

Überblick Nichtlineare Optimierung

zur Vorlesung von Christoph Helmberg

Inhaltsübersicht

Freie Nichtlineare Optimierung

Restringierte Nichtlineare Optimierung: Grundlagen

Restringierte Optimierung: Verfahren

Ableitungsfreie Optimierung/Direkte Suchverfahren

Inhaltsübersicht

Freie Nichtlineare Optimierung

Restringierte Nichtlineare Optimierung: Grundlagen

Restringierte Optimierung: Verfahren

Ableitungsfreie Optimierung/Direkte Suchverfahren

1 Freie Nichtlineare Optimierung

Verfahren zur Minimierung glatter Funktionen ohne Nebenbedingungen,

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

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:

- Nichtlineare 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?

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.1 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.

1.1 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.

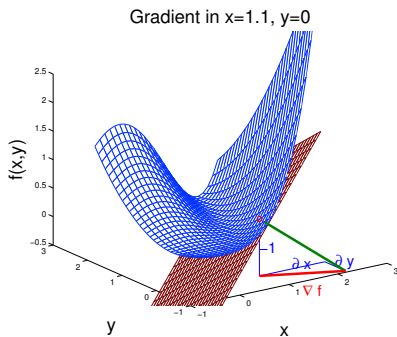
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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:

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

- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x .
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



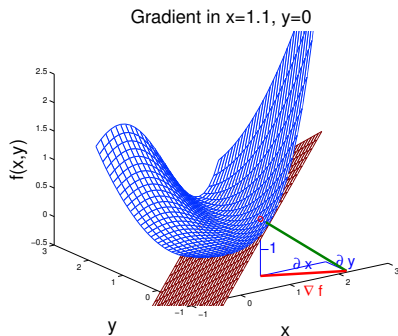
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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:

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

- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x .
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



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

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

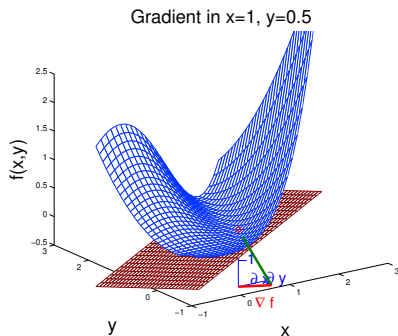
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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:

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

- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x .
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



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

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

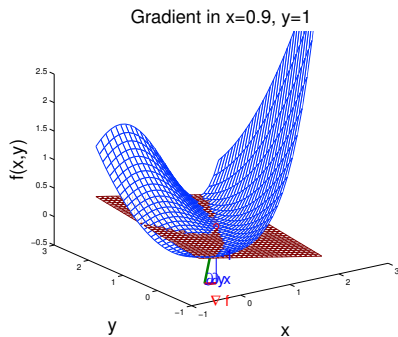
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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:

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

- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x .
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



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

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

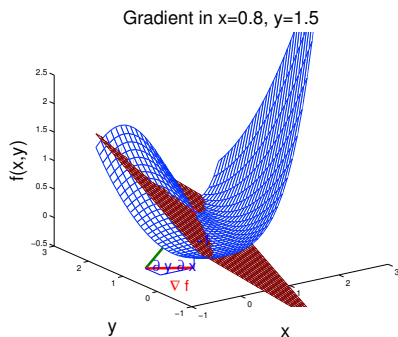
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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:

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

- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x .
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



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

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

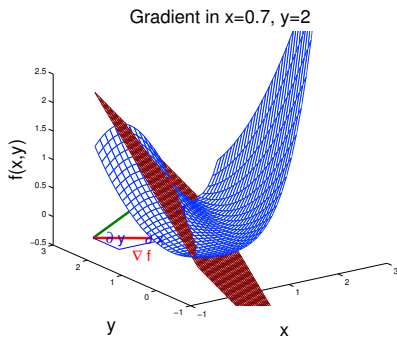
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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:

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

- $\nabla f(x)$ zeigt in Richtung des steilsten Anstiegs von f in x .
- $\|\nabla f(x)\|$ misst die Größe des Anstiegs.
- $\begin{bmatrix} \nabla f(x) \\ -1 \end{bmatrix}$ 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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.



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

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

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 $\Phi : \mathbb{R} \rightarrow \mathbb{R}$, $\Phi(\alpha) := f(\bar{x} + \alpha h)$]

Gradient und Richtungsableitung, lineares Modell

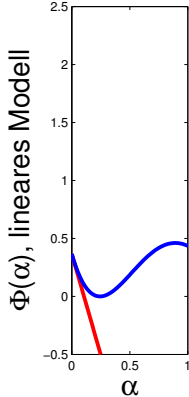
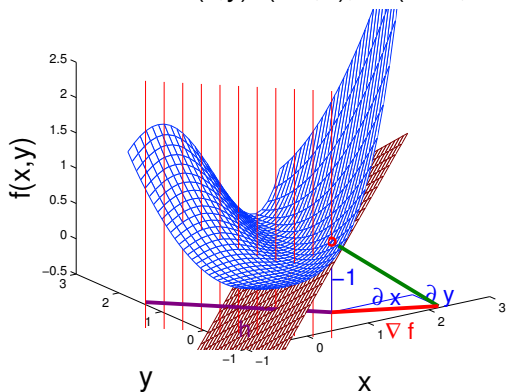
Das lineare Modell von f in \bar{x} hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in \bar{x} : die **Richtungsableitung von f in \bar{x} in Richtung h** ,

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

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

Gradient in $(x,y)=(1.1,0)$, $h=(-1.8,1.8)$

$(x,y)=(1.1,0)$, $h=(-1.8,1.8)$



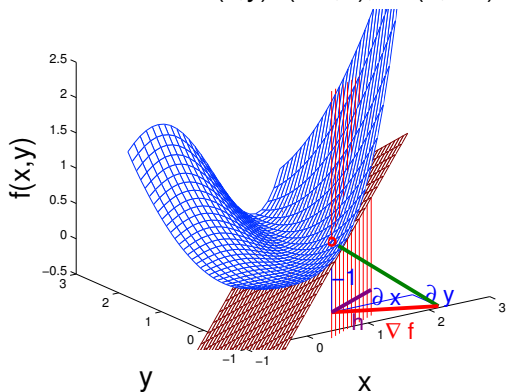
Gradient und Richtungsableitung, lineares Modell

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 ,**

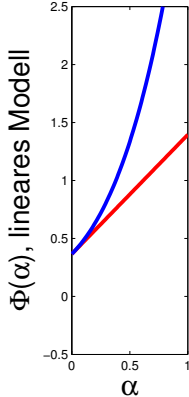
$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

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

Gradient in $(x,y)=(1.1,0)$, $h=(1,0.5)$



$(x,y)=(1.1,0)$, $h=(1,0.5)$



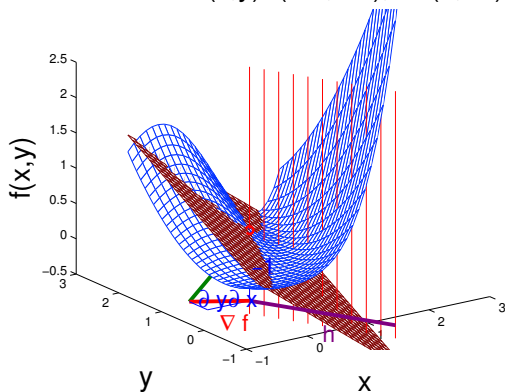
Gradient und Richtungsableitung, lineares Modell

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 ,**

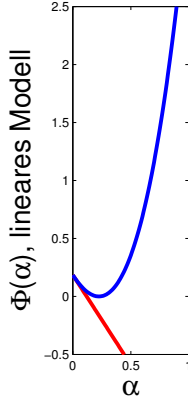
$$\nabla f(\bar{x})^T h = \lim_{\alpha \downarrow 0} \frac{f(\bar{x} + \alpha h) - f(\bar{x})}{\alpha} =: D_h f(\bar{x})$$

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

Gradient in $(x,y)=(0.8,1.5)$, $h=(1,-2)$



$(x,y)=(0.8,1.5)$, $h=(1,-2)$



Gradient und Richtungsableitung, lineares Modell

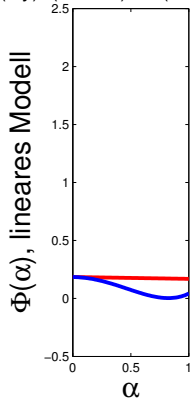
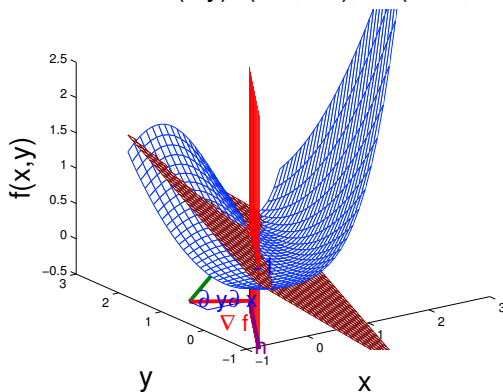
Das lineare Modell von f in \bar{x} hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in \bar{x} : die **Richtungsableitung von f in \bar{x} in Richtung h ,**

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

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

Gradient in $(x,y)=(0.8,1.5)$, $h=(-1.1,-1.8)$

$(x,y)=(0.8,1.5)$, $h=(-1.1,-1.8)$



Gradient und Richtungsableitung, lineares Modell

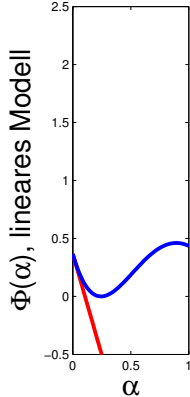
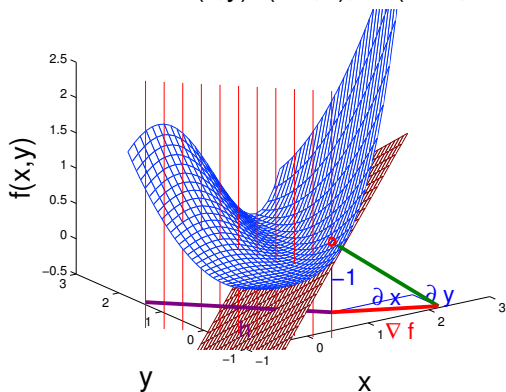
Das lineare Modell von f in \bar{x} hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in \bar{x} : die **Richtungsableitung von f in \bar{x} in Richtung h ,**

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

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

Gradient in $(x,y)=(1.1,0)$, $h=(-1.8,1.8)$

$(x,y)=(1.1,0)$, $h=(-1.8,1.8)$



Gradient und Richtungsableitung, lineares Modell

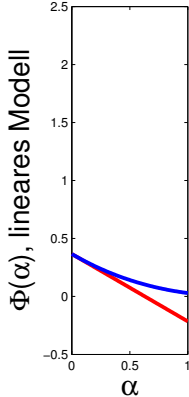
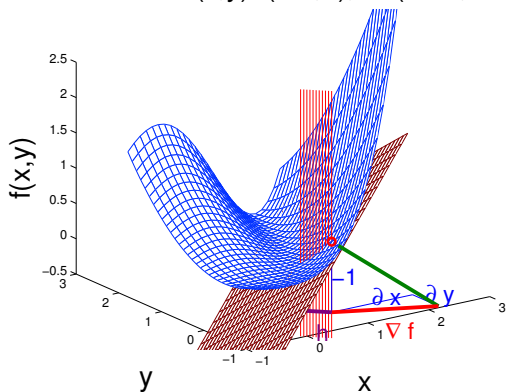
Das lineare Modell von f in \bar{x} hat in jede Richtung $h \in \mathbb{R}^n$ den gleichen Anstieg wie f in \bar{x} : die **Richtungsableitung von f in \bar{x} in Richtung h ,**

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

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

Gradient in $(x,y)=(1.1,0)$, $h=(-0.3,0.3)$

$(x,y)=(1.1,0)$, $h=(-0.3,0.3)$



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

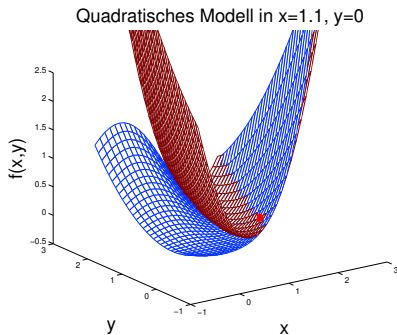
Für $\bar{x} \in \mathbb{R}^n$ werden $f(\bar{x})$, $\nabla f(\bar{x})$ und **Hessematrix** $\nabla^2 f(\bar{x})$ (2. Abl.) berechnet.

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.

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

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



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

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

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

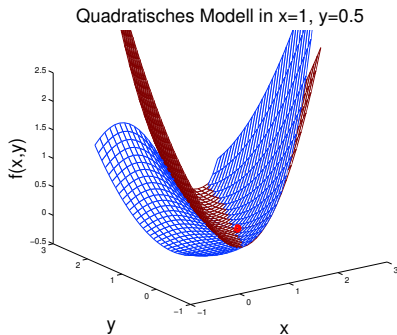
$$\lim_{x \rightarrow \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x}) - \frac{1}{2} (x - \bar{x})^T \nabla^2 f(\bar{x}) (x - \bar{x})}{\|x - \bar{x}\|^2} = 0.$$

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.

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

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



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

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

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

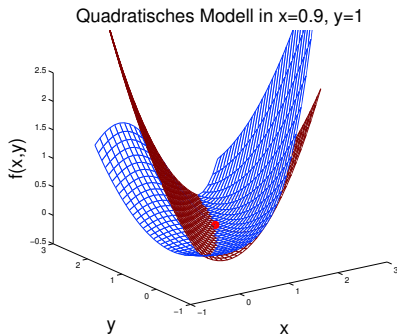
$$\lim_{x \rightarrow \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x}) - \frac{1}{2} (x - \bar{x})^T \nabla^2 f(\bar{x}) (x - \bar{x})}{\|x - \bar{x}\|^2} = 0.$$

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.

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

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



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

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

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

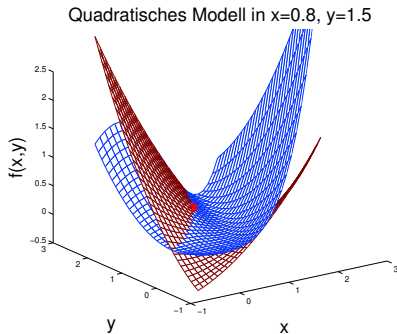
$$\lim_{x \rightarrow \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x}) - \frac{1}{2} (x - \bar{x})^T \nabla^2 f(\bar{x}) (x - \bar{x})}{\|x - \bar{x}\|^2} = 0.$$

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.

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

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



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

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

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

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

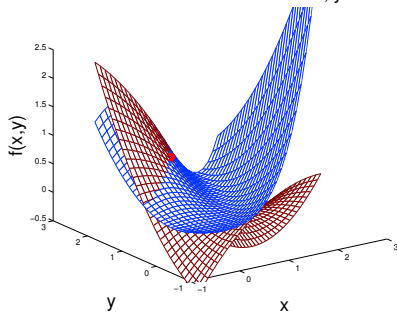
Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

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.

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

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

Quadratisches Modell in $x=0.7, y=2$



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

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

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

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

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} .

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

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

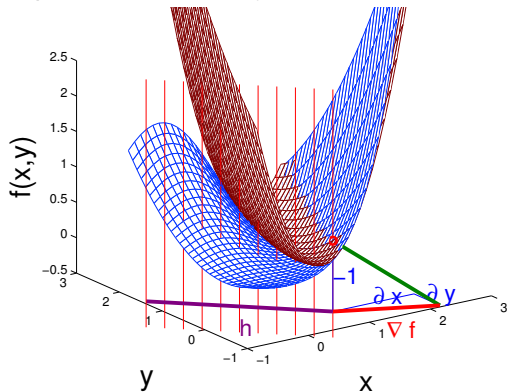
Quadratisches Modell in Richtung h

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} .

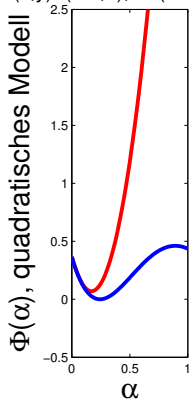
$$\tilde{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

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

quad. Modell in $(x,y)=(1.1,0)$, $h=(-1.8,1.8)$



$(x,y)=(1.1,0)$, $h=(-1.8,1.8)$



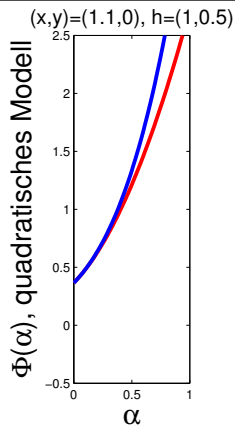
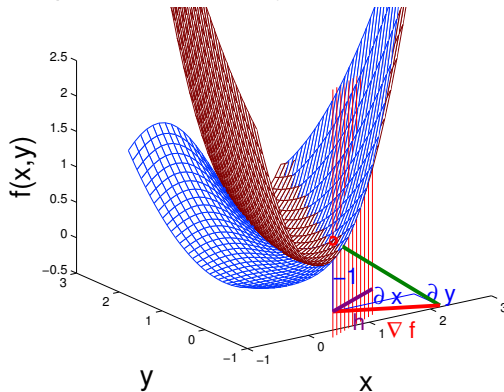
Quadratisches Modell in Richtung h

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} .

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

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

quad. Modell in $(x,y)=(1.1,0)$, $h=(1,0.5)$



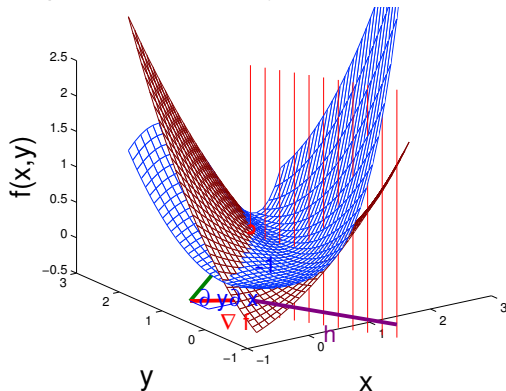
Quadratisches Modell in Richtung h

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} .

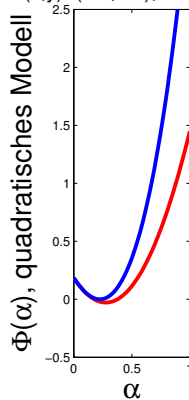
$$\check{f}_{\bar{x}}(\bar{x} + \alpha h) = f(\bar{x}) + \alpha \nabla f(\bar{x})^T h + \frac{\alpha^2}{2} h^T \nabla^2 f(\bar{x}) h.$$

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

quad. Modell in $(x,y)=(0.8,1.5)$, $h=(1,-2)$



$(x,y)=(0.8,1.5)$, $h=(1,-2)$



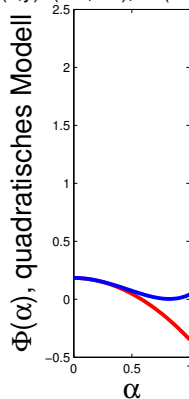
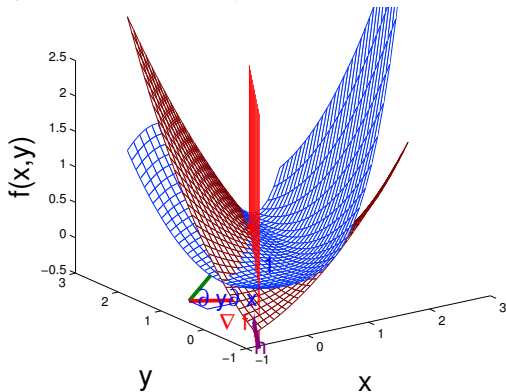
Quadratisches Modell in Richtung h

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} .

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

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

quad. Modell in $(x,y)=(0.8,1.5)$, $h=(-1.1,-1.8)$ $(x,y)=(0.8,1.5)$, $h=(-1.1,-1.8)$



Der Satz von Taylor/Mittelwertsatz

Satz (Taylor/Mittelwertsatz)

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

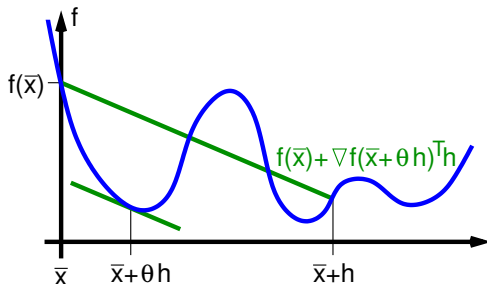
$$\exists \theta_1 \in (0, 1): f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x} + \theta_1 h)^T h,$$

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

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

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

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

Lipschitz-Stetigkeit, „Klein-o-Notation“

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

$$f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \mathbf{o}(\|h\|),$$

$$f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \frac{1}{2} h^T \nabla^2 f(\bar{x}) h + \mathbf{o}(\|h\|^2)$$

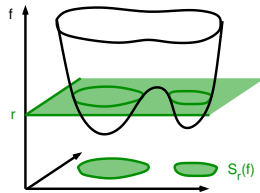
Das **Landau-Symbol** $\mathbf{o}(g(y))$ steht immer als Ersatz für eine nicht weiter interessierende Funktion $g'(y)$ mit der Eigenschaft $\lim_{y \rightarrow 0} \frac{g'(y)}{g(y)} \rightarrow 0$, also ein g' , das schneller klein wird als g .

Bei Folgen $g(y^{(k)})$ steht es für ein $g'(y^{(k)})$ mit $\lim_{k \rightarrow \infty} \frac{g'(y^{(k)})}{g(y^{(k)})} \rightarrow 0$.

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 $r \in \mathbb{R}$,

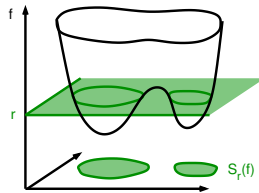
$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$



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“]

$$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\} \text{ (contour plots)}$$

helfen, Verfahren zu illustrieren.

[Höhenlinien in Landkarten, Wetterkarten]

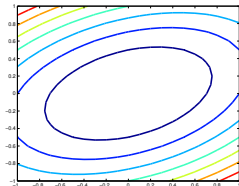
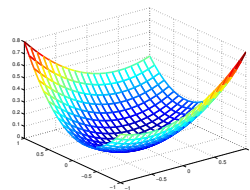
Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$

$$\text{gilt } 0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|).$$

Bsp: quadratische Funktion

$$\frac{1}{2}x^T Qx + q^T x + d$$

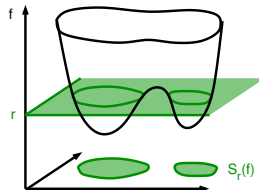
mit Q positiv definit.



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 $r \in \mathbb{R}$,

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

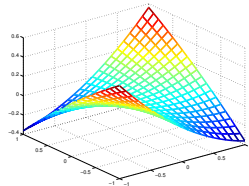


Funktionsdarstellungen über Niveaulinien [„Linien“]

$$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\} \text{ (contour plots)}$$

helfen, Verfahren zu illustrieren.

[Höhenlinien in Landkarten, Wetterkarten]



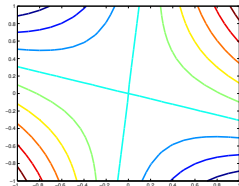
Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$

$$\text{gilt } 0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|).$$

Bsp: quadratische Funktion

$$\frac{1}{2}x^T Qx + q^T x + d$$

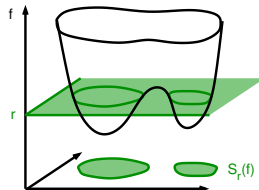
mit Q indefinit.



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“]

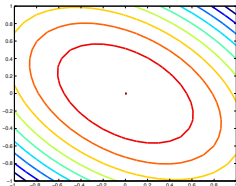
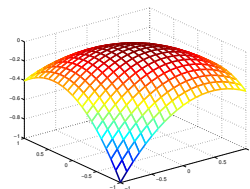
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (contour plots)

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

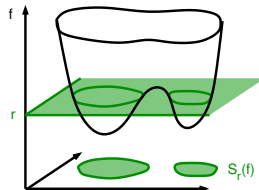
$$\frac{1}{2}x^T Qx + q^T x + d$$

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“]

$$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\} \text{ (contour plots)}$$

helfen, Verfahren zu illustrieren.

[Höhenlinien in Landkarten, Wetterkarten]

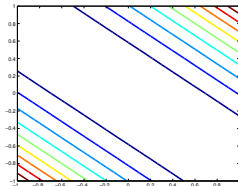
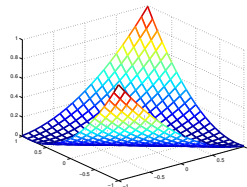
Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$

$$\text{gilt } 0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|).$$

Bsp: quadratische Funktion

$$\frac{1}{2}x^T Qx + q^T x + d$$

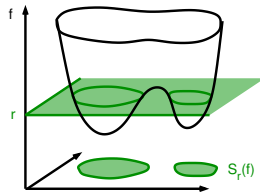
mit Q positiv semidefinit.



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“]

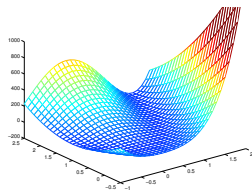
$N_r(f) := \{x \in \mathbb{R}^n : f(x) = r\}$ (contour plots)

helfen, Verfahren zu illustrieren.

[Höhenlinien in Landkarten, Wetterkarten]

Beachte: Der Gradient ist immer orthogonal zur Niveaulinie, denn für $x, x+h \in N_r(f)$

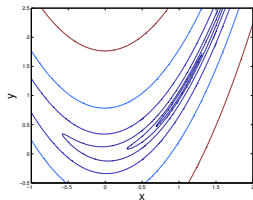
gilt $0 = f(x+h) - f(x) = \nabla f(x)^T h + \mathbf{o}(\|h\|)$.



Bsp: Rosenbrock-Funktion (*banana shape*)

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

Minimum wird in (1,1) angenommen.



Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen**
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

1.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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 $f(x) = -x^2$.

1.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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!

1.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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: $f(x) = \frac{1}{2}x^T Qx + q^T x + c$ mit $Q \succ 0$ ist (streng) konvex.

Mit $\nabla f(x) = Qx + q$ bestimmt $\nabla f(x^*) = 0$ das Minimum x^* eindeutig,

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

1.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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.

1.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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 $f : \mathbb{R}^n \rightarrow \mathbb{R}$, gilt

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

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

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

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

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

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 λ_{\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

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

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

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

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

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

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 $\nabla^2 f \succeq 0$ (> 0) durch $\nabla^2 f \preceq 0$ (< 0) ersetzen, der Rest bleibt gleich.

Bemerkungen

- In Optimalitätsbedingungen für Maximierungsprobleme muss man nur $\nabla^2 f \succeq 0$ (> 0) durch $\nabla^2 f \preceq 0$ (< 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 $\nabla^2 f \succeq 0$ (> 0) durch $\nabla^2 f \preceq 0$ (< 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.

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren**
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.3 Das Newton-Verfahren

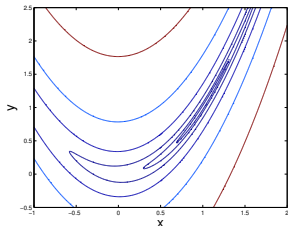
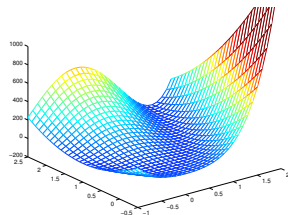
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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,

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

$$[= c + q^T h + \frac{1}{2} h^T Qh]$$



1.3 Das Newton-Verfahren

[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (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,

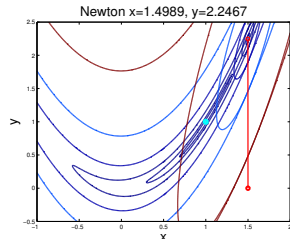
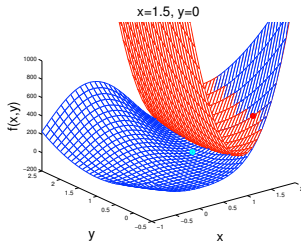
$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

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

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

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

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



1.3 Das Newton-Verfahren

[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (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,

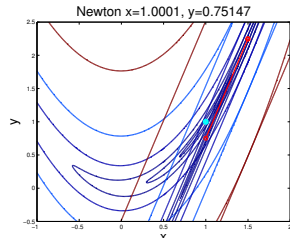
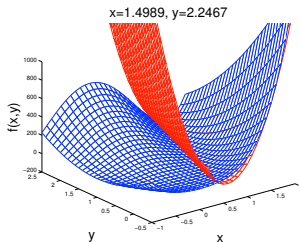
$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

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

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

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

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



1.3 Das Newton-Verfahren

[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (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,

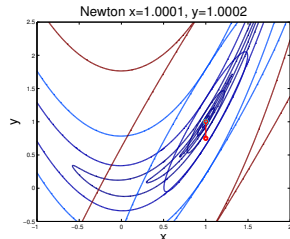
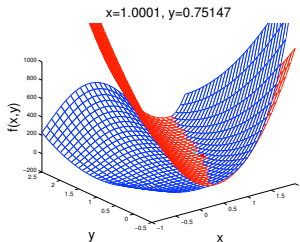
$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

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

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

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

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



1.3 Das Newton-Verfahren

[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (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,

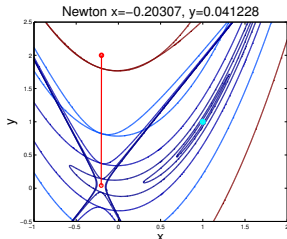
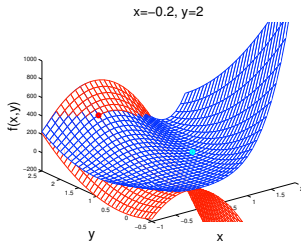
$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

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

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

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

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



1.3 Das Newton-Verfahren

[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (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,

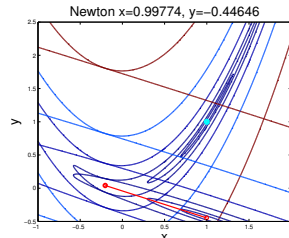
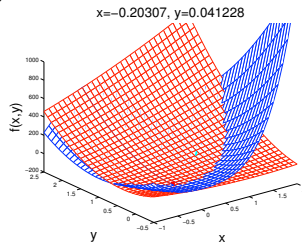
$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

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

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

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

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



1.3 Das Newton-Verfahren

[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ (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,

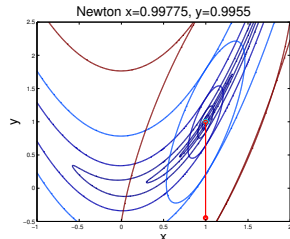
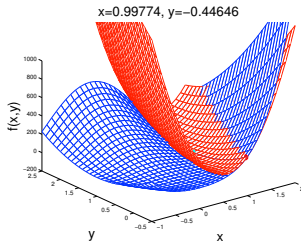
$$f(x^{(k)}) + \nabla f(x^{(k)})^T h + \frac{1}{2} h^T \nabla^2 f(x^{(k)}) h$$

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

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

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

$h_N^{(k)}$ 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).
- 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 $\|\nabla f(x^{(k)})\| \rightarrow 0$ erzeugt. [Das kann auch $\|x^{(k)}\| \rightarrow \infty$ 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 $\|\nabla f(x^{(k)})\| \rightarrow 0$ erzeugt. [Das kann auch $\|x^{(k)}\| \rightarrow \infty$ 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 \rightarrow 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 $\|\nabla f(x^{(k)})\| \rightarrow 0$ erzeugt. [Das kann auch $\|x^{(k)}\| \rightarrow \infty$ 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 \rightarrow 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 $\nabla^2 f(x^{(k)})$.

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren**
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

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

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

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:

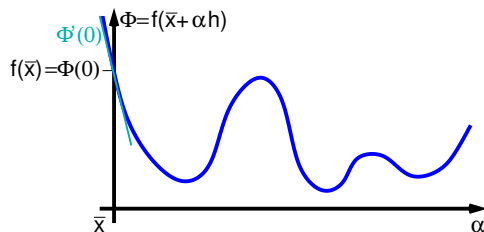
$$\exists \delta > 0 : \frac{-\nabla f_k^T}{\|\nabla f_k\|} \frac{h^{(k)}}{\|h^{(k)}\|} = \cos \angle(-\nabla f_k, h^{(k)}) \geq \delta > 0 \text{ für } k > 0.$$

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

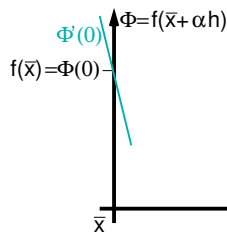


Line-Search für Abstiegsrichtung h

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 $\Phi'(0) = \nabla f(\bar{x})^T h$



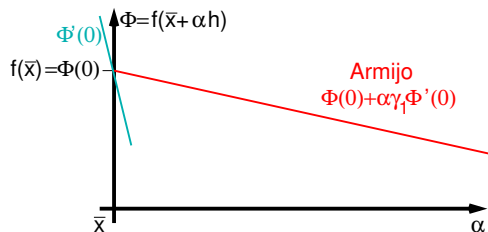
Line-Search für Abstiegsrichtung h

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]



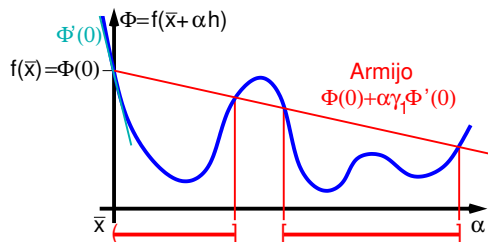
Line-Search für Abstiegsrichtung h

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

Berechnung von $\bar{\alpha} \in \text{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (**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]



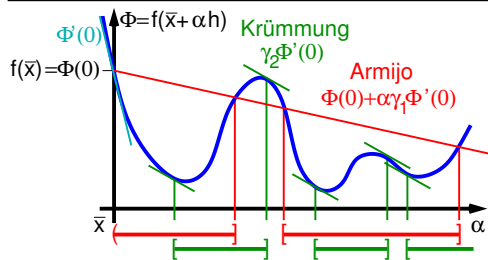
Line-Search für Abstiegsrichtung h

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

Berechnung von $\bar{\alpha} \in \text{Argmin}_{\alpha \geq 0} \Phi(\alpha)$ (**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]



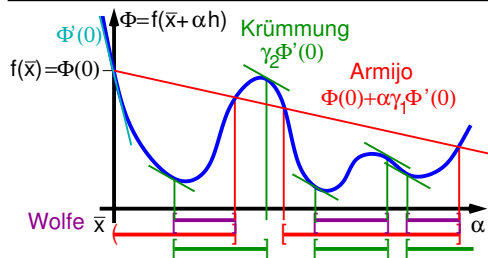
Line-Search für Abstiegsrichtung h

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

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

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

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

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 $\|\nabla f_k\| \rightarrow 0$.

Wolfe-Bedingungen und globale Konvergenz

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

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

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 $\|\nabla f_k\| \rightarrow 0$.

Vorsicht: Man hofft auf Konvergenz gegen ein Minimum, aber sowohl $\|x^{(k)}\| \rightarrow \infty$ 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.
- 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 $0 < \sigma < 1$).

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg**
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.5 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} ($B \succ 0$):

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

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

1.5 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} ($B \succ 0$):

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

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

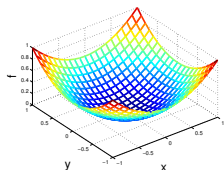
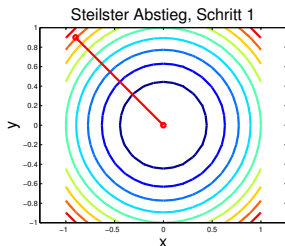
Ideal skaliert ($Q = I$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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 ($Q = I$):

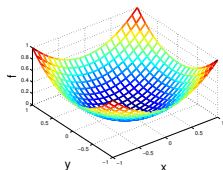
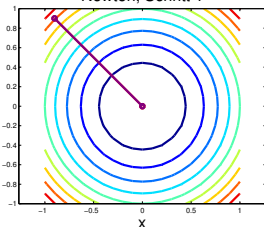
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad q = 0, \quad c = 0$$

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

Newton ($B = Q$):

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

Newton, Schritt 1



1.5 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} ($B \succ 0$):

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

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

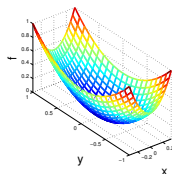
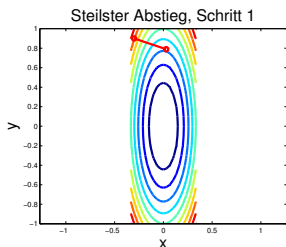
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

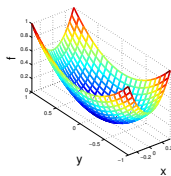
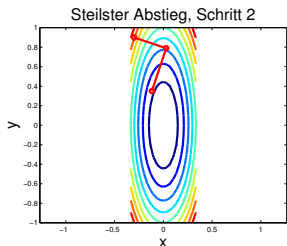
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

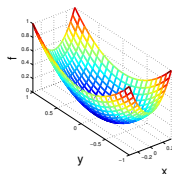
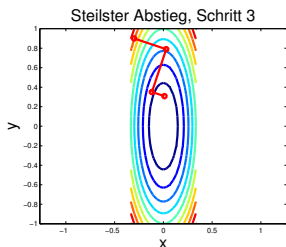
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

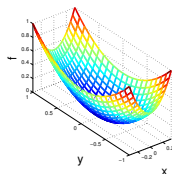
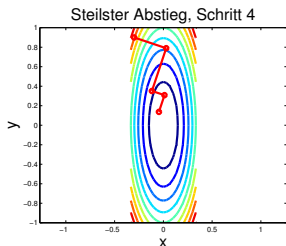
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

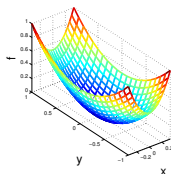
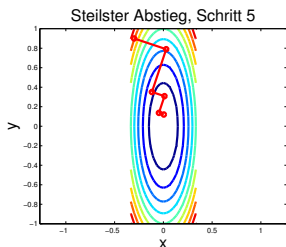
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

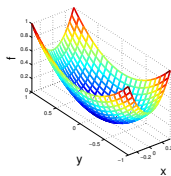
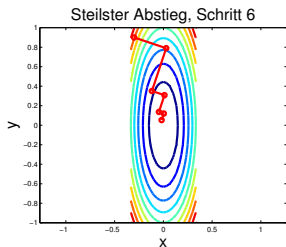
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

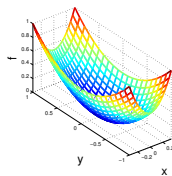
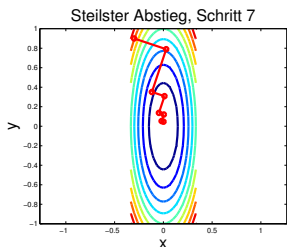
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

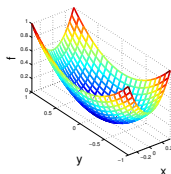
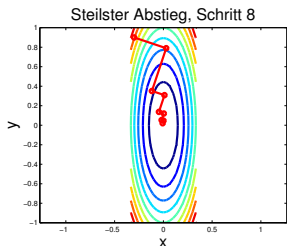
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

Steilster Abstieg ($B = I$):

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



1.5 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} ($B \succ 0$):

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

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

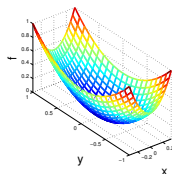
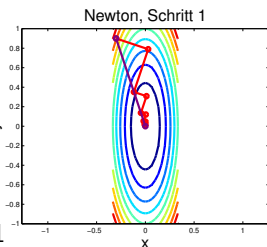
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

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

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

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

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



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 $f(x) = \frac{1}{2}x^T Qx + q^T x + c$ 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 $f(x) = \frac{1}{2}x^T Qx + q^T x + c$ 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

- steilster Abstieg konvergiert am Ende fast immer schlecht
- 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 $f(x) = \frac{1}{2}x^T Qx + q^T x + c$ 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

→ steilster Abstieg konvergiert am Ende fast immer schlecht

→ 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.

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton**
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.6 Quasi-Newton-Verfahren

Ein Verfahren **konvergiert superlinear** gegen x^* , falls $\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0$
 (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, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.*

1.6 Quasi-Newton-Verfahren

Ein Verfahren **konvergiert superlinear** gegen x^* , falls $\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0$
(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, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.*

⇒ Superlineare Konvergenz erfordert Approximation der Newton-Richtung.

Für $h = -B^{-1}\nabla f$ sollte B also die Hessematrix nachbauen. Diese erfüllt

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

1.6 Quasi-Newton-Verfahren

Ein Verfahren **konvergiert superlinear** gegen x^* , falls $\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0$
(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, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

⇒ 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: $h^{(k+1)} = -B_{k+1}^{-1} \nabla f_{k+1}$

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

1.6 Quasi-Newton-Verfahren

Ein Verfahren **konvergiert superlinear** gegen x^* , falls $\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0$
(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, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

⇒ 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: $h^{(k+1)} = -B_{k+1}^{-1} \nabla f_{k+1}$

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

→ 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:

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

1.6 Quasi-Newton-Verfahren

Ein Verfahren **konvergiert superlinear** gegen x^* , falls $\lim_{k \rightarrow \infty} \frac{\|x^{(k+1)} - x^*\|}{\|x^{(k)} - x^*\|} = 0$
(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, $\|h^{(k)} - h_N^{(k)}\| = \mathbf{o}(\|h^{(k)}\|)$.

⇒ 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: $h^{(k+1)} = -B_{k+1}^{-1} \nabla f_{k+1}$

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

→ 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:

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

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:

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

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 **B**royden, **F**letcher, **G**oldfarb und **S**hanno:

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

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:

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

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 **B**royden, **F**letcher, **G**oldfarb und **S**hanno:

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

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:

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

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 **B**royden, **F**letcher, **G**oldfarb und **S**hanno:

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

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.

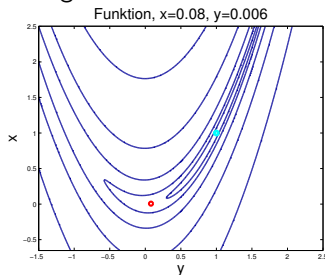
Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren**
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.7 Trust-Region-Verfahren

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

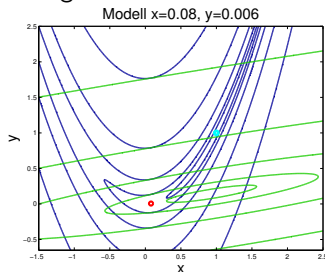


1.7 Trust-Region-Verfahren

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]



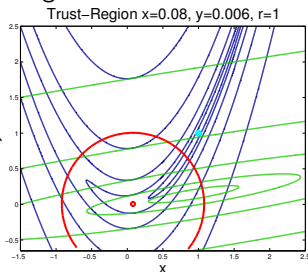
1.7 Trust-Region-Verfahren

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

$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$



1.7 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]

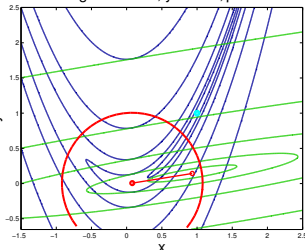
und einen Trust-Region-Radius Δ_0 mit

$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

1. Löse das Trust-Region-Unterproblem:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

Trust-Region $x=0.08, y=0.006, \rho=-66.1145$



1.7 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

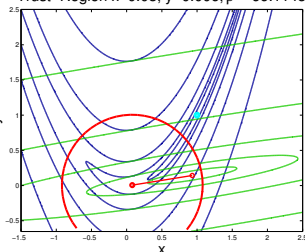
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

1. Löse das Trust-Region-Unterproblem:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

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

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

Trust-Region $x=0.08, y=0.006, \rho=-66.1145$ 

1.7 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

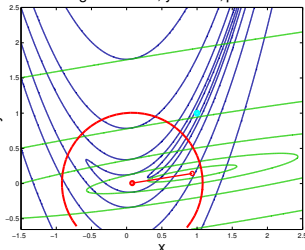
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

1. Löse das Trust-Region-Unterproblem:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

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

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

Trust-Region $x=0.08, y=0.006, \rho=-66.1145$ 

1.7 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 $x^{(0)}$,

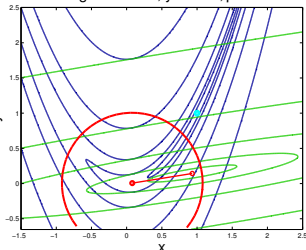
$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
 [B_0 z.B. aus Newton oder Quasi-Newton]
 und einen Trust-Region-Radius Δ_0 mit

$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

1. Löse das Trust-Region-Unterproblem:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

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 $\Delta_{k+1} := \begin{cases} \frac{1}{4} \|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} & [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k & [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} & [\text{Wert ok}] \end{cases}$
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
 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, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

Trust-Region $x=0.08, y=0.006, \rho=-66.1145$ 

1.7 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

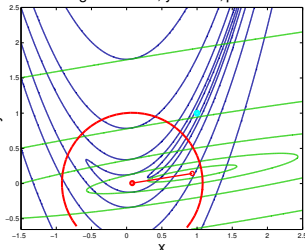
$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

1. Löse das Trust-Region-Unterproblem:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

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 $\eta \in [0, 1)$,
 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, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

Trust-Region $x=0.08, y=0.006, \rho=-66.1145$ 

1.7 Trust-Region-Verfahren

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

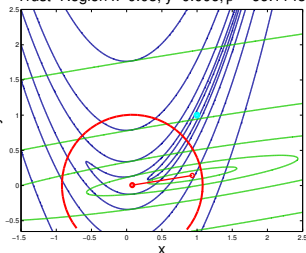
0. Wähle ein Model m_0 von f um $x^{(0)}$,

$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
 [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:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

Trust-Region $x=0.08, y=0.006, \rho=-66.1145$ 

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 $\Delta_{k+1} := \begin{cases} \frac{1}{4} \|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} & [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k & [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} & [\text{Wert ok}] \end{cases}$
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
 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.

1.7 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 $x^{(0)}$,

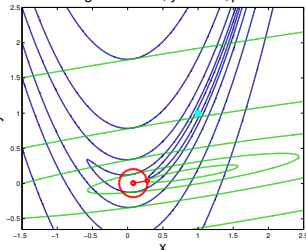
$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
 [B_0 z.B. aus Newton oder Quasi-Newton]
 und einen Trust-Region-Radius Δ_0 mit

$$m_0(h) \approx f(x^{(0)} + h) \quad \forall \|h\| \leq \Delta_0$$

1. Löse das Trust-Region-Unterproblem:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$

Trust-Region $x=0.08, y=0.006, \rho=0.52888$



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 $\Delta_{k+1} := \begin{cases} \frac{1}{4} \|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} & [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k & [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} & [\text{Wert ok}] \end{cases}$
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
 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.

1.7 Trust-Region-Verfahren

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

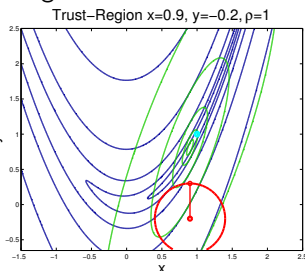
0. Wähle ein Model m_0 von f um $x^{(0)}$,

$$m_0(h) = f_0 + \nabla f_0^T h + \frac{1}{2} h^T B_0 h$$
 [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:

$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\text{Argmin}} m_k(h) \quad [\text{nur annähern}]$$



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 $\Delta_{k+1} := \begin{cases} \frac{1}{4} \|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} & [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k & [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} & [\text{Wert ok}] \end{cases}$
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
 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

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

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

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

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

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

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 $h = -\frac{\nabla f_k}{\|\nabla f_k\|}$

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

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

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

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

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

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 $h = -\frac{\nabla f_k}{\|\nabla f_k\|}$

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

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

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

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)

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren**
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.8 Numerisches Differenzieren

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

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

1.8 Numerisches Differenzieren

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

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

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

1.8 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 $\text{float}(1 + u) = \text{float}(1)$), 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.

1.8 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 $\text{float}(1 + u) = \text{float}(1)$), 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 Berechnung von $\nabla^2 f$ oder der Jacobimatrix J_g einer Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ 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.

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren**
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

$$x_4 := x_1 x_2$$

$$x_5 := e^{x_4}$$

$$x_6 := x_4 + x_5$$

$$x_7 := x_6 / x_3$$

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

$$x_5 := e^{x_4}$$

$$x_6 := x_4 + x_5$$

$$x_7 := x_6 / x_3$$

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

$$x_6 := x_4 + x_5$$

$$x_7 := x_6 / x_3$$

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

$$x_7 := x_6 / x_3$$

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

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

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

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

Beachte:

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

1.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

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

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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ 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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ 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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ Richtungsableitung für h
 Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,
 $x_4 := x_1 x_2 \rightarrow D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$
 $x_5 := e^{x_4}$

$$x_6 := x_4 + x_5$$

$$x_7 := x_6 / x_3$$

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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ Richtungsableitung für h
 Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,
 $x_4 := x_1 x_2 \rightarrow D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$
 $x_5 := e^{x_4} \rightarrow D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$
 $x_6 := x_4 + x_5$
 $x_7 := x_6 / x_3$

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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ Richtungsableitung für h
 Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,
 $x_4 := x_1 x_2 \rightarrow D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$
 $x_5 := e^{x_4} \rightarrow D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$
 $x_6 := x_4 + x_5 \rightarrow D_h x_6 = \dots = D_h x_4 + D_h x_5$
 $x_7 := x_6 / x_3$

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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ Richtungsableitung für h
 Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,
 $x_4 := x_1 x_2 \rightarrow D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$
 $x_5 := e^{x_4} \rightarrow D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$
 $x_6 := x_4 + x_5 \rightarrow D_h x_6 = \dots = D_h x_4 + D_h x_5$
 $x_7 := x_6 / x_3 \rightarrow D_h x_7 = \dots = x_3^{-1} D_h x_6 - x_6 x_3^{-2} D_h x_3$

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: $f(x) = (x_1 x_2 + e^{x_1 x_2})/x_3$, $D_h \dots$ Richtungsableitung für h
 Beginne mit $D_h x_1 := \nabla_x x_1^T h = e_1^T h = h_1$, $D_h x_2 = \nabla_x x_2^T h = h_2$, $D_h x_3 = h_3$,
 $x_4 := x_1 x_2 \rightarrow D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 = x_2 h_1 + x_1 h_2$
 $x_5 := e^{x_4} \rightarrow D_h x_5 = e^{x_4} \nabla_x x_4^T h = e^{x_4} D_h x_4$
 $x_6 := x_4 + x_5 \rightarrow D_h x_6 = \dots = D_h x_4 + D_h x_5$
 $x_7 := x_6 / x_3 \rightarrow D_h x_7 = \dots = x_3^{-1} D_h x_6 - x_6 x_3^{-2} D_h x_3$

$D_h f(x) = \nabla f(x)^T h$ 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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$		$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
$x_1 = 2$	{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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
$x_1 = 2$	{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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
$x_1 = 2$	{4}	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$ $\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$ $\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$ $\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$
$x_2 = 0$	{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	— ($f = x_7$!!!)	

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_4}$

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

Taucht x_j 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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
$x_1 = 2$	{4}	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$ $\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$ $\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$ $\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$ $\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$
$x_2 = 0$	{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}	$\frac{\partial x_7}{\partial x_5} = \frac{1}{x_3} = \frac{1}{3}$	
$x_7 := x_6 / x_3 = \frac{1}{3}$	\emptyset	— ($f = x_7$!!!)	

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_3}$

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

Taucht x_j 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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_j}$
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: $\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3} \right)^T = \left(0, \frac{4}{3}, \frac{1}{9} \right)^T$

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).
- $\nabla f^T h$ ($\nabla^2 f h$) 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).
- $\nabla f^T h$ ($\nabla^2 f h$) 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).
- $\nabla f^T h$ ($\nabla^2 f h$) 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“.

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren**
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.10 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.

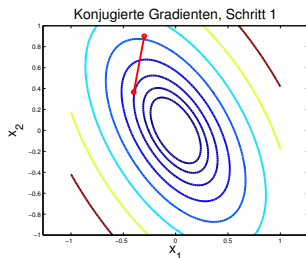
1.10 Das lineare Konjugierte-Gradienten-Verfahren

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.



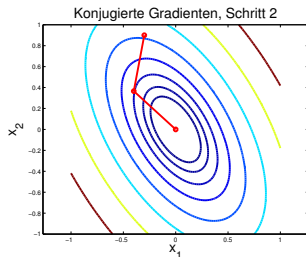
1.10 Das lineare Konjugierte-Gradienten-Verfahren

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.



1.10 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.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

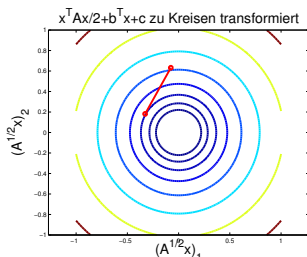
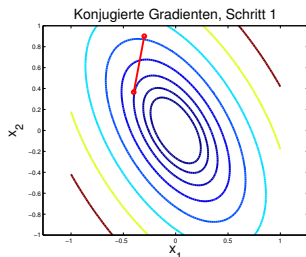
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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.



1.10 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.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

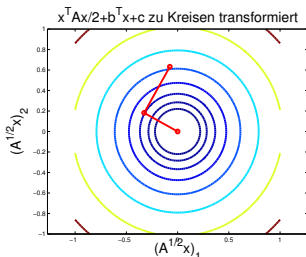
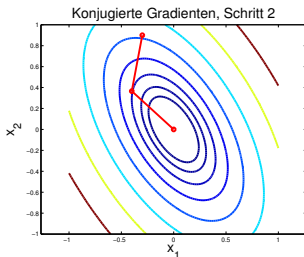
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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!]

1.10 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.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

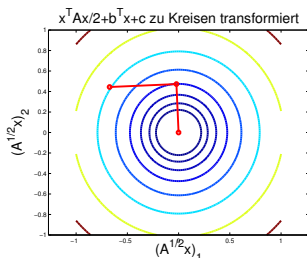
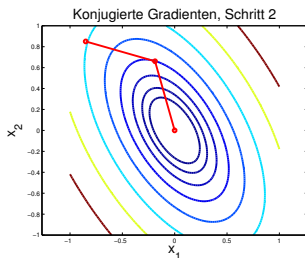
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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!]

1.10 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.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

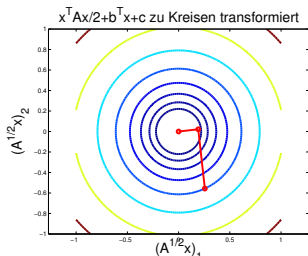
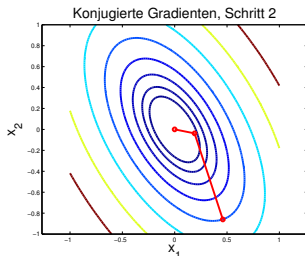
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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!]

1.10 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.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$

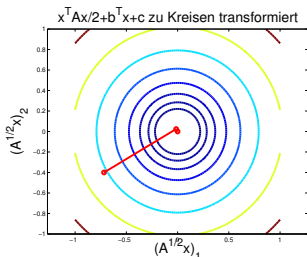
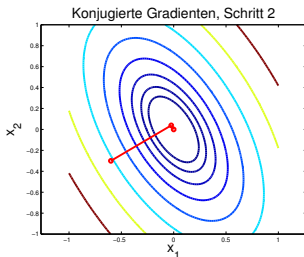
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$

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: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$

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. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1} r^{(0)}\}]$
 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 $x^{(k+1)} = x^{(i+1)} + \sum_{j=i+1}^k \alpha_j h^{(j)}$,

$$(h^{(i)})^T (r^{(k+1)}) = \underbrace{(h^{(i)})^T [Ax^{(i+1)} - b]}_{=(h^{(i)})^T r^{(i+1)}=0} + \sum_{j=i+1}^k \alpha_j \underbrace{(h^{(i)})^T A h^{(j)}}_{=0} = 0$$

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. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1} r^{(0)}\}]$
Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1} r^{(0)}\}]$
 $\beta_{k+1} := \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$. 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: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T A h^{(k)}}$
 3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
 4. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1} r^{(0)}\}]$
 Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
 5. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1} r^{(0)}\}]$
 $\beta_{k+1} := \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$. 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

- $k = 0$, Residuum $r^{(0)} = Ax^{(0)} - b [= \nabla f(x^{(0)})]$, $h^{(0)} = -r^{(0)}$
- Falls $\|r^{(k)}\| < \varepsilon$ STOP.
- exakter Line-Search: $\alpha_k = \operatorname{argmin}_{\alpha \geq 0} f(x^{(k)} + \alpha h^{(k)}) = -\frac{(r^{(k)})^T h^{(k)}}{(h^{(k)})^T Ah^{(k)}}$
- $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
- $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k Ah^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.
- $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)} \quad [\in \operatorname{span}\{r^{(0)}, \dots, A^{k+1}r^{(0)}\}]$
 $\beta_{k+1} := \frac{(r^{(k+1)})^T Ah^{(k)}}{(h^{(k)})^T Ah^{(k)}}$. Für $i \leq k$ gilt $(h^{(i)})^T Ah^{(k+1)} = 0 = (r^{(i)})^T r^{(k+1)}$.
- $k \leftarrow k + 1$, gehe zu 1.

Satz (Lineares CG-Verfahren)

Ist in Iteration k der Gradient $r_k \neq 0$, so gilt

$$\operatorname{span}\{r^{(0)}, \dots, A^k r^{(0)}\} = \operatorname{span}\{r^{(0)}, \dots, r^{(k)}\} = \operatorname{span}\{h^{(0)}, \dots, h^{(k)}\}$$

und $(r^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T Ah^{(k)} = 0$ für $0 \leq i < k$.

Insbesondere ist $x^{(k)} = \operatorname{argmin} \{f(x) : x \in x^{(0)} + \operatorname{span}\{r^{(0)}, \dots, A^{k-1}r^{(0)}\}\}$

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:

$$\text{Für } \alpha: (r^{(k)})^T h^{(k)} = (r^{(k)})^T (-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$$

$$\text{Für } \beta: (r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T (r^{(k+1)} - r^{(k)}) / \alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$$

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:

$$\text{Für } \alpha: (r^{(k)})^T h^{(k)} = (r^{(k)})^T (-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$$

$$\text{Für } \beta: (r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T (r^{(k+1)} - r^{(k)}) / \alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$$

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:

$$\text{Für } \alpha: (r^{(k)})^T h^{(k)} = (r^{(k)})^T (-r^{(k)} + \beta h^{(k-1)}) = -(r^{(k)})^T r^{(k)}$$

$$\text{Für } \beta: (r^{(k+1)})^T A h^{(k)} = (r^{(k+1)})^T (r^{(k+1)} - r^{(k)}) / \alpha_k = \frac{(r^{(k+1)})^T r^{(k+1)}}{\alpha_k}$$

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.
 → In der Praxis ist nur mit Prädiktionierung gute Genauigkeit erzielbar.
[Prädik. \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 Abstiegschritt \rightarrow global konvergent

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren**
- 1.12 Nichtlineare kleinste Quadrate
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.11 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.

1.11 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

$$(INC) \quad \|r^{(k)}\| \leq \eta_k \|\nabla f_k\|$$

und geben dafür eine **forcing sequence** $\eta_k \in (0, 1)$ vor.

1.11 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

$$(INC) \quad \|r^{(k)}\| \leq \eta_k \|\nabla f_k\|$$

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 $\eta_k \rightarrow 0$, 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, $\nabla^2 f_k \neq 0$]
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, $\nabla^2 f_k \neq 0$]

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, $\nabla^2 f_k \neq 0$]
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, $\nabla^2 f_k \neq 0$]
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 $\text{span}\{d^{(0)}, \dots, d^{(i-1)}\}$.
 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, $\nabla^2 f_k \neq 0$]
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. $\alpha_j := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [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 $\|p^{(0)}\| < \dots < \|p^{(i)}\| < \dots \leq \Delta \Rightarrow$ lokal inexakter Newton

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate**
 - Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen

1.12 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 $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ glatt.

1.12 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 $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 .

1.12 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 $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ 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

$$p(\tilde{y}; x, t) := \prod_{j=1}^m \varphi(\tilde{y}_j - \Phi(t_j; x)) = \left(\frac{1}{2\pi\sigma^2}\right)^{\frac{m}{2}} \exp\left(-\frac{1}{2\sigma^2} \sum_{j=1}^m [\tilde{y}_j - \Phi(t_j; x)]^2\right)$$

aufzutreten.

1.12 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 $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

1.12 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 $r_j : \mathbb{R}^n \rightarrow \mathbb{R}$ 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 $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

$$\longrightarrow \text{Finde } \bar{x} \in \operatorname{Argmin} f(x) := \frac{1}{2} \sum_{j=1}^m r_j^2(x) \text{ mit } r_j(x) := \tilde{y}_j - \Phi(t_j; x).$$

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\rightarrow \min_x f(x) := \frac{1}{2} \|Ax - b\|^2$

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\rightarrow \min_x f(x) := \frac{1}{2} \|Ax - b\|^2$

Wegen $\frac{1}{2} \|Ax - b\|^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} x^T A^T A x - x^T A^T b + \frac{1}{2} b^T b$
und $A^T A \succeq 0$, ist das Problem konvex quadratisch und $\nabla f(x) = 0$ ist
hinreichende Optimalitätsbedingung.

x^* ist Optimallösung $\Leftrightarrow x^*$ erfüllt die **Normalgleichungen** $A^T A x = A^T b$.

Spezialfall: Lineare kleinste Quadrate

Sind alle r_j linear/affin $\rightarrow \min_x f(x) := \frac{1}{2} \|Ax - b\|^2$

Wegen $\frac{1}{2} \|Ax - b\|^2 = \frac{1}{2} (Ax - b)^T (Ax - b) = \frac{1}{2} x^T A^T A x - x^T A^T b + \frac{1}{2} b^T b$
und $A^T A \succeq 0$, ist das Problem konvex quadratisch und $\nabla f(x) = 0$ ist
hinreichende Optimalitätsbedingung.

x^* ist Optimallösung $\Leftrightarrow x^*$ erfüllt die **Normalgleichungen** $A^T A x = A^T b$.

Da $A^T A \succeq 0$, bieten sich viele numerische Verfahren zur Lösung an:

- Cholesky-Faktorisierung von $A^T A$ (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 $A^T w$ 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):

$$\nabla f(x) = \sum_{j=1}^m r_j(x) \nabla r_j(x) = J_r(x)^T r(x)$$

Für eine Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $g(x) = (g_1(x), \dots, g_m(x))^T$ ist

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix} = \begin{bmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix}$$

die Jacobimatrix von g in x . So ist z.B. $\nabla^2 f(x) = J_{\nabla f}(x)$.

$g(x+h) = g(x) + J_g(x)h + \mathbf{o}(h) \rightarrow$ 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):

$$\begin{aligned} \nabla f(x) &= \sum_{j=1}^m r_j(x) \nabla r_j(x) = J_r(x)^T r(x) \\ \nabla^2 f(x) &= \sum_{j=1}^m \nabla r_j(x) \nabla r_j(x)^T + \sum_{j=1}^m r_j(x) \nabla^2 r_j(x) \\ &= J_r(x)^T J_r(x) + \underbrace{\sum_{j=1}^m r_j(x) \nabla^2 r_j(x)}_{\text{klein}} \approx J_r(x)^T J_r(x) \end{aligned}$$

→ die Jacobimatrix liefert eine gute Näherung für die Hessematrix.

Für eine Funktion $g : \mathbb{R}^n \rightarrow \mathbb{R}^m$ mit $g(x) = (g_1(x), \dots, g_m(x))^T$ ist

$$J_g(x) = \begin{bmatrix} \frac{\partial g_1}{\partial x_1}(x) & \dots & \frac{\partial g_1}{\partial x_n}(x) \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial x_1}(x) & \dots & \frac{\partial g_m}{\partial x_n}(x) \end{bmatrix} = \begin{bmatrix} \nabla g_1(x)^T \\ \vdots \\ \nabla g_m(x)^T \end{bmatrix}$$

die Jacobimatrix von g in x . So ist z.B. $\nabla^2 f(x) = J_{\nabla f}(x)$.

$g(x+h) = g(x) + J_g(x)h + \mathbf{o}(h)$ → das lineare Modell von g in x .

1.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), r_k := r(x^{(k)})$$

1.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$

1.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

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 $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.

1.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

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.

1.12.1 Das Gauss-Newton-Verfahren

Zur Lösung von $\min_x f(x) := \frac{1}{2} \|r(x)\|^2$ mit $r : \mathbb{R}^n \rightarrow \mathbb{R}^m$ wähle statt der Newton-Schrittrichtung $\nabla^2 f_k h_k^N = -\nabla f_k$ die Gauss-Newton-Richtung

$$J_k^T J_k h_k^{GN} = -J_k^T r_k \quad \text{wobei } J_k := J_r(x^{(k)}), \quad r_k := r(x^{(k)})$$

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.]

Inhaltsübersicht

Freie Nichtlineare Optimierung

- 1.1 Orakel, lineares/quadratisches Modell
- 1.2 Optimalitätsbedingungen
- 1.3 Das Newton-Verfahren
- 1.4 Line-Search-Verfahren
- 1.5 Skalierung und Steilster Abstieg
- 1.6 Quasi-Newton
- 1.7 Trust-Region-Verfahren
- 1.8 Numerisches Differenzieren
- 1.9 Automatisches Differenzieren
- 1.10 Das Konjugierte-Gradienten-Verfahren
- 1.11 Inexakte Newton-Verfahren
- 1.12 Nichtlineare kleinste Quadrate
Das Gauss-Newton-Verfahren
- 1.13 Newton für nichtlineare Gleichungen**

1.13 Newton für nichtlineare Gleichungen

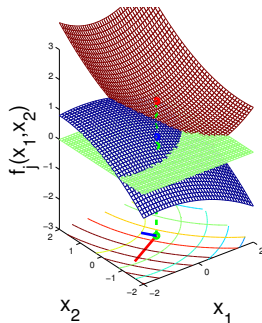
Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Funktionswerte in $x_1=0, x_2=0$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

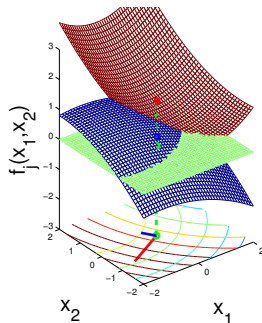
Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Bsp1: Primal-Duales System für LP

$$F \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Funktionswerte in $x_1=0, x_2=0$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

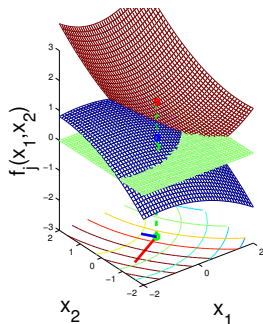
Bsp1: Primal-Duales System für LP

$$F \left(\begin{bmatrix} x \\ y \\ z \end{bmatrix} \right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$$F(x) := \nabla f(x) = 0$$

Funktionswerte in $x_1=0, x_2=0$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

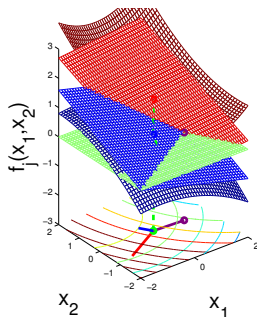
$$F(x) := \nabla f(x) = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Newton-Schritt zu $x_1^N = 1.1053$, $x_2^N = -0.0592$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

$$F(x) := \nabla f(x) = 0$$

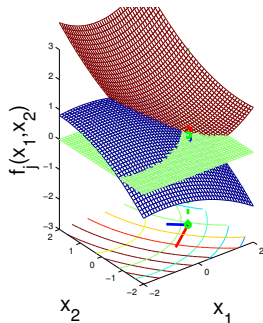
Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Funktionswerte in $x_1=1.1053$, $x_2=-0.05921$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

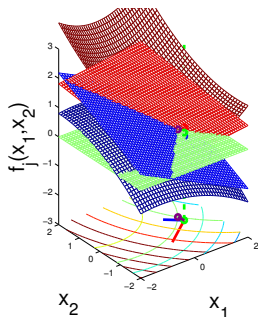
$$F(x) := \nabla f(x) = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Newton-Schritt zu $x_1^N = 1.0756$, $x_2^N = 0.2043$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

$$F(x) := \nabla f(x) = 0$$

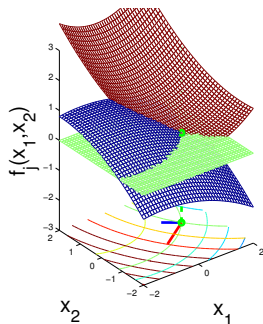
Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Funktionswerte in $x_1=1.0756$, $x_2=0.2043$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

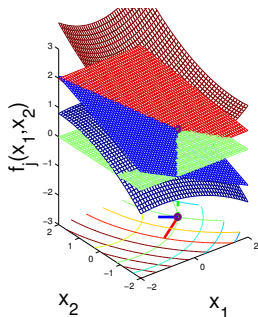
$$F(x) := \nabla f(x) = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Newton-Schritt zu $x_1^N = 1.0736$, $x_2^N = 0.22139$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

$$F(x) := \nabla f(x) = 0$$

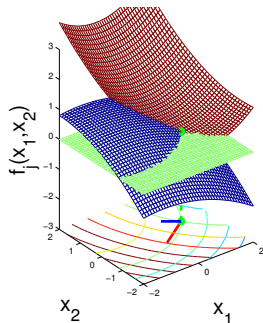
Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Funktionswerte in $x_1=1.0736, x_2=0.22139$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

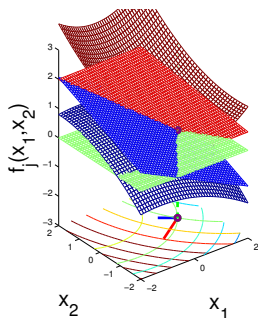
$$F(x) := \nabla f(x) = 0$$

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 $x^{(k+1)} := x^{(k)} + h_{NF}^{(k)}$.

Newton-Schritt zu $x_1^N = 1.0736$, $x_2^N = 0.22146$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $x^{(k)}$,

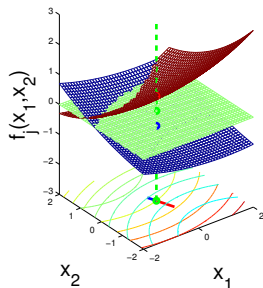
$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

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.

Funktionswerte in $x_1=0, x_2=0$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $\min_x f(x)$

$$F(x) := \nabla f(x) = 0$$

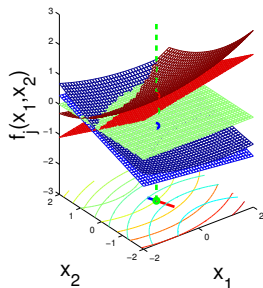
Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

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.

Newton-Schritt in $x_1=0, x_2=0$



1.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$

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 $x^{(k)}$,

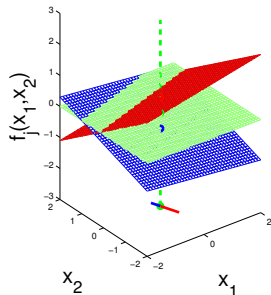
$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

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.

Rang(Jacobimatrix)=1



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

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \text{Diag}(z) & 0 & \text{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

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

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \text{Diag}(z) & 0 & \text{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

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

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \text{Diag}(z) & 0 & \text{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

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

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \text{Diag}(z) & 0 & \text{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

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!

Inhaltsübersicht

Freie Nichtlineare Optimierung

Restringierte Nichtlineare Optimierung: Grundlagen

Restringierte Optimierung: Verfahren

Ableitungsfreie Optimierung/Direkte Suchverfahren

Inhaltsübersicht

Restringierte Nichtlineare Optimierung: Grundlagen

2.1 Aufgabenstellung

2.2 Zulässige Richtungen, Tangential- und Polarkegel

2.3 Notwendige Optimalitätsbedingung

2.4 Der linearisierte Tangentialkegel

2.5 KKT-Bedingungen

2.6 Bedingungen 2. Ordnung

2.7 Sensitivität

2.1 Restringierte Nichtlineare Optimierung

Aufgabenstellung:

$$\begin{array}{ll} \min & f(x) & \text{Zielfunktion} \\ (P) \quad \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbedingungen} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbedingungen} \\ & x \in \mathbb{R}^n & \text{Grundmenge (meist } \mathbb{R}^n, \text{ manchmal } [0, 1]^n) \end{array}$$

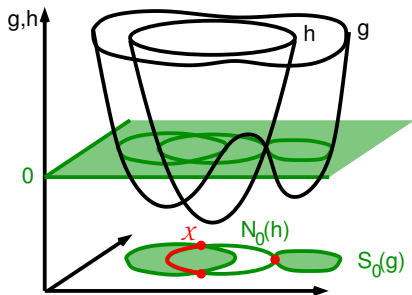
mit $f, g_i, h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ hinreichend glatt und \mathcal{E}, \mathcal{I} endliche Indexmengen.

2.1 Restringierte Nichtlineare Optimierung

Aufgabenstellung:

$$\begin{array}{ll}
 \min & f(x) & \text{Zielfunktion} \\
 \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbedingungen} \\
 & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbedingungen} \\
 & x \in \mathbb{R}^n & \text{Grundmenge (meist } \mathbb{R}^n, \text{ manchmal } [0, 1]^n)
 \end{array}$$

mit $f, g_i, h_i : \mathbb{R}^n \rightarrow \mathbb{R}$ hinreichend glatt und \mathcal{E}, \mathcal{I} endliche Indexmengen.
 Die zulässige Menge $\mathcal{X} := \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$
 ist Schnitt der Niveaulinien/-mengen, $\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i)$.



2.1 Restringierte Nichtlineare Optimierung

Aufgabenstellung:

$$\begin{array}{ll}
 \min & f(x) & \text{Zielfunktion} \\
 \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbedingungen} \\
 & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbedingungen} \\
 & x \in \mathbb{R}^n & \text{Grundmenge (meist } \mathbb{R}^n, \text{ manchmal } [0, 1]^n)
 \end{array}$$

mit $f, g_i, h_i: \mathbb{R}^n \rightarrow \mathbb{R}$ hinreichend glatt und \mathcal{E}, \mathcal{I} endliche Indexmengen.

Die zulässige Menge $\mathcal{X} := \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$

ist Schnitt der Niveaulinien/-mengen, $\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i)$.

Wir suchen ein lokal optimales $x^* \in \mathcal{X}$, also ein x^* mit

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{X} \cap B_\varepsilon(x^*) := \{x : \|x - x^*\| \leq \varepsilon\} \text{ für ein } \varepsilon > 0.$$

Für Algorithmen ist diese Beschreibung ungeeignet. Wir benötigen eine algebraische Beschreibung der lokalen Optimalität, die für gegebenes $\bar{x} \in \mathbb{R}^n$ nur auf den Funktionswerten und Gradienten (und eventuell ∇^2) von f, g_i, h_i in \bar{x} beruht (Orakel 1./2. Ordnung für f, g_i, h_i).

Idee: Beschreibe die Richtungen, in die man sich von \bar{x} aus innerhalb \mathcal{X} noch ein kleines Stück bewegen kann. Ist die Richtungsableitung in all diesen Richtungen positiv, gibt es in der Nähe keinen besseren Punkt.

Inhaltsübersicht

Restringierte Nichtlineare Optimierung: Grundlagen

2.1 Aufgabenstellung

2.2 Zulässige Richtungen, Tangential- und Polarkegel

2.3 Notwendige Optimalitätsbedingung

2.4 Der linearisierte Tangentialkegel

2.5 KKT-Bedingungen

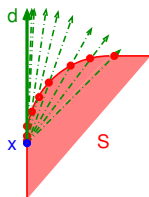
2.6 Bedingungen 2. Ordnung

2.7 Sensitivität

2.2 Zulässige Richtungen und Tangentialkegel

Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

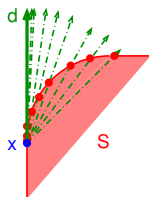
[Die α_k strecken die Vektoren auf die Länge von d]



2.2 Zulässige Richtungen und Tangentialkegel

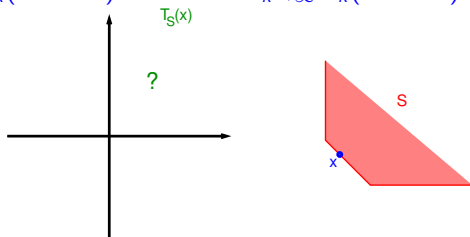
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

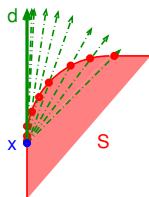
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

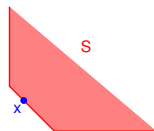
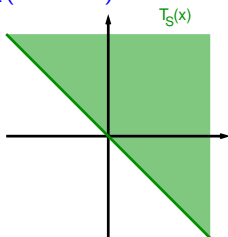
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k (x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

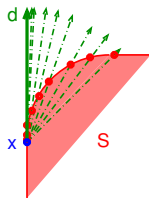
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k (x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k (x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

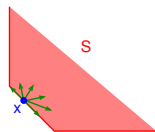
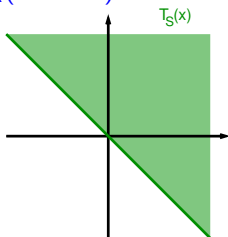
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

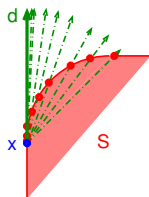
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

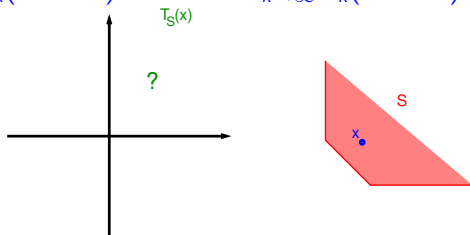
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

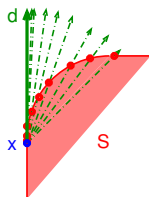
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

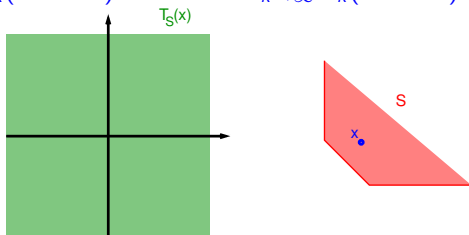
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

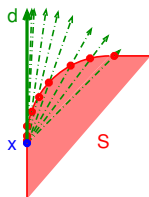
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

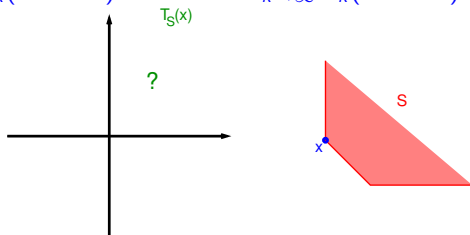
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

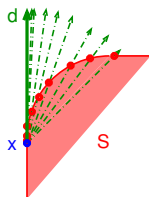
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

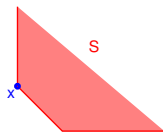
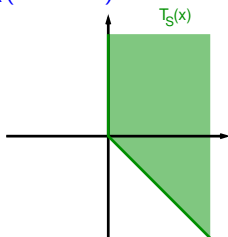
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

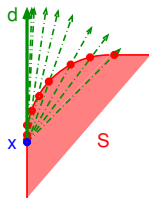
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

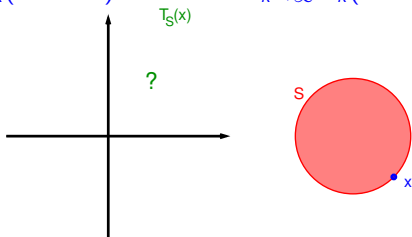
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

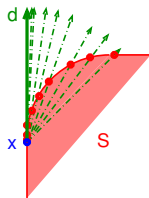
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

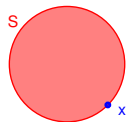
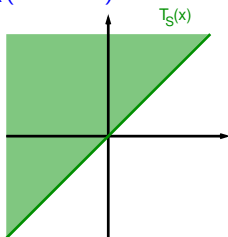
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

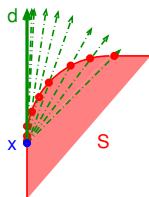
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

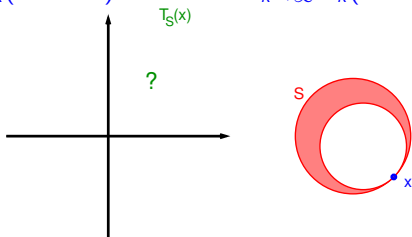
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

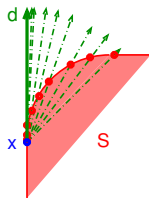
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

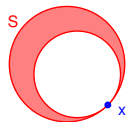
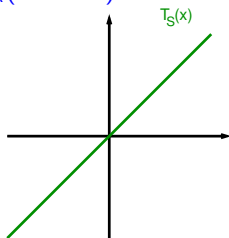
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

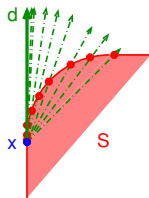
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

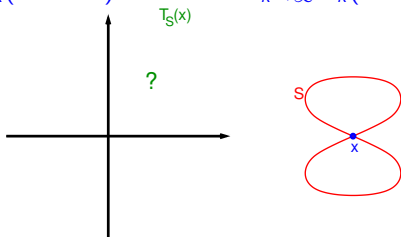
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

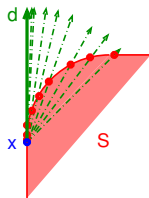
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



2.2 Zulässige Richtungen und Tangentialkegel

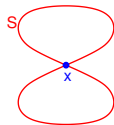
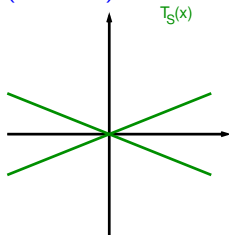
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ **zulässige Richtung (Tangentialrichtung)**, wenn es eine Folge $x^{(k)} \in S$ ($k \in \mathbb{N}$) mit $\lim_{k \rightarrow \infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$.

[Die α_k strecken die Vektoren auf die Länge von d]



Die Menge aller zulässigen Richtungen für ein x in S bildet den **Tangentialkegel** $T_S(x)$ von S in x .

[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k \rightarrow \infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k \rightarrow \infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha \alpha_k$.]



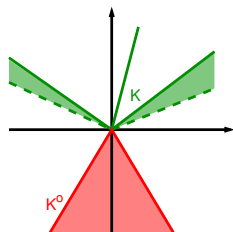
Der Polarkegel

Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

den **Polarkegel** zu K .

[Für K konvex ist $K^\circ = -K^*$ der negative Dualkegel. Er wird auch Normalkegel zu K genannt.]



Der Polarkegel

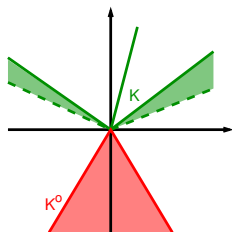
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

den **Polarkegel** zu K .

[Für K konvex ist $K^\circ = -K^*$ der negative Dualkegel. Er wird auch Normalkegel zu K genannt.]

Beachte: Aus $K \subseteq \bar{K}$ folgt $\bar{K}^\circ \subseteq K^\circ$.



Der Polarkegel

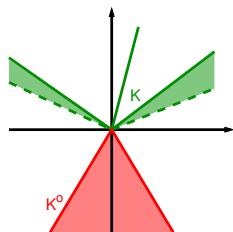
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

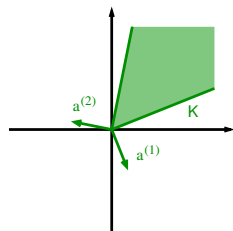
den **Polarkegel** zu K .

[Für K konvex ist $K^\circ = -K^*$ der negative Dualkegel. Er wird auch Normalkegel zu K genannt.]

Beachte: Aus $K \subseteq \bar{K}$ folgt $\bar{K}^\circ \subseteq K^\circ$.



Gegeben $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \dots, m\}$, was ist K° ?



Der Polarkegel

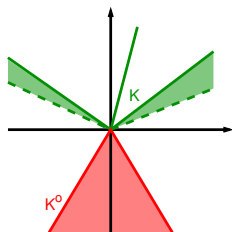
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

den **Polarkegel** zu K .

[Für K konvex ist $K^\circ = -K^*$ der negative Dualkegel. Er wird auch Normalkegel zu K genannt.]

Beachte: Aus $K \subseteq \bar{K}$ folgt $\bar{K}^\circ \subseteq K^\circ$.

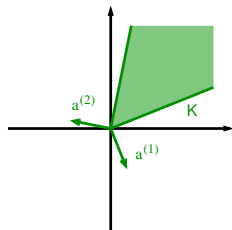


Gegeben $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \dots, m\}$, was ist K° ?

Setze dazu $A := [a^{(1)}, \dots, a^{(m)}]$.

$$p \in K^\circ \Leftrightarrow$$

$$\begin{aligned} 0 &= \max && p^T d \\ &\text{s.t.} && A^T d \leq 0 \\ &&& d \text{ frei} \end{aligned}$$



Der Polarkegel

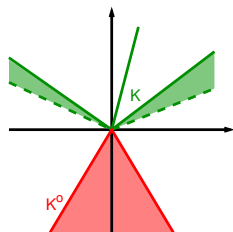
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

den **Polarkegel** zu K .

[Für K konvex ist $K^\circ = -K^*$ der negative Dualkegel. Er wird auch Normalkegel zu K genannt.]

Beachte: Aus $K \subseteq \bar{K}$ folgt $\bar{K}^\circ \subseteq K^\circ$.

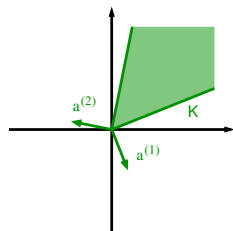


Gegeben $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \dots, m\}$, was ist K° ?

Setze dazu $A := [a^{(1)}, \dots, a^{(m)}]$.

$$p \in K^\circ \Leftrightarrow$$

$$\begin{aligned} 0 &= \max & p^T d & & 0 &= \min & 0^T \lambda \\ \text{s.t.} & & A^T d \leq 0 & \Leftrightarrow & \text{s.t.} & & A \lambda = p \\ & & d \text{ frei} & & & & \lambda \geq 0 \end{aligned}$$



Der Polarkegel

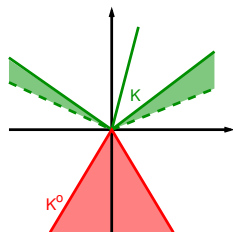
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

den **Polarkegel** zu K .

[Für K konvex ist $K^\circ = -K^*$ der negative Dualkegel. Er wird auch Normalkegel zu K genannt.]

Beachte: Aus $K \subseteq \bar{K}$ folgt $\bar{K}^\circ \subseteq K^\circ$.



Gegeben $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \dots, m\}$, was ist K° ?

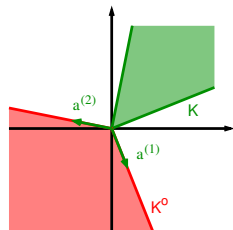
Setze dazu $A := [a^{(1)}, \dots, a^{(m)}]$.

$$p \in K^\circ \Leftrightarrow$$

$$\begin{array}{ll} 0 = \max & p^T d \\ \text{s.t.} & A^T d \leq 0 \\ & d \text{ frei} \end{array} \Leftrightarrow \begin{array}{ll} 0 = \min & 0^T \lambda \\ \text{s.t.} & A \lambda = p \\ & \lambda \geq 0 \end{array}$$

$$\Leftrightarrow \exists \lambda \geq 0 : A \lambda = p$$

Also ist $K^\circ = \{p \in \mathbb{R}^n : p = \sum_{i=1}^m \lambda_i a^{(i)}, \lambda_i \geq 0\}$.



Inhaltsübersicht

Restringierte Nichtlineare Optimierung: Grundlagen

2.1 Aufgabenstellung

2.2 Zulässige Richtungen, Tangential- und Polarkegel

2.3 Notwendige Optimalitätsbedingung

2.4 Der linearisierte Tangentialkegel

2.5 KKT-Bedingungen

2.6 Bedingungen 2. Ordnung

2.7 Sensitivität

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k(x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$f(x^*) \leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|)$$

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k(x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$\begin{aligned} f(x^*) &\leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\ \Rightarrow \quad \nabla f(x^*)^T (x^{(k)} - x^*) &\geq -\mathbf{o}(\|x^{(k)} - x^*\|) \quad | \cdot \alpha_k \geq 0 \end{aligned}$$

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k(x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$\begin{aligned} f(x^*) &\leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\ \Rightarrow \quad \nabla f(x^*)^T (x^{(k)} - x^*) &\geq -\mathbf{o}(\|x^{(k)} - x^*\|) \quad | \cdot \alpha_k \geq 0 \\ \Rightarrow \quad \underbrace{\nabla f(x^*)^T [\alpha_k (x^{(k)} - x^*)]}_{\lim \rightarrow \nabla f(x^*)^T d} &\geq -\underbrace{\mathbf{o}(\alpha_k \|x^{(k)} - x^*\|)}_{\lim \rightarrow \mathbf{o}(\|d\|)=0} \end{aligned}$$

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k (x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$\begin{aligned} f(x^*) &\leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\ \Rightarrow \quad \nabla f(x^*)^T (x^{(k)} - x^*) &\geq -\mathbf{o}(\|x^{(k)} - x^*\|) \quad | \cdot \alpha_k \geq 0 \\ \Rightarrow \quad \underbrace{\nabla f(x^*)^T [\alpha_k (x^{(k)} - x^*)]}_{\lim \rightarrow \nabla f(x^*)^T d} &\geq -\underbrace{\mathbf{o}(\alpha_k \|x^{(k)} - x^*\|)}_{\lim \rightarrow \mathbf{o}(\|d\|)=0} \end{aligned}$$

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist x^* ein lokales Optimum von (P) , so ist jede Richtungsableitung in eine zulässige Richtung nichtnegativ,

$$\nabla f(x^*)^T d \geq 0 \quad \forall d \in T_{\mathcal{X}}(x^*).$$

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k(x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$\begin{aligned} f(x^*) &\leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\ \Rightarrow \nabla f(x^*)^T (x^{(k)} - x^*) &\geq -\mathbf{o}(\|x^{(k)} - x^*\|) \quad | \cdot \alpha_k \geq 0 \\ \Rightarrow \underbrace{\nabla f(x^*)^T [\alpha_k (x^{(k)} - x^*)]}_{\lim \rightarrow \nabla f(x^*)^T d} &\geq -\underbrace{\mathbf{o}(\alpha_k \|x^{(k)} - x^*\|)}_{\lim \rightarrow \mathbf{o}(\|d\|)=0} \end{aligned}$$

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist x^* ein lokales Optimum von (P) , so ist jede Richtungsableitung in eine zulässige Richtung nichtnegativ,

$$\nabla f(x^*)^T d \geq 0 \quad \forall d \in T_{\mathcal{X}}(x^*).$$

[Wie in der freien Optimierung ist dies nicht hinreichend!]

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k (x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$\begin{aligned} f(x^*) &\leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\ \Rightarrow \quad \nabla f(x^*)^T (x^{(k)} - x^*) &\geq -\mathbf{o}(\|x^{(k)} - x^*\|) \quad | \cdot \alpha_k \geq 0 \\ \Rightarrow \quad \underbrace{\nabla f(x^*)^T [\alpha_k (x^{(k)} - x^*)]}_{\lim \rightarrow \nabla f(x^*)^T d} &\geq -\underbrace{\mathbf{o}(\alpha_k \|x^{(k)} - x^*\|)}_{\lim \rightarrow \mathbf{o}(\|d\|)=0} \end{aligned}$$

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist x^* ein lokales Optimum von (P) , so ist jede Richtungsableitung in eine zulässige Richtung nichtnegativ,

$$\nabla f(x^*)^T d \geq 0 \quad \forall d \in T_{\mathcal{X}}(x^*).$$

[Wie in der freien Optimierung ist dies nicht hinreichend!]

Äquivalent: $-\nabla f(x^*) \in T_{\mathcal{X}}(x^*)^\circ$, der negative Gradient liegt im Polarkegel.

2.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \min_{x \in \mathcal{X}} f(x)$

Ist $x^* \in \mathcal{X}$ lokales Optimum und $d \in T_{\mathcal{X}}(x^*)$ mit $d = \lim \alpha_k(x^{(k)} - x^*)$,
dann gilt, wegen $\mathcal{X} \ni x^{(k)} \rightarrow x^*$, für große k

$$\begin{aligned} f(x^*) &\leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\ \Rightarrow \quad \nabla f(x^*)^T (x^{(k)} - x^*) &\geq -\mathbf{o}(\|x^{(k)} - x^*\|) \quad | \cdot \alpha_k \geq 0 \\ \Rightarrow \quad \underbrace{\nabla f(x^*)^T [\alpha_k(x^{(k)} - x^*)]}_{\lim \rightarrow \nabla f(x^*)^T d} &\geq -\underbrace{\mathbf{o}(\alpha_k \|x^{(k)} - x^*\|)}_{\lim \rightarrow \mathbf{o}(\|d\|)=0} \end{aligned}$$

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist x^* ein lokales Optimum von (P) , so ist jede Richtungsableitung in eine zulässige Richtung nichtnegativ,

$$\nabla f(x^*)^T d \geq 0 \quad \forall d \in T_{\mathcal{X}}(x^*).$$

[Wie in der freien Optimierung ist dies nicht hinreichend!]

Äquivalent: $-\nabla f(x^*) \in T_{\mathcal{X}}(x^*)^\circ$, der negative Gradient liegt im Polarkegel.

Ein $\bar{x} \in \mathcal{X}$ mit $-\nabla f(\bar{x}) \in T_{\mathcal{X}}(\bar{x})^\circ$ heißt **stationärer Punkt** von (P) .

Das Ziel der Optimierungsalgorithmen ist, stationäre Punkte zu finden, aber lässt sich $T_{\mathcal{X}}(x)$ durch die Gradienten in x beschreiben?

Inhaltsübersicht

Restringierte Nichtlineare Optimierung: Grundlagen

2.1 Aufgabenstellung

2.2 Zulässige Richtungen, Tangential- und Polarkegel

2.3 Notwendige Optimalitätsbedingung

2.4 Der linearisierte Tangentialkegel

2.5 KKT-Bedingungen

2.6 Bedingungen 2. Ordnung

2.7 Sensitivität

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

[Alle $x^{(k)} \in \mathcal{X}$ für d sind auch in den anderen Mengen.]

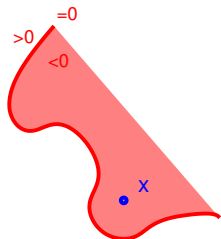
2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als
 $\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i)$, sieht man leicht

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) =$



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als
 $\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i)$, sieht man leicht

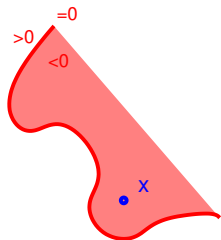
$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

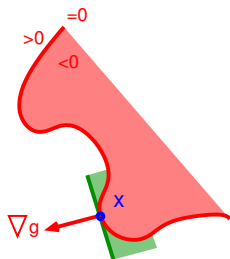
Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) =$



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

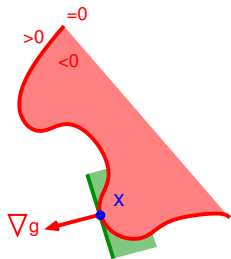
$$1. \ g_i(x) < 0: T_{S_0(g_i)}(x) = \mathbb{R}^n$$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

$$2. \ g_i(x) = 0: T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\},$$

falls $\nabla g_i(x) \neq 0!$



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

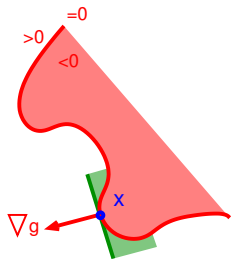
Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\}$,

falls $\nabla g_i(x) \neq 0$! Ist $\nabla g_i(x) = 0$, gilt $T_{S_0(g_i)}(x) \subseteq \mathbb{R}^n$

und Gleichheit nur, wenn x lokales Maximum von g_i ist!

Eine Ungleichung mit $g_i(x) = 0$ heißt **aktiv** (in x).



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

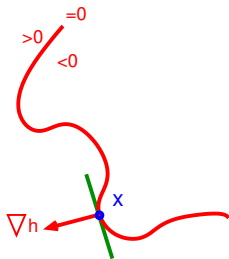
2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\}$,

falls $\nabla g_i(x) \neq 0$! Ist $\nabla g_i(x) = 0$, gilt $T_{S_0(g_i)}(x) \subseteq \mathbb{R}^n$

und Gleichheit nur, wenn x lokales Maximum von g_i ist!

Eine Ungleichung mit $g_i(x) = 0$ heißt **aktiv** (in x).

3. $h_i(x) = 0$: $T_{N_0(h_i)}(x) =$



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

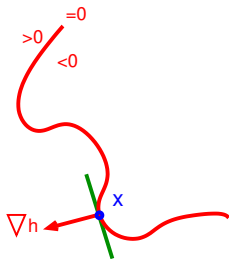
2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\}$,

falls $\nabla g_i(x) \neq 0$! Ist $\nabla g_i(x) = 0$, gilt $T_{S_0(g_i)}(x) \subseteq \mathbb{R}^n$ und Gleichheit nur, wenn x lokales Maximum von g_i ist!

Eine Ungleichung mit $g_i(x) = 0$ heißt **aktiv** (in x).

3. $h_i(x) = 0$: $T_{N_0(h_i)}(x) = \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0\}$,

falls $\nabla h_i(x) \neq 0$ (für $\nabla h_i(x) = 0$ nur $T_{N_0(h_i)}(x) \subseteq \mathbb{R}^n$).



Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\}$,

falls $\nabla g_i(x) \neq 0$! Ist $\nabla g_i(x) = 0$, gilt $T_{S_0(g_i)}(x) \subseteq \mathbb{R}^n$ und Gleichheit nur, wenn x lokales Maximum von g_i ist!

Eine Ungleichung mit $g_i(x) = 0$ heißt **aktiv** (in x).

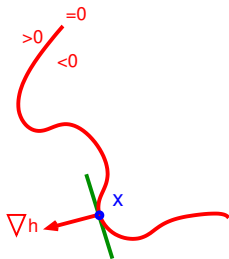
3. $h_i(x) = 0$: $T_{N_0(h_i)}(x) = \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0\}$,

falls $\nabla h_i(x) \neq 0$ (für $\nabla h_i(x) = 0$ nur $T_{N_0(h_i)}(x) \subseteq \mathbb{R}^n$).

Im Folgenden bezeichnet $\mathcal{A}(x) = \{i \in \mathcal{I} : g_i(x) = 0\}$ die **aktive Menge**.

Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$



2.4 Der linearisierte Tangentialkegel

Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \cap \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \Rightarrow d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \text{ und } d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

Keine Einschränkung der Richtungen!

Eine Ungleichung mit $g_i(x) < 0$ heißt **inaktiv** (in x).

2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\}$,

falls $\nabla g_i(x) \neq 0$! Ist $\nabla g_i(x) = 0$, gilt $T_{S_0(g_i)}(x) \subseteq \mathbb{R}^n$ und Gleichheit nur, wenn x lokales Maximum von g_i ist!

Eine Ungleichung mit $g_i(x) = 0$ heißt **aktiv** (in x).

3. $h_i(x) = 0$: $T_{N_0(h_i)}(x) = \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0\}$,

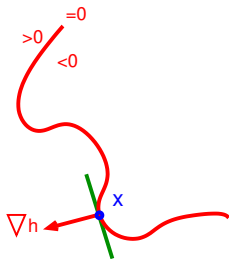
falls $\nabla h_i(x) \neq 0$ (für $\nabla h_i(x) = 0$ nur $T_{N_0(h_i)}(x) \subseteq \mathbb{R}^n$).

Im Folgenden bezeichnet $\mathcal{A}(x) = \{i \in \mathcal{I} : g_i(x) = 0\}$ die **aktive Menge**.

Der Schnitt der von den Gradienten der h_i und aktiven g_i induzierten Unter- und Halbräume ist der **linearisierte Tangentialkegel** zu (P) in x ,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

$$T_P(x)^\circ := \{p \in \mathbb{R}^n : p = \sum_{i \in \mathcal{E}} \mu_i \nabla h_i(x) + \sum_{i \in \mathcal{A}(x)} \lambda_i \nabla g_i(x), \mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{A}(x)}\}.$$



Wann ist $T_{\mathcal{X}}(x) = T_P(x)$?

Wir wissen, $T_{\mathcal{X}}(x) \subseteq T_P(x)$ für alle x , aber für ein bestimmtes \bar{x} kann $T_P(\bar{x})$ zu groß sein, wenn z.B. $\nabla g_i(\bar{x}) = 0$ für ein $i \in \mathcal{A}(\bar{x})$ oder wenn \bar{x} ungünstig am Rand mehrerer Niveaumengen liegt. Dann gilt nur $T_P(x)^\circ \subseteq T_{\mathcal{X}}(x)^\circ$ und für ein Minimum \bar{x} mit $-\nabla f(\bar{x}) \in T_{\mathcal{X}}(\bar{x})^\circ$ ist eventuell $-\nabla f(\bar{x}) \notin T_P(\bar{x})^\circ$, also ist \bar{x} schlecht als stationär erkennbar!

Wann ist $T_{\mathcal{X}}(x) = T_P(x)$?

Wir wissen, $T_{\mathcal{X}}(x) \subseteq T_P(x)$ für alle x , aber für ein bestimmtes \bar{x} kann $T_P(\bar{x})$ zu groß sein, wenn z.B. $\nabla g_i(\bar{x}) = 0$ für ein $i \in \mathcal{A}(\bar{x})$ oder wenn \bar{x} ungünstig am Rand mehrerer Niveaumengen liegt. Dann gilt nur $T_P(x)^\circ \subseteq T_{\mathcal{X}}(x)^\circ$ und für ein Minimum \bar{x} mit $-\nabla f(\bar{x}) \in T_{\mathcal{X}}(\bar{x})^\circ$ ist eventuell $-\nabla f(\bar{x}) \notin T_P(\bar{x})^\circ$, also ist \bar{x} schlecht als stationär erkennbar!

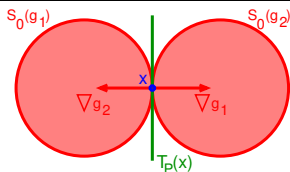
Bsp: $\bar{x} = 0$ und 2 Nebenbedingungen:

$$g_1(x) := \frac{1}{2} \|x + e_1\|^2 - 1 \leq 0, \quad \nabla g_1(\bar{x}) = e_1,$$

$$g_2(x) := \frac{1}{2} \|x - e_1\|^2 - 1 \leq 0, \quad \nabla g_2(\bar{x}) = -e_1,$$

$$\Rightarrow T_P(\bar{x}) = \{d \in \mathbb{R}^n : e_1^T d \leq 0, -e_1^T d \leq 0\} \\ = \{d \in \mathbb{R}^n : d_1 = 0\},$$

aber $T_{\mathcal{X}}(\bar{x}) = \{0\}$. Betrachte $f(x) := x_1^2 + x_2$.



Wann ist $T_{\mathcal{X}}(x) = T_P(x)$?

Wir wissen, $T_{\mathcal{X}}(x) \subseteq T_P(x)$ für alle x , aber für ein bestimmtes \bar{x} kann $T_P(\bar{x})$ zu groß sein, wenn z.B. $\nabla g_i(\bar{x}) = 0$ für ein $i \in \mathcal{A}(\bar{x})$ oder wenn \bar{x} ungünstig am Rand mehrerer Niveaumengen liegt. Dann gilt nur $T_P(x)^\circ \subseteq T_{\mathcal{X}}(x)^\circ$ und für ein Minimum \bar{x} mit $-\nabla f(\bar{x}) \in T_{\mathcal{X}}(\bar{x})^\circ$ ist eventuell $-\nabla f(\bar{x}) \notin T_P(\bar{x})^\circ$, also ist \bar{x} schlecht als stationär erkennbar!

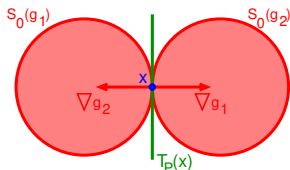
Bsp: $\bar{x} = 0$ und 2 Nebenbedingungen:

$$g_1(x) := \frac{1}{2}\|x + e_1\|^2 - 1 \leq 0, \quad \nabla g_1(\bar{x}) = e_1,$$

$$g_2(x) := \frac{1}{2}\|x - e_1\|^2 - 1 \leq 0, \quad \nabla g_2(\bar{x}) = -e_1,$$

$$\Rightarrow T_P(\bar{x}) = \{d \in \mathbb{R}^n : e_1^T d \leq 0, -e_1^T d \leq 0\} \\ = \{d \in \mathbb{R}^n : d_1 = 0\},$$

aber $T_{\mathcal{X}}(\bar{x}) = \{0\}$. Betrachte $f(x) := x_1^2 + x_2$.



In $\bar{x} \in \mathcal{X}$ ist die **Regularitätsbedingung der linearen Unabhängigkeit** (*linear independence constraint qualification*, kurz **LICQ**) erfüllt, wenn die Gradienten $\nabla h_i(\bar{x})$ ($i \in \mathcal{E}$) und $\nabla g_i(\bar{x})$ ($i \in \mathcal{A}(\bar{x})$) linear unabhängig sind.

Wann ist $T_{\mathcal{X}}(x) = T_P(x)$?

Wir wissen, $T_{\mathcal{X}}(x) \subseteq T_P(x)$ für alle x , aber für ein bestimmtes \bar{x} kann $T_P(\bar{x})$ zu groß sein, wenn z.B. $\nabla g_i(\bar{x}) = 0$ für ein $i \in \mathcal{A}(\bar{x})$ oder wenn \bar{x} ungünstig am Rand mehrerer Niveaumengen liegt. Dann gilt nur $T_P(x)^\circ \subseteq T_{\mathcal{X}}(x)^\circ$ und für ein Minimum \bar{x} mit $-\nabla f(\bar{x}) \in T_{\mathcal{X}}(\bar{x})^\circ$ ist eventuell $-\nabla f(\bar{x}) \notin T_P(\bar{x})^\circ$, also ist \bar{x} schlecht als stationär erkennbar!

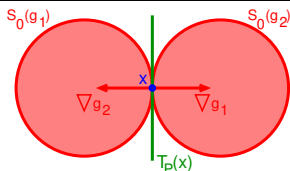
Bsp: $\bar{x} = 0$ und 2 Nebenbedingungen:

$$g_1(x) := \frac{1}{2}\|x + e_1\|^2 - 1 \leq 0, \quad \nabla g_1(\bar{x}) = e_1,$$

$$g_2(x) := \frac{1}{2}\|x - e_1\|^2 - 1 \leq 0, \quad \nabla g_2(\bar{x}) = -e_1,$$

$$\begin{aligned} \Rightarrow T_P(\bar{x}) &= \{d \in \mathbb{R}^n : e_1^T d \leq 0, -e_1^T d \leq 0\} \\ &= \{d \in \mathbb{R}^n : d_1 = 0\}, \end{aligned}$$

aber $T_{\mathcal{X}}(\bar{x}) = \{0\}$. Betrachte $f(x) := x_1^2 + x_2$.



In $\bar{x} \in \mathcal{X}$ ist die **Regularitätsbedingung der linearen Unabhängigkeit** (*linear independence constraint qualification*, kurz **LICQ**) erfüllt, wenn die Gradienten $\nabla h_i(\bar{x})$ ($i \in \mathcal{E}$) und $\nabla g_i(\bar{x})$ ($i \in \mathcal{A}(\bar{x})$) linear unabhängig sind.

Satz

Für ein $\bar{x} \in \mathcal{X}$ sei (LICQ) erfüllt, dann gilt $T_{\mathcal{X}}(\bar{x}) = T_P(\bar{x})$.

[Der Beweis zeigt für jedes $d \in T_P(\bar{x})$ die Existenz einer geeigneten Folge $x^{(k)} \in \mathcal{X}$ mittels des Satzes über implizite Funktionen.]

Inhaltsübersicht

Restringierte Nichtlineare Optimierung: Grundlagen

2.1 Aufgabenstellung

2.2 Zulässige Richtungen, Tangential- und Polarkegel

2.3 Notwendige Optimalitätsbedingung

2.4 Der linearisierte Tangentialkegel

2.5 KKT-Bedingungen

2.6 Bedingungen 2. Ordnung

2.7 Sensitivität

2.5 Existenz von Lagrange-Multiplikatoren

Ein Minimum x^* , das (LICQ) erfüllt, kann über $-\nabla f(x^*) \in T_P(x^*)^\circ$ als stationär erkannt werden.

Satz (Karush-Kuhn-Tucker)

Sei x^* ein lokales Minimum von (P) , in dem (LICQ) erfüllt ist. Dann gibt es (eindeutige) **Lagrange-Multiplikatoren** $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und $\lambda^* \in \mathbb{R}_+^{\mathcal{I}}$, sodass

$$\begin{aligned} \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad [\text{Kompl.}] \end{aligned}$$

2.5 Existenz von Lagrange-Multiplikatoren

Ein Minimum x^* , das (LICQ) erfüllt, kann über $-\nabla f(x^*) \in T_P(x^*)^\circ$ als stationär erkannt werden.

Satz (Karush-Kuhn-Tucker)

Sei x^* ein lokales Minimum von (P) , in dem (LICQ) erfüllt ist. Dann gibt es (eindeutige) **Lagrange-Multiplikatoren** $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und $\lambda^* \in \mathbb{R}_+^{\mathcal{I}}$, sodass

$$\begin{aligned} \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad [\text{Kompl.}] \end{aligned}$$

Beweis: Da x^* lokales Minimum ist, gilt $-\nabla f(x^*) \in T_{\mathcal{X}}(x^*)^\circ$.

Wegen (LICQ) ist $T_P(x^*)^\circ = T_{\mathcal{X}}(x^*)^\circ$, also gibt es $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und

$\lambda^* \in \mathbb{R}^{\mathcal{A}(x^*)}$ mit $-\nabla f(x^*) = \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla g_i(x^*)$.

Für $i \in \mathcal{I} \setminus \mathcal{A}(x^*)$ setze $\lambda_i^* := 0$. □

2.5 Existenz von Lagrange-Multiplikatoren

Ein Minimum x^* , das (LICQ) erfüllt, kann über $-\nabla f(x^*) \in T_P(x^*)^\circ$ als stationär erkannt werden.

Satz (Karush-Kuhn-Tucker)

Sei x^* ein lokales Minimum von (P) , in dem (LICQ) erfüllt ist. Dann gibt es (eindeutige) **Lagrange-Multiplikatoren** $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und $\lambda^* \in \mathbb{R}_+^{\mathcal{I}}$, sodass

$$\begin{aligned} \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad [\text{Kompl.}] \end{aligned}$$

Beweis: Da x^* lokales Minimum ist, gilt $-\nabla f(x^*) \in T_{\mathcal{X}}(x^*)^\circ$.

Wegen (LICQ) ist $T_P(x^*)^\circ = T_{\mathcal{X}}(x^*)^\circ$, also gibt es $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und

$\lambda^* \in \mathbb{R}^{\mathcal{A}(x^*)}$ mit $-\nabla f(x^*) = \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla g_i(x^*)$.

Für $i \in \mathcal{I} \setminus \mathcal{A}(x^*)$ setze $\lambda_i^* := 0$. □

„In einem lokalen Minimum, in dem (LICQ) erfüllt ist, liegt der negative Gradient der Zielfunktion im Kegel, der von den Gradienten der aktiven Nebenbedingungen aufgespannt wird.“

2.5 Existenz von Lagrange-Multiplikatoren

Ein Minimum x^* , das (LICQ) erfüllt, kann über $-\nabla f(x^*) \in T_P(x^*)^\circ$ als stationär erkannt werden.

Satz (Karush-Kuhn-Tucker)

Sei x^* ein lokales Minimum von (P) , in dem (LICQ) erfüllt ist. Dann gibt es (eindeutige) **Lagrange-Multiplikatoren** $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und $\lambda^* \in \mathbb{R}_+^{\mathcal{I}}$, sodass

$$\begin{aligned} \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad [\text{Kompl.}] \end{aligned}$$

Beweis: Da x^* lokales Minimum ist, gilt $-\nabla f(x^*) \in T_{\mathcal{X}}(x^*)^\circ$.

Wegen (LICQ) ist $T_P(x^*)^\circ = T_{\mathcal{X}}(x^*)^\circ$, also gibt es $\mu^* \in \mathbb{R}^{\mathcal{E}}$ und

$\lambda^* \in \mathbb{R}^{\mathcal{A}(x^*)}$ mit $-\nabla f(x^*) = \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{A}(x^*)} \lambda_i^* \nabla g_i(x^*)$.

Für $i \in \mathcal{I} \setminus \mathcal{A}(x^*)$ setze $\lambda_i^* := 0$. □

„In einem lokalen Minimum, in dem (LICQ) erfüllt ist, liegt der negative Gradient der Zielfunktion im Kegel, der von den Gradienten der aktiven Nebenbedingungen aufgespannt wird.“

Warum „Lagrange“-Multiplikatoren?

Lagrange-Funktion und Sattelpunkte

Bestrafung der Verletzung von $h_i(x) = 0$ mit Multiplikatoren $\mu_i \in \mathbb{R}$ und von $g_i(x) \leq 0$ mit $\lambda_i \geq 0$ in der Zielfunktion ergibt die **Lagrange-Funktion**

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x).$$

Es gilt $v(P) = \inf_{x \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(x, \mu, \lambda)$

$\bar{x} \in \mathcal{X}$: $\sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(\bar{x}, \mu, \lambda) = f(\bar{x})$, da $\lambda_i = 0$ für $g_i(\bar{x}) < 0$ optimal,

$\bar{x} \notin \mathcal{X}$: $\sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(\bar{x}, \mu, \lambda) = \infty$, da $h_i(\bar{x}) \neq 0$ oder $g_i(\bar{x}) > 0$ für ein i]

Lagrange-Funktion und Sattelpunkte

Bestrafung der Verletzung von $h_i(x) = 0$ mit Multiplikatoren $\mu_i \in \mathbb{R}$ und von $g_i(x) \leq 0$ mit $\lambda_i \geq 0$ in der Zielfunktion ergibt die **Lagrange-Funktion**

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x).$$

Es gilt $v(P) = \inf_{x \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(x, \mu, \lambda)$

Ein Punkt $(\bar{x}, \bar{\mu}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ heißt (lokaler) **Sattelpunkt** von \mathcal{L} , falls

$$\mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \mu, \lambda)$$

für alle $\lambda \geq 0$, μ und alle x in einer Umgebung von \bar{x} .

Lagrange-Funktion und Sattelpunkte

Bestrafung der Verletzung von $h_i(x) = 0$ mit Multiplikatoren $\mu_i \in \mathbb{R}$ und von $g_i(x) \leq 0$ mit $\lambda_i \geq 0$ in der Zielfunktion ergibt die **Lagrange-Funktion**

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x).$$

Es gilt $v(P) = \inf_{x \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(x, \mu, \lambda)$

Ein Punkt $(\bar{x}, \bar{\mu}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ heißt (lokaler) **Sattelpunkt** von \mathcal{L} , falls

$$\mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \mu, \lambda)$$

für alle $\lambda \geq 0$, μ und alle x in einer Umgebung von \bar{x} .

Eigenschaften von Sattelpunkten $(\bar{x}, \bar{\mu}, \bar{\lambda})$ der Lagrange-Funktion:

$(\bar{\mu}, \bar{\lambda})$ ist Maximum des linearen Problems $\max_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(\bar{x}, \mu, \lambda)$:

Lagrange-Funktion und Sattelpunkte

Bestrafung der Verletzung von $h_i(x) = 0$ mit Multiplikatoren $\mu_i \in \mathbb{R}$ und von $g_i(x) \leq 0$ mit $\lambda_i \geq 0$ in der Zielfunktion ergibt die **Lagrange-Funktion**

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x).$$

Es gilt $v(P) = \inf_{x \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(x, \mu, \lambda)$

Ein Punkt $(\bar{x}, \bar{\mu}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ heißt (lokaler) **Sattelpunkt** von \mathcal{L} , falls

$$\mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \mu, \bar{\lambda})$$

für alle $\lambda \geq 0$, μ und alle x in einer Umgebung von \bar{x} .

Eigenschaften von Sattelpunkten $(\bar{x}, \bar{\mu}, \bar{\lambda})$ der Lagrange-Funktion:

$(\bar{\mu}, \bar{\lambda})$ ist Maximum des linearen Problems $\max_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(\bar{x}, \mu, \lambda)$:

$$\nabla_{\mu} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \Leftrightarrow h_i(\bar{x}) = 0$$

$$\nabla_{\lambda} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \leq 0 \text{ und } \nabla_{\lambda} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda})^T \bar{\lambda} = 0 \Leftrightarrow g_i(\bar{x}) \leq 0 \text{ und } g_i(\bar{x}) \bar{\lambda}_i = 0$$

[Komplementarität: ($i \notin \mathcal{A}(\bar{x}) \Rightarrow \bar{\lambda}_i = 0$) und ($\bar{\lambda}_i > 0 \Rightarrow g_i(\bar{x}) = 0$)]

Lagrange-Funktion und Sattelpunkte

Bestrafung der Verletzung von $h_i(x) = 0$ mit Multiplikatoren $\mu_i \in \mathbb{R}$ und von $g_i(x) \leq 0$ mit $\lambda_i \geq 0$ in der Zielfunktion ergibt die **Lagrange-Funktion**

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x).$$

Es gilt $v(P) = \inf_{x \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(x, \mu, \lambda)$

Ein Punkt $(\bar{x}, \bar{\mu}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ heißt (lokaler) **Sattelpunkt** von \mathcal{L} , falls

$$\mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \mu, \lambda)$$

für alle $\lambda \geq 0$, μ und alle x in einer Umgebung von \bar{x} .

Eigenschaften von Sattelpunkten $(\bar{x}, \bar{\mu}, \bar{\lambda})$ der Lagrange-Funktion:

$(\bar{\mu}, \bar{\lambda})$ ist Maximum des linearen Problems $\max_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(\bar{x}, \mu, \lambda)$:

$$\nabla_{\mu} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \Leftrightarrow h_i(\bar{x}) = 0$$

$\nabla_{\lambda} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \leq 0$ und $\nabla_{\lambda} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda})^T \bar{\lambda} = 0 \Leftrightarrow g_i(\bar{x}) \leq 0$ und $g_i(\bar{x}) \bar{\lambda}_i = 0$

[Komplementarität: ($i \notin \mathcal{A}(\bar{x}) \Rightarrow \bar{\lambda}_i = 0$) und ($\bar{\lambda}_i > 0 \Rightarrow g_i(\bar{x}) = 0$)]

\bar{x} ist lokales Minimum des unrestringierten Problems $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \bar{\mu}, \bar{\lambda})$:

Lagrange-Funktion und Sattelpunkte

Bestrafung der Verletzung von $h_i(x) = 0$ mit Multiplikatoren $\mu_i \in \mathbb{R}$ und von $g_i(x) \leq 0$ mit $\lambda_i \geq 0$ in der Zielfunktion ergibt die **Lagrange-Funktion**

$$\mathcal{L}(x, \mu, \lambda) = f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x).$$

Es gilt $v(P) = \inf_{x \in \mathbb{R}^n} \sup_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(x, \mu, \lambda)$

Ein Punkt $(\bar{x}, \bar{\mu}, \bar{\lambda}) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ heißt (lokaler) **Sattelpunkt** von \mathcal{L} , falls

$$\mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \geq \mathcal{L}(\bar{x}, \mu, \lambda)$$

für alle $\lambda \geq 0$, μ und alle x in einer Umgebung von \bar{x} .

Eigenschaften von Sattelpunkten $(\bar{x}, \bar{\mu}, \bar{\lambda})$ der Lagrange-Funktion:

$(\bar{\mu}, \bar{\lambda})$ ist Maximum des linearen Problems $\max_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}(\bar{x}, \mu, \lambda)$:

$$\nabla_{\mu} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \Leftrightarrow h_i(\bar{x}) = 0$$

$\nabla_{\lambda} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \leq 0$ und $\nabla_{\lambda} \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda})^T \bar{\lambda} = 0 \Leftrightarrow g_i(\bar{x}) \leq 0$ und $g_i(\bar{x}) \bar{\lambda}_i = 0$

[Komplementarität: $(i \notin \mathcal{A}(\bar{x}) \Rightarrow \bar{\lambda}_i = 0)$ und $(\bar{\lambda}_i > 0 \Rightarrow g_i(\bar{x}) = 0)$]

\bar{x} ist lokales Minimum des unrestringierten Problems $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \bar{\mu}, \bar{\lambda})$:

$\nabla_x \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \Leftrightarrow \nabla f(\bar{x}) + \sum_{i \in \mathcal{E}} \bar{\mu}_i \nabla h_i(\bar{x}) + \sum_{i \in \mathcal{I}} \bar{\lambda}_i \nabla g_i(\bar{x}) = 0$,
also $-\nabla f(\bar{x}) \in T_P(\bar{x})^\circ \subseteq T_{\mathcal{X}}(\bar{x})^\circ$ und \bar{x} ist lokales Minimum von (P) .

Die KKT-Bedingungen

Optimierungsverfahren suchen nach Lösungen der **KKT-Bedingungen**

$$\begin{array}{rcl}
 \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) = 0, \\
 \nabla_\mu \mathcal{L} = & h_i(x) = 0, \quad i \in \mathcal{E}, \\
 (KKT) \quad \nabla_\lambda \mathcal{L} = & g_i(x) \leq 0, \quad i \in \mathcal{I}, \\
 & \lambda_i \geq 0, \quad i \in \mathcal{I}, \\
 & g_i(x) \lambda_i = 0, \quad i \in \mathcal{I},
 \end{array}$$

also nach stationären Punkten $(x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ der Lagrange-Funktion. Jeder solche Punkt ist auch stationärer Punkt von (P) (ein Sattelpunkt sogar lokales Minimum), und jeder stationäre Punkt von (P), in dem (LICQ) erfüllt ist, lässt sich so finden.

Die KKT-Bedingungen

Optimierungsverfahren suchen nach Lösungen der **KKT-Bedingungen**

$$\begin{array}{rcl}
 \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) = 0, \\
 \nabla_\mu \mathcal{L} = & h_i(x) = 0, \quad i \in \mathcal{E}, \\
 (KKT) \quad \nabla_\lambda \mathcal{L} = & g_i(x) \leq 0, \quad i \in \mathcal{I}, \\
 & \lambda_i \geq 0, \quad i \in \mathcal{I}, \\
 & g_i(x) \lambda_i = 0, \quad i \in \mathcal{I},
 \end{array}$$

also nach stationären Punkten $(x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ der Lagrange-Funktion. Jeder solche Punkt ist auch stationärer Punkt von (P) (ein Sattelpunkt sogar lokales Minimum), und jeder stationäre Punkt von (P), in dem (LICQ) erfüllt ist, lässt sich so finden.

Gilt $T_P(x^*) = T_{\mathcal{X}}(x^*)$ für ein lokales Optimum x^* , dann gibt es Lagrange-Multiplikatoren (μ^*, λ^*) , sodass (x^*, μ^*, λ^*) die KKT-Bedingungen erfüllt.

Die KKT-Bedingungen

Optimierungsverfahren suchen nach Lösungen der **KKT-Bedingungen**

$$\begin{array}{l}
 \nabla_x \mathcal{L} = \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) = 0, \\
 \nabla_\mu \mathcal{L} = h_i(x) = 0, \quad i \in \mathcal{E}, \\
 \nabla_\lambda \mathcal{L} = g_i(x) \leq 0, \quad i \in \mathcal{I}, \\
 \lambda_i \geq 0, \quad i \in \mathcal{I}, \\
 g_i(x) \lambda_i = 0, \quad i \in \mathcal{I},
 \end{array}
 \quad (KKT)$$

also nach stationären Punkten $(x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ der Lagrange-Funktion. Jeder solche Punkt ist auch stationärer Punkt von (P) (ein Sattelpunkt sogar lokales Minimum), und jeder stationäre Punkt von (P), in dem (LICQ) erfüllt ist, lässt sich so finden.

Gilt $T_P(x^*) = T_{\mathcal{X}}(x^*)$ für ein lokales Optimum x^* , dann gibt es Lagrange-Multiplikatoren (μ^*, λ^*) , sodass (x^*, μ^*, λ^*) die KKT-Bedingungen erfüllt. (LICQ) ist dafür hinreichend, aber nicht notwendig. Es gibt schwächere Bedingungen und in Spezialfällen sind keine notwendig:

Die KKT-Bedingungen

Optimierungsverfahren suchen nach Lösungen der **KKT-Bedingungen**

$$\begin{array}{rcl}
 \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) = 0, \\
 \nabla_{\mu} \mathcal{L} = & h_i(x) = 0, \quad i \in \mathcal{E}, \\
 (KKT) \quad \nabla_{\lambda} \mathcal{L} = & g_i(x) \leq 0, \quad i \in \mathcal{I}, \\
 & \lambda_i \geq 0, \quad i \in \mathcal{I}, \\
 & g_i(x) \lambda_i = 0, \quad i \in \mathcal{I},
 \end{array}$$

also nach stationären Punkten $(x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ der Lagrange-Funktion. Jeder solche Punkt ist auch stationärer Punkt von (P) (ein Sattelpunkt sogar lokales Minimum), und jeder stationäre Punkt von (P), in dem (LICQ) erfüllt ist, lässt sich so finden.

Gilt $T_P(x^*) = T_{\mathcal{X}}(x^*)$ für ein lokales Optimum x^* , dann gibt es Lagrange-Multiplikatoren (μ^*, λ^*) , sodass (x^*, μ^*, λ^*) die KKT-Bedingungen erfüllt. (LICQ) ist dafür hinreichend, aber nicht notwendig. Es gibt schwächere Bedingungen und in Spezialfällen sind keine notwendig:

- Sind alle Nebenbedingungen g_i und h_i linear/affin, dann gilt $T_P(x) = T_{\mathcal{X}}(x)$ für alle $x \in \mathcal{X}$.

Die KKT-Bedingungen

Optimierungsverfahren suchen nach Lösungen der **KKT-Bedingungen**

$$\begin{array}{rcl}
 \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) = 0, \\
 \nabla_\mu \mathcal{L} = & h_i(x) = 0, \quad i \in \mathcal{E}, \\
 (KKT) \quad \nabla_\lambda \mathcal{L} = & g_i(x) \leq 0, \quad i \in \mathcal{I}, \\
 & \lambda_i \geq 0, \quad i \in \mathcal{I}, \\
 & g_i(x) \lambda_i = 0, \quad i \in \mathcal{I},
 \end{array}$$

also nach stationären Punkten $(x, \mu, \lambda) \in \mathbb{R}^n \times \mathbb{R}^{\mathcal{E}} \times \mathbb{R}_+^{\mathcal{I}}$ der Lagrange-Funktion. Jeder solche Punkt ist auch stationärer Punkt von (P) (ein Sattelpunkt sogar lokales Minimum), und jeder stationäre Punkt von (P), in dem (LICQ) erfüllt ist, lässt sich so finden.

Gilt $T_P(x^*) = T_{\mathcal{X}}(x^*)$ für ein lokales Optimum x^* , dann gibt es Lagrange-Multiplikatoren (μ^*, λ^*) , sodass (x^*, μ^*, λ^*) die KKT-Bedingungen erfüllt. (LICQ) ist dafür hinreichend, aber nicht notwendig. Es gibt schwächere Bedingungen und in Spezialfällen sind keine notwendig:

- Sind alle Nebenbedingungen g_i und h_i linear/affin, dann gilt $T_P(x) = T_{\mathcal{X}}(x)$ für alle $x \in \mathcal{X}$.
- Sind alle h_i affin, alle g_i konvex und existiert ein **Slater-Punkt** $\bar{x} \in \mathcal{X}$ mit $g_i(\bar{x}) < 0$ ($i \in \mathcal{I}$), dann gilt $T_P(x) = T_{\mathcal{X}}(x)$ für alle $x \in \mathcal{X}$.

Spezialfall: Opt.-Bed. für (glatte) konvexe Optimierung

Ein (glattes) **konvexes Optimierungsproblem** hat die Gestalt

$$\begin{array}{ll}
 \min & f(x) & f \text{ konvex, glatt} \\
 \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \quad \text{jedes } h_i \text{ affin} \\
 & g_i(x) \leq 0 & i \in \mathcal{I} \quad \text{jedes } g_i \text{ konvex, glatt} \\
 & x \in \Omega & \mathbb{R}^n \text{ oder „einfaches“ Polyeder}
 \end{array}$$

(CP)

Spezialfall: Opt.-Bed. für (glatte) konvexe Optimierung

Ein (glattes) **konvexes Optimierungsproblem** hat die Gestalt

$$\begin{array}{ll}
 \min & f(x) & f \text{ konvex, glatt} \\
 \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} & \text{jedes } h_i \text{ affin} \\
 & g_i(x) \leq 0 & i \in \mathcal{I} & \text{jedes } g_i \text{ konvex, glatt} \\
 & x \in \Omega & & \mathbb{R}^n \text{ oder „einfaches“ Polyeder}
 \end{array}$$

(CP)

Die zulässige Menge \mathcal{X} ist Schnitt konvexer Mengen, also konvex.

Spezialfall: Opt.-Bed. für (glatte) konvexe Optimierung

Ein (glattes) **konvexes Optimierungsproblem** hat die Gestalt

$$\begin{array}{ll}
 \min & f(x) & f \text{ konvex, glatt} \\
 \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \quad \text{jedes } h_i \text{ affin} \\
 & g_i(x) \leq 0 & i \in \mathcal{I} \quad \text{jedes } g_i \text{ konvex, glatt} \\
 & x \in \Omega & \mathbb{R}^n \text{ oder „einfaches“ Polyeder}
 \end{array}$$

(CP)

Die zulässige Menge \mathcal{X} ist Schnitt konvexer Mengen, also konvex.

Satz

Sei (CP) ein konvexes Optimierungsproblem und es gebe einen Slater-Punkt oder alle g_i seien affin, dann sind die KKT-Bedingungen notwendig und hinreichend für die Optimalität.

Spezialfall: Opt.-Bed. für (glatte) konvexe Optimierung

Ein (glattes) **konvexes Optimierungsproblem** hat die Gestalt

$$\begin{array}{ll}
 \min & f(x) & f \text{ konvex, glatt} \\
 \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \quad \text{jedes } h_i \text{ affin} \\
 & g_i(x) \leq 0 & i \in \mathcal{I} \quad \text{jedes } g_i \text{ konvex, glatt} \\
 & x \in \Omega & \mathbb{R}^n \text{ oder „einfaches“ Polyeder}
 \end{array}$$

(CP)

Die zulässige Menge \mathcal{X} ist Schnitt konvexer Mengen, also konvex.

Satz

Sei (CP) ein konvexes Optimierungsproblem und es gebe einen Slater-Punkt oder alle g_i seien affin, dann sind die KKT-Bedingungen notwendig und hinreichend für die Optimalität.

Bsp: Innere-Punkte-Verfahren für LP $\min\{c^T x : Ax = b, x \geq 0\}$.

$f(x) := c^T x - \mu \log x$ (μ Barriereparameter!), $h_i(x) := [b - Ax]_i$, $\Omega := \mathbb{R}_+^n$,
Lagrange-Funktion (Multiplikator y für h): $\mathcal{L}(x, y) := f(x) + y^T(b - Ax)$

Spezialfall: Opt.-Bed. für (glatte) konvexe Optimierung

Ein (glattes) **konvexes Optimierungsproblem** hat die Gestalt

$$\begin{array}{ll}
 \min & f(x) & f \text{ konvex, glatt} \\
 \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \quad \text{jedes } h_i \text{ affin} \\
 & g_i(x) \leq 0 & i \in \mathcal{I} \quad \text{jedes } g_i \text{ konvex, glatt} \\
 & x \in \Omega & \mathbb{R}^n \text{ oder „einfaches“ Polyeder}
 \end{array}
 \quad (CP)$$

Die zulässige Menge \mathcal{X} ist Schnitt konvexer Mengen, also konvex.

Satz

Sei (CP) ein konvexes Optimierungsproblem und es gebe einen Slater-Punkt oder alle g_i seien affin, dann sind die KKT-Bedingungen notwendig und hinreichend für die Optimalität.

Bsp: Innere-Punkte-Verfahren für LP $\min\{c^T x : Ax = b, x \geq 0\}$.

$f(x) := c^T x - \mu \log x$ (μ Barriereparameter!), $h_i(x) := [b - Ax]_i$, $\Omega := \mathbb{R}_+^n$,

Lagrange-Funktion (Multiplikator y für h): $\mathcal{L}(x, y) := f(x) + y^T(b - Ax)$

KKT-Bedingungen (notwendig und hinreichend):

$$\nabla_x \mathcal{L} = c - \mu x^{-1} - A^T y = 0$$

$$\nabla_y \mathcal{L} = b - Ax = 0$$

Mit $z := \mu x^{-1}$ bzw. $z \circ x = \mu \mathbf{1}$ \longrightarrow primal-duales KKT-System.

Inhaltsübersicht

Restringierte Nichtlineare Optimierung: Grundlagen

- 2.1 Aufgabenstellung
- 2.2 Zulässige Richtungen, Tangential- und Polarkegel
- 2.3 Notwendige Optimalitätsbedingung
- 2.4 Der linearisierte Tangentialkegel
- 2.5 KKT-Bedingungen
- 2.6 Bedingungen 2. Ordnung**
- 2.7 Sensitivität

2.6 Notwendige Optimalitätsbedingung 2. Ordnung

Sei x^* ein lokales Minimum von (P), in dem (LICQ) erfüllt ist, und seien (μ^*, λ^*) die Lagrange-Multiplikatoren. Da $\nabla f(x^*)^T d \geq 0$ für alle $d \in T_P(x^*) = T_{\mathcal{X}}(x^*)$, sind nun die Richtungen interessant mit

$$0 = d^T \nabla f(x^*) \stackrel{(KKT)}{=} \sum \mu_i^* \underbrace{d^T \nabla h_i(x^*)}_{=0} + \sum \lambda_i^* \underbrace{d^T \nabla g_i(x^*)}_{\leq 0},$$

also Richtungen aus dem Teilkegel

$$T'_P(x^*, \lambda^*) := \{d \in T_P(x^*) : d^T \nabla g_i(x^*) = 0 \text{ mit } \lambda_i^* > 0 (i \in \mathcal{A}(x^*))\}.$$

2.6 Notwendige Optimalitätsbedingung 2. Ordnung

Sei x^* ein lokales Minimum von (P), in dem (LICQ) erfüllt ist, und seien (μ^*, λ^*) die Lagrange-Multiplikatoren. Da $\nabla f(x^*)^T d \geq 0$ für alle $d \in T_P(x^*) = T_{\mathcal{X}}(x^*)$, sind nun die Richtungen interessant mit

$$0 = d^T \nabla f(x^*) \stackrel{(KKT)}{=} \sum \mu_i^* \underbrace{d^T \nabla h_i(x^*)}_{=0} + \sum \lambda_i^* \underbrace{d^T \nabla g_i(x^*)}_{\leq 0},$$

also Richtungen aus dem Teilkegel

$$T'_P(x^*, \lambda^*) := \{d \in T_P(x^*) : d^T \nabla g_i(x^*) = 0 \text{ mit } \lambda_i^* > 0 (i \in \mathcal{A}(x^*))\}.$$

Satz über implizite Funktionen: Für jedes $d \in T'_P$ gibt es $x^{(k)} \in \mathcal{X}$,

$\alpha_k \geq 0$ mit $d = \lim \alpha_k (x^{(k)} - x^*)$ und $f(x^{(k)}) = \mathcal{L}(x^{(k)}, \mu^*, \lambda^*)$.

Setze $h^{(k)} := x^{(k)} - x^* \rightarrow 0$, nutze $\mathcal{L}_* := \mathcal{L}(x^*, \mu^*, \lambda^*) = f(x^*) =: f_*$,

$$\mathcal{L}(x^* + h^{(k)}, \mu^*, \lambda^*) = \underbrace{\mathcal{L}_*}_{=f_*} + \underbrace{\nabla_x \mathcal{L}_*^T h^{(k)}}_{=0 \text{ (KKT)}} + \frac{1}{2} \underbrace{(h^{(k)})^T \nabla_{xx} \mathcal{L}_* h^{(k)}}_{\Rightarrow \lim \rightarrow d^T \nabla_{xx} \mathcal{L}_* d \geq 0} + \mathbf{o}(\|h\|^2) \geq f_*$$

2.6 Notwendige Optimalitätsbedingung 2. Ordnung

Sei x^* ein lokales Minimum von (P), in dem (LICQ) erfüllt ist, und seien (μ^*, λ^*) die Lagrange-Multiplikatoren. Da $\nabla f(x^*)^T d \geq 0$ für alle $d \in T_P(x^*) = T_{\mathcal{X}}(x^*)$, sind nun die Richtungen interessant mit

$$0 = d^T \nabla f(x^*) \stackrel{(KKT)}{=} \sum \mu_i^* \underbrace{d^T \nabla h_i(x^*)}_{=0} + \sum \lambda_i^* \underbrace{d^T \nabla g_i(x^*)}_{\leq 0},$$

also Richtungen aus dem Teilkegel

$$T'_P(x^*, \lambda^*) := \{d \in T_P(x^*) : d^T \nabla g_i(x^*) = 0 \text{ mit } \lambda_i^* > 0 (i \in \mathcal{A}(x^*))\}.$$

Satz über implizite Funktionen: Für jedes $d \in T'_P$ gibt es $x^{(k)} \in \mathcal{X}$, $\alpha_k \geq 0$ mit $d = \lim \alpha_k (x^{(k)} - x^*)$ und $f(x^{(k)}) = \mathcal{L}(x^{(k)}, \mu^*, \lambda^*)$.

Setze $h^{(k)} := x^{(k)} - x^* \rightarrow 0$, nutze $\mathcal{L}_* := \mathcal{L}(x^*, \mu^*, \lambda^*) = f(x^*) =: f_*$,

$$\mathcal{L}(x^* + h^{(k)}, \mu^*, \lambda^*) = \underbrace{\mathcal{L}_*}_{=f_*} + \underbrace{\nabla_x \mathcal{L}_*^T h^{(k)}}_{=0 \text{ (KKT)}} + \frac{1}{2} \underbrace{(h^{(k)})^T \nabla_{xx} \mathcal{L}_* h^{(k)}}_{\Rightarrow \lim \rightarrow d^T \nabla_{xx} \mathcal{L}_* d \geq 0} + \mathbf{o}(\|h\|^2) \geq f_*$$

Satz (Notwendige Optimalitätsbedingung 2. Ordnung)

Ist x^* ein lokales Minimum von (P), das (LICQ) erfüllt, und sind (μ^*, λ^*) die Lagrange-Multiplikatoren zu x^* , dann gilt

$$d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*, \lambda^*) d \geq 0 \quad \text{für alle } d \in T'_P(x^*, \lambda^*).$$

Hinreichende Optimalitätsbedingungen

Wieder nutzt man, dass die Lagrange-Funktion für die korrekten Multiplikatoren ein gutes unrestringiertes Modell für (P) um x^* ist.

Satz (Hinreichende Optimalitätsbedingungen)

Erfüllt $x^ \in \mathcal{X}$ mit (μ^*, λ^*) die KKT-Bedingungen und gilt*

$$d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*, \lambda^*) d > 0 \quad \text{für alle } d \in T'_P(x^*, \lambda^*) \setminus \{0\},$$

so ist x^ ein lokales Minimum von (P).*

Inhaltsübersicht

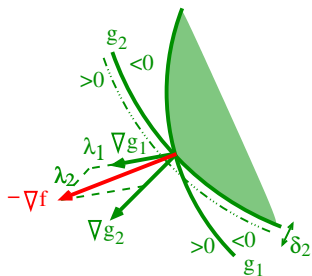
Restringierte Nichtlineare Optimierung: Grundlagen

- 2.1 Aufgabenstellung
- 2.2 Zulässige Richtungen, Tangential- und Polarkegel
- 2.3 Notwendige Optimalitätsbedingung
- 2.4 Der linearisierte Tangentialkegel
- 2.5 KKT-Bedingungen
- 2.6 Bedingungen 2. Ordnung
- 2.7 **Sensitivität**

2.7 Sensitivität

Wie ändert sich die Lösung, wenn man an den rechten Seiten der Nebenbedingungen wackelt?

Sind die Multiplikatoren in einer Umgebung der Lösung eindeutig (dies benötigt LICQ) und strenge Komplementarität, $\lambda_i^* > 0 \Leftrightarrow g_i(x^*) = 0$, geben sie dazu viel Information.



2.7 Sensitivität

Wie ändert sich die Lösung, wenn man an den rechten Seiten der Nebenbedingungen wackelt?

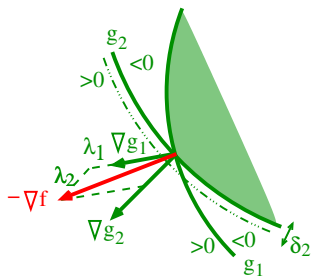
Sind die Multiplikatoren in einer Umgebung der Lösung eindeutig (dies benötigt LICQ) und strenge Komplementarität, $\lambda_i^* > 0 \Leftrightarrow g_i(x^*) = 0$, geben sie dazu viel Information.

Satz (Sensitivität von Optimallösungen)

Für $\delta \in \mathbb{R}^{\mathcal{E} \cup \mathcal{I}}$ bezeichne (P_δ)

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = \delta_i, \quad i \in \mathcal{E}, \\ & g_i(x) \leq \delta_i, \quad i \in \mathcal{I}, \\ & x \in \mathbb{R}^n. \end{aligned}$$

Für (P_0) erfülle ein Punkt x^* (LICQ), die hinr. Opt.-Bed. und strenge Komplementarität bzgl. der Lagrange-Mult. (μ^*, λ^*) . Dann gibt es eine Umgebung $U \subseteq \mathbb{R}^{\mathcal{E} \cup \mathcal{I}}$ um $\delta = 0$ und eine stetige Funktion $x^*(\delta)$ mit $x^*(0) = x^*$ und der Eigenschaft, dass für jedes $\delta \in U$ der Punkt $x^*(\delta)$ lokales Minimum von (P_δ) ist und $\nabla_\delta [f(x^*(\cdot))](0) = - \begin{bmatrix} \mu^* \\ \lambda^* \end{bmatrix}$.



2.7 Sensitivität

Wie ändert sich die Lösung, wenn man an den rechten Seiten der Nebenbedingungen wackelt?

Sind die Multiplikatoren in einer Umgebung der Lösung eindeutig (dies benötigt LICQ) und strenge Komplementarität, $\lambda_i^* > 0 \Leftrightarrow g_i(x^*) = 0$, geben sie dazu viel Information.

Satz (Sensitivität von Optimallösungen)

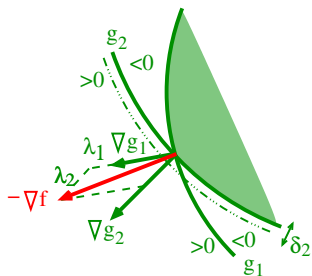
Für $\delta \in \mathbb{R}^{\mathcal{E} \cup \mathcal{I}}$ bezeichne (P_δ)

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = \delta_i, \quad i \in \mathcal{E}, \\ & g_i(x) \leq \delta_i, \quad i \in \mathcal{I}, \\ & x \in \mathbb{R}^n. \end{aligned}$$

Für (P_0) erfülle ein Punkt x^* (LICQ), die hinr. Opt.-Bed. und strenge Komplementarität bzgl. der Lagrange-Mult. (μ^*, λ^*) . Dann gibt es eine Umgebung $U \subseteq \mathbb{R}^{\mathcal{E} \cup \mathcal{I}}$ um $\delta = 0$ und eine stetige Funktion $x^*(\delta)$ mit $x^*(0) = x^*$ und der Eigenschaft, dass für jedes $\delta \in U$ der Punkt $x^*(\delta)$ lokales Minimum von (P_δ) ist und $\nabla_\delta [f(x^*(\cdot))](0) = - \begin{bmatrix} \mu^* \\ \lambda^* \end{bmatrix}$.

Für sehr kleine δ ändert sich der Funktionswert also um etwa $-\delta^T \begin{bmatrix} \mu^* \\ \lambda^* \end{bmatrix}$.

Ein gutes Maß für den Einfluss einer Ungleichung ist $\lambda_i \|\nabla g_i\|$ (Skalierung!).



Inhaltsübersicht

Freie Nichtlineare Optimierung

Restringierte Nichtlineare Optimierung: Grundlagen

Restringierte Optimierung: Verfahren

Ableitungsfreie Optimierung/Direkte Suchverfahren

Inhaltsübersicht

Restringierte Optimierung: Verfahren

3.1 Strafverfahren

3.2 Augmentierte-Lagrange-Verfahren

3.3 Barriere-Verfahren

3.4 Quadratische Optimierung

Konvexe quadratische Optimierung

Gleichungsbeschränkte Quadratische Optimierung

Aktive-Mengen-Verfahren für Quadratische Optimierung

3.5 SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Lokale quadratische Konvergenz über Newton

Globalisierung mit Merit-Funktionen

Globalisierung mit Trust-Region-Ansätzen

Globalisierung mit einem Filter-Ansatz

3.6 Anwendungsbeispiel: Optimalsteuerung

Mathematisches Modell

Zeitoptimale Steuerung

Diskretisierung

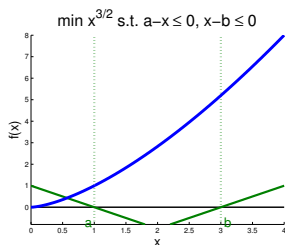
3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

Bsp: $\min f(x) := x^{\frac{3}{2}}$
 s.t. $g_1(x) := a - x \leq 0$
 $g_2(x) := x - b \leq 0$



3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

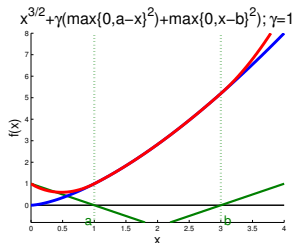
$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

Unzul. mit Strafparameter $\gamma > 0$ bestrafen:

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\}^2 + \max\{0, g_2(x)\}^2]$$

Unrestringiertes Problem: $\min_{x \in \mathbb{R}^n} f_\gamma(x)$

Suche Minimum x_γ^* , vergrößere γ , etc.



3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

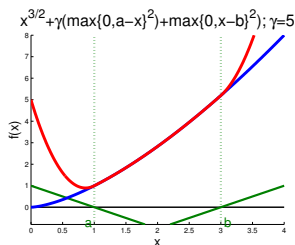
$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

Unzul. mit Strafparameter $\gamma > 0$ bestrafen:

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\}^2 + \max\{0, g_2(x)\}^2]$$

Unrestringiertes Problem: $\min_{x \in \mathbb{R}^n} f_\gamma(x)$

Suche Minimum x_γ^* , vergrößere γ , etc.



3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

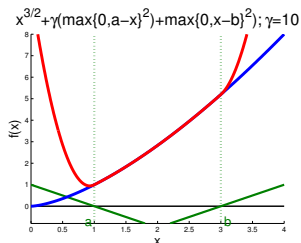
$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

Unzul. mit Strafparameter $\gamma > 0$ bestrafen:

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\}^2 + \max\{0, g_2(x)\}^2]$$

Unrestringiertes Problem: $\min_{x \in \mathbb{R}^n} f_\gamma(x)$

Suche Minimum x_γ^* , vergrößere γ , etc.



3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

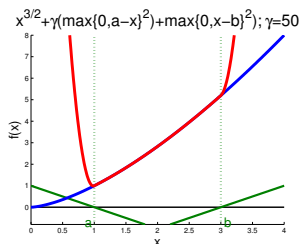
$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

Unzul. mit Strafparameter $\gamma > 0$ bestrafen:

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\}^2 + \max\{0, g_2(x)\}^2]$$

Unrestringiertes Problem: $\min_{x \in \mathbb{R}^n} f_\gamma(x)$

Suche Minimum x_γ^* , vergrößere γ , etc.



3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

$$\begin{array}{ll} \min f(x) & \rightarrow \min_{x \in \mathbb{R}^n} f_\gamma(x) := f(x) + \gamma \left[\underbrace{\sum_{i \in \mathcal{E}} \Psi(h_i(x)) + \sum_{i \in \mathcal{I}} \Phi(g_i(x))}_{=: \sigma(x) \dots \text{Straffunktion}} \right] \\ \text{s.t. } h_i(x) = 0 \quad i \in \mathcal{E} & \\ g_i(x) \leq 0 \quad i \in \mathcal{I} & \\ x \in \mathbb{R}^n & \end{array}$$

$\gamma \geq 0 \dots$ **Strafparameter**

mit $\begin{cases} \Psi(y) > 0 & \text{für } y \neq 0, \\ \Psi(y) = 0 & \text{für } y = 0, \end{cases}$ und $\begin{cases} \Phi(y) > 0 & \text{für } y > 0, \\ \Phi(y) = 0 & \text{für } y \leq 0. \end{cases}$ [Relax.!]

Strafproblem: $\max_{\gamma \geq 0} \Theta(\gamma)$ mit $\Theta(\gamma) := \inf_{x \in \mathbb{R}^n} f_\gamma(x)$

3.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

$$\begin{array}{ll} \min f(x) & \rightarrow \min_{x \in \mathbb{R}^n} f_\gamma(x) := f(x) + \gamma \left[\underbrace{\sum_{i \in \mathcal{E}} \Psi(h_i(x)) + \sum_{i \in \mathcal{I}} \Phi(g_i(x))}_{=: \sigma(x) \dots \text{Straffunktion}} \right] \\ \text{s.t. } h_i(x) = 0 \quad i \in \mathcal{E} & \\ g_i(x) \leq 0 \quad i \in \mathcal{I} & \\ x \in \mathbb{R}^n & \end{array}$$

$\gamma \geq 0 \dots$ **Strafparameter**

mit $\begin{cases} \Psi(y) > 0 & \text{für } y \neq 0, \\ \Psi(y) = 0 & \text{für } y = 0, \end{cases}$ und $\begin{cases} \Phi(y) > 0 & \text{für } y > 0, \\ \Phi(y) = 0 & \text{für } y \leq 0. \end{cases}$ [Relax.!]

Strafproblem: $\max_{\gamma \geq 0} \Theta(\gamma)$ mit $\Theta(\gamma) := \inf_{x \in \mathbb{R}^n} f_\gamma(x)$

Satz (Korrektheit von Strafverfahren)

Ist das Originalproblem zulässig und gibt es für jedes $\gamma \geq 0$ ein globales Optimum x_γ , für das $\Theta(\gamma)$ angenommen wird, und ist $\{x_\gamma : \gamma \geq 0\}$ beschränkt und abgeschlossen, dann sind alle Häufungspunkte von $\{x_\gamma\}$ globale Optimallösungen des Originalproblems.

Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die l_1 -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

Satz (Exaktheit der l_1 -Straffunktion für konvexe Probleme)

Sei (x^*, μ^*, λ^*) ein KKT-Punkt eines konvexen Problems (P) , dann ist für $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$ der Punkt x^* auch globales Optimum von $f_\gamma(x) := f(x) + \gamma [\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$.

Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die l_1 -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

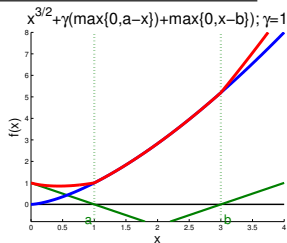
Satz (Exaktheit der l_1 -Straffunktion für konvexe Probleme)

Sei (x^*, μ^*, λ^*) ein KKT-Punkt eines konvexen Problems (P), dann ist für $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$ der Punkt x^* auch globales Optimum von $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$.

Bsp:
$$\begin{aligned} \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$$

In $x^* = 1$ ist g_1 aktiv mit Lagrangemult. $\lambda_1 = \frac{3}{2}$
 [KKT: $0 = \nabla f(1) + \lambda_1 \nabla g_1(1) = \frac{3}{2} - \lambda_1$]



Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die l_1 -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

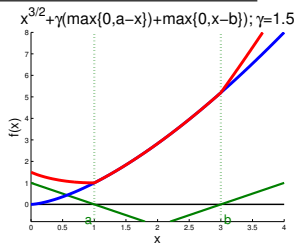
Satz (Exaktheit der l_1 -Straffunktion für konvexe Probleme)

Sei (x^*, μ^*, λ^*) ein KKT-Punkt eines konvexen Problems (P), dann ist für $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$ der Punkt x^* auch globales Optimum von $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$.

$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$$

$$\begin{aligned} \text{In } x^* = 1 \text{ ist } g_1 \text{ aktiv mit Lagrangemult. } \lambda_1 = \frac{3}{2} \\ \text{[KKT: } 0 = \nabla f(1) + \lambda_1 \nabla g_1(1) = \frac{3}{2} - \lambda_1 \end{aligned}$$



Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die l_1 -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

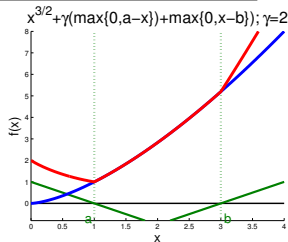
Satz (Exaktheit der l_1 -Straffunktion für konvexe Probleme)

Sei (x^*, μ^*, λ^*) ein KKT-Punkt eines konvexen Problems (P) , dann ist für $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$ der Punkt x^* auch globales Optimum von $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$.

Bsp:
$$\begin{aligned} \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$$

In $x^* = 1$ ist g_1 aktiv mit Lagrangemult. $\lambda_1 = \frac{3}{2}$
 [KKT: $0 = \nabla f(1) + \lambda_1 \nabla g_1(1) = \frac{3}{2} - \lambda_1$]



Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die l_1 -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

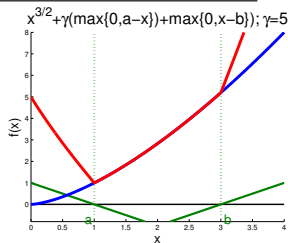
Satz (Exaktheit der l_1 -Straffunktion für konvexe Probleme)

Sei (x^*, μ^*, λ^*) ein KKT-Punkt eines konvexen Problems (P), dann ist für $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$ der Punkt x^* auch globales Optimum von $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$.

Bsp:
$$\begin{aligned} \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$$

In $x^* = 1$ ist g_1 aktiv mit Lagrangemult. $\lambda_1 = \frac{3}{2}$
 [KKT: $0 = \nabla f(1) + \lambda_1 \nabla g_1(1) = \frac{3}{2} - \lambda_1$]



Exakte Strafverfahren und l_1 -Straffunktion

Gibt es für jede lokale Optimallösung x^* des Originalproblems ein endliches $\gamma_{x^*} > 0$, sodass x^* auch lokales Optimum von $f_\gamma(\cdot)$ für alle $\gamma \geq \gamma_{x^*}$ ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die l_1 -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

Satz (Exaktheit der l_1 -Straffunktion für konvexe Probleme)

Sei (x^*, μ^*, λ^*) ein KKT-Punkt eines konvexen Problems (P) , dann ist für $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$ der Punkt x^* auch globales Optimum von $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$.

Was hat γ mit den Lagrangemultiplikatoren zu tun?

Um x^* bestraft $\mathcal{L}(x, \mu^*, \lambda^*) = f(x) + \sum \mu_i^* h_i(x) + \sum \lambda_i^* g_i(x)$ unzulässige Richtungen, in die sich f verbessert, mit $\mu_i^* h_i(x) \geq 0$ bzw. $\lambda_i^* g_i(x) \geq 0$ gerade so, dass der Anstieg $\nabla_x \mathcal{L}(x^*, \mu^*, \lambda^*) = 0$ ist. Jeder größere Anstieg γ macht es nur noch unattraktiver.

Nachteil der l_1 -Straffunktion: f_γ ist nicht differenzierbar!

Die quadratische Straffunktion (quadratic penalty)

Verwende $\Psi(y) := \frac{1}{2}y^2$ und $\Phi(y) := \frac{1}{2} \max\{0, y\}^2$, also

$$f_\gamma(x) = f(x) + \frac{\gamma}{2} \left[\sum_{i \in \mathcal{E}} h_i(x)^2 + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}^2 \right].$$

- f_γ ist stetig diffbar, aber für $\mathcal{I} \neq \emptyset$ i.A. nicht zweimal stetig diffbar.
- Falls $\mathcal{I} = \emptyset$, kann jedes glatte unrestringierte Verfahren zur Bestimmung eines x_γ^* verwendet werden.
- Gerade für Gleichungsnebenbedingungen führt die quadratische Straffunktion zu bananenförmigen Teilproblemen \rightarrow ungünstige Konvergenzeigenschaften.
- Für große γ wird die zweite Ableitung in unzulässige Richtungen sehr groß \rightarrow numerische Probleme.
- Die quadratische Straffunktion ist i.A. nicht exakt, selbst für sehr großes γ bleibt x_γ^* unzulässig (s. Anfangsbeispiel).

Inhaltsübersicht

Restringierte Optimierung: Verfahren

3.1 Strafverfahren

3.2 Augmentierte-Lagrange-Verfahren

3.3 Barriere-Verfahren

3.4 Quadratische Optimierung

Konvexe quadratische Optimierung

Gleichungsbeschränkte Quadratische Optimierung

Aktive-Mengen-Verfahren für Quadratische Optimierung

3.5 SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Lokale quadratische Konvergenz über Newton

Globalisierung mit Merit-Funktionen

Globalisierung mit Trust-Region-Ansätzen

Globalisierung mit einem Filter-Ansatz

3.6 Anwendungsbeispiel: Optimalsteuerung

Mathematisches Modell

Zeitoptimale Steuerung

Diskretisierung

3.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator μ^* exakt.

3.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator μ^* exakt.

Erfüllt x^* die hinreichenden Opt.-bed. mit Multiplikator μ^* , dann gilt

$$\nabla_x \mathcal{L}(x^*, \mu^*) = 0 \text{ und } d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*) d > 0 \quad \forall d \in T_P(x^*) [\Leftrightarrow J_h(x^*) d = 0].$$

3.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator μ^* exakt.

Erfüllt x^* die hinreichenden Opt.-bed. mit Multiplikator μ^* , dann gilt

$$\nabla_x \mathcal{L}(x^*, \mu^*) = 0 \text{ und } d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*) d > 0 \quad \forall d \in T_P(x^*) [\Leftrightarrow J_h(x^*) d = 0].$$

In allen Richtungen orthogonal dazu $[d = J_h(x^*)^T u \text{ für } u \in \mathbb{R}^{\mathcal{E}}]$ sorgt

$$\text{wegen } \nabla_{xx} \mathcal{L}_\gamma(x, \mu) = \nabla_{xx} \mathcal{L}(x, \mu) + \gamma \sum_{i \in \mathcal{E}} \underbrace{h_i(x)}_{=0 \text{ in } x^*} \nabla^2 h_i(x) + \gamma J_h(x)^T J_h(x)$$

der Summand $J_h(x^*)^T J_h(x^*)$ für γ groß genug dafür, dass insgesamt

$$\nabla_x \mathcal{L}_\gamma(x^*, \mu^*) = 0 \text{ und } \nabla_{xx} \mathcal{L}_\gamma(x^*, \mu^*) \succ 0.$$

\Rightarrow Für γ groß genug erfüllt x^* die hinr. freien Opt.-Bed. für $\mathcal{L}_\gamma(\cdot, \mu^*)$.

3.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator μ^* exakt.

Erfüllt x^* die hinreichenden Opt.-bed. mit Multiplikator μ^* , dann gilt

$$\nabla_x \mathcal{L}(x^*, \mu^*) = 0 \text{ und } d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*) d > 0 \quad \forall d \in T_P(x^*) [\Leftrightarrow J_h(x^*) d = 0].$$

In allen Richtungen orthogonal dazu [$d = J_h(x^*)^T u$ für $u \in \mathbb{R}^{\mathcal{E}}$] sorgt

$$\text{wegen } \nabla_{xx} \mathcal{L}_\gamma(x, \mu) = \nabla_{xx} \mathcal{L}(x, \mu) + \gamma \sum_{i \in \mathcal{E}} \underbrace{h_i(x)}_{=0 \text{ in } x^*} \nabla^2 h_i(x) + \gamma J_h(x)^T J_h(x)$$

der Summand $J_h(x^*)^T J_h(x^*)$ für γ groß genug dafür, dass insgesamt

$$\nabla_x \mathcal{L}_\gamma(x^*, \mu^*) = 0 \text{ und } \nabla_{xx} \mathcal{L}_\gamma(x^*, \mu^*) \succ 0.$$

\Rightarrow Für γ groß genug erfüllt x^* die hinr. freien Opt.-Bed. für $\mathcal{L}_\gamma(\cdot, \mu^*)$.

Geometrisch: $\mathcal{L}(\cdot, \mu^*)$ ist wegen $\nabla_x \mathcal{L}_\gamma(x^*, \mu^*) = 0$ in alle Richtungen

um x^* flach und in zulässige Richtungen lokal konvex. Der Strafterm

$\frac{\gamma}{2} \|h(x)\|^2$ macht die Funktion in den unzulässigen Richtungen lokal konvex.

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2$$

Satz (Augmentierte-Lagrange-Verfahren bei Gleichungen)

Erfüllt x^* die hinreichenden Optimalitätsbedingungen für ein restringiertes Problem ohne Ungleichungen mit Multiplikator μ^* , dann gibt es ein $\underline{\gamma} > 0$, sodass x^* für jedes $\gamma > \underline{\gamma}$ ein lokales Minimum von $\min_{x \in \mathbb{R}^n} \mathcal{L}_\gamma(x, \mu^*)$ ist, das die hinreichenden freien Optimalitätsbedingungen erfüllt.

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2$$

Satz (Augmentierte-Lagrange-Verfahren bei Gleichungen)

Erfüllt x^* die hinreichenden Optimalitätsbedingungen für ein restringiertes Problem ohne Ungleichungen mit Multiplikator μ^* , dann gibt es ein $\underline{\gamma} > 0$, sodass x^* für jedes $\gamma > \underline{\gamma}$ ein lokales Minimum von $\min_{x \in \mathbb{R}^n} \mathcal{L}_\gamma(x, \mu^*)$ ist, das die hinreichenden freien Optimalitätsbedingungen erfüllt.

Ist μ^* bekannt \rightarrow lokale freie Optimierung. Aber μ^* ist unbekannt!
Wie wählt man μ im Verfahren? (iterativ, Schritt $k \rightarrow k + 1$)

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2$$

Satz (Augmentierte-Lagrange-Verfahren bei Gleichungen)

Erfüllt x^* die hinreichenden Optimalitätsbedingungen für ein restringiertes Problem ohne Ungleichungen mit Multiplikator μ^* , dann gibt es ein $\underline{\gamma} > 0$, sodass x^* für jedes $\gamma > \underline{\gamma}$ ein lokales Minimum von $\min_{x \in \mathbb{R}^n} \mathcal{L}_\gamma(x, \mu^*)$ ist, das die hinreichenden freien Optimalitätsbedingungen erfüllt.

Ist μ^* bekannt \rightarrow lokale freie Optimierung. Aber μ^* ist unbekannt!

Wie wählt man μ im Verfahren? (iterativ, Schritt $k \rightarrow k+1$)
 μ^* und x^* erfüllen $0 = \nabla_x \mathcal{L}(x^*, \mu^*) = \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*)$.

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2$$

Satz (Augmentierte-Lagrange-Verfahren bei Gleichungen)

Erfüllt x^* die hinreichenden Optimalitätsbedingungen für ein restringiertes Problem ohne Ungleichungen mit Multiplikator μ^* , dann gibt es ein $\underline{\gamma} > 0$, sodass x^* für jedes $\gamma > \underline{\gamma}$ ein lokales Minimum von $\min_{x \in \mathbb{R}^n} \mathcal{L}_\gamma(x, \mu^*)$ ist, das die hinreichenden freien Optimalitätsbedingungen erfüllt.

Ist μ^* bekannt \rightarrow lokale freie Optimierung. Aber μ^* ist unbekannt!

Wie wählt man μ im Verfahren? (iterativ, Schritt $k \rightarrow k+1$)
 μ^* und x^* erfüllen $0 = \nabla_x \mathcal{L}(x^*, \mu^*) = \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*)$.

Sei $x^{(k)}$ die berechnete Lösung zu $\min_{x \in \mathbb{R}^n} \mathcal{L}_{\gamma_k}(x, \mu^{(k)})$, dann ist

$$0 \approx \nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)}) = \nabla f(x^{(k)}) + \underbrace{\sum_{i \in \mathcal{E}} [\mu_i^{(k)} + \gamma_k h_i(x^{(k)})]}_{\approx \mu^*} \nabla h_i(x^{(k)})$$

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2$$

Satz (Augmentierte-Lagrange-Verfahren bei Gleichungen)

Erfüllt x^* die hinreichenden Optimalitätsbedingungen für ein restringiertes Problem ohne Ungleichungen mit Multiplikator μ^* , dann gibt es ein $\underline{\gamma} > 0$, sodass x^* für jedes $\gamma > \underline{\gamma}$ ein lokales Minimum von $\min_{x \in \mathbb{R}^n} \mathcal{L}_\gamma(x, \mu^*)$ ist, das die hinreichenden freien Optimalitätsbedingungen erfüllt.

Ist μ^* bekannt \rightarrow lokale freie Optimierung. Aber μ^* ist unbekannt!

Wie wählt man μ im Verfahren? (iterativ, Schritt $k \rightarrow k+1$)
 μ^* und x^* erfüllen $0 = \nabla_x \mathcal{L}(x^*, \mu^*) = \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*)$.

Sei $x^{(k)}$ die berechnete Lösung zu $\min_{x \in \mathbb{R}^n} \mathcal{L}_{\gamma_k}(x, \mu^{(k)})$, dann ist

$$0 \approx \nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)}) = \nabla f(x^{(k)}) + \sum_{i \in \mathcal{E}} \underbrace{[\mu_i^{(k)} + \gamma_k h_i(x^{(k)})]}_{\approx \mu^*} \nabla h_i(x^{(k)})$$

$$\rightarrow \mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \quad (i \in \mathcal{E})$$

Alg. Schema für Augm.-Lagr.-Verf. mit Gleichungen

0. Wähle Startpunkt $\bar{x}^{(0)}$, $\mu^{(0)}$, $\gamma_0 > 0$, Genauigkeit $\varepsilon_0 > 0$, $k := 0$
1. Bestimme Näherungslösung $x^{(k)}$ zu $\min_{x \in \mathbb{R}^n} \mathcal{L}_{\gamma_k}(x, \mu^{(k)})$ mit Startpunkt $\bar{x}^{(k)}$, so dass $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\| \leq \varepsilon_k$.
2. Ist $\|h(x^{(k)})\|$ und $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\|$ klein genug, STOP.
(u.U. Neustart, falls kein Fortschritt Richtung Zulässigkeit, etc.)
3. Aktualisiere die Multiplikatoren: $\mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \quad (i \in \mathcal{E})$
4. Wähle einen neuen Strafparameter $\gamma_{k+1} \in [\gamma_k, \infty)$
5. Wähle die nächste Genauigkeit $\varepsilon_{k+1} \in (0, \varepsilon_k]$
6. Wähle den nächsten Startpunkt, meist $\bar{x}^{(k+1)} := x^{(k)}$
7. $k \leftarrow k + 1$, GOTO 1.

Alg. Schema für Augm.-Lagr.-Verf. mit Gleichungen

0. Wähle Startpunkt $\bar{x}^{(0)}$, $\mu^{(0)}$, $\gamma_0 > 0$, Genauigkeit $\varepsilon_0 > 0$, $k := 0$
1. Bestimme Näherungslösung $x^{(k)}$ zu $\min_{x \in \mathbb{R}^n} \mathcal{L}_{\gamma_k}(x, \mu^{(k)})$ mit Startpunkt $\bar{x}^{(k)}$, so dass $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\| \leq \varepsilon_k$.
2. Ist $\|h(x^{(k)})\|$ und $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\|$ klein genug, STOP.
(u.U. Neustart, falls kein Fortschritt Richtung Zulässigkeit, etc.)
3. Aktualisiere die Multiplikatoren: $\mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \quad (i \in \mathcal{E})$
4. Wähle einen neuen Strafparameter $\gamma_{k+1} \in [\gamma_k, \infty)$
5. Wähle die nächste Genauigkeit $\varepsilon_{k+1} \in (0, \varepsilon_k]$
6. Wähle den nächsten Startpunkt, meist $\bar{x}^{(k+1)} := x^{(k)}$
7. $k \leftarrow k + 1$, GOTO 1.

Ungleichungen:

werden z.B. in LANCELOT mit Schlupfvariablen in Gln umgewandelt:

$$g_i(x) + s_i = 0, \quad s_i \geq 0 \quad (i \in \mathcal{I})$$

In Schritt 1 wird das Problem dann mit Vorzeichenbedingungen gelöst,

$$\min_{x \in \mathbb{R}^n, s \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}_{\gamma_k}(x, s, \tilde{\mu}^{(k)})$$

(z.B. über ein quadratisches Modell mit linearen Ungln, s. später).

Inhaltsübersicht

Restringierte Optimierung: Verfahren

3.1 Strafverfahren

3.2 Augmentierte-Lagrange-Verfahren

3.3 Barriere-Verfahren

3.4 Quadratische Optimierung

Konvexe quadratische Optimierung

Gleichungsbeschränkte Quadratische Optimierung

Aktive-Mengen-Verfahren für Quadratische Optimierung

3.5 SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Lokale quadratische Konvergenz über Newton

Globalisierung mit Merit-Funktionen

Globalisierung mit Trust-Region-Ansätzen

Globalisierung mit einem Filter-Ansatz

3.6 Anwendungsbeispiel: Optimalsteuerung

Mathematisches Modell

Zeitoptimale Steuerung

Diskretisierung

3.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

3.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp:

3.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp: Innere-Punkte-Verfahren für Lineare Optimierung über Kegeln

3.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp: Innere-Punkte-Verfahren für Lineare Optimierung über Kegeln

Das zulässige Innere $\overset{\circ}{\mathcal{X}} := \{x \in \mathbb{R}^n : g_i(x) < 0, i \in \mathcal{I}\}$ sei nicht leer, $\overset{\circ}{\mathcal{X}} \neq \emptyset$.
Barriere-Funktionen erfüllen folgende Eigenschaften:

- Für $x \notin \overset{\circ}{\mathcal{X}}$ haben sie den Wert ∞ .
- Innerhalb von $\overset{\circ}{\mathcal{X}}$ sind sie glatt.
- Geht eine Folge $x^{(k)} \in \overset{\circ}{\mathcal{X}}$ gegen den Rand von $\overset{\circ}{\mathcal{X}}$, geht der Wert gegen ∞ .

3.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp: Innere-Punkte-Verfahren für Lineare Optimierung über Kegeln

Das zulässige Innere $\overset{\circ}{\mathcal{X}} := \{x \in \mathbb{R}^n : g_i(x) < 0, i \in \mathcal{I}\}$ sei nicht leer, $\overset{\circ}{\mathcal{X}} \neq \emptyset$.
 Barriere-Funktionen erfüllen folgende Eigenschaften:

- Für $x \notin \overset{\circ}{\mathcal{X}}$ haben sie den Wert ∞ .
 - Innerhalb von $\overset{\circ}{\mathcal{X}}$ sind sie glatt.
 - Geht eine Folge $x^{(k)} \in \overset{\circ}{\mathcal{X}}$ gegen den Rand von $\overset{\circ}{\mathcal{X}}$, geht der Wert gegen ∞ .
- Beispiele für $y < 0$: $-\frac{1}{y}$, $-\log(-y)$

3.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp: Innere-Punkte-Verfahren für Lineare Optimierung über Kegeln

Das zulässige Innere $\overset{\circ}{\mathcal{X}} := \{x \in \mathbb{R}^n : g_i(x) < 0, i \in \mathcal{I}\}$ sei nicht leer, $\overset{\circ}{\mathcal{X}} \neq \emptyset$.
Barriere-Funktionen erfüllen folgende Eigenschaften:

- Für $x \notin \overset{\circ}{\mathcal{X}}$ haben sie den Wert ∞ .
- Innerhalb von $\overset{\circ}{\mathcal{X}}$ sind sie glatt.
- Geht eine Folge $x^{(k)} \in \overset{\circ}{\mathcal{X}}$ gegen den Rand von $\overset{\circ}{\mathcal{X}}$, geht der Wert gegen ∞ .

Beispiele für $y < 0$: $-\frac{1}{y}$, $-\log(-y)$

Der Einfluss der Barriere-Funktion wird durch einen **Barriereparameter** β kontrolliert und man sucht ein lokales Minimum für

$$\min_{x \in \mathbb{R}^n} f_\beta(x) := f(x) - \beta \sum_{i \in \mathcal{I}} \log(-g_i(x))$$

ausgehend von einem Startpunkt im Inneren.

Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt $\bar{x}^{(0)} \in \overset{\circ}{\mathcal{X}}$, $\beta_0 > 0$, Genauigkeit $\varepsilon_0 > 0$, $k := 0$
1. Bestimme Näherungslösung $x^{(k)}$ zu $\min_{x \in \mathbb{R}^n} f_{\beta_k}(x)$ mit Startpunkt $\bar{x}^{(k)}$, so dass $\|\nabla_x f_{\beta_k}(x^{(k)})\| \leq \varepsilon_k$.
2. Sind die KKT-Bedingungen näherungsweise erfüllt, STOP.
(teste ob Lösungen nach unendlich gehen, etc.)
3. Wähle einen neuen Barriereparameter $\beta_{k+1} \in (0, \beta_k)$
4. Wähle die nächste Genauigkeit $\varepsilon_{k+1} \in (0, \varepsilon_k]$
5. Wähle den nächsten Startpunkt, meist $\bar{x}^{(k+1)} := x^{(k)}$
6. $k \leftarrow k + 1$, GOTO 1.

Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt $\bar{x}^{(0)} \in \overset{\circ}{\mathcal{X}}$, $\beta_0 > 0$, Genauigkeit $\varepsilon_0 > 0$, $k := 0$
1. Bestimme Näherungslösung $x^{(k)}$ zu $\min_{x \in \mathbb{R}^n} f_{\beta_k}(x)$ mit Startpunkt $\bar{x}^{(k)}$, so dass $\|\nabla_x f_{\beta_k}(x^{(k)})\| \leq \varepsilon_k$.
2. Sind die KKT-Bedingungen näherungsweise erfüllt, STOP.
(teste ob Lösungen nach unendlich gehen, etc.)
3. Wähle einen neuen Barriereparameter $\beta_{k+1} \in (0, \beta_k)$
4. Wähle die nächste Genauigkeit $\varepsilon_{k+1} \in (0, \varepsilon_k]$
5. Wähle den nächsten Startpunkt, meist $\bar{x}^{(k+1)} := x^{(k)}$
6. $k \leftarrow k + 1$, GOTO 1.

KKT-Bedingungen? Sei $x(\beta)$ lokale OL zu $\min_{x \in \mathbb{R}^n} f_{\beta}(x)$, dann ist

$$0 = \nabla f_{\beta}(x(\beta)) = \nabla f(x(\beta)) + \underbrace{\sum_{i \in \mathcal{I}} \frac{\beta}{-g_i(x(\beta))}}_{=: \lambda_i(\beta) \geq 0} \nabla g_i(x(\beta))$$

Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt $\bar{x}^{(0)} \in \overset{\circ}{\mathcal{X}}$, $\beta_0 > 0$, Genauigkeit $\varepsilon_0 > 0$, $k := 0$
1. Bestimme Näherungslösung $x^{(k)}$ zu $\min_{x \in \mathbb{R}^n} f_{\beta_k}(x)$ mit Startpunkt $\bar{x}^{(k)}$, so dass $\|\nabla_x f_{\beta_k}(x^{(k)})\| \leq \varepsilon_k$.
2. Sind die KKT-Bedingungen näherungsweise erfüllt, STOP.
(teste ob Lösungen nach unendlich gehen, etc.)
3. Wähle einen neuen Barriereparameter $\beta_{k+1} \in (0, \beta_k)$
4. Wähle die nächste Genauigkeit $\varepsilon_{k+1} \in (0, \varepsilon_k]$
5. Wähle den nächsten Startpunkt, meist $\bar{x}^{(k+1)} := x^{(k)}$
6. $k \leftarrow k + 1$, GOTO 1.

KKT-Bedingungen? Sei $x(\beta)$ lokale OL zu $\min_{x \in \mathbb{R}^n} f_{\beta}(x)$, dann ist

$$0 = \nabla f_{\beta}(x(\beta)) = \nabla f(x(\beta)) + \underbrace{\sum_{i \in \mathcal{I}} \frac{\beta}{-g_i(x(\beta))}}_{=: \lambda_i(\beta) \geq 0} \nabla g_i(x(\beta))$$

also ist $\nabla f(x(\beta)) + \sum_{i \in \mathcal{I}} \lambda_i(\beta) \nabla g_i(x(\beta)) = 0$

$$g_i(x(\beta)) \leq 0 \quad (i \in \mathcal{I})$$

$$\lambda_i(\beta) \geq 0 \quad (i \in \mathcal{I})$$

$$-\lambda_i(\beta) g_i(x(\beta)) = \beta \quad (i \in \mathcal{I}) \quad [\text{pert. Kompl.}]$$

Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt $\bar{x}^{(0)} \in \overset{\circ}{\mathcal{X}}$, $\beta_0 > 0$, Genauigkeit $\varepsilon_0 > 0$, $k := 0$
1. Bestimme Näherungslösung $x^{(k)}$ zu $\min_{x \in \mathbb{R}^n} f_{\beta_k}(x)$ mit Startpunkt $\bar{x}^{(k)}$, so dass $\|\nabla_x f_{\beta_k}(x^{(k)})\| \leq \varepsilon_k$.
2. Sind die KKT-Bedingungen näherungsweise erfüllt, STOP.
(teste ob Lösungen nach unendlich gehen, etc.)
3. Wähle einen neuen Barriereparameter $\beta_{k+1} \in (0, \beta_k)$
4. Wähle die nächste Genauigkeit $\varepsilon_{k+1} \in (0, \varepsilon_k]$
5. Wähle den nächsten Startpunkt, meist $\bar{x}^{(k+1)} := x^{(k)}$
6. $k \leftarrow k + 1$, GOTO 1.

KKT-Bedingungen? Sei $x(\beta)$ lokale OL zu $\min_{x \in \mathbb{R}^n} f_{\beta}(x)$, dann ist

$$0 = \nabla f_{\beta}(x(\beta)) = \nabla f(x(\beta)) + \underbrace{\sum_{i \in \mathcal{I}} \frac{\beta}{-g_i(x(\beta))}}_{=: \lambda_i(\beta) \geq 0} \nabla g_i(x(\beta))$$

also ist $\nabla f(x(\beta)) + \sum_{i \in \mathcal{I}} \lambda_i(\beta) \nabla g_i(x(\beta)) = 0$

$$g_i(x(\beta)) \leq 0 \quad (i \in \mathcal{I})$$

$$\lambda_i(\beta) \geq 0 \quad (i \in \mathcal{I})$$

$$-\lambda_i(\beta) g_i(x(\beta)) = \beta \quad (i \in \mathcal{I}) \quad [\text{pert. Kompl.}]$$

Für $\beta \rightarrow 0$ wird bei Konvergenz ein KKT-Punkt gefunden!

Satz (Fiacco und McCormick 1968)

Sei $\overset{\circ}{\mathcal{X}} \neq \emptyset$ und x^* ein lokales Minimum, in dem (LICQ) und die hinreichenden Optimalitätsbed. mit streng komplementären Multiplikatoren λ^* erfüllt sind. Dann gilt:

- (i) Für $\bar{\beta}$ klein genug gibt es eine diffbare Funktion $x(\beta): (0, \bar{\beta}) \rightarrow \mathbb{R}^n$ mit $x(\beta) \xrightarrow{\beta \rightarrow 0} x^*$ und $x(\beta)$ ist lokales Minimum von f_β . [\[zent. Pfad\]](#)
- (ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \xrightarrow{\beta \rightarrow 0} \lambda_i^*$ für $i \in \mathcal{I}$.
- (iii) Für β klein genug erfüllt die Hessematrix $\nabla_x^2 f_\beta(x(\beta)) \succ 0$.

Satz (Fiacco und McCormick 1968)

Sei $\overset{\circ}{\mathcal{X}} \neq \emptyset$ und x^* ein lokales Minimum, in dem (LICQ) und die hinreichenden Optimalitätsbed. mit streng komplementären Multiplikatoren λ^* erfüllt sind. Dann gilt:

- (i) Für $\bar{\beta}$ klein genug gibt es eine diffbare Funktion $x(\beta): (0, \bar{\beta}) \rightarrow \mathbb{R}^n$ mit $x(\beta) \xrightarrow{\beta \rightarrow 0} x^*$ und $x(\beta)$ ist lokales Minimum von f_β . [\[zent. Pfad\]](#)
- (ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \xrightarrow{\beta \rightarrow 0} \lambda_i^*$ für $i \in \mathcal{I}$.
- (iii) Für β klein genug erfüllt die Hessematrix $\nabla_x^2 f_\beta(x(\beta)) \succ 0$.

Bemerkungen:

- Die Glattheit von $x(\beta)$ kann zur Beschleunigung durch Prädiktor-Korrektor-Verfahren genutzt werden.

Satz (Fiacco und McCormick 1968)

Sei $\overset{\circ}{\mathcal{X}} \neq \emptyset$ und x^* ein lokales Minimum, in dem (LICQ) und die hinreichenden Optimalitätsbed. mit streng komplementären Multiplikatoren λ^* erfüllt sind. Dann gilt:

- (i) Für $\bar{\beta}$ klein genug gibt es eine diffbare Funktion $x(\beta): (0, \bar{\beta}) \rightarrow \mathbb{R}^n$ mit $x(\beta) \xrightarrow{\beta \rightarrow 0} x^*$ und $x(\beta)$ ist lokales Minimum von f_β . [\[zent. Pfad\]](#)
- (ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \xrightarrow{\beta \rightarrow 0} \lambda_i^*$ für $i \in \mathcal{I}$.
- (iii) Für β klein genug erfüllt die Hessematrix $\nabla_x^2 f_\beta(x(\beta)) \succ 0$.

Bemerkungen:

- Die Glattheit von $x(\beta)$ kann zur Beschleunigung durch Prädiktor-Korrektor-Verfahren genutzt werden.
- $\nabla_x^2 f_\beta$ wird auch schlecht skaliert, aber das ist theoretisch begründbar numerisch meist noch akzeptabel.

Satz (Fiacco und McCormick 1968)

Sei $\overset{\circ}{\mathcal{X}} \neq \emptyset$ und x^* ein lokales Minimum, in dem (LICQ) und die hinreichenden Optimalitätsbed. mit streng komplementären Multiplikatoren λ^* erfüllt sind. Dann gilt:

- (i) Für $\bar{\beta}$ klein genug gibt es eine diffbare Funktion $x(\beta): (0, \bar{\beta}) \rightarrow \mathbb{R}^n$ mit $x(\beta) \xrightarrow{\beta \rightarrow 0} x^*$ und $x(\beta)$ ist lokales Minimum von f_β . [\[zent. Pfad\]](#)
- (ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \xrightarrow{\beta \rightarrow 0} \lambda_i^*$ für $i \in \mathcal{I}$.
- (iii) Für β klein genug erfüllt die Hessematrix $\nabla_x^2 f_\beta(x(\beta)) \succ 0$.

Bemerkungen:

- Die Glattheit von $x(\beta)$ kann zur Beschleunigung durch Prädiktor-Korrektor-Verfahren genutzt werden.
- $\nabla_x^2 f_\beta$ wird auch schlecht skaliert, aber das ist theoretisch begründbar numerisch meist noch akzeptabel.
- Gleichungsnebenbedingungen werden meist durch einen Strafterm eingebunden.

Inhaltsübersicht

Restringierte Optimierung: Verfahren

3.1 Strafverfahren

3.2 Augmentierte-Lagrange-Verfahren

3.3 Barriere-Verfahren

3.4 Quadratische Optimierung

Konvexe quadratische Optimierung

Gleichungsbeschränkte Quadratische Optimierung

Aktive-Mengen-Verfahren für Quadratische Optimierung

3.5 SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Lokale quadratische Konvergenz über Newton

Globalisierung mit Merit-Funktionen

Globalisierung mit Trust-Region-Ansätzen

Globalisierung mit einem Filter-Ansatz

3.6 Anwendungsbeispiel: Optimalsteuerung

Mathematisches Modell

Zeitoptimale Steuerung

Diskretisierung

3.4 Quadratische Optimierung (Quadratic Programming)

QPs sind die typischen Unterprobleme der restringierten Optimierung
(wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll}
 \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\
 \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b \\
 & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\
 & x \in \mathbb{R}^n
 \end{array}$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

3.4 Quadratische Optimierung (Quadratic Programming)

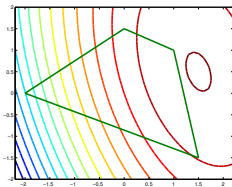
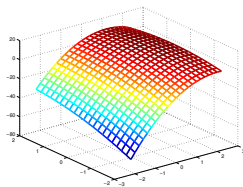
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

$f(\cdot)$ konkav ($Q \prec 0$):
 Alle OL werden in Ecken (bzw. auf minimalen Seiten) angenommen.



3.4 Quadratische Optimierung (Quadratic Programming)

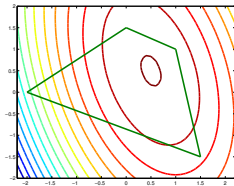
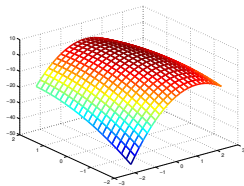
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

$f(\cdot)$ konkav ($Q \prec 0$):
 Alle OL werden in Ecken (bzw. auf minimalen Seiten) angenommen.



3.4 Quadratische Optimierung (Quadratic Programming)

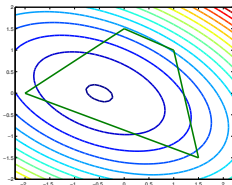
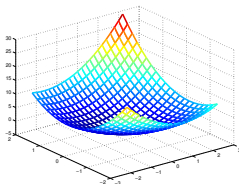
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

$f(\cdot)$ konvex ($Q \succ 0$):
 OL in der Mitte, oder



3.4 Quadratische Optimierung (Quadratic Programming)

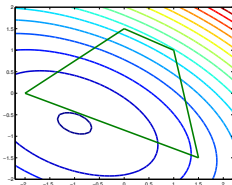
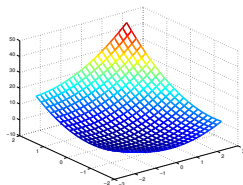
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

$f(\cdot)$ konvex ($Q \succ 0$):
 OL auf einer Seite, oder



3.4 Quadratische Optimierung (Quadratic Programming)

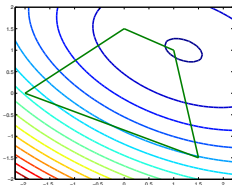
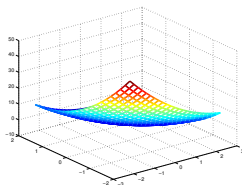
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

$f(\cdot)$ konvex ($Q \succ 0$):
 OL in einer Ecke



3.4 Quadratische Optimierung (Quadratic Programming)

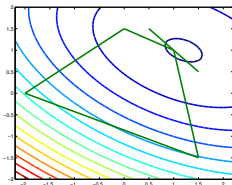
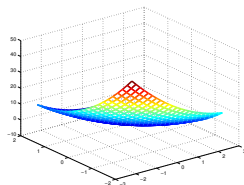
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

degenerierte Lösung:
 aktive Nebenbedingungen
 sind linear abhängig, oder



3.4 Quadratische Optimierung (Quadratic Programming)

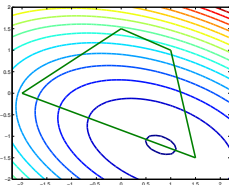
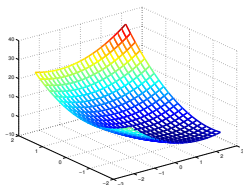
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
 \Rightarrow Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

Einige typische Fälle:

degenerierte Lösung:
 aktive Nebenbedingungen sind linear abhängig, oder die unrestringierte OL liegt auf einer Seite



3.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$(PQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax \geq b$$

3.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$(PQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax \geq b$$

Lagrange-Funktion: $\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Qx + q^T x + \lambda^T (b - Ax)$, ($\lambda \geq 0$)

primales Problem: $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$ [Für x^* : $\exists \lambda^*$ mit $(b - Ax^*)^T \lambda^* = 0$]

3.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$(PQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax \geq b$$

Lagrange-Funktion: $\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Qx + q^T x + \lambda^T (b - Ax), (\lambda \geq 0)$

primales Problem: $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$ [Für x^* : $\exists \lambda^*$ mit $(b - Ax^*)^T \lambda^* = 0$]

duales Problem: $\sup_{\lambda \geq 0} \inf_x \mathcal{L}(x, \lambda) = \sup_{\lambda \geq 0} \left(b^T \lambda + \underbrace{\inf_x \left[\frac{1}{2}x^T Qx + (q - A^T \lambda)^T x \right]}_{\nabla_x = 0 \Rightarrow Qx + q - A^T \lambda = 0, \text{ sonst } -\infty} \right)$

3.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$(PQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax \geq b$$

Lagrange-Funktion: $\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Qx + q^T x + \lambda^T (b - Ax)$, ($\lambda \geq 0$)

primales Problem: $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$ [Für x^* : $\exists \lambda^*$ mit $(b - Ax^*)^T \lambda^* = 0$]

duales Problem: $\sup_{\lambda \geq 0} \inf_x \mathcal{L}(x, \lambda) = \sup_{\lambda \geq 0} (b^T \lambda + \underbrace{\inf_x [\frac{1}{2}x^T Qx + (q - A^T \lambda)^T x]}_{\nabla_x = 0 \Rightarrow Qx + q - A^T \lambda = 0, \text{ sonst } -\infty})$

I.A. enthält das duale QP auch noch x -Variable(!):

$$(DQP) \quad \begin{array}{ll} \max & \frac{1}{2}x^T Qx + q^T x + (b - Ax)^T \lambda \\ \text{s.t.} & Qx + q - A^T \lambda = 0 \\ & x \in \mathbb{R}^n, \lambda \geq 0 \end{array} \quad \text{[erfüllt starke Dualität!]}$$

3.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$\text{(PQP)} \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax \geq b$$

Lagrange-Funktion: $\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Qx + q^T x + \lambda^T (b - Ax)$, ($\lambda \geq 0$)

primales Problem: $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$ [Für x^* : $\exists \lambda^*$ mit $(b - Ax^*)^T \lambda^* = 0$]

duales Problem: $\sup_{\lambda \geq 0} \inf_x \mathcal{L}(x, \lambda) = \sup_{\lambda \geq 0} \left(b^T \lambda + \underbrace{\inf_x \left[\frac{1}{2}x^T Qx + (q - A^T \lambda)^T x \right]}_{\nabla_x = 0 \Rightarrow Qx + q - A^T \lambda = 0, \text{ sonst } -\infty} \right)$

I.A. enthält das duale QP auch noch x -Variable(!):

$$\begin{aligned} \text{(DQP)} \quad & \max \quad \frac{1}{2}x^T Qx + q^T x + (b - Ax)^T \lambda \\ & \text{s.t.} \quad Qx + q - A^T \lambda = 0 \\ & \quad \quad x \in \mathbb{R}^n, \lambda \geq 0 \end{aligned} \quad \text{[erfüllt starke Dualität!]}$$

Für $Q \succ 0$ ist x eindeutig bestimmt: $x(\lambda) = Q^{-1}(A^T \lambda - q)$,

$$\longrightarrow \max_{\lambda \geq 0} -\frac{1}{2}\lambda^T (AQ^{-1}A^T)\lambda + (b + AQ^{-1}q)^T \lambda - \frac{1}{2}q^T Q^{-1}q$$

Nur Vorzeichenbedingungen! Oft besonders effizient lösbar $\rightarrow \lambda^*, x^* = x(\lambda^*)$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverfahren, $A = [a_1, \dots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \frac{1}{2}x^T Qx + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverfahren, $A = [a_1, \dots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \frac{1}{2}x^T Qx + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

Stationarität des Barriere-Unterproblems für $\gamma > 0$ ($\nabla_x = 0$):

$$Qx + q - \gamma \sum a_i \frac{1}{a_i^T x - b_i} = 0$$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverfahren, $A = [a_1, \dots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \frac{1}{2}x^T Qx + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

Stationarität des Barriere-Unterproblems für $\gamma > 0$ ($\nabla_x = 0$):

$$Qx + q - \gamma \sum a_i \frac{1}{a_i^T x - b_i} = 0$$

Führt man Schlupfvariablen $s_i := a_i^T x - b_i \geq 0$

und Dualvariablen (Lagrange-Multiplikatoren) $\lambda_i := \gamma \frac{1}{s_i}$

ein, erhält man ein primal-duales KKT-System:

$$\begin{array}{ll} Qx + q - A^T \lambda & = 0 \quad \text{„duale Zulässigkeit“} \\ Ax - s & = b \quad \text{„primale Zulässigkeit“} \\ s \circ \lambda & = \gamma \mathbf{1} \quad \text{„perturb. Kompl.“} \end{array}$$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverfahren, $A = [a_1, \dots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \frac{1}{2}x^T Qx + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

Stationarität des Barriere-Unterproblems für $\gamma > 0$ ($\nabla_x = 0$):

$$Qx + q - \gamma \sum a_i \frac{1}{a_i^T x - b_i} = 0$$

Führt man Schlupfvariablen $s_i := a_i^T x - b_i \geq 0$

und Dualvariablen (Lagrange-Multiplikatoren) $\lambda_i := \gamma \frac{1}{s_i}$

ein, erhält man ein primal-duales KKT-System:

$$\begin{array}{ll} Qx + q - A^T \lambda & = 0 \quad \text{„duale Zulässigkeit“} \\ Ax - s & = b \quad \text{„primale Zulässigkeit“} \\ s \circ \lambda & = \gamma \mathbf{1} \quad \text{„perturb. Kompl.“} \end{array}$$

Starte mit $s, \lambda > 0$, löse das System näherungsweise mit Newton, bewahre Positivität durch Line-Search, verkleinere γ , etc.

(wie in LP, ε -OL in gleicher polynomialer Anzahl an Iterationen).

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverfahren, $A = [a_1, \dots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + q^T x & \rightarrow \quad \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2}x^T Qx + q^T x - \gamma \sum \log(a_i^T x - b_i) \\ \text{s.t.} \quad & Ax \geq b \end{aligned}$$

Stationarität des Barriere-Unterproblems für $\gamma > 0$ ($\nabla_x = 0$):

$$Qx + q - \gamma \sum a_i \frac{1}{a_i^T x - b_i} = 0$$

Führt man Schlupfvariablen $s_i := a_i^T x - b_i \geq 0$

und Dualvariablen (Lagrange-Multiplikatoren) $\lambda_i := \gamma \frac{1}{s_i}$

ein, erhält man ein primal-duales KKT-System:

$$\begin{aligned} Qx + q - A^T \lambda &= 0 && \text{„duale Zulässigkeit“} \\ Ax - s &= b && \text{„primale Zulässigkeit“} \\ s \circ \lambda &= \gamma \mathbf{1} && \text{„perturb. Kompl.“} \end{aligned}$$

Starte mit $s, \lambda > 0$, löse das System näherungsweise mit Newton, bewahre Positivität durch Line-Search, verkleinere γ , etc.

(wie in LP, ε -OL in gleicher polynomialer Anzahl an Iterationen).

Innere-Punkte-Verf. sind auch auf nicht-konvexe QPs recht gut anwendbar.

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum $\ker A$ habe Dim. k und die Spalten von $Z \in \mathbb{R}^{n \times k}$ als Basis, dann ist $\ker A = \{Zu : u \in \mathbb{R}^k\}$,

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum $\ker A$ habe Dim. k und die Spalten von $Z \in \mathbb{R}^{n \times k}$ als Basis, dann ist $\ker A = \{Zu : u \in \mathbb{R}^k\}$, $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum $\ker A$ habe Dim. k und die Spalten von $Z \in \mathbb{R}^{n \times k}$ als Basis, dann ist $\ker A = \{Zu : u \in \mathbb{R}^k\}$, $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$ und

$$(EQP) \quad \Leftrightarrow \quad \min_{u \in \mathbb{R}^k} \frac{1}{2}u^T Z^T QZ u + (Q\bar{x} + q)^T Z u + \frac{1}{2}\bar{x}^T Q\bar{x} + q^T \bar{x}.$$

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum $\ker A$ habe Dim. k und die Spalten von $Z \in \mathbb{R}^{n \times k}$ als Basis, dann ist $\ker A = \{Zu : u \in \mathbb{R}^k\}$, $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$ und

$$(EQP) \Leftrightarrow \min_{u \in \mathbb{R}^k} \frac{1}{2}u^T Z^T QZ u + (Q\bar{x} + q)^T Z u + \frac{1}{2}\bar{x}^T Q\bar{x} + q^T \bar{x}.$$

\Rightarrow endliche OL nur, falls $Z^T QZ \succeq 0$ und $Z^T QZ u + Z^T(Q\bar{x} + q) = 0$ lösbar,

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum $\ker A$ habe Dim. k und die Spalten von $Z \in \mathbb{R}^{n \times k}$ als Basis, dann ist $\ker A = \{Zu : u \in \mathbb{R}^k\}$, $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$ und

$$(EQP) \Leftrightarrow \min_{u \in \mathbb{R}^k} \frac{1}{2}u^T Z^T QZ u + (Q\bar{x} + q)^T Z u + \frac{1}{2}\bar{x}^T Q\bar{x} + q^T \bar{x}.$$

\Rightarrow endliche OL nur, falls $Z^T QZ \succeq 0$ und $Z^T QZ u + Z^T(Q\bar{x} + q) = 0$ lösbar, eindeutig für $Z^T QZ \succ 0$: $u^* = -(Z^T QZ)^{-1} Z^T(Q\bar{x} + q)$ und $x^* = \bar{x} + Zu$.

3.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein $\bar{x} \in \mathcal{X}$ ($A\bar{x} = b$), ist jedes $x \in \mathcal{X}$ darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum $\ker A$ habe Dim. k und die Spalten von $Z \in \mathbb{R}^{n \times k}$ als Basis, dann ist $\ker A = \{Zu : u \in \mathbb{R}^k\}$, $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$ und

$$(EQP) \Leftrightarrow \min_{u \in \mathbb{R}^k} \frac{1}{2}u^T Z^T QZ u + (Q\bar{x} + q)^T Z u + \frac{1}{2}\bar{x}^T Q\bar{x} + q^T \bar{x}.$$

\Rightarrow endliche OL nur, falls $Z^T QZ \succeq 0$ und $Z^T QZ u + Z^T(Q\bar{x} + q) = 0$ lösbar, eindeutig für $Z^T QZ \succ 0$: $u^* = -(Z^T QZ)^{-1} Z^T(Q\bar{x} + q)$ und $x^* = \bar{x} + Zu$.

Satz (QP mit Gleichungsbedingungen)

Seien die Spalten von Z eine Basis von $\ker A$, $Z^T QZ \succ 0$, und $\bar{x} \in \mathcal{X}$. Ist $x^* = \bar{x} + d^*$ die Lösung des KKT Systems zu (EQP),

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \Leftrightarrow \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -d^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} q + Q\bar{x} \\ A\bar{x} - b \end{bmatrix},$$

dann ist x^* die eindeutige Optimallösung von (EQP).

3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

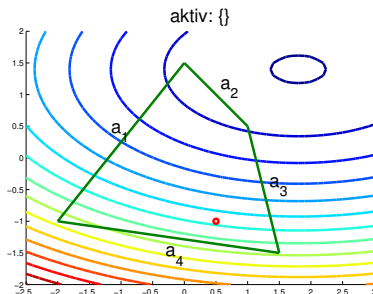
3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 0$, Punkt $x^{(0)}$,
 aktive Menge: \emptyset ,
 Richtung: ?



3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

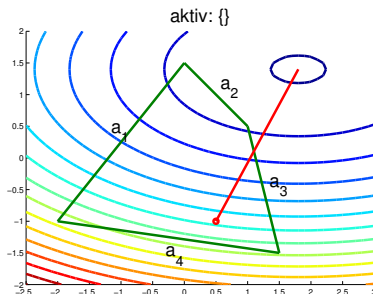
Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 0$, Punkt $x^{(0)}$,

aktive Menge: \emptyset ,

Richtung: $d = -Q^{-1}q - x^{(0)}$,

aktiv wird ?



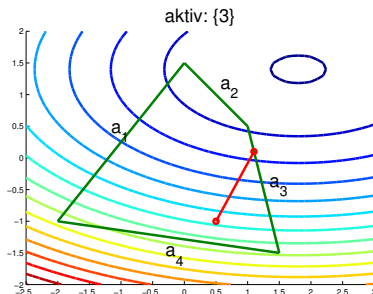
3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 1$, Punkt $x^{(1)}$,
 aktive Menge: $\{3\}$,
 Richtung ?



3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

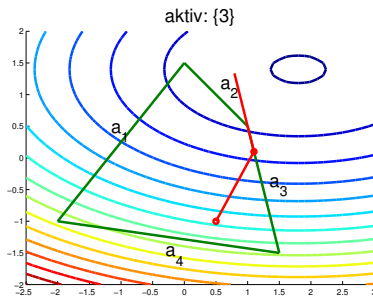
$k = 1$, Punkt $x^{(1)}$,

aktive Menge: $\{3\}$,

Richtung:

$$\begin{bmatrix} Q & a_3 \\ a_3^T & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} q + Qx^{(1)} \\ a_3^T x^{(1)} - b_3 \end{bmatrix},$$

aktiv wird ?



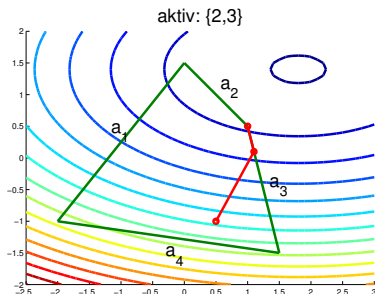
3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 2$, Punkt $x^{(2)}$,
 aktive Menge: $\{2, 3\}$,
 Richtung ?



3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) \rightarrow **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 2$, Punkt $x^{(2)}$,

aktive Menge: $\{2, 3\}$,

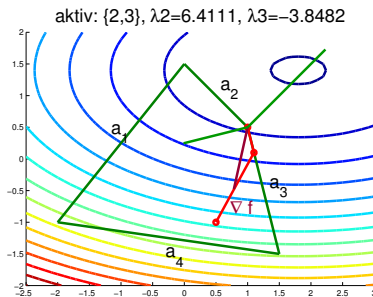
Richtung:

$$\begin{bmatrix} Q & a_2 & a_3 \\ a_2^T & 0 & 0 \\ a_3^T & 0 & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} q + Qx^{(2)} \\ a_2^T x^{(2)} - b_2 \\ a_3^T x^{(2)} - b_3 \end{bmatrix}$$

$\rightarrow d = 0, \lambda_3 < 0$ (!),

$x^{(3)} := x^{(2)}$,

aktiv wird ?



3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

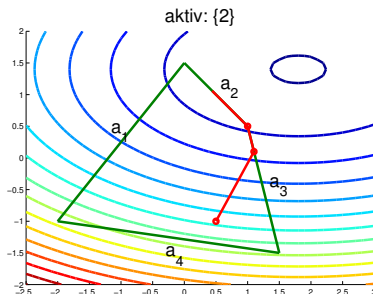
Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 3$, Punkt $x^{(3)}$,
aktive Menge: $\{2\}$,

Richtung:

$$\begin{bmatrix} Q & a_2 \\ a_2^T & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} q + Qx^{(3)} \\ a_2^T x^{(3)} - b_2 \end{bmatrix},$$

aktiv wird ?



3.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2}x^T Qx + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

$k = 3$, Punkt $x^{(3)}$,
aktive Menge: $\{2\}$,

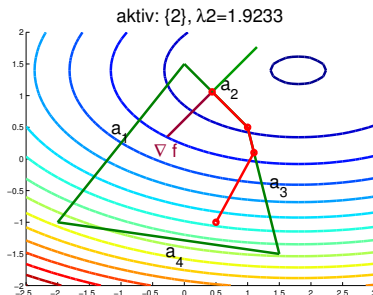
Richtung:

$$\begin{bmatrix} Q & a_2 \\ a_2^T & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} q + Qx^{(2)} \\ a_2^T x^{(3)} - b_2 \end{bmatrix},$$

$$x^{(4)} := x^{(3)} + d,$$

$$\nabla_x f(x^{(4)}) + \lambda_2(-a_2) = 0, \lambda_2 \geq 0,$$

KKT erfüllt!



Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Qx + q^T x \quad s.t. \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Qx + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Qx + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Qx + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Qx + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.

Sonst ($d \neq 0$)

- a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Q x + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.

Sonst ($d \neq 0$)

- a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]
- b) Ist $\alpha_k > 1$, setze $x^{(k+1)} := x^{(k)} + d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$,
sonst $x^{(k+1)} := x^{(k)} + \alpha_k d$ und $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$ mit \hat{i} erzeugt α_k .

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Qx + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
 1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
 2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.
 Sonst ($d \neq 0$)
 - a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]
 - b) Ist $\alpha_k > 1$, setze $x^{(k+1)} := x^{(k)} + d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$,
sonst $x^{(k+1)} := x^{(k)} + \alpha_k d$ und $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$ mit \hat{i} erzeugt α_k .
 3. Setze $k \leftarrow k + 1$, GOTO 1.
-

Bemerkungen:

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Q x + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
 1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
 2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.
 Sonst ($d \neq 0$)
 - a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]
 - b) Ist $\alpha_k > 1$, setze $x^{(k+1)} := x^{(k)} + d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$,
sonst $x^{(k+1)} := x^{(k)} + \alpha_k d$ und $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$ mit \hat{i} erzeugt α_k .
 3. Setze $k \leftarrow k + 1$, GOTO 1.
-

Bemerkungen:

- Wie beim Simplex-Verfahren kann Kreisen auftreten.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Q x + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
 1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
 2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.
 Sonst ($d \neq 0$)
 - a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]
 - b) Ist $\alpha_k > 1$, setze $x^{(k+1)} := x^{(k)} + d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$,
sonst $x^{(k+1)} := x^{(k)} + \alpha_k d$ und $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$ mit \hat{i} erzeugt α_k .
 3. Setze $k \leftarrow k + 1$, GOTO 1.
-

Bemerkungen:

- Wie beim Simplex-Verfahren kann Kreisen auftreten.
- Für $Q \succ 0$ und ohne Degeneriertheiten endet der Alg. für endliches k .

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Q x + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
 1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
 2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.
 Sonst ($d \neq 0$)
 - a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]
 - b) Ist $\alpha_k > 1$, setze $x^{(k+1)} := x^{(k)} + d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$,
sonst $x^{(k+1)} := x^{(k)} + \alpha_k d$ und $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$ mit \hat{i} erzeugt α_k .
 3. Setze $k \leftarrow k + 1$, GOTO 1.
-

Bemerkungen:

- Wie beim Simplex-Verfahren kann Kreisen auftreten.
- Für $Q \succ 0$ und ohne Degeneriertheiten endet der Alg. für endliches k .
- Effiziente Implementationen nutzen, dass sich (EQP) nur wenig ändert.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: $\min x^T Q x + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

0. Bestimme zulässigen Startpunkt $x^{(0)} \in \{x : Ax \geq b\}$ (Simplex/IP), wähle aktive Menge $\mathcal{A}_0 \subseteq \mathcal{A}(x)$, setze $k := 0$.
 1. Bestimme d, λ für (EQP) mit Gleichungsbedingungen $i \in \mathcal{A}_k$
 2. Falls $d = 0$:
 - a) Ist $\lambda \geq 0$ (KKT erfüllt), STOP mit $x^* = x^{(k)}$.
 - b) Wähle $\hat{i} \in \mathcal{A}_k$ mit $\lambda_{\hat{i}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$, $x^{(k+1)} := x^{(k)}$.
 Sonst ($d \neq 0$)
 - a) bestimme $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$. [$+\infty$ für \emptyset]
 - b) Ist $\alpha_k > 1$, setze $x^{(k+1)} := x^{(k)} + d$, $\mathcal{A}_{k+1} := \mathcal{A}_k$,
sonst $x^{(k+1)} := x^{(k)} + \alpha_k d$ und $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$ mit \hat{i} erzeugt α_k .
 3. Setze $k \leftarrow k + 1$, GOTO 1.
-

Bemerkungen:

- Wie beim Simplex-Verfahren kann Kreisen auftreten.
- Für $Q \succ 0$ und ohne Degeneriertheiten endet der Alg. für endliches k .
- Effiziente Implementationen nutzen, dass sich (EQP) nur wenig ändert.
- Für indefinites Q nutzt man z.B. Richtungen zu $\lambda_i(Q) < 0$.

Es gibt viele weitere Varianten für QP,
z.B. Gradienten-Projektions-Verfahren, falls $Ax \geq b$ besonders
einfache Struktur hat ($x \geq 0$ oder $x \in [a, b]$) ...

Kleine QPs sind extrem effizient lösbar (im Millisekunden-Bereich),
große QPs können sehr anspruchsvoll werden.

Konvexe QPs sind auch als SOCPs formulierbar (s. dort), das ist
aber meist weniger effizient als passende QP-Löser zu verwenden.

Inhaltsübersicht

Restringierte Optimierung: Verfahren

- 3.1 Strafverfahren
- 3.2 Augmentierte-Lagrange-Verfahren
- 3.3 Barriere-Verfahren
- 3.4 Quadratische Optimierung
 - Konvexe quadratische Optimierung
 - Gleichungsbeschränkte Quadratische Optimierung
 - Aktive-Mengen-Verfahren für Quadratische Optimierung
- 3.5 SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)
 - Lokale quadratische Konvergenz über Newton
 - Globalisierung mit Merit-Funktionen
 - Globalisierung mit Trust-Region-Ansätzen
 - Globalisierung mit einem Filter-Ansatz
- 3.6 Anwendungsbeispiel: Optimalsteuerung
 - Mathematisches Modell
 - Zeitoptimale Steuerung
 - Diskretisierung

3.5 SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

Idee: Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

3.5 SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

Idee: Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

In 2 Schritten:

1. Lokale Konvergenz:

Das QP-Modell leitet sich aus dem Newton-Verfahren zur Bestimmung eines stationären Punktes der Lagrange-Funktion (= KKT-Bed.) her
→ Newton führt zu lokal quadratischer Konvergenz von SQP-Verf.

3.5 SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

Idee: Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

In 2 Schritten:

1. Lokale Konvergenz:

Das QP-Modell leitet sich aus dem Newton-Verfahren zur Bestimmung eines stationären Punktes der Lagrange-Funktion (= KKT-Bed.) her
→ Newton führt zu lokal quadratischer Konvergenz von SQP-Verf.

2. Globalisierungs-Ansätze:

Fortschritt bezüglich Erfüllung der Nebenbedingungen und Verbesserung der Zielfunktion kann durch Merit-Funktion, Trust-Region-Ansatz oder Filter-Verfahren kontrolliert werden.

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

$$\text{Newton: } \begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$$

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

$$\text{Newton: } \begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$$

$$J_F(x, \mu) = \begin{bmatrix} \nabla_{xx} \mathcal{L} & \nabla_{x\mu} \mathcal{L} \\ \nabla_{\mu x} \mathcal{L} & \nabla_{\mu\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx} \mathcal{L}(x, \mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. [$h: \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}$]

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_{\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

$$\text{Newton: } \begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = 0$$

$$J_F(x, \mu) = \begin{bmatrix} \nabla_{xx} \mathcal{L} & \nabla_{x\mu} \mathcal{L} \\ \nabla_{\mu x} \mathcal{L} & \nabla_{\mu\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx} \mathcal{L}(x, \mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

$$\text{Iteration } k: \text{ Löse } \begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

$$\text{mit } Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}), A_k := J_h(x^{(k)}), h^{(k)} := [h_1(x^{(k)}), \dots, h_{|\mathcal{E}|}(x^{(k)})]^T.$$

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. [$h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}$]

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_{\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

$$\text{Newton: } \begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = 0$$

$$J_F(x, \mu) = \begin{bmatrix} \nabla_{xx} \mathcal{L} & \nabla_{x\mu} \mathcal{L} \\ \nabla_{\mu x} \mathcal{L} & \nabla_{\mu\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx} \mathcal{L}(x, \mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

$$\text{Iteration } k: \text{ Löse } \begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

mit $Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \dots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\begin{aligned} \min \quad & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} \quad & A_k d_x = -h^{(k)} \quad [\text{Zeile } i: h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0] \end{aligned}$$

3.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. [$h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}$]

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_{\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

$$\text{Newton: } \begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = 0$$

$$J_F(x, \mu) = \begin{bmatrix} \nabla_{xx} \mathcal{L} & \nabla_{x\mu} \mathcal{L} \\ \nabla_{\mu x} \mathcal{L} & \nabla_{\mu\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx} \mathcal{L}(x, \mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

$$\text{Iteration } k: \text{ Löse } \begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

mit $Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \dots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\min \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x$$

$$\text{s.t. } A_k d_x = -h^{(k)} \quad [\text{Zeile } i: h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0]$$

Dafür bestimmt sich eine Optimallösung mit Multiplikatoren y durch

$$\begin{bmatrix} Q_k - A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x \\ y \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -h^{(k)} \end{bmatrix}, \text{ setze } y = -(\mu^{(k)} + d_{\mu}) \Leftrightarrow \text{Newton-System}$$

Das QP des SQP-Verfahrens

$$(P) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} , \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{array} \quad , \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x),$$

Das QP des SQP-Verfahrens

$$\begin{aligned}
 & \min f(x) \\
 (P) \quad & \text{s.t. } h_i(x) = 0 \quad i \in \mathcal{E}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x), \\
 & \quad \quad g_i(x) \leq 0 \quad i \in \mathcal{I}
 \end{aligned}$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}, Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$\begin{aligned}
 & \min \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\
 (QP_k) \quad & \text{s.t. } h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\
 & \quad \quad g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I}
 \end{aligned}$$

Das QP des SQP-Verfahrens

$$\begin{aligned} \min & f(x) \\ (P) \text{ s.t. } & h_i(x) = 0 \quad i \in \mathcal{E}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x), \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{aligned}$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}, Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$\begin{aligned} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ (QP_k) \text{ s.t. } & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \rightarrow \text{s.t. } A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} & & A_k^g d_x \leq -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}] \end{aligned}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Das QP des SQP-Verfahrens

$$\begin{aligned} \min & f(x) \\ (P) \text{ s.t. } & h_i(x) = 0 \quad i \in \mathcal{E}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x), \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{aligned}$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}, Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$\begin{aligned} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ (QP_k) \text{ s.t. } & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} & \rightarrow \text{s.t. } & A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} & & A_k^g d_x \leq -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}] \end{aligned}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Bemerkungen:

- Ist $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ nahe genug an (x^*, μ^*, λ^*) , das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert **lokal quadratische Konvergenz**.

Das QP des SQP-Verfahrens

$$(P) \quad \min f(x)$$

$$\text{s.t.} \quad h_i(x) = 0 \quad i \in \mathcal{E}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x),$$

$$g_i(x) \leq 0 \quad i \in \mathcal{I}$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}, Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$(QP_k) \quad \min \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \quad \min \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x$$

$$\text{s.t.} \quad h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \quad \rightarrow \quad \text{s.t.} \quad A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}]$$

$$g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} \quad \quad \quad A_k^g d_x \leq -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}]$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Bemerkungen:

- Ist $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ nahe genug an (x^*, μ^*, λ^*) , das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert **lokal quadratische Konvergenz**.
- Q_k enthält Krümmung von f **und** den aktiven Nebenbedingungen,

$$Q_k = \nabla^2 f(x^{(k)}) + \sum_{i \in \mathcal{E}} \mu_i^{(k)} \nabla^2 h_i(x^{(k)}) + \sum_{i \in \mathcal{I}} \lambda_i^{(k)} \nabla^2 g_i(x^{(k)}),$$
 → kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.

Das QP des SQP-Verfahrens

$$\begin{aligned} \min & f(x) \\ (P) \text{ s.t. } & h_i(x) = 0 \quad i \in \mathcal{E}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x), \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{aligned}$$

bestimme $x^{(k+1)} := x^{(k)} + d_x^{(k)}$ für geg. $\mu^{(k)}, \lambda^{(k)}, Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ aus

$$\begin{aligned} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ (QP_k) \text{ s.t. } & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \rightarrow \text{s.t. } A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} & & A_k^g d_x \leq -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}] \end{aligned}$$

wobei $A_k^h = J_h(x^{(k)})$, $A_k^g = J_g(x^{(k)})$, setze $\mu^{(k+1)} := -y_h^{(k)}$, $\lambda^{(k+1)} := -y_g^{(k)}$.

Bemerkungen:

- Ist $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$ nahe genug an (x^*, μ^*, λ^*) , das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert **lokal quadratische Konvergenz**.
- Q_k enthält Krümmung von f **und** den aktiven Nebenbedingungen,

$$Q_k = \nabla^2 f(x^{(k)}) + \sum_{i \in \mathcal{E}} \mu_i^{(k)} \nabla^2 h_i(x^{(k)}) + \sum_{i \in \mathcal{I}} \lambda_i^{(k)} \nabla^2 g_i(x^{(k)}),$$
 → kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.
- Die Hessematrizen in Q_k sind durch BFGS, etc. approximierbar.

3.5.2 Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

3.5.2 Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

l_1 -Merit-Funktion: Für geg. Strafparameter $\gamma > 0$ verwende

$$f_\gamma(x) := f(x) + \gamma \left[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\} \right]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

3.5.2 Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

l_1 -Merit-Funktion: Für geg. Strafparameter $\gamma > 0$ verwende

$$f_\gamma(x) := f(x) + \gamma \left[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\} \right]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

Augmentierte-Lagrange-Merit-Funktion von Fletcher: Für geg. $\gamma > 0$,

$$\mathcal{L}_\gamma(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x) + \gamma \left[\sum_{i \in \mathcal{E}} h_i(x)^2 + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}^2 \right]$$

- differenzierbar und für γ groß genug ist $d_x^{(k)}$ Abstiegsrichtung
- Die Wahl von γ hat starken Einfluss auf das Konvergenzverhalten ...

3.5.2 Globalisierung mit Merit-Funktionen

Verwende die Lösung $d_x^{(k)}$ von (QP_k) als Suchrichtung und bestimme $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$ durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

I_1 -Merit-Funktion: Für geg. Strafparameter $\gamma > 0$ verwende

$$f_\gamma(x) := f(x) + \gamma \left[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\} \right]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

Augmentierte-Lagrange-Merit-Funktion von Fletcher: Für geg. $\gamma > 0$,

$$\mathcal{L}_\gamma(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x) + \gamma \left[\sum_{i \in \mathcal{E}} h_i(x)^2 + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}^2 \right]$$

- differenzierbar und für γ groß genug ist $d_x^{(k)}$ Abstiegsrichtung
- Die Wahl von γ hat starken Einfluss auf das Konvergenzverhalten ...

Generelles Problem: **Maratos-Effekt**

Line-Search bzgl. Merit-Funktion verbietet oft Schrittweite 1 im Gebiet der lokalen quadratischen Konvergenz \rightarrow schlechte lokale Konvergenz.

Es gibt heuristische Gegenstrategien ...

3.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

3.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)

Idee: x ist zulässig, wenn $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.

3.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)

Idee: x ist zulässig, wenn $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.

→ Löse als Trust-Region-Unterproblem für Strafparameter $\gamma > 0$

$$\begin{array}{ll}
 \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s \\
 \text{s.t.} & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s \quad i \in \mathcal{E} \\
 & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s \quad i \in \mathcal{I} \\
 & -\Delta \leq [d_x]_i \leq \Delta \quad (i = 1, \dots, n), s \geq 0
 \end{array}$$

3.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \leq \Delta$.
- Schwierigkeit: Ist $x^{(k)}$ unzulässig, kann durch $\|d_x\|_\infty \leq \Delta$ auch (QP_k) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann Δ zu wählen, etc.

Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)

Idee: x ist zulässig, wenn $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.

→ Löse als Trust-Region-Unterproblem für Strafparameter $\gamma > 0$

$$\begin{aligned}
 \min \quad & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s \\
 \text{s.t.} \quad & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s \quad i \in \mathcal{E} \\
 & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s \quad i \in \mathcal{I} \\
 & -\Delta \leq [d_x]_i \leq \Delta \quad (i = 1, \dots, n), s \geq 0
 \end{aligned}$$

Der Trust-Region Algorithmus vergleicht Fortschritt der Funktion $f_\gamma(d) := f(x^{(k)} + d) + \gamma \max(\{|h_i(x^{(k)} + d)| : i \in \mathcal{E}\} \cup \{g_i(x^{(k)} + d) : i \in \mathcal{I}\} \cup \{0\})$ zum Fortschritt im Modell und passt Δ entsprechend an. Strafparameter γ wird nur bei Verletzung der Nebenbed. vergrößert, wenn der Fortschritt Richtung Zulässigkeit im Verhältnis zur Schrittlänge zu klein ist.

3.5.4 Globalisierung mit einem Filter-Ansatz

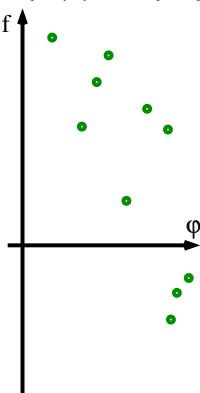
Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .



3.5.4 Globalisierung mit einem Filter-Ansatz

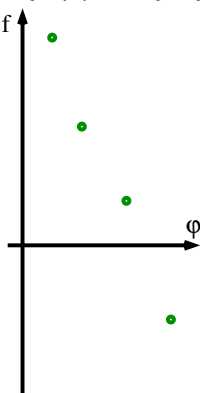
Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

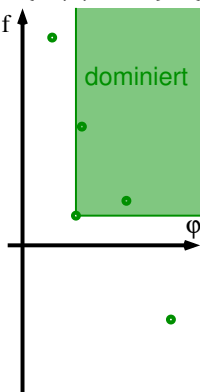
Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

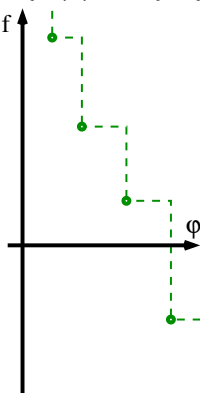
Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 . f

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

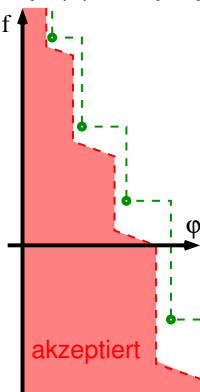
$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

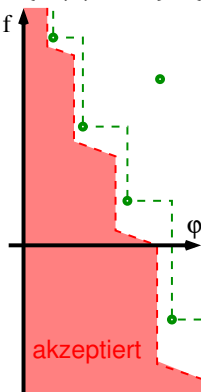
$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

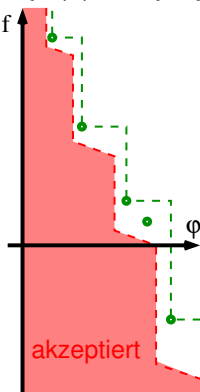
$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

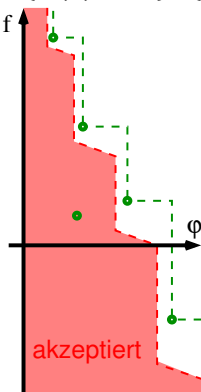
$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 . f

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

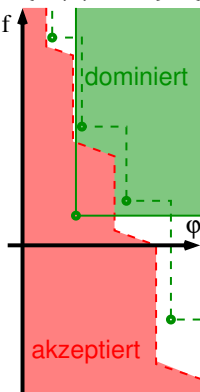
$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von \mathcal{F}_k akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

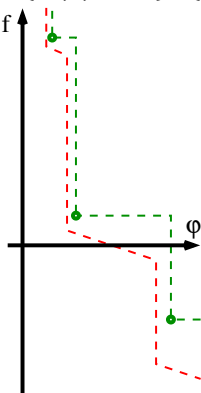
$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von \mathcal{F}_k akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$



3.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$... Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$ als Punkt in \mathbb{R}^2 .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ **dominiert** $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von \mathcal{F}_k für gegebene

$0 < \gamma_1 < \gamma_2 < 1$ **akzeptiert**, falls

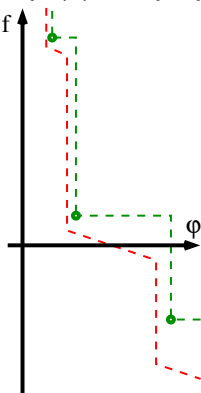
$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von \mathcal{F}_k akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$

[Der Filter sichert monotone Verbesserung und sammelt Erfahrung über gute Werte.]



Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$\begin{array}{ll}
 \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\
 \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\
 & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} \\
 & -\Delta \leq [d_x]_i \leq \Delta \quad i \in \{1, \dots, n\}
 \end{array}$$

(QP_k^Δ)

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$\begin{array}{ll}
 \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\
 \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\
 & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} \\
 & -\Delta \leq [d_x]_i \leq \Delta \quad i \in \{1, \dots, n\}
 \end{array}$$

(QP_k^Δ)

Falls (QP_k^Δ) unzulässig ist (in $\|d_x\|_\infty \leq \Delta$ ist kein zul. Punkt), führe **Wiederherstellungs-Phase** (restauration phase) durch:

Minimiere nur Verletzung $\varphi(x)$, bis für Filter akzeptabel \rightarrow neues Δ , aber
 \Rightarrow der Filter darf nur unzulässige Punkte $(\hat{f}, \hat{\varphi})$ mit $\hat{\varphi} > 0$ enthalten!

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$\begin{array}{ll}
 \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\
 \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\
 & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} \\
 & -\Delta \leq [d_x]_i \leq \Delta \quad i \in \{1, \dots, n\}
 \end{array}
 \quad (QP_k^\Delta)$$

Falls (QP_k^Δ) unzulässig ist (in $\|d_x\|_\infty \leq \Delta$ ist kein zul. Punkt), führe **Wiederherstellungs-Phase** (restauration phase) durch:

Minimiere nur Verletzung $\varphi(x)$, bis für Filter akzeptabel \rightarrow neues Δ , aber
 \Rightarrow der Filter darf nur unzulässige Punkte $(\hat{f}, \hat{\varphi})$ mit $\hat{\varphi} > 0$ enthalten!

Verwende dazu folgende Sonderregel:

Vom Filter akzeptierte Punkte werden nur dann zum Filter hinzugefügt, wenn der nächste Schritt zu einer Wiederherstellung führt $\Rightarrow \varphi(x^{(k)}) > 0$.

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$\begin{aligned}
 (QP_k^\Delta) \quad & \min \quad \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\
 & \text{s.t.} \quad h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\
 & \quad \quad g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} \\
 & \quad \quad -\Delta \leq [d_x]_i \leq \Delta \quad i \in \{1, \dots, n\}
 \end{aligned}$$

Falls (QP_k^Δ) unzulässig ist (in $\|d_x\|_\infty \leq \Delta$ ist kein zul. Punkt), führe **Wiederherstellungs-Phase** (restauration phase) durch:

Minimiere nur Verletzung $\varphi(x)$, bis für Filter akzeptabel \rightarrow neues Δ , aber
 \Rightarrow der Filter darf nur unzulässige Punkte $(\hat{f}, \hat{\varphi})$ mit $\hat{\varphi} > 0$ enthalten!

Verwende dazu folgende Sonderregel:

Vom Filter akzeptierte Punkte werden nur dann zum Filter hinzugefügt, wenn der nächste Schritt zu einer Wiederherstellung führt $\Rightarrow \varphi(x^{(k)}) > 0$.

Der Algorithmus dazu nutzt weitere Parameter:

- $\underline{\Delta}$... Mindestradius nach Wiederherstellung
- u ... maximale Unzulässigkeit für φ im Filter
- $\sigma \in (0, 1)$... Modellqualität für Reduktion von Δ

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x))$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
 1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
 2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.
 3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
 4. Wird $(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x))$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
 5. Setze $\Delta q := -\frac{1}{2}d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
- [Der Fortschritt in f ist im Vergleich zum Modell zu gering.]

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x))$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2}d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
[Der Fortschritt in f ist im Vergleich zum Modell zu gering.]
6. Ist $\Delta q < 0$, [$\Rightarrow x^{(k)}$ ist unzulässig, sonst wäre $d_x = 0$ besser]
setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, sonst $\mathcal{F}_k := \mathcal{F}_{k-1}$

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x))$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2}d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
[Der Fortschritt in f ist im Vergleich zum Modell zu gering.]
6. Ist $\Delta q < 0$, [$\Rightarrow x^{(k)}$ ist unzulässig, sonst wäre $d_x = 0$ besser]
setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, sonst $\mathcal{F}_k := \mathcal{F}_{k-1}$
7. Setze $x^{(k+1)} := x^{(k)} + d_x$, $k \leftarrow k + 1$, wähle $\Delta \geq \underline{\Delta}$, GOTO 2.

Filter-SQP-Algorithmus

0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0, 1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von \mathcal{F}_{k-1} akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \rightarrow d_x$.
Falls (QP_k^Δ) unzul., setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, $k \leftarrow k + 1$, GOTO 1.
3. Ist $d_x = 0$, STOP, $x^{(k)}$ ist stationärer Punkt.
4. Wird $(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x))$ nicht vom erweiterten Filter $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$ akzeptiert, setze $\Delta \leftarrow \Delta/2$, GOTO 2.
5. Setze $\Delta q := -\frac{1}{2}d_x^T Q_k d_x - \nabla f_k^T d_x$ und $\Delta f := f_k - f(x^{(k)} + d_x)$.
Ist $\Delta q > 0$ und $\Delta f < \sigma \Delta q$, setze $\Delta \leftarrow \Delta/2$ und GOTO 2.
[Der Fortschritt in f ist im Vergleich zum Modell zu gering.]
6. Ist $\Delta q < 0$, [$\Rightarrow x^{(k)}$ ist unzulässig, sonst wäre $d_x = 0$ besser]
setze $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$, sonst $\mathcal{F}_k := \mathcal{F}_{k-1}$
7. Setze $x^{(k+1)} := x^{(k)} + d_x$, $k \leftarrow k + 1$, wähle $\Delta \geq \underline{\Delta}$, GOTO 2.

Unter geeigneten Voraussetzungen kann Konvergenz gegen einen stationären Punkt gezeigt werden.

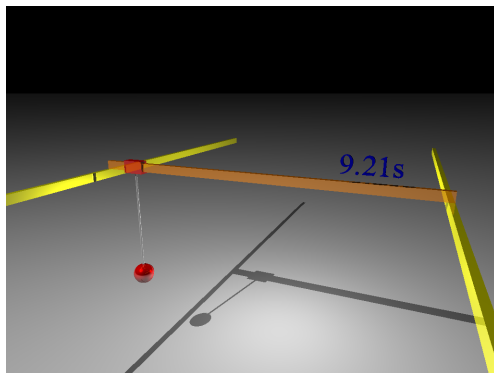
Inhaltsübersicht

Restringierte Optimierung: Verfahren

- 3.1 Strafverfahren
- 3.2 Augmentierte-Lagrange-Verfahren
- 3.3 Barriere-Verfahren
- 3.4 Quadratische Optimierung
 - Konvexe quadratische Optimierung
 - Gleichungsbeschränkte Quadratische Optimierung
 - Aktive-Mengen-Verfahren für Quadratische Optimierung
- 3.5 SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)
 - Lokale quadratische Konvergenz über Newton
 - Globalisierung mit Merit-Funktionen
 - Globalisierung mit Trust-Region-Ansätzen
 - Globalisierung mit einem Filter-Ansatz
- 3.6 Anwendungsbeispiel: Optimalsteuerung
 - Mathematisches Modell
 - Zeitoptimale Steuerung
 - Diskretisierung

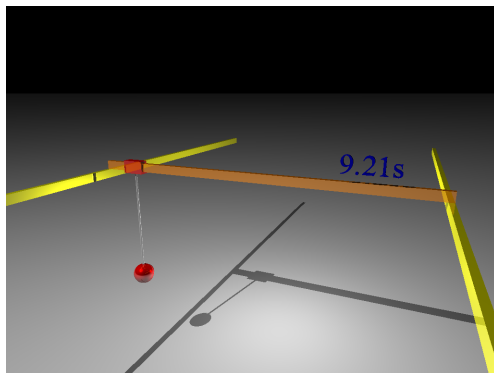
3.6 Anwendungsbeispiel: Optimalsteuerung

Wie muss die Laufkatze eines Industriekrans gesteuert werden, damit eine Last aus dem Ruhezustand in Punkt P möglichst schnell zu Punkt Q transportiert wird und dort wieder ruhig hängt?



3.6 Anwendungsbeispiel: Optimalsteuerung

Wie muss die Laufkatze eines Industriekrans gesteuert werden, damit eine Last aus dem Ruhezustand in Punkt P möglichst schnell zu Punkt Q transportiert wird und dort wieder ruhig hängt?

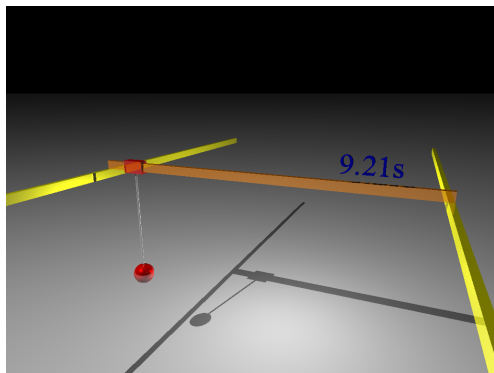


Die erste Idee – bis zur Hälfte voll beschleunigen, dann voll bremsen – ist nicht richtig.

Illustration

3.6 Anwendungsbeispiel: Optimalsteuerung

Wie muss die Laufkatze eines Industriekrans gesteuert werden, damit eine Last aus dem Ruhezustand in Punkt P möglichst schnell zu Punkt Q transportiert wird und dort wieder ruhig hängt?



Wie lässt sich das mathematisch modellieren?

(Material und Illustrationen von Prof. Dr. Roland Herzog)

3.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze

$d(t)$: horizontale Position der Last

M : die Masse der Laufkatze

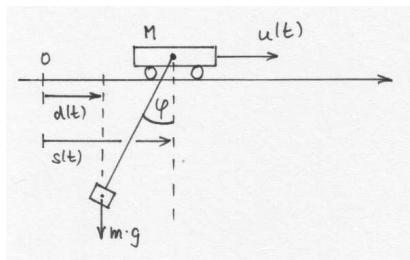
m : die Masse der Last

$u(t)$: **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$: Winkel der Last zur Vertikalen

g : die Erdbeschleunigung

L : die Länge des Seiles



3.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze

$d(t)$: horizontale Position der Last

M : die Masse der Laufkatze

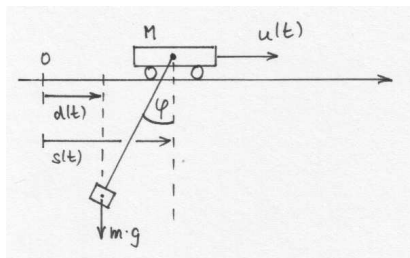
m : die Masse der Last

$u(t)$: **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$: Winkel der Last zur Vertikalen

g : die Erdbeschleunigung

L : die Länge des Seiles



Der **Zustand** $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \quad [t \text{ wird meist weggelassen}]$$

3.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze

$d(t)$: horizontale Position der Last

M : die Masse der Laufkatze

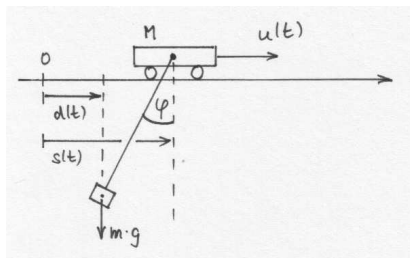
m : die Masse der Last

$u(t)$: **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$: Winkel der Last zur Vertikalen

g : die Erdbeschleunigung

L : die Länge des Seiles



Der **Zustand** $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \quad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

$$\ddot{s}(t) = -\frac{m}{M} \frac{g}{L} z + \frac{u(t)}{M}$$

$$\ddot{z}(t) = -\frac{(m+M)}{M} \frac{g}{L} z + \frac{u(t)}{M}$$

3.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze

$d(t)$: horizontale Position der Last

M : die Masse der Laufkatze

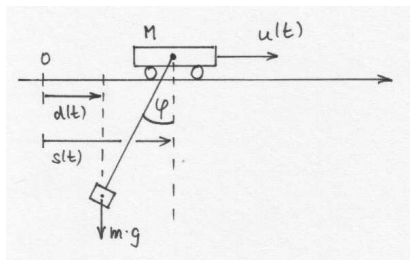
m : die Masse der Last

$u(t)$: **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$: Winkel der Last zur Vertikalen

g : die Erdbeschleunigung

L : die Länge des Seiles



Der **Zustand** $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \quad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

$$\ddot{s}(t) = -\frac{m}{M} \frac{g}{L} z + \frac{u(t)}{M}$$

$$\ddot{z}(t) = -\frac{(m+M)}{M} \frac{g}{L} z + \frac{u(t)}{M}$$

$$\Leftrightarrow \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

3.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: horizontale Position der Laufkatze

$d(t)$: horizontale Position der Last

M : die Masse der Laufkatze

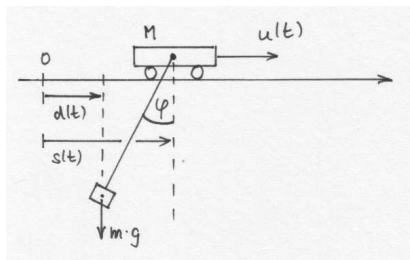
m : die Masse der Last

$u(t)$: **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$: Winkel der Last zur Vertikalen

g : die Erdbeschleunigung

L : die Länge des Seiles



Der **Zustand** $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \quad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

lineare gewöhnliche
Differentialgleichung

$$\dot{y} = Ay + Bu \leftrightarrow \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Für $t \in [0, T]$ ist eine Steuerfunktion $u(t)$ gesucht, die den Zustand $y(t)$ von einem Anfangszustand $y(0) = y_0$ in einen Endzustand $y(T) = y_T$ überführt, dabei Zulässigkeitsbedingungen $h(t, y(t), u(t)) \leq 0$ erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

Optimalstearaufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Für $t \in [0, T]$ ist eine Steuerfunktion $u(t)$ gesucht, die den Zustand $y(t)$ von einem Anfangszustand $y(0) = y_0$ in einen Endzustand $y(T) = y_T$ überführt, dabei Zulässigkeitsbedingungen $h(t, y(t), u(t)) \leq 0$ erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

$$\begin{array}{ll} \min_u & g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt & \text{Zielfunktion(al)} \\ \text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\ & y(0) = y_0 & \text{Anfangsbedingungen} \\ & y(T) = y_T & \text{Endbedingungen} \\ & h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.} \end{array}$$

Optimalsteueraufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands $y(t)$ in Abhängigkeit von Zeit t und Steuerung $u(t)$.

Für $t \in [0, T]$ ist eine Steuerfunktion $u(t)$ gesucht, die den Zustand $y(t)$ von einem Anfangszustand $y(0) = y_0$ in einen Endzustand $y(T) = y_T$ überführt, dabei Zulässigkeitsbedingungen $h(t, y(t), u(t)) \leq 0$ erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

$$\begin{array}{ll} \min_u & g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt & \text{Zielfunktion(al)} \\ \text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\ & y(0) = y_0 & \text{Anfangsbedingungen} \\ & y(T) = y_T & \text{Endbedingungen} \\ & h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.} \end{array}$$

Die **Endzeit** T kann dabei fest gegeben oder auch frei (d.h. Teil der zu optimierenden Größen) sein, ebenso könnten y_0 oder y_T teilweise frei sein.

3.6.2 Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

→ Zielfunktion:

3.6.2 Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

→ Zielfunktion: $\min \int_0^T 1 dt [= T]$

→ $y(0) =$

3.6.2 Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

→ Zielfunktion: $\min \int_0^T 1 dt [= T]$

→ $y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Position der Laufkatze
Relativposition der Last zur Laufkatze
Geschwindigkeit der Laufkatze
Relativgeschwindigkeit der Last zur Laufkatze

→ $y(T) =$

3.6.2 Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

→ Zielfunktion: $\min \int_0^T 1 dt [= T]$

→ $y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Position der Laufkatze
 Relativposition der Last zur Laufkatze
 Geschwindigkeit der Laufkatze
 Relativgeschwindigkeit der Last zur Laufkatze

→ $y(T) = (Q, 0, 0, 0)^T$

Steuerbeschränkungen: $-1 \leq u(t) \leq 1$ [keine unendliche Beschleunigung]

Es könnte auch **Zustandsbeschränkungen** für $y(t)$ (z.B. Hindernisse) geben.

3.6.2 Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit T vom Stillstand am Punkt P zum Stillstand an einem Punkt Q gebracht werden.

→ Zielfunktion: $\min \int_0^T 1 dt [= T]$

→ $y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Position der Laufkatze
 Relativposition der Last zur Laufkatze
 Geschwindigkeit der Laufkatze
 Relativgeschwindigkeit der Last zur Laufkatze

→ $y(T) = (Q, 0, 0, 0)^T$

Steuerbeschränkungen: $-1 \leq u(t) \leq 1$ [keine unendliche Beschleunigung]

Es könnte auch **Zustandsbeschränkungen** für $y(t)$ (z.B. Hindernisse) geben.

Hauptproblem noch:

T ist eine Variable und gibt zugleich den Definitionsbereich von u an!

Transformation von freier Endzeit auf feste Endzeit

$$\min_{u, T} \quad g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt$$

$$\text{s.t.} \quad \dot{y}(t) = f(t, y(t), u(t))$$

$$y(0) = y_0$$

$$y(T) = y_T$$

$$h(t, y(t), u(t)) \leq 0$$

Zielfunktion(al)

Differentialgleichung

Anfangsbedingungen

Endbedingungen

Beschränkungen.

Transformation von freier Endzeit auf feste Endzeit

$\min_{u, T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \leq 0$	Beschränkungen.

Führe eine neue Zeitvariable $\tau \in [0, 1]$ ein: $t = \tau T, \tau \in [0, 1]$

Definiere neue Zustände/Steuerung für τ : $\tilde{y}(\tau) := y(\tau T), \tilde{u}(\tau) := u(\tau T)$

Transformation von freier Endzeit auf feste Endzeit

$\min_{u, T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \leq 0$	Beschränkungen.

Führe eine neue Zeitvariable $\tau \in [0, 1]$ ein: $t = \tau T, \tau \in [0, 1]$

Definiere neue Zustände/Steuerung für τ : $\tilde{y}(\tau) := y(\tau T), \tilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\tilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)).$$

Transformation von freier Endzeit auf feste Endzeit

$\min_{u, T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt$	Zielfunktion(al)
s.t.	$\dot{y}(t) = f(t, y(t), u(t))$	Differentialgleichung
	$y(0) = y_0$	Anfangsbedingungen
	$y(T) = y_T$	Endbedingungen
	$h(t, y(t), u(t)) \leq 0$	Beschränkungen.

Führe eine neue Zeitvariable $\tau \in [0, 1]$ ein: $t = \tau T, \tau \in [0, 1]$

Definiere neue Zustände/Steuerung für τ : $\tilde{y}(\tau) := y(\tau T), \tilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\tilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = T f(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt \quad \rightarrow \quad g(\tilde{y}(1)) + T \int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) d\tau.$$

Transformation von freier Endzeit auf feste Endzeit

$$\begin{array}{ll}
 \min_{u, T} & g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt & \text{Zielfunktion(al)} \\
 \text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
 & y(0) = y_0 & \text{Anfangsbedingungen} \\
 & y(T) = y_T & \text{Endbedingungen} \\
 & h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.}
 \end{array}$$

Führe eine neue Zeitvariable $\tau \in [0, 1]$ ein: $t = \tau T, \tau \in [0, 1]$

Definiere neue Zustände/Steuerung für τ : $\tilde{y}(\tau) := y(\tau T), \tilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\tilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt \rightarrow g(\tilde{y}(1)) + T \int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) d\tau.$$

$$\begin{array}{ll}
 \min_{\tilde{u}, T} & g(\tilde{y}(1)) + T \int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) d\tau \\
 \text{s.t.} & \dot{\tilde{y}}(\tau) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) \\
 & \tilde{y}(0) = y_0 \\
 & \tilde{y}(1) = y_T \\
 & h(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) \leq 0.
 \end{array}$$

[Def-Bereich von \tilde{u} unabhängig von T , T ist nun „normale“ Variable!]

3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

$$\begin{aligned}
 & \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} && T \\
 & \text{s.t.} && \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad [\text{nichtlineare Nebenbed.}] \\
 & && y(0) = (P, 0, 0, 0)^T \\
 & && y(1) = (Q, 0, 0, 0)^T \\
 & && -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{aligned}$$

3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

$$\begin{array}{ll}
 \min & T \\
 u: [0, 1] \rightarrow \mathbb{R}, T \in \mathbb{R} & \\
 \text{s.t.} & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\
 & y(0) = (P, 0, 0, 0)^T \\
 & y(1) = (Q, 0, 0, 0)^T \\
 & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{array}$$

Variablen:

3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

$$\begin{array}{ll}
 \min & T \\
 u: [0, 1] \rightarrow \mathbb{R}, T \in \mathbb{R} & \\
 \text{s.t.} & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad [\text{nichtlineare Nebenbed.}] \\
 & y(0) = (P, 0, 0, 0)^T \\
 & y(1) = (Q, 0, 0, 0)^T \\
 & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{array}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ $\rightarrow \infty$ -dimensional

3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

$$\begin{aligned}
 & \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} && T \\
 & \text{s.t.} && \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\
 & && y(0) = (P, 0, 0, 0)^T \\
 & && y(1) = (Q, 0, 0, 0)^T \\
 & && -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{aligned}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ \rightarrow ∞ -dimensional

Wie optimiert man über Funktionen?

3.6.3 Das Laufkatzenproblem nach Transformation

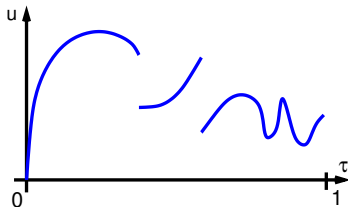
Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

$$\begin{aligned}
 & \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} && T \\
 & \text{s.t.} && \dot{y}(\tau) = T Ay(\tau) + T Bu(\tau) && \text{[nichtlineare Nebenbed.!]} \\
 & && y(0) = (P, 0, 0, 0)^T \\
 & && y(1) = (Q, 0, 0, 0)^T \\
 & && -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{aligned}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.



3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

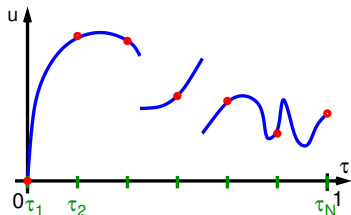
$$\begin{aligned} \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1]. \end{aligned}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.

\rightarrow **Diskretisierung**



3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

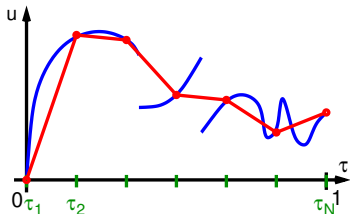
$$\begin{aligned} \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = T Ay(\tau) + T Bu(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1]. \end{aligned}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.

\rightarrow **Diskretisierung** und, z.B., **lineare Interpolation** \rightarrow **Diskretisierungsfehler**



3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

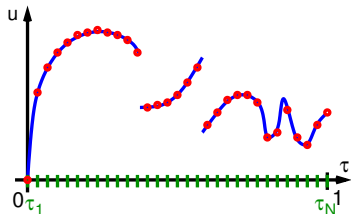
$$\begin{aligned} \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1]. \end{aligned}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.

\rightarrow Diskretisierung und, z.B., lineare Interpolation \rightarrow Diskretisierungsfehler



3.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne \sim aber mit neuer Zeit $\tau \in [0, 1]$)

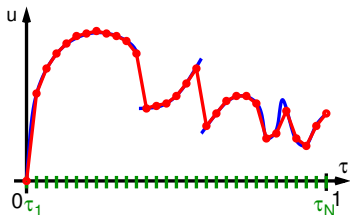
$$\begin{aligned} \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1]. \end{aligned}$$

Variablen: $T, u(\tau)$ für $\tau \in [0, 1]$ $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.

\rightarrow Diskretisierung und, z.B., lineare Interpolation \rightarrow Diskretisierungsfehler



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

$$\text{s.t. } \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1]$$

$$y(0) = (P, 0, 0, 0)^T$$

$$y(1) = (Q, 0, 0, 0)^T$$

$$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+.$$

Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1] \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+. \end{aligned}$$

Was geschieht mit der Differentialgleichung?

Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$\min T$$

$$\text{s.t. } \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1]$$

$$y(0) = (P, 0, 0, 0)^T$$

$$y(1) = (Q, 0, 0, 0)^T$$

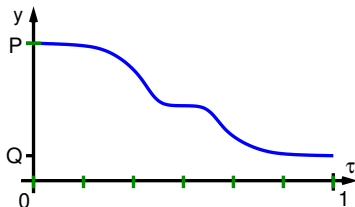
$$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+.$$

Was geschieht mit der Differentialgleichung?

Berechne

$$y(\tau) = y(0) + \int_0^\tau T A y(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

$$\text{s.t. } \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1]$$

$$y(0) = (P, 0, 0, 0)^T$$

$$y(1) = (Q, 0, 0, 0)^T$$

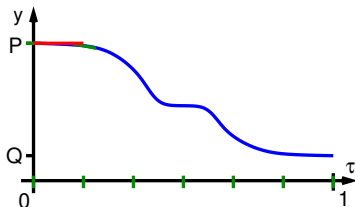
$$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+.$$

Was geschieht mit der Differentialgleichung?

Berechne

$$y(\tau) = y(0) + \int_0^\tau T A y(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

s.t. $\dot{y}(\tau) = T Ay(\tau) + T B u(\tau) \quad \tau \in [0, 1]$

$y(0) = (P, 0, 0, 0)^T$

$y(1) = (Q, 0, 0, 0)^T$

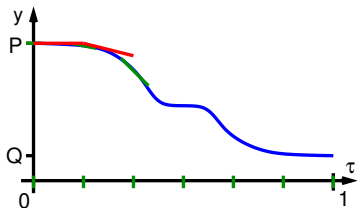
$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+$.

Was geschieht mit der Differentialgleichung?

Berechne

$$y(\tau) = y(0) + \int_0^\tau T Ay(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

s.t. $\dot{y}(\tau) = T Ay(\tau) + T B u(\tau) \quad \tau \in [0, 1]$

$y(0) = (P, 0, 0, 0)^T$

$y(1) = (Q, 0, 0, 0)^T$

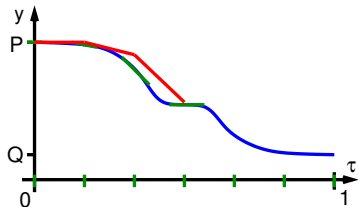
$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+$.

Was geschieht mit der Differentialgleichung?

Berechne

$$y(\tau) = y(0) + \int_0^\tau T Ay(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

s.t. $\dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1]$

$y(0) = (P, 0, 0, 0)^T$

$y(1) = (Q, 0, 0, 0)^T$

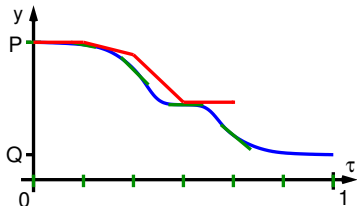
$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+$.

Was geschieht mit der Differentialgleichung?

Berechne

$$y(\tau) = y(0) + \int_0^\tau T A y(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

s.t. $\dot{y}(\tau) = T Ay(\tau) + T B u(\tau) \quad \tau \in [0, 1]$

$y(0) = (P, 0, 0, 0)^T$

$y(1) = (Q, 0, 0, 0)^T$

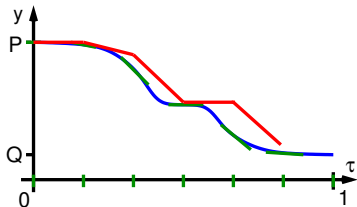
$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+$.

Was geschieht mit der Differentialgleichung?

Berechne

$$y(\tau) = y(0) + \int_0^\tau T Ay(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0, 1] \rightarrow \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

min T

s.t. $\dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1]$

$y(0) = (P, 0, 0, 0)^T$

$y(1) = (Q, 0, 0, 0)^T$

$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+$.

Was geschieht mit der Differentialgleichung?

Berechne

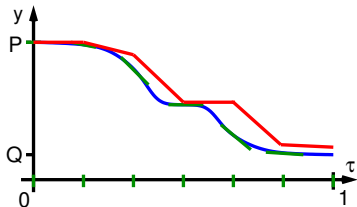
$$y(\tau) = y(0) + \int_0^\tau T A y(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).

→ Funktionsorakel $y(\tau) = R(u, \tau)$, das $y(0) = (P, 0, 0, 0)^T$ erfüllt.

→ Es bleibt nur eine Gls-Nebenbedingung: $R(u, 1) - (Q, 0, 0, 0)^T = 0$

Funktionsauswertung mit Gradienten über automatisches Differenzieren!



Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u, 1) - (Q, 0, 0, 0)^T = 0 \\ & -1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+. \end{aligned}$$

Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

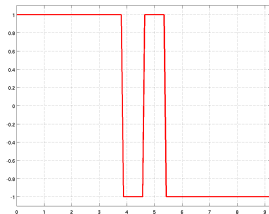
Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u, 1) - (Q, 0, 0, 0)^T = 0 \\ & -1 \leq u_i \leq 1 \quad (i = 1, \dots, N), \quad T \in \mathbb{R}_+. \end{aligned}$$

Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

→ Approximation der zeitoptimalen Steuerung
 $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$

Illustration



Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u, 1) - (Q, 0, 0, 0)^T = 0 \\ & -1 \leq u_i \leq 1 \quad (i = 1, \dots, N), \quad T \in \mathbb{R}_+. \end{aligned}$$

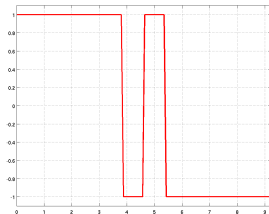
Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

→ Approximation der zeitoptimalen Steuerung
 $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$

Illustration

Ganz analog sind zeitoptimale 2D-Steuerungen berechenbar,

2D-Illustration



Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u, 1) - (Q, 0, 0, 0)^T = 0 \\ & -1 \leq u_i \leq 1 \quad (i = 1, \dots, N), \quad T \in \mathbb{R}_+. \end{aligned}$$

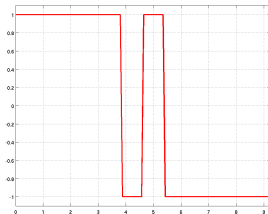
Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren $R(\cdot)$ über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

→ Approximation der zeitoptimalen Steuerung
 $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$

Illustration

Ganz analog sind zeitoptimale 2D-Steuerungen berechenbar,

2D-Illustration



Die Qualität der Lösung und der notwendige Aufwand zur Bestimmung der Lösung sind stark von der Wahl der Diskretisierung abhängig!

Inhaltsübersicht

Freie Nichtlineare Optimierung

Restringierte Nichtlineare Optimierung: Grundlagen

Restringierte Optimierung: Verfahren

Ableitungsfreie Optimierung/Direkte Suchverfahren

4 Ableitungsfreie Optimierung/Direkte Suchverfahren

umfasst Verfahren zu $\min_{x \in \mathbb{R}^n} f(x)$, die nur ein Orakel 0. Ordnung (nur Funktionswerte) benötigen. Sie werden in der Praxis oft eingesetzt, weil

- Funktionen durch Simulationsprogramme ausgewertet werden, die wirklich kein automatisches Differenzieren erlauben,
- Benutzer nicht wissen, was eine Ableitung ist, wozu sie gut ist, und wie man sie bekommt,
- schon bei der Modellbildung auf Optimierungswünsche und Ableitungsinformation geachtet werden müsste (für Optimierung geeignete Modelle zu bauen verlangt oft viel Erfahrung!).

4 Ableitungsfreie Optimierung/Direkte Suchverfahren

umfasst Verfahren zu $\min_{x \in \mathbb{R}^n} f(x)$, die nur ein Orakel 0. Ordnung (nur Funktionswerte) benötigen. Sie werden in der Praxis oft eingesetzt, weil

- Funktionen durch Simulationsprogramme ausgewertet werden, die wirklich kein automatisches Differenzieren erlauben,
 - Benutzer nicht wissen, was eine Ableitung ist, wozu sie gut ist, und wie man sie bekommt,
 - schon bei der Modellbildung auf Optimierungswünsche und Ableitungsinformation geachtet werden müsste (für Optimierung geeignete Modelle zu bauen verlangt oft viel Erfahrung!).
-

Bei Funktionen, die nicht stückweise glatt sind und erratisch springen, hilft höchstens zufälliges Suchen, sie kommen aber in der Praxis nicht vor.

4 Ableitungsfreie Optimierung/Direkte Suchverfahren

umfasst Verfahren zu $\min_{x \in \mathbb{R}^n} f(x)$, die nur ein Orakel 0. Ordnung (nur Funktionswerte) benötigen. Sie werden in der Praxis oft eingesetzt, weil

- Funktionen durch Simulationsprogramme ausgewertet werden, die wirklich kein automatisches Differenzieren erlauben,
 - Benutzer nicht wissen, was eine Ableitung ist, wozu sie gut ist, und wie man sie bekommt,
 - schon bei der Modellbildung auf Optimierungswünsche und Ableitungsinformation geachtet werden müsste (für Optimierung geeignete Modelle zu bauen verlangt oft viel Erfahrung!).
-

Bei Funktionen, die nicht stückweise glatt sind und erratisch springen, hilft höchstens zufälliges Suchen, sie kommen aber in der Praxis nicht vor.

Meist sind Funktionen stückweise glatt mit Knick- und/oder Sprungstellen.

Für stetige Funktionen mit Knickstellen sollte man Verfahren der **nicht-glatten Optimierung (nonsmooth optimization)** verwenden, die sogenannte Subdifferentialinformation aus Subgradienten nutzen.

Sprungstellen sind nur durch Gebietsunterteilung zu kontrollieren. Auf den stetigen Teilen hilft wieder glatte oder nichtglatte Optimierung.

All das benötigt ein gewisses Problemverständnis. Ohne dieses probiert man zuerst Direkte Suchverfahren.

4 Ableitungsfreie Optimierung/Direkte Suchverfahren

umfasst Verfahren zu $\min_{x \in \mathbb{R}^n} f(x)$, die nur ein Orakel 0. Ordnung (nur Funktionswerte) benötigen. Sie werden in der Praxis oft eingesetzt, weil

- Funktionen durch Simulationsprogramme ausgewertet werden, die wirklich kein automatisches Differenzieren erlauben,
 - Benutzer nicht wissen, was eine Ableitung ist, wozu sie gut ist, und wie man sie bekommt,
 - schon bei der Modellbildung auf Optimierungswünsche und Ableitungsinformation geachtet werden müsste (für Optimierung geeignete Modelle zu bauen verlangt oft viel Erfahrung!).
-

Bei Funktionen, die nicht stückweise glatt sind und erratisch springen, hilft höchstens zufälliges Suchen, sie kommen aber in der Praxis nicht vor.

Alle Suchverfahren nehmen an, dass bekannte Funktionswerte Aufschluss über die lokale Funktionsentwicklung geben, haben explizit oder implizit das Ziel, ein lokales Modell der Funktion zu erstellen (Gradienten/Subdifferenziale tun genau das). Verbreitete/wichtige Verfahren sind

- Das „Simplex“-Verfahren (Nelder-Mead, neuer: Torczon) [verbreitet]
- Das Kriging-Verfahren [für wenige Auswertungen]
- NEWUOA von Powell [wichtig]

Inhaltsübersicht

Ableitungsfreie Optimierung/Direkte Suchverfahren

- 4.1 Das Simplex-Verfahren von Nelder und Mead
- 4.2 Das Kriging-Verfahren
- 4.3 NEWUOA von Powell

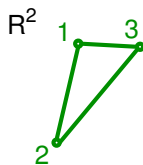
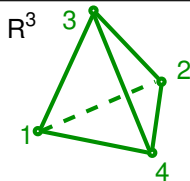
4.1 Das Simplex-Verfahren von Nelder und Mead

... hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

4.1 Das Simplex-Verfahren von Nelder und Mead

... hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \rightarrow \mathbb{R}$ zu minimieren, wähle $n + 1$ affin unabhängige Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n -dimensionalen Simplex.



4.1 Das Simplex-Verfahren von Nelder und Mead

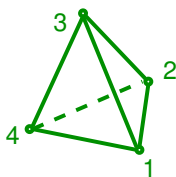
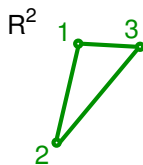
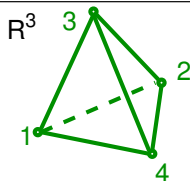
... hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \rightarrow \mathbb{R}$ zu minimieren, wähle $n + 1$ affin unabhängige Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n -dimensionalen Simplex.

Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \dots \leq f(x^{(n+1)}).$$

Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \dots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.



4.1 Das Simplex-Verfahren von Nelder und Mead

... hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

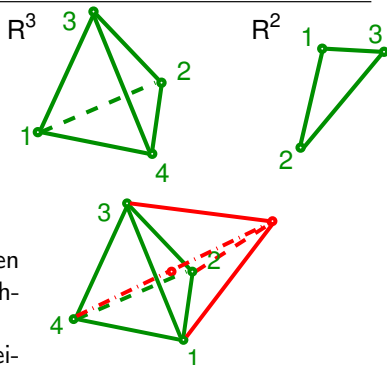
Idee: Um $f : \mathbb{R}^n \rightarrow \mathbb{R}$ zu minimieren, wähle $n + 1$ affin unabhängige Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n -dimensionalen Simplex.

Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \dots \leq f(x^{(n+1)}).$$

Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \dots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.

→ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).



4.1 Das Simplex-Verfahren von Nelder und Mead

... hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

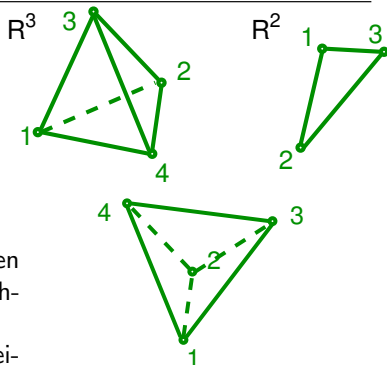
Idee: Um $f : \mathbb{R}^n \rightarrow \mathbb{R}$ zu minimieren, wähle $n + 1$ affin unabhängige Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n -dimensionalen Simplex.

Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \dots \leq f(x^{(n+1)}).$$

Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \dots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.

→ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).



4.1 Das Simplex-Verfahren von Nelder und Mead

... hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \rightarrow \mathbb{R}$ zu minimieren, wähle $n + 1$ affin unabhängige Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n -dimensionalen Simplex.

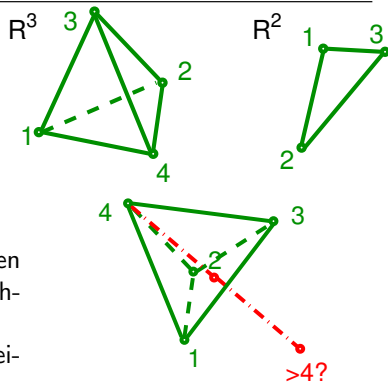
Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \dots \leq f(x^{(n+1)}).$$

Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \dots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.

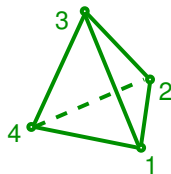
→ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).

Ist der Punkt immer noch am schlechtesten, schrumpfe den Simplex in dieser Richtung, ist er sehr gut, strecke die Richtung.



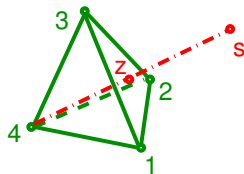
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$
[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]



Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$
[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).



Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$

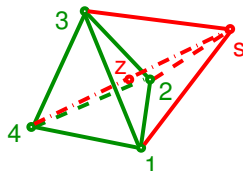
[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]

1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).

2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$:

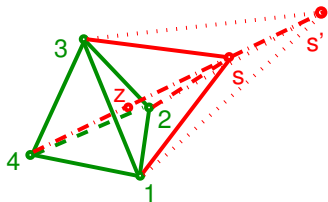
(Spiegelung, reflection)

Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.



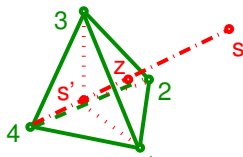
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$
[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.



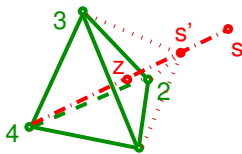
Algorithmisches Schema des Nelder-Mead-Verfahrens:

- Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$
[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
- Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
- Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
- Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
- Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)**
Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$
Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,



Algorithmisches Schema des Nelder-Mead-Verfahrens:

- Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$
[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]
- Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).
- Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
- Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.
- Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)**
Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$
Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,



Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$

[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]

1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).

2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
 Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.

3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
 Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
 Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.

4. Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)**

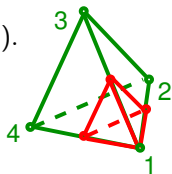
Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$

Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,

sonst schrumpfe den Simplex hin zu $x^{(1)}$:

$x^{(i)} \leftarrow \frac{1}{2}(x^{(1)} + x^{(i)}), (i = 2, \dots, n + 1).$

Sortiere neu, GOTO 1.



Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \dots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \dots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$

[z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]

1. Setze $z \leftarrow \frac{1}{n} \sum_{i=1}^n x_i$ (Baryzentrum der „guten“ Punkte),
 $s \leftarrow z + \alpha(z - x^{(n+1)})$ (gespiegelter Punkt).

2. Falls $f(x^{(1)}) \leq f(s) \leq f(z)$: **(Spiegelung, reflection)**
 Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.

3. Falls $f(s) < f(x^{(1)})$: **(Streckung, expansion)**
 Teste $s' \leftarrow z + \beta(s - z)$. Gilt $f(s') < f(s)$, setze $s \leftarrow s'$.
 Setze $x^{(n+1)} \leftarrow s$, sortiere neu, GOTO 1.

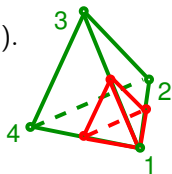
4. Falls $f(x^{(n)}) < f(s)$: **(Schrumpfung, contraction)**

Teste $s' \leftarrow \begin{cases} z + \gamma(x^{(n+1)} - z) & \text{falls } f(s) \geq f(x^{(n+1)}), \\ z + \gamma(s - z) & \text{falls } f(s) < f(x^{(n+1)}). \end{cases}$

Gilt $f(s') < \min\{f(x^{(n+1)}), f(s)\}$, setze $x^{(n+1)} \leftarrow s'$,
 sonst schrumpfe den Simplex hin zu $x^{(1)}$:

$x^{(i)} \leftarrow \frac{1}{2}(x^{(1)} + x^{(i)}), (i = 2, \dots, n + 1).$

Sortiere neu, GOTO 1.



Abbruchkriterien?

Vorschläge zu Abbruchkriterien:

1. Nelder und Mead: Setze $\bar{f} := \frac{1}{n+1} \sum_{i=1}^{n+1} f(x^{(i)})$

Falls $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - \bar{f}]^2 \leq \varepsilon$ STOP (f auf Simplex fast konstant?)

Vorschläge zu Abbruchkriterien:

1. Nelder und Mead: Setze $\bar{f} := \frac{1}{n+1} \sum_{i=1}^{n+1} f(x^{(i)})$
Falls $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - \bar{f}]^2 \leq \varepsilon$ STOP (f auf Simplex fast konstant?)
2. Powell: Starte alle k Iteration erneut mit regelmäßigem Simplex um $x^{(1)}$ und Seitenlänge $\|x^{(1)} - x^{(n+1)}\|$ (sonst zu lang und dünn).
Stoppe, falls keine Verbesserung in $2n$ Iterationen.

Vorschläge zu Abbruchkriterien:

1. Nelder und Mead: Setze $\bar{f} := \frac{1}{n+1} \sum_{i=1}^{n+1} f(x^{(i)})$
Falls $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - \bar{f}]^2 \leq \varepsilon$ STOP (f auf Simplex fast konstant?)
 2. Powell: Starte alle k Iteration erneut mit regelmäßigem Simplex um $x^{(1)}$ und Seitenlänge $\|x^{(1)} - x^{(n+1)}\|$ (sonst zu lang und dünn).
Stoppe, falls keine Verbesserung in $2n$ Iterationen.
-

Bemerkungen:

- sehr einfach zu implementieren
- benötigt sehr viele Auswertungen, schon $n + 1$ um zu starten $\rightarrow n$ klein

Vorschläge zu Abbruchkriterien:

1. Nelder und Mead: Setze $\bar{f} := \frac{1}{n+1} \sum_{i=1}^{n+1} f(x^{(i)})$
Falls $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - \bar{f}]^2 \leq \varepsilon$ STOP (f auf Simplex fast konstant?)
 2. Powell: Starte alle k Iteration erneut mit regelmäßigem Simplex um $x^{(1)}$ und Seitenlänge $\|x^{(1)} - x^{(n+1)}\|$ (sonst zu lang und dünn).
Stoppe, falls keine Verbesserung in $2n$ Iterationen.
-

Bemerkungen:

- sehr einfach zu implementieren
- benötigt sehr viele Auswertungen, schon $n + 1$ um zu starten $\rightarrow n$ klein
- Konvergenz ist nicht garantiert, es gibt streng konvexe Gegenbeispiele!
- sehr skalierungsabhängig!

Vorschläge zu Abbruchkriterien:

1. Nelder und Mead: Setze $\bar{f} := \frac{1}{n+1} \sum_{i=1}^{n+1} f(x^{(i)})$
Falls $\frac{1}{n+1} \sum_{i=1}^{n+1} [f(x^{(i)}) - \bar{f}]^2 \leq \varepsilon$ STOP (f auf Simplex fast konstant?)
 2. Powell: Starte alle k Iteration erneut mit regelmäßigem Simplex um $x^{(1)}$ und Seitenlänge $\|x^{(1)} - x^{(n+1)}\|$ (sonst zu lang und dünn).
Stoppe, falls keine Verbesserung in $2n$ Iterationen.
-

Bemerkungen:

- sehr einfach zu implementieren
- benötigt sehr viele Auswertungen, schon $n + 1$ um zu starten $\rightarrow n$ klein
- Konvergenz ist nicht garantiert, es gibt streng konvexe Gegenbeispiele!
- sehr skalierungsabhängig!
- Erweiterung auf Nebenbedingungen: Setze $f \gg$ für unzulässige Punkte.
- Torczon: Konvergente Variante für konvexes f .

Inhaltsübersicht

Ableitungsfreie Optimierung/Direkte Suchverfahren

- 4.1 Das Simplex-Verfahren von Nelder und Mead
- 4.2 Das Kriging-Verfahren
- 4.3 NEWUOA von Powell

4.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen → wird gerne verwendet, wenn jede Funktionsauswertung einem realen Experiment oder einer aufwendigen Simulation entspricht.

4.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen → wird gerne verwendet, wenn jede Funktionsauswertung einem realen Experiment oder einer aufwendigen Simulation entspricht.

Idee: Beachte folgende Leitlinien bei Entwurf von Modell und Verfahren:

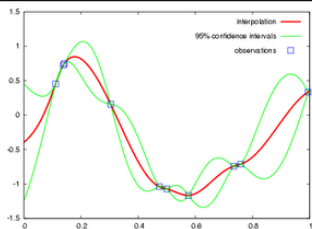
- Das Modell sollte die Funktion in bekannten Punkten interpolieren.
- Oszillationen sollten im Modell vermieden werden.
- In der Nähe bekannter Punkte sind Modellpunkte vertrauenswürdiger.
- Optimierungsaufwand ist vernachlässigbar gegenüber Auswertungen.

4.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen → wird gerne verwendet, wenn jede Funktionsauswertung einem realen Experiment oder einer aufwendigen Simulation entspricht.

Idee: Beachte folgende Leitlinien bei Entwurf von Modell und Verfahren:

- Das Modell sollte die Funktion in bekannten Punkten interpolieren.
- Oszillationen sollten im Modell vermieden werden.
- In der Nähe bekannter Punkte sind Modellpunkte vertrauenswürdiger.
- Optimierungsaufwand ist vernachlässigbar gegenüber Auswertungen.



(22.1.2010, <http://en.wikipedia.org/wiki/Kriging>)

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \dots, f_k \in \mathbb{R}$.

Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^k f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \dots, f_k \in \mathbb{R}$.

Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^k f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

→ Für distanzabhängige Gewichtsfunktionen $b_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $b_i(x) := e^{-\|x-x^{(i)}\|^2}$

bestimme Koeffizienten $\alpha_i \in \mathbb{R}$ so, dass das Modell

$$\hat{f}(x) := \mu + \sum_{i=1}^k \alpha_i b_i(x)$$

die Funktionswerte in den $x^{(i)}$ annimmt, $\hat{f}(x^{(i)}) = f_i$ ($i = 1, \dots, k$).

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \dots, f_k \in \mathbb{R}$.

Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^k f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

→ Für distanzabhängige Gewichtsfunktionen $b_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $b_i(x) := e^{-\|x-x^{(i)}\|^2}$

bestimme Koeffizienten $\alpha_i \in \mathbb{R}$ so, dass das Modell

$$\hat{f}(x) := \mu + \sum_{i=1}^k \alpha_i b_i(x)$$

die Funktionswerte in den $x^{(i)}$ annimmt, $\hat{f}(x^{(i)}) = f_i$ ($i = 1, \dots, k$).

$$\rightarrow \text{Löse } \begin{bmatrix} b_1(x^{(1)}) & \dots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \dots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \dots, f_k \in \mathbb{R}$.
Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^k f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

→ Für distanzabhängige Gewichtsfunktionen $b_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $b_i(x) := e^{-\|x - x^{(i)}\|^2}$
bestimme Koeffizienten $\alpha_i \in \mathbb{R}$ so, dass das Modell

$$\hat{f}(x) := \mu + \sum_{i=1}^k \alpha_i b_i(x)$$

die Funktionswerte in den $x^{(i)}$ annimmt, $\hat{f}(x^{(i)}) = f_i$ ($i = 1, \dots, k$).

$$\rightarrow \text{Löse } \begin{bmatrix} b_1(x^{(1)}) & \dots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \dots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.

→ **Skalierung:** Bestimme eine Norm $\|d\|_A^2 := d^T A d$ mit $A \succ 0$ wie folgt:

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \dots, f_k \in \mathbb{R}$.
Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^k f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

→ Für distanzabhängige Gewichtsfunktionen $b_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $b_i(x) := e^{-\|x-x^{(i)}\|^2}$
bestimme Koeffizienten $\alpha_i \in \mathbb{R}$ so, dass das Modell

$$\hat{f}(x) := \mu + \sum_{i=1}^k \alpha_i b_i(x)$$

die Funktionswerte in den $x^{(i)}$ annimmt, $\hat{f}(x^{(i)}) = f_i$ ($i = 1, \dots, k$).

$$\rightarrow \text{Löse } \begin{bmatrix} b_1(x^{(1)}) & \dots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \dots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.

→ **Skalierung:** Bestimme eine Norm $\|d\|_A^2 := d^T A d$ mit $A \succ 0$ wie folgt:
 $\|\cdot\|_A$ ist gut, wenn für jedes $j \in \{1, \dots, k\}$ das Modell der anderen Punkte

$$\hat{f}_A^j(x) := \mu + \sum_{i=1}^{j-1} \alpha_i^{A,j} b_i^A(x) + \sum_{i=j+1}^k \alpha_i^{A,j} b_i^A(x)$$

in $x^{(j)}$ möglichst kleinen Fehler $e_A^j := \hat{f}_A^j(x^{(j)}) - f_j$ hat.

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \dots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \dots, f_k \in \mathbb{R}$.
Für von allen $x^{(i)}$ weit entfernte Punkte wird angenommen:

Der Durchschnittswert $\mu = \frac{1}{k} \sum_{i=1}^k f_i$ ist guter Schätzer.

Für Punkte nahe an $x^{(i)}$ sollte der Einfluss von f_i groß sein.

→ Für distanzabhängige Gewichtsfunktionen $b_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $b_i(x) := e^{-\|x-x^{(i)}\|^2}$
bestimme Koeffizienten $\alpha_i \in \mathbb{R}$ so, dass das Modell

$$\hat{f}(x) := \mu + \sum_{i=1}^k \alpha_i b_i(x)$$

die Funktionswerte in den $x^{(i)}$ annimmt, $\hat{f}(x^{(i)}) = f_i$ ($i = 1, \dots, k$).

$$\rightarrow \text{Löse} \quad \begin{bmatrix} b_1(x^{(1)}) & \dots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \dots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.

→ **Skalierung:** Bestimme eine Norm $\|d\|_A^2 := d^T A d$ mit $A \succ 0$ wie folgt:
 $\|\cdot\|_A$ ist gut, wenn für jedes $j \in \{1, \dots, k\}$ das Modell der anderen Punkte

$$\hat{f}_A^j(x) := \mu + \sum_{i=1}^{j-1} \alpha_i^{A,j} b_i^A(x) + \sum_{i=j+1}^k \alpha_i^{A,j} b_i^A(x)$$

in $x^{(j)}$ möglichst kleinen Fehler $e_A^j := \hat{f}_A^j(x^{(j)}) - f_j$ hat.

→ Löse $\min_{A \succ 0} \max_{1 \leq j \leq k} e_A^j$ (ein nichtglattes Problem) $\rightarrow A_*, b_i^*, \hat{f}_*$

Man hofft nun, dass die Fehler $e_{A_*}^j$ auch gute **Fehlerschätzer** für die Abweichung des Modells \hat{f}_* von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}_{A_*}^j$ über die Distanz $d_j^* := \min_{1 \leq i \leq k, i \neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$ einen Fehler von $e_{A_*}^j$.

Man hofft nun, dass die Fehler $e_{A_*}^j$ auch gute **Fehlerschätzer** für die Abweichung des Modells \hat{f}_* von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}_{A_*}^j$ über die Distanz $d_j^* := \min_{1 \leq i \leq k, i \neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$

einen Fehler von $e_{A_*}^j$. Pro Distanzeinheit von $\|\cdot\|_{A_*}$ weicht damit \hat{f}_* von f hoffentlich nur um

$$\bar{s} := \max_{1 \leq j \leq k} \frac{e_{A_*}^j}{d_j^*},$$

ab, vielleicht noch multipliziert mit einer Konstanten $\nu > 1$, also

$$\text{hoffentlich } f(x) \geq \hat{f}_*(x) - \nu \bar{s} \min_{0 \leq i \leq k} \|x - x^{(i)}\|_{A_*} =: g(x) \text{ für } x \in \Omega,$$

wobei die Grundmenge $\Omega \subset \mathbb{R}^n$ als kompakt angenommen wird.

Man hofft nun, dass die Fehler $e_{A_*}^j$ auch gute **Fehlerschätzer** für die Abweichung des Modells \hat{f}_* von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}_{A_*}^j$ über die Distanz $d_j^* := \min_{1 \leq i \leq k, i \neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$

einen Fehler von $e_{A_*}^j$. Pro Distanzeinheit von $\|\cdot\|_{A_*}$ weicht damit \hat{f}_* von f hoffentlich nur um

$$\bar{s} := \max_{1 \leq j \leq k} \frac{e_{A_*}^j}{d_j^*},$$

ab, vielleicht noch multipliziert mit einer Konstanten $\nu > 1$, also

$$\text{hoffentlich } f(x) \geq \hat{f}_*(x) - \nu \bar{s} \min_{0 \leq i \leq k} \|x - x^{(i)}\|_{A_*} =: g(x) \text{ für } x \in \Omega,$$

wobei die Grundmenge $\Omega \subset \mathbb{R}^n$ als kompakt angenommen wird.

Das Optimierungsproblem des Kriging-Verfahrens:

Finde ein globales Minimum zu $\min_{x \in \Omega} g(x)$.

Eine Optimallösung dieses Problems wird als nächster Auswertungspunkt gewählt. Dann wird für die nun $k + 1$ Punkte das Modell erneut berechnet, etc.

Inhaltsübersicht

Ableitungsfreie Optimierung/Direkte Suchverfahren

- 4.1 Das Simplex-Verfahren von Nelder und Mead
- 4.2 Das Kriging-Verfahren
- 4.3 NEWUOA von Powell

4.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n + 1$ Funktionswerte von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

4.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n + 1$ Funktionswerte von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

Gegeben: Startpunkt $x^{(0)}$ und Intervall $[\underline{\Delta}, \bar{\Delta}]$ für den Trustregion-Radius, wobei $\underline{\Delta}$ auch die Abbruchgenauigkeit für die Lösungskordinaten sei.

4.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n + 1$ Funktionswerte von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

Gegeben: Startpunkt $x^{(0)}$ und Intervall $[\underline{\Delta}, \bar{\Delta}]$ für den Trustregion-Radius, wobei $\underline{\Delta}$ auch die Abbruchgenauigkeit für die Lösungskordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \frac{1}{2}x^T Q^{(k)}x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, ($i = 0, \dots, 2n$) die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert.

4.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n + 1$ Funktionswerte von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

Gegeben: Startpunkt $x^{(0)}$ und Intervall $[\underline{\Delta}, \bar{\Delta}]$ für den Trustregion-Radius, wobei $\underline{\Delta}$ auch die Abbruchgenauigkeit für die Lösungskordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \frac{1}{2}x^T Q^{(k)}x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, ($i = 0, \dots, 2n$) die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \quad \begin{array}{ll} \min & \sum_{1 \leq i, j \leq n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2 \\ \text{s.t.} & \frac{1}{2}(x^{(i)})^T Q^{(k)}x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \dots, 2n, \\ & Q^{(k)} \in \mathcal{S}^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}. \end{array}$$

4.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n + 1$ Funktionswerte von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

Gegeben: Startpunkt $x^{(0)}$ und Intervall $[\underline{\Delta}, \bar{\Delta}]$ für den Trustregion-Radius, wobei $\underline{\Delta}$ auch die Abbruchgenauigkeit für die Lösungskordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \frac{1}{2}x^T Q^{(k)}x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, ($i = 0, \dots, 2n$) die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \quad \begin{array}{ll} \min & \sum_{1 \leq i, j \leq n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2 \\ \text{s.t.} & \frac{1}{2}(x^{(i)})^T Q^{(k)}x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \dots, 2n, \\ & Q^{(k)} \in \mathcal{S}^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}. \end{array}$$

Pro Iteration ändert sich nur ein $x^{(i)}$, daher lässt sich (QP_m^k) mit vielen Tricks in Konstante mal $(2n + 1)^2$ Fließkomma-Operationen lösen.

4.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n + 1$ Funktionswerte von $f : \mathbb{R}^n \rightarrow \mathbb{R}$ durch ein quadratisches Modell, das sich möglichst wenig vom Vorgänger-Modell unterscheidet (wie bei BFGS), und bestimmt den nächsten Punkt als Näherungslösung eines Trustregion-Problems zu diesem Modell.

Gegeben: Startpunkt $x^{(0)}$ und Intervall $[\underline{\Delta}, \bar{\Delta}]$ für den Trustregion-Radius, wobei $\underline{\Delta}$ auch die Abbruchgenauigkeit für die Lösungskordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \frac{1}{2}x^T Q^{(k)}x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, ($i = 0, \dots, 2n$) die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \quad \min \quad \sum_{1 \leq i, j \leq n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2$$

$$\text{s.t.} \quad \frac{1}{2}(x^{(i)})^T Q^{(k)}x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \dots, 2n,$$

$$Q^{(k)} \in \mathcal{S}^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}.$$

Pro Iteration ändert sich nur ein $x^{(i)}$, daher lässt sich (QP_m^k) mit vielen Tricks in Konstante mal $(2n + 1)^2$ Fließkomma-Operationen lösen.

Die Initialisierung von $m^{(0)}$ verwendet die Punkte $x^{(0)}, x^{(2i-1)} := x^{(0)} - \bar{\Delta}e_i, x^{(2i)} := x^{(0)} + \bar{\Delta}e_i$ ($i = 1, \dots, n$) $\rightarrow c^{(0)} = f_0, q^{(0)}$, Diagonalmatrix $Q^{(0)}$.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem
 $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{i})}\|$ maximal und GOTO 4.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem
 $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.
 Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{i})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{i}} - f(x^{(\hat{i})} + d)}{f_{\hat{i}} - m^{(k)}(x^{(\hat{i})} + d)}$, passe Δ ($\geq \delta$) entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{i})} + d$ für geeignet gewähltes j .

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.
 Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{i})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{i}} - f(x^{(\hat{i})} + d)}{f_{\hat{i}} - m^{(k)}(x^{(\hat{i})} + d)}$, passe Δ ($\geq \delta$) entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{i})} + d$ für geeignet gewähltes j .
4. Bestimme $m^{(k+1)}$ aus (QP_m^{k+1}) mit neuem $x^{(j)}$, $k \leftarrow k + 1$, GOTO 1.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.
 Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{i})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{i}} - f(x^{(\hat{i})} + d)}{f_{\hat{i}} - m^{(k)}(x^{(\hat{i})} + d)}$, passe Δ ($\geq \delta$) entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{i})} + d$ für geeignet gewähltes j .
4. Bestimme $m^{(k+1)}$ aus (QP_m^{k+1}) mit neuem $x^{(j)}$, $k \leftarrow k + 1$, GOTO 1.

Bemerkungen:

- Der adaptive Parameter δ wird nur verkleinert, wenn Schrittweiten $\geq \delta$ keine Verbesserung mehr zu bringen scheinen.

Grober Ablauf von NEUWOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.
 Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{i})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{i}} - f(x^{(\hat{i})} + d)}{f_{\hat{i}} - m^{(k)}(x^{(\hat{i})} + d)}$, passe Δ ($\geq \delta$) entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{i})} + d$ für geeignet gewähltes j .
4. Bestimme $m^{(k+1)}$ aus (QP_m^{k+1}) mit neuem $x^{(j)}$, $k \leftarrow k + 1$, GOTO 1.

Bemerkungen:

- Der adaptive Parameter δ wird nur verkleinert, wenn Schrittweiten $\geq \delta$ keine Verbesserung mehr zu bringen scheinen.
- Jede Wahl neuer Punkte $x^{(i)}$ soll etwa Abstand δ haben.

Grober Ablauf von NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
Bestimme $x^{(i)}$, f_i ($i = 1, \dots, 2n$) und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{i} \in \{0, \dots, 2n\}$ mit $f_{\hat{i}} \leq f_i$ ($\forall i$) und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{i})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
Ist zum 3. Mal $|f(x^{(\hat{i})} + d) - m^{(k)}(x^{(\hat{i})} + d)|$ klein, dann
 - (a) falls $\delta = \underline{\Delta}$, STOP,
 - (b) sonst setze $\delta \leftarrow \max\{\underline{\Delta}, \delta/10\}$, $\Delta \leftarrow \delta$, GOTO 1.
Setze $\Delta \leftarrow \max\{\delta, \Delta/10\}$,
finde Ersatzpunkt für j mit $\|x^{(j)} - x^{(\hat{i})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{i}} - f(x^{(\hat{i})} + d)}{f_{\hat{i}} - m^{(k)}(x^{(\hat{i})} + d)}$, passe Δ ($\geq \delta$) entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{i})} + d$ für geeignet gewähltes j .
4. Bestimme $m^{(k+1)}$ aus (QP_m^{k+1}) mit neuem $x^{(j)}$, $k \leftarrow k + 1$, GOTO 1.

Bemerkungen:

- Der adaptive Parameter δ wird nur verkleinert, wenn Schrittweiten $\geq \delta$ keine Verbesserung mehr zu bringen scheinen.
- Jede Wahl neuer Punkte $x^{(i)}$ soll etwa Abstand δ haben.
- Viele weitere Details für Effizienz, Numerik, etc. ...
- Bester ableitungsfreier Code in Benchmark von Moré und Wild, 2009.