_j; x)]^2\right)$$

auftreten. Betrachtet man $p(\tilde{y}; x, t)$ als Likelihood-Funktion dafür, dass x die korrekten Parameter sind, ist die beste Parameterwahl für $\tilde{y}$ und t in diesem Sinn ein Maximum-Likelihood-Schätzer $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

$$\longrightarrow \text{Finde } \bar{x} \in \operatorname{Argmin} f(x) := \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit } r_j(x) := \tilde{y}_j - \Phi(t_j; x).$$

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\quad\rightarrow\quad$ $\min_x f(x) := \frac{1}{2}\|Ax - b\|^2$

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\quad\rightarrow\quad \min_x f(x) := \frac{1}{2}\|Ax-b\|^2$

Wegen $\frac{1}{2}\|Ax-b\|^2 = \frac{1}{2}(Ax-b)^T(Ax-b) = \frac{1}{2}x^TA^TAx - x^TA^Tb + \frac{1}{2}b^Tb$ und $A^TA \succeq 0$, ist das Problem konvex quadratisch und $\nabla f(x) = 0$ ist hinreichende Optimalitätsbedingung.

x^* ist Optimallösung $\Leftrightarrow$ x^* erfüllt die Normalgleichungen $A^TAx = A^Tb$.

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\quad\rightarrow\quad \min_x f(x) := \frac{1}{2}\|Ax-b\|^2$

Wegen $\frac{1}{2}\|Ax-b\|^2 = \frac{1}{2}(Ax-b)^T(Ax-b) = \frac{1}{2}x^TA^TAx - x^TA^Tb + \frac{1}{2}b^Tb$ und $A^TA \succeq 0$, ist das Problem konvex quadratisch und $\nabla f(x) = 0$ ist hinreichende Optimalitätsbedingung.

x^* ist Optimallösung $\Leftrightarrow$ x^* erfüllt die Normalgleichungen $A^TAx = A^Tb$.

Da $A^TA \succeq 0$, bieten sich viele numerische Verfahren zur Lösung an:

- Cholesky-Faktorisierung von A^TA (mit pivotisieren),
- QR-Faktorisierung von A (recht stabil, wird aber dicht besetzt),
- SVD-Dekomposition (am stabilsten und teuersten)
- Präkonditionierte CG-Verfahren (PCG)
  (für Näherungslösung im *large scale*-Bereich, falls Av und A^Tw billig)

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).
Betrachte Gradienten und Hessematrix von f (Kettenregel nutzen):

$$\nabla f(x) \quad = \quad \sum_{j=1}^{m} r_j(x)\nabla r_j(x) = J_r(x)^T r(x)$$

Für eine Funktion $g : \mathbb{R}^n \to \mathbb{R}^m$ mit $g(x) = (g_1(x), \ldots, g_m(x))^T$ ist

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \ldots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \ldots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix} = \begin{bmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix}$$

die Jacobimatrix von g in x. So ist z.B. $\nabla^2 f(x) = J_{\nabla f}(x)$.
$g(x+h) = g(x) + J_g(x)h + \mathbf{o}(h) \quad \longrightarrow$ das lineare Modell von g in x.

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).
Betrachte Gradienten und Hessematrix von f (Kettenregel nutzen):

$$\begin{aligned} \nabla f(x) &= \textstyle\sum_{j=1}^m r_j(x)\nabla r_j(x) = J_r(x)^T r(x) \\ \nabla^2 f(x) &= \textstyle\sum_{j=1}^m \nabla r_j(x)\nabla r_j(x)^T + \sum_{j=1}^m r_j(x)\nabla^2 r_j(x) \\ &= J_r(x)^T J_r(x) + \textstyle\sum_{j=1}^m \underbrace{r_j(x)}_{\text{klein}} \nabla^2 r_j(x) \approx J_r(x)^T J_r(x) \end{aligned}$$

$\rightarrow$ die Jacobimatrix liefert eine gute Näherung für die Hessematrix.

Für eine Funktion $g : \mathbb{R}^n \to \mathbb{R}^m$ mit $g(x) = (g_1(x), \ldots, g_m(x))^T$ ist

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \cdots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \cdots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix} = \begin{bmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix}$$

die Jacobimatrix von g in x. So ist z.B. $\nabla^2 f(x) = J_{\nabla f}(x)$.

$g(x+h) = g(x) + J_g(x)h + \mathbf{o}(h) \quad \longrightarrow$ das lineare Modell von g in x.

6.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2}\|r(x)\|^2$ mit $r : \mathbb{R}^n \to \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k \; h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}),\; r_k := r(x^{(k)})$$

6.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2}\|r(x)\|^2$ mit $r : \mathbb{R}^n \to \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k \; h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}),\; r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$

6.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2}\|r(x)\|^2$ mit $r : \mathbb{R}^n \to \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k \; h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}),\; r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\text{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) und $\nabla f_k \neq 0$, so ist h_k^{GN} Abstiegsrichtung: $(h_k^{GN})^T \nabla f_k = (h_k^{GN})^T J_k^T r_k = -(h_k^{GN})^T J_k^T J_k \; h_k^{GN} = -\|J_k h_k^{GN}\|^2 < 0$, denn wegen $J_k^T J_k \succ 0$ ist $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.

6.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2}\|r(x)\|^2$ mit $r : \mathbb{R}^n \to \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k \; h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}),\; r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\text{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) und $\nabla f_k \neq 0$, so ist h_k^{GN} Abstiegsrichtung:
  $(h_k^{GN})^T \nabla f_k = (h_k^{GN})^T J_k^T r_k = -(h_k^{GN})^T J_k^T J_k \; h_k^{GN} = -\|J_k h_k^{GN}\|^2 < 0$,
  denn wegen $J_k^T J_k \succ 0$ ist $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.
- Die Gleichung für h_k^{GN} hat die Form $A^T A x = A^T b$ (Normalgleichungen), h_k^{GN} ist Lösung des linearen kleinste Quadrate Problems
  $$\min_h \|J_k^T h - r_k\|^2$$
  $\Rightarrow$ alle numerischen Verfahren des linearen Falls sind verwendbar.

6.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2}\|r(x)\|^2$ mit $r : \mathbb{R}^n \to \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k \; h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}),\; r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\text{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) und $\nabla f_k \neq 0$, so ist h_k^{GN} Abstiegsrichtung:
  $(h_k^{GN})^T \nabla f_k = (h_k^{GN})^T J_k^T r_k = -(h_k^{GN})^T J_k^T J_k \; h_k^{GN} = -\|J_k h_k^{GN}\|^2 < 0$,
  denn wegen $J_k^T J_k \succ 0$ ist $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.
- Die Gleichung für h_k^{GN} hat die Form $A^T A x = A^T b$ (Normalgleichungen), h_k^{GN} ist Lösung des linearen kleinste Quadrate Problems
  $$\min_h \|J_k^T h - r_k\|^2$$
  $\Rightarrow$ alle numerischen Verfahren des linearen Falls sind verwendbar.
- Ist $\text{Rang}(J_k) < n$, verwendet man im Levenberg-Marquardt-Verfahren
  $$(J_k^T J_k + \lambda I)h = -J_k r_k,$$
  mit ähnlichen Anpassungsregeln für λ wie für die Trust-Region.
  [Interpretiere λ als Lagrange-Multiplikator für die Trust-Region-Constraint.]

Inhalt

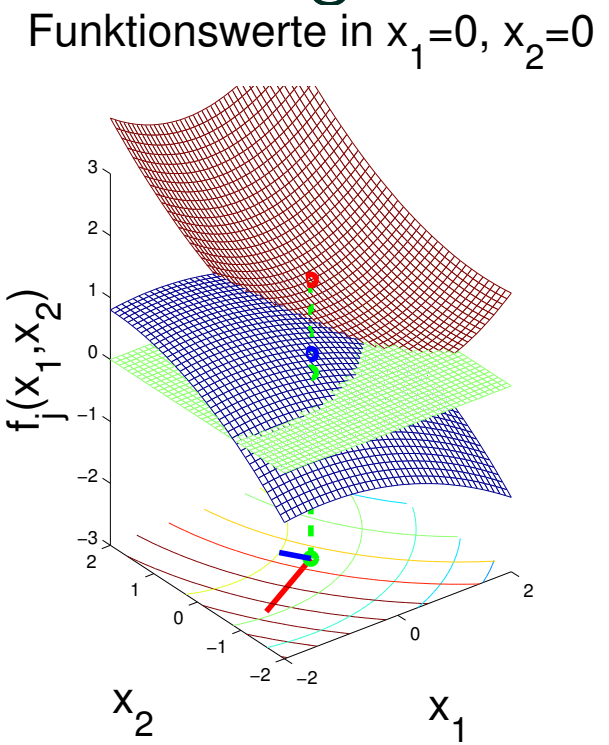
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

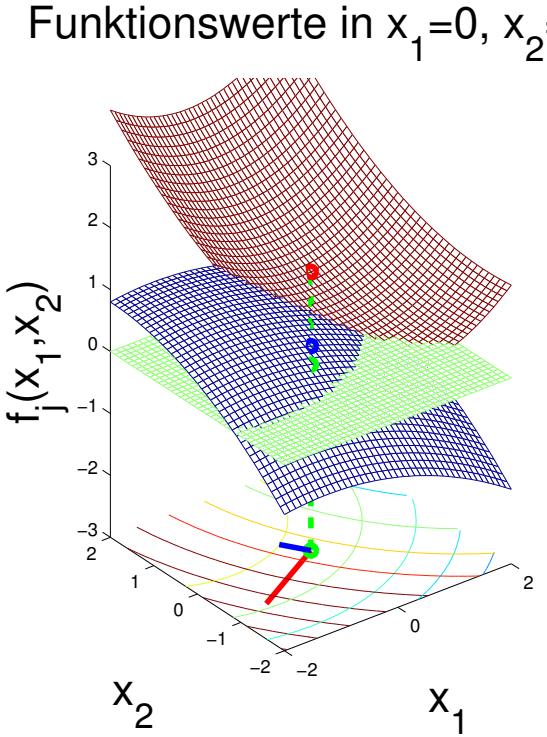
(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$



6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

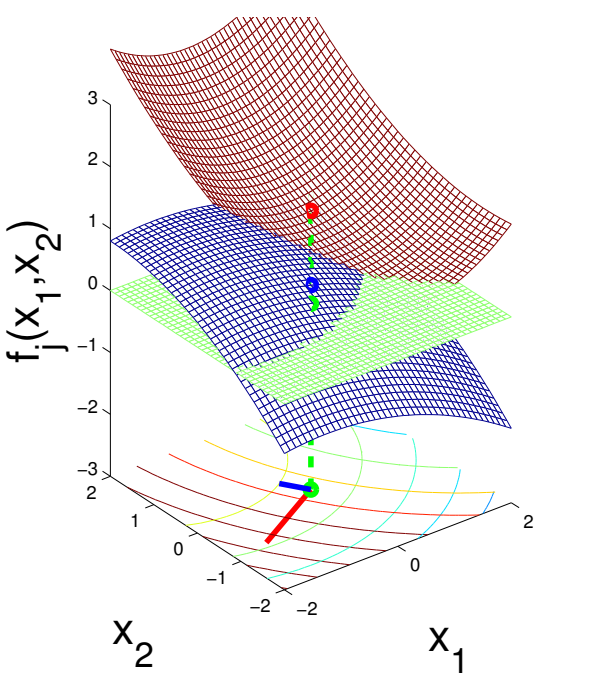
Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$



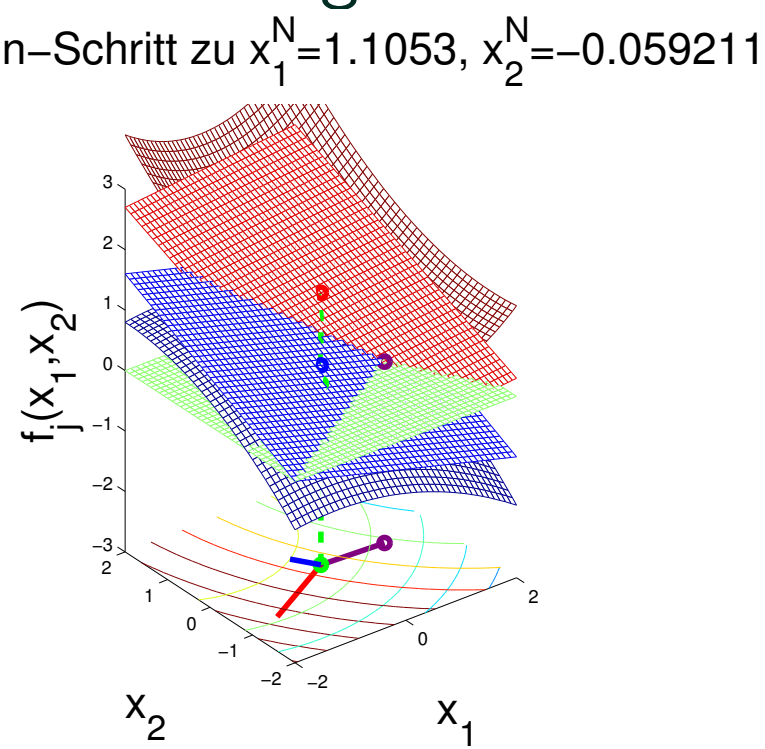
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

6.13 Newton für nichtlineare Gleichungen

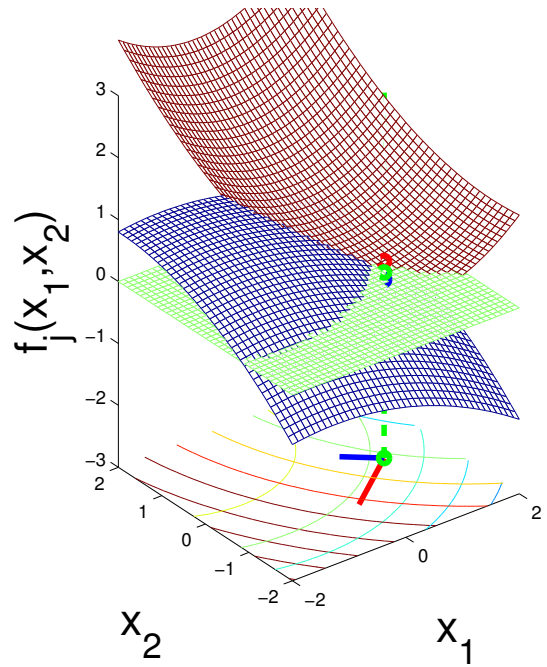
Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Funktionswerte in x_1=1.1053, x_2=–0.059211



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

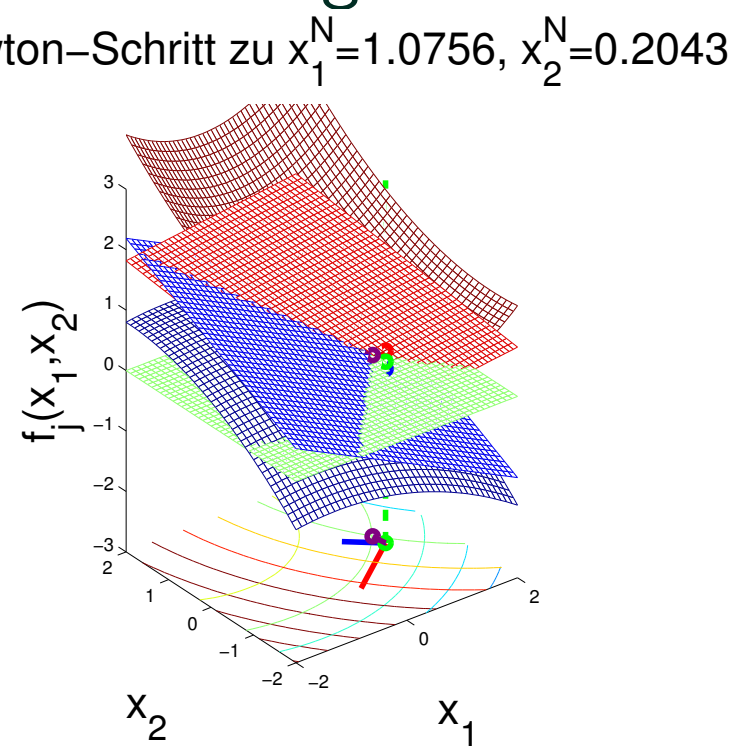
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

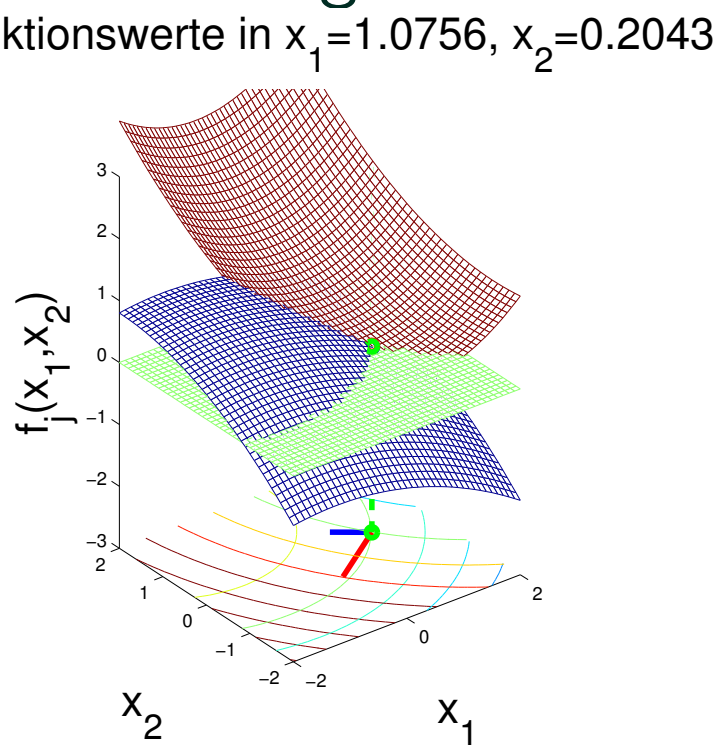
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

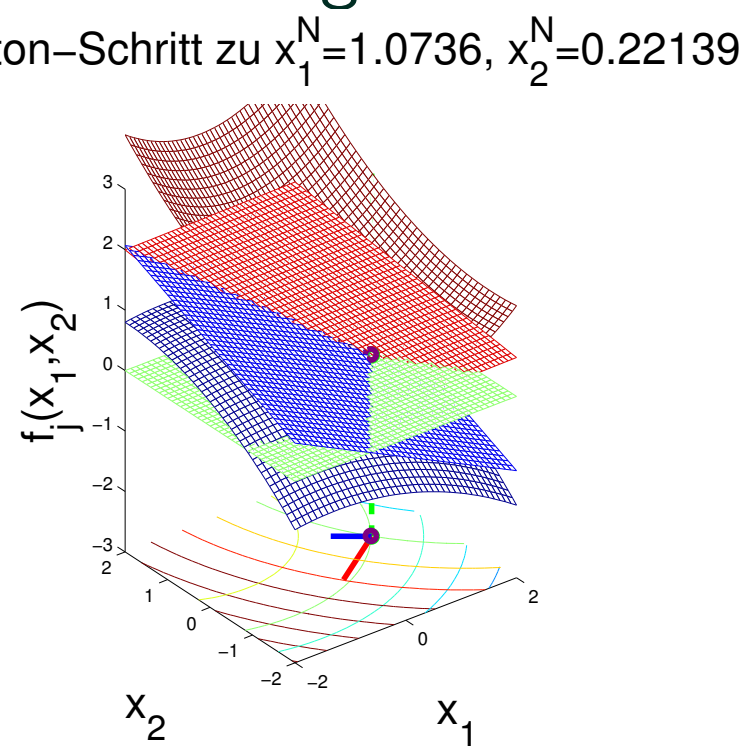
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

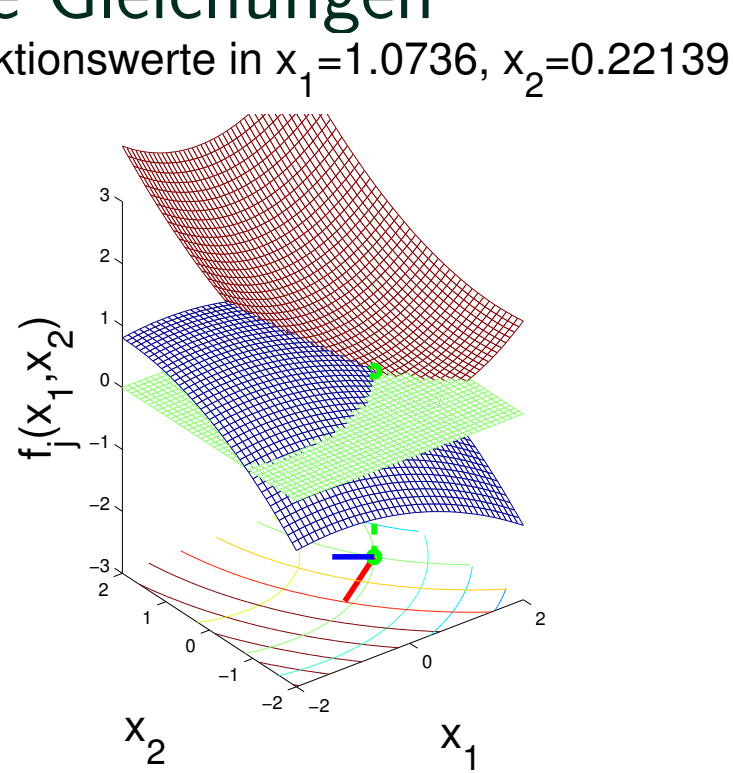
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

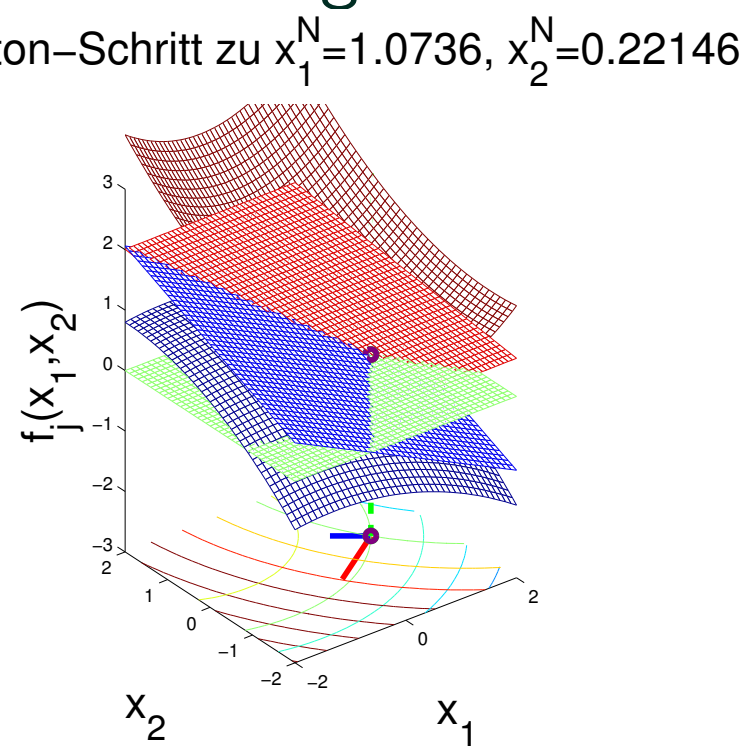
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

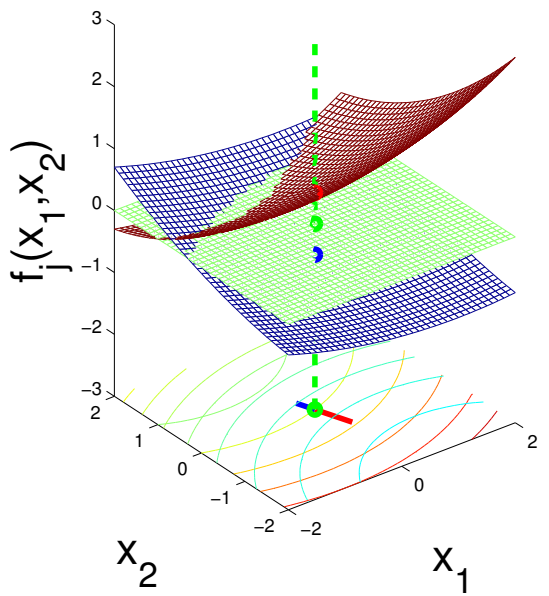
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

Ist $J_F(x^{(k)})$ singulär ($\Leftrightarrow$ die ∇f_j lin. abh.), versagt es.

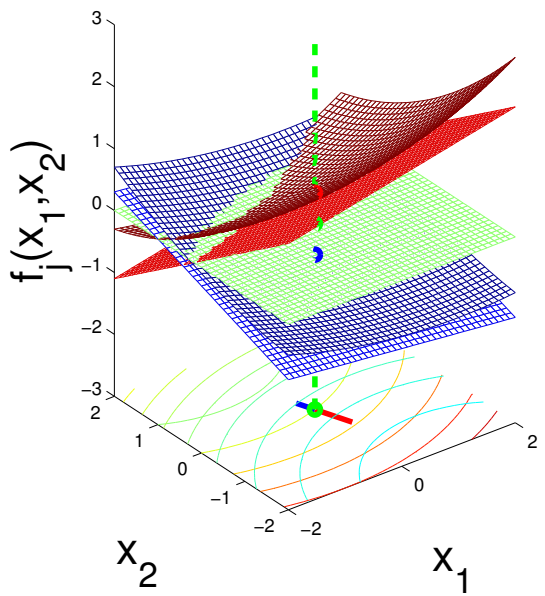
6.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

Ist $J_F(x^{(k)})$ singulär ($\Leftrightarrow$ die ∇f_j lin. abh.), versagt es.

6.13 Newton für nichtlineare Gleichungen

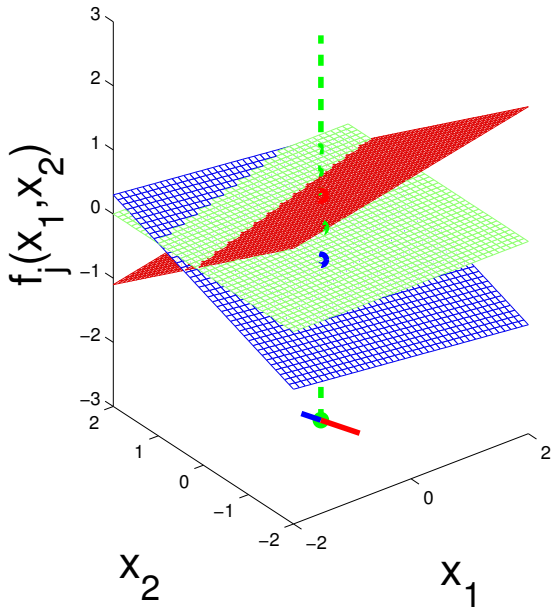
Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Rang(Jacobimatrix)=1



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

Ist $J_F(x^{(k)})$ singulär ($\Leftrightarrow$ die ∇f_j lin. abh.), versagt es.

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$, $F(x) := \nabla f(x) = 0$, $J_F(x) = \nabla^2 f(x)$

$$0 = F(x) + J_F(x)h = \nabla f(x) + \nabla^2 f(x) h \Rightarrow h_N = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$, $F(x) := \nabla f(x) = 0$, $J_F(x) = \nabla^2 f(x)$

$$0 = F(x) + J_F(x)h = \nabla f(x) + \nabla^2 f(x) h \Rightarrow h_N = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Zur Globalisierung wird der Fortschritt über eine *merit-function*, meist $f(x) := \frac{1}{2}\|F(x)\|^2$ (s. nichtlin. kleinste Quadrate), bewertet. h_{NF} ist (falls definiert) Abstiegsrichtung für f: $\nabla f_k^T h_{NF}^{(k)} = -F_k^T J_k J_k^{-1} F_k = -\|F_k\|^2 < 0$.

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$, $F(x) := \nabla f(x) = 0$, $J_F(x) = \nabla^2 f(x)$

$$0 = F(x) + J_F(x)h = \nabla f(x) + \nabla^2 f(x) h \Rightarrow h_N = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Zur Globalisierung wird der Fortschritt über eine *merit-function*, meist $f(x) := \frac{1}{2}\|F(x)\|^2$ (s. nichtlin. kleinste Quadrate), bewertet. h_{NF} ist (falls definiert) Abstiegsrichtung für f: $\nabla f_k^T h_{NF}^{(k)} = -F_k^T J_k J_k^{-1} F_k = -\|F_k\|^2 < 0$.
Vorsicht: Lokale Minima der Merit-Funktion sind i.A. keine Lösungen!