Optimierung für Nichtmathematiker

(für Master)

Vorlesung: Christoph Helmberg
Übung: Anja Lau

Ziele: Einführung in

- richtige Einordnung von Optimierungsproblemen
- Modellierungstechniken
- praktische Umsetzung mit Optimierungssoftware
- grobes Verständnis für Funktionsweise und Grenzen der Optimierungsverfahren

ET/IT, MB, WiWi $\rightarrow$ Anwendungen in allen drei Gebieten
(z.B. Signalverarbeitung, Truss Topology Design, Operations Research)

Inhaltsübersicht

2: $2 \in [2,2]$

Inhaltsübersicht

3: $3 \in [3,3]$

Inhaltsübersicht

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P) \qquad \begin{array}{lll}
\text{Minimiere} & f(x) & \text{Zielfunktion} \\
\text{unter (s.t.)} & h_i(x) = 0, \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0, \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P)\qquad
\begin{array}{lllll}
\text{Minimiere} & f(x) & & & \text{Zielfunktion} \\
\text{unter (s.t.)} & h_i(x) = 0, & i \in \mathcal{E} & & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0, & i \in \mathcal{I} & & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & & & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P)\qquad
\begin{array}{lllll}
\text{Minimiere} & f(x) & & & \text{Zielfunktion} \\
\text{unter (s.t.)} & h_i(x) = 0, & i \in \mathcal{E} & & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0, & i \in \mathcal{I} & & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & & & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$

$(f(x) = x - x^2,\ \mathcal{E} = \emptyset,\ g_1(x) = x^2 - 1,\ \mathcal{I} = \{1\},\ \Omega = \mathbb{R})$

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P)\qquad
\begin{array}{lllll}
\text{Minimiere} & f(x) & & & \text{Zielfunktion} \\
\text{unter (s.t.)} & h_i(x) = 0, & i \in \mathcal{E} & & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0, & i \in \mathcal{I} & & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & & & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$

$(f(x) = x - x^2,\ \mathcal{E} = \emptyset,\ g_1(x) = x^2 - 1,\ \mathcal{I} = \{1\},\ \Omega = \mathbb{R})$

(P2) $\min x - x^2$ s.t. $-1 \leq x \leq 1,\ x \in \mathbb{R}$

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P)\qquad
\begin{array}{lllll}
\text{Minimiere} & f(x) & & & \text{Zielfunktion} \\
\text{unter (s.t.)} & h_i(x) = 0, & i \in \mathcal{E} & & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0, & i \in \mathcal{I} & & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & & & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$

$(f(x) = x - x^2,\ \mathcal{E} = \emptyset,\ g_1(x) = x^2 - 1,\ \mathcal{I} = \{1\},\ \Omega = \mathbb{R})$

(P2) $\min x - x^2$ s.t. $-1 \leq x \leq 1,\ x \in \mathbb{R}$

$(f,\ \mathcal{E},\ \Omega$ *w.o.*, $g_1(x) = -1 - x,\ g_2(x) = x - 1,\ \mathcal{I} = \{1, 2\})$

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P)\qquad
\begin{array}{lllll}
\text{Minimiere} & f(x) & & & \text{Zielfunktion} \\
\text{unter (s.t.)} & h_i(x) = 0, & i \in \mathcal{E} & & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0, & i \in \mathcal{I} & & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & & & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$

$(f(x) = x - x^2,\ \mathcal{E} = \emptyset,\ g_1(x) = x^2 - 1,\ \mathcal{I} = \{1\},\ \Omega = \mathbb{R})$

(P2) $\min x - x^2$ s.t. $-1 \leq x \leq 1,\ x \in \mathbb{R}$

$(f,\ \mathcal{E},\ \Omega$ *w.o.*, $g_1(x) = -1 - x,\ g_2(x) = x - 1,\ \mathcal{I} = \{1, 2\})$

(P3) $\min x - x^2$ s.t. $x \in [-1, 1]$

1.1 Das Optimierungsproblem in allgemeiner Form

$$
(P)\qquad
\begin{array}{lllll}
\text{Minimiere} & f(x) & & & \text{Zielfunktion}\\
\text{unter (s.t.)} & h_i(x) = 0, & i \in \mathcal{E} & & \text{Gleichungsnebenbed.}\\
& g_i(x) \le 0, & i \in \mathcal{I} & & \text{Ungleichungsnebenbed.}\\
& x \in \Omega & & & \text{Variablen, Grundmenge}
\end{array}
$$

Bei uns meist:

$\Omega \subseteq \mathbb{R}^n$ (geht aber auch $\mathbb{C}^n$,$\mathbb{Z}^n$, Funktionenräume ...)

$f, g_i, h_i : \mathbb{R}^n \to \mathbb{R}$ (linear, konvex, hinreichend glatt, ...)

$|\mathcal{E}|$, $|\mathcal{I}|$ endliche Indexmengen (abzählbar, überabzählbar)

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \le 1,\ x \in \mathbb{R}$

$(f(x) = x - x^2,\ \mathcal{E} = \emptyset,\ g_1(x) = x^2 - 1,\ \mathcal{I} = \{1\},\ \Omega = \mathbb{R})$

(P2) $\min x - x^2$ s.t. $-1 \le x \le 1,\ x \in \mathbb{R}$

$(f,\ \mathcal{E},\ \Omega$ *w.o.*, $g_1(x) = -1 - x,\ g_2(x) = x - 1,\ \mathcal{I} = \{1, 2\})$

(P3) $\min x - x^2$ s.t. $x \in [-1, 1]$ $(\mathcal{I} = \mathcal{E} = \emptyset,\ \Omega = [-1, 1])$

Terminologie: Zulässige Menge, Optimalwert

$$
(P) \qquad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Bigg\} \text{Restriktionen}
$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge*
  $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$

Terminologie: Zulässige Menge, Optimalwert

$$
(P) \qquad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Bigg\} \text{Restriktionen}
$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge* $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$
- *Ist* $\mathcal{X}(P) = \emptyset$*, heißt das Problem* (P) *unzulässig.*

Terminologie: Zulässige Menge, Optimalwert

$$(P) \quad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Big\} \text{ Restriktionen}$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge*
  $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$
- *Ist $\mathcal{X}(P) = \emptyset$, heißt das Problem (P) unzulässig.*
- *$v(P) := \inf_{x \in \mathcal{X}} f(x)$ ist der Optimalwert von (P).*
  Falls $v(P) = -\infty$ heißt das Problem (P) unbeschränkt.
  Für unzulässiges (P) ist $v(P) = \infty$.

Terminologie: Zulässige Menge, Optimalwert

$$(P) \quad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Big\} \text{ Restriktionen}$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge*
  $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$
- *Ist $\mathcal{X}(P) = \emptyset$, heißt das Problem (P) unzulässig.*
- *$v(P) := \inf_{x \in \mathcal{X}} f(x)$ ist der Optimalwert von (P).*
  Falls $v(P) = -\infty$ heißt das Problem (P) unbeschränkt.
  Für unzulässiges (P) ist $v(P) = \infty$.

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$ $\qquad \mathcal{X} =$

Terminologie: Zulässige Menge, Optimalwert

$$
(P) \quad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Big\} \text{ Restriktionen}
$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge*
  $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$
- *Ist $\mathcal{X}(P) = \emptyset$, heißt das Problem (P) unzulässig.*
- *$v(P) := \inf_{x \in \mathcal{X}} f(x)$ ist der Optimalwert von (P).*
  Falls $v(P) = -\infty$ heißt das Problem (P) unbeschränkt.
  Für unzulässiges (P) ist $v(P) = \infty$.

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$ $\qquad \mathcal{X} = [-1, 1],\ v(P1) =$

Terminologie: Zulässige Menge, Optimalwert

$$(P) \quad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Big\} \text{ Restriktionen}$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge* $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$
- *Ist* $\mathcal{X}(P) = \emptyset$, *heißt das Problem* (P) *unzulässig.*
- $v(P) := \inf_{x \in \mathcal{X}} f(x)$ *ist der Optimalwert von* (P).
  Falls $v(P) = -\infty$ *heißt das Problem* (P) *unbeschränkt.*
  Für unzulässiges (P) ist $v(P) = \infty$.

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$ $\quad \mathcal{X} = [-1, 1]$, $v(P1) = -2$,
ebenso (P2) $(-1 \leq x \leq 1,\ x \in \mathbb{R})$ *und (P3)* $(x \in [-1, 1])$.
(P4) $\min x - x^2$ s.t. $x^2 = 1,\ x \in \mathbb{R}$ $\quad \mathcal{X} =$

Terminologie: Zulässige Menge, Optimalwert

$$(P) \quad \begin{array}{lllll} \text{Minimiere} & f(x) & & & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Glgsnebenbed.} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Unglgsnebenbed.} \\ & x \in \Omega & & & \text{Grundmenge} \end{array} \Big\} \text{ Restriktionen}$$

Definition

- *Punkte, die alle Bedingungen erfüllen, bilden die zulässige Menge* $\mathcal{X}(P) := \{x \in \Omega : h_i(x) = 0, i \in \mathcal{E}, g_j(x) \leq 0, j \in \mathcal{I}\}$
- *Ist* $\mathcal{X}(P) = \emptyset$, *heißt das Problem* (P) *unzulässig.*
- $v(P) := \inf_{x \in \mathcal{X}} f(x)$ *ist der Optimalwert von* (P).
  Falls $v(P) = -\infty$ *heißt das Problem* (P) *unbeschränkt.*
  Für unzulässiges (P) ist $v(P) = \infty$.

Beispiel

(P1) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$ $\quad \mathcal{X} = [-1, 1],\ v(P1) = -2,$
ebenso (P2) $(-1 \leq x \leq 1,\ x \in \mathbb{R})$ *und (P3)* $(x \in [-1, 1])$.
(P4) $\min x - x^2$ s.t. $x^2 = 1,\ x \in \mathbb{R}$ $\quad \mathcal{X} = \{-1, 1\},\ v(P4) = -2$

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll}
\text{Minimiere} & f(x) & \text{Zielfunktion} \\
\text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & \text{Grundmenge}
\end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll}
\text{Minimiere} & f(x) & \text{Zielfunktion} \\
\text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & \text{Grundmenge}
\end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*
- *Ein* $x \in \mathcal{X}$ *heißt lokale Optimallösung, wenn es eine (kleine) Umgebung* $U(x)$ *um* x *gibt mit* $f(x) \leq f(y)\ \forall y \in \mathcal{X} \cap U(x)$

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll}
\text{Minimiere} & f(x) & \text{Zielfunktion} \\
\text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & \text{Grundmenge}
\end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*
- *Ein* $x \in \mathcal{X}$ *heißt lokale Optimallösung, wenn es eine (kleine) Umgebung* $U(x)$ *um* x *gibt mit* $f(x) \leq f(y)\ \forall y \in \mathcal{X} \cap U(x)$

Beispiel

(P1-P3) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$
globale OL:

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll} \text{Minimiere} & f(x) & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\ & x \in \Omega & \text{Grundmenge} \end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*
- *Ein* $x \in \mathcal{X}$ *heißt lokale Optimallösung, wenn es eine (kleine) Umgebung* $U(x)$ *um* x *gibt mit* $f(x) \leq f(y)\ \forall y \in \mathcal{X} \cap U(x)$

Beispiel

(P1-P3) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$
globale OL: $x \in \{-1\}$*, lokale OL:*

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll} \text{Minimiere} & f(x) & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\ & x \in \Omega & \text{Grundmenge} \end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*
- *Ein* $x \in \mathcal{X}$ *heißt lokale Optimallösung, wenn es eine (kleine) Umgebung* $U(x)$ *um* x *gibt mit* $f(x) \leq f(y)\ \forall y \in \mathcal{X} \cap U(x)$

Beispiel

(P1-P3) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$
globale OL: $x \in \{-1\}$*, lokale OL:* $x \in \{-1, 1\}$

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll} \text{Minimiere} & f(x) & \text{Zielfunktion} \\ \text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\ & x \in \Omega & \text{Grundmenge} \end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*
- *Ein* $x \in \mathcal{X}$ *heißt lokale Optimallösung, wenn es eine (kleine) Umgebung* $U(x)$ *um* x *gibt mit* $f(x) \leq f(y)\ \forall y \in \mathcal{X} \cap U(x)$

Beispiel

(P1-P3) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$
globale OL: $x \in \{-1\}$*, lokale OL:* $x \in \{-1, 1\}$
(P4) $\min x - x^2$ s.t. $x^2 = 1,\ x \in \mathbb{R}$ *???*

Terminologie: Lösungen, globale/lokale Optimallösungen

$$
(P) \quad \begin{array}{lll}
\text{Minimiere} & f(x) & \text{Zielfunktion} \\
\text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbed.} \\
& g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbed.} \\
& x \in \Omega & \text{Grundmenge}
\end{array}
$$

Definition

- $x \in \mathcal{X}(P)$ *. . . heißt Lösung oder zulässiger Punkt*
- *Ein* $x \in \mathcal{X}$ *mit* $f(x) = v(P)$ *heißt (globale) Optimallösung (also* $f(x) \leq f(y)\ \forall y \in \mathcal{X}$*)*
- *Ein* $x \in \mathcal{X}$ *heißt lokale Optimallösung, wenn es eine (kleine) Umgebung* $U(x)$ *um* x *gibt mit* $f(x) \leq f(y)\ \forall y \in \mathcal{X} \cap U(x)$

Beispiel

(P1-P3) $\min x - x^2$ s.t. $x^2 \leq 1,\ x \in \mathbb{R}$
globale OL: $x \in \{-1\}$*, lokale OL:* $x \in \{-1, 1\}$
(P4) $\min x - x^2$ s.t. $x^2 = 1,\ x \in \mathbb{R}$ *??? ebenso*

Beachte:
Optimalwert gibt es nur einen, Optimallösungen u.U. viele!

Oft wird Optimalwert/lösung mit einem $*$ gekennzeichnet, z.B.:

$$f^* = f(x^*) = \inf_{x \in \mathcal{X}} f(x)$$

Es bezeichnet

- $x^* = \operatorname{argmin}\{f(x) : x \in \mathcal{X}\}$ die eindeutige Optimallösung (wenn wir schon wissen, dass es genau eine gibt),
- $\operatorname{Argmin}\{f(x) : x \in \mathcal{X}\}$ die Menge aller Optimallösungen (diese kann auch leer sein).

Optimal ist nicht steigerungsfähig,
„noch optimaler“ ist sinnlos und schlechter Sprachgebrauch!

Inhaltsübersicht

1.2 Überblick über spezielle Klassen von Optimierungsproblemen

Verfahren gibt es nur für eingeschränkte Problemklassen, diese unterscheiden sich nach

- den Eigenschaften der Funktionen f, g_i, h_i
- den Eigenschaften der Grundmenge Ω
- der Form, in der die Problemdaten gegeben sind
- den Ansprüchen an die Lösung (lokal/global/multikriteriell)

Verfahren/Löser für viele wichtige Problemklassen gibt es auf dem

NEOS Server for Optimization

Nichtlineare Optimierung (NonLinear Programming)

$$\begin{array}{ll} \text{Minimiere} & f(x) \\ \text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \\ & x \in \Omega \end{array}$$

f, g_i, h_i	„hinreichend glatt“, $C_1(\mathbb{R}^n)$ oder $C_2(\mathbb{R}^n)$, d.h., mindestens einmal oder zweimal stetig differenzierbar,
$\mathcal{E}$ und $\mathcal{I}$	endliche Mengen (unendlich: „Semiinfinite Opt.“) falls $\mathcal{E} = \mathcal{I} = \emptyset$: freie/unrestringierte Optimierung sonst restringierte Optimierung oder Opt. mit Nebenbed.
$\Omega = \mathbb{R}^n$	(meist)
Ziel:	lokales Optimum (aber oft schon Zulässigkeit schwer!)
Anw.:	Optimalsteuerung, Shape Optimization, Parameterschätzung (nichtlin.), Lösung nichtlinearer Gleichungssysteme, ...
Verf.:	für lokal gute Konvergenz: Newton, Quasinewton, ... zur Suche lokaler Mulden: Line-Search, Trust-Region, CG, ...
Input:	Unterroutinen für Funktionswert, Gradient, (Hessematrix)
Größe :	einige 100 bis einige 1000 Variablen (mehr bei spez. Struktur)

Konvexe Optimierung (Convex Optimization)

$$\begin{array}{ll} \text{Minimiere} & f(x) \\ \text{unter} & Ax = b \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \\ & x \in \Omega \end{array}$$

f, g_i	konvexe Funktionen
$Ax = b$	nur lineare Gleichungsnebenbedingungen!
	falls f, g_i glatt $\rightarrow$ smooth convex opt.
	falls f, g_i nicht notw. diffb. $\rightarrow$ nonsmooth convex opt.
$\mathcal{I}$	endliche Menge
$\mathcal{C}$	„einfache“ konvexe Menge (Box[=Intervall], $\mathbb{R}^n$, ...)
Ziel:	globales Optimum
Anw.:	Portfolio Design, Experimental Design, Optimalsteuerung, Signal Processing, Berechnung von Schranken für nichtlineare Probleme, ...
Verf.:	smooth: Newton, Quasinewton, ...
	nonsmooth: Subgradienten-, Bündel-Verfahren
Input:	Unterroutinen für Funktionswert, (Sub)Gradient, (Hessematrix)
Größe :	einige 100 bis einige 10000 Var. (mehr bei spez. Struktur)

Konvexe Opt. mit Struktur (Structured Convex Opt.)

Minimiere $q(x)$
unter $Ax = b$
$x \in \mathcal{K}$

$q(\cdot)$	lineare (affine) oder konvex-quadratische Zielfunktion $q(x) = c^T x$ ($+\frac{1}{2} x^T Q x$ mit Q symmetrisch positiv semidefinit)
$Ax = b$	lineare Gleichungs- oder auch Ungleichungsnebenbedingungen
$\mathcal{K}$	konvexe Kegel spezieller Struktur q linear, $\mathcal{K} = R^n_+$: Lineare Opt. (Linear Programming) q linear, $\mathcal{K} = \mathcal{Q}^n_+$: Second Order Cone Opt. (SOCP) q linear, $\mathcal{K} = S^n_+$: Semidefinite Opt. (SemiDefnite P.) q quadrat., $\mathcal{K} = \mathbb{R}^n_+$: (Konvexe) Quadratische Opt. (QP)
Ziel:	globales Optimum in „kurzer“ Zeit
Anw.:	Portfolio Design, Experimental Design, Optimalsteuerung, Signal Processing, Berechnung von Schranken für ganzz. Probleme, ...
Verf.:	LP, SOCP, SDP, QP: Innere-Punkte-Verf. (Newton) LP: Simplex
Input:	Koeffizienten der Matrizen und Vektoren (evtl. Kegeltyp)
Größe :	einige 1000 bis Millionen Variablen

Ganzzahlige Optimierung (Integer Programming)

Minimiere $c^T x$
unter $Ax \leq b$
$x \in \mathbb{Z}^n$

$c^T x$	lineare Zielfunktion
$Ax \leq b$	lineare Gleichungs- oder auch Ungleichungsnebenbedingungen
$x \in \mathbb{Z}^n$	nur ganzzahlige Lösungen! verwandte Varianten:
	Binary Integer P.: $x \in \{0,1\}^n$ ($\approx$ kombinatorische Opt.)
	Mixed Integer P.: $x \in \mathbb{R}^{n_1} \times \mathbb{Z}^{n_2}$
	Mixed Integer NonLinear P.: f, g_i, h_i nichtlin.
Ziel:	sehr problemabhängig (meist *NP*-schwer), „gute“ Lösung mit Gütegarantie
Anw.:	Probleme mit Entscheidungskomponenten, z.B., Flüsse in Netzwerken, Zuweisungs-, Transport-, Standortprobleme, VLSI-Design, Basisauswahl, ...
Verf.:	konvexe/lineare Relaxation und lokale Suche/Rundungsheuristiken, exakte Lösung durch „Branch and Bound“ (effiz. Enumerieren), für sehr spezielle Probleme: exakte Algorithmen
Input:	von Koeffizienten der Matrizen bis hin zu Struktur-nutzenden Zusatzroutinen
Größe :	extrem problemabhängig, von unter 100 bis zu Millionen

Globale Optimierung (Global Optimization)

$$\begin{array}{ll} \text{Minimiere} & f(x) \\ \text{unter} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \\ & x \in \Omega \end{array}$$

f, g_i, h_i	hinreichend glatt von bekannter Struktur (z.B. Polynome) so dass auf Intervallen Unterschätzer konstruierbar sind
$\mathcal{E}$ und $\mathcal{I}$	(kleine) endliche Mengen
Ω	„einfache“ konvexe Menge (Box)
Ziel:	globales Optimum (!!! i.A. zu schwer, nur sehr kleine Dimension!!!)
Anw.:	kleine nichtlineare Optimalsteuerungsprobleme . . .
Verf.:	Branch and Bound: pro Intervall der Unterteilung untere Schranken durch Lösung konvexer Relaxation und obere Schranken durch NLP-Löser
Input:	algebraische Beschreibung der Funktionen
Größe :	etwa 10-30 Variable (je nach spez. Struktur u.U. auch mehr)

Einige weitere Klassen/Forschungsgebiete

Meist durch spezielle Anwendungsforderungen motiviert:

Multikriterielle Optimierung (Mehrziel-Opt.):

- Bspe: Portfolio soll Gewinn maximieren und Risiko minimieren, Auto soll möglichst schnell mit möglichst wenig Treibstoff fahren, größte Stabilität bei geringstem Materialeinsatz, etc.
- Darstellung konkurrierende Ziele werden durch $f : \mathbb{R}^n \to \mathbb{R}^m$ und eine durch einen spitzen Kegel induzierte Partialordnung auf $\mathbb{R}^m$.
- „Pareto-optimale Lösung“: bzgl. Partialordnung nicht verbesserbar
- Für m klein, Berechnung der „Paretofront“, sonst Rückführung auf Standardverf. durch Skalarisierung (gewichtete Linearkombination) oder lexikographisches Optimieren mittels neuer Nebenbedingungen

Ableitungsfreie Optimierung (derivative-free opt.):

- Es ist jeweils nur $f(x)$ bestimmbar (wird durch Simulation, Messung, Bohrung, etc. ermittelt), aber keine Ableitungsinformation
- $f(x)$ billig: numerisches Differenzieren oder Verf. von Nelder-Mead
- $f(x)$ teuer: Modellerstellende Verfahren (Kriging, Powell, ...)

Einige weitere Klassen/Forschungsgebiete

Stochastische Optimierung:

- Statistische Daten in Entscheidungen einbeziehen: Ein-/Ausschalten von Kraftwerken für stochstisches Verbrauchsmodell, Portfoliooptimierung für stochastische Finanzmodelle, Logistik-Optimierung nach stochastischem Bedarfsmodell
- oft Einteilung in gewichtete mehrstufige Szenarien, rekursives Lösen mit Standardverfahren

Robuste Optimierung:

- Gegen Datenunsicherheit, Mindestanforderungen, oder Ungenauigkeiten in der realen Umsetzung absichern: leichteste Brücke für unterschiedliche Lasten, Entwurf von Antennen-Arrays, Mindestproduktionskapazitäten bei Maschinenausfällen
- geschickte Modellierung erlaubt oft den Einsatz von Standardverfahren

Inhaltsübersicht

Inhaltsübersicht

2 Lineare Optimierung

Zwei typische Schreibweisen für ein „Lineares Programm“ (LP)

LP in Standardform

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array}$$

LP in kanonischer Form

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array}$$

Wenn nicht anders erwähnt, dann $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$.

$x \geq 0$ und $Ax \geq b$ sind jeweils elementweise zu vestehen,
also $x \geq 0 :\Leftrightarrow x \in \mathbb{R}^n_+ = \{x \in \mathbb{R}^n : x_i \geq 0, i = 1, \ldots n\}$

Egal ob man

- minimieren oder maximieren will,
- Gleichungen oder Ungleichungen hat,
- Variablen mit oder ohne Nichtnegativitätsbedingungen hat,

man kann jedes LP in so eine Form bringen.

Umformen von LPs

- maximieren $\longrightarrow$ minimieren:

  $$\max c^T x \text{ s.t. } Ax = b, x \geq 0 \;\Leftrightarrow\; -\min(-c)^T x \text{ s.t. } Ax = b, x \geq 0$$

- Gleichungen $\longrightarrow$ Ungleichungen:

  $$Ax = b \quad \Leftrightarrow \quad \begin{bmatrix} A \\ -A \end{bmatrix} x \geq \begin{bmatrix} b \\ -b \end{bmatrix}$$

- Ungleichungen $\longrightarrow$ Gleichungen: durch Schlupfvariablen $s \geq 0$

  $$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax \geq b \\ & x \geq 0 \end{array} \;\Leftrightarrow\; \begin{array}{ll} \min & c^T x \\ \text{s.t.} & [A, I] \begin{bmatrix} x \\ s \end{bmatrix} = b \\ & x \geq 0, s \geq 0 \end{array} \qquad [I \text{ Identität}]$$

- Statt jeder freien Variable $x_i \in \mathbb{R}$ zwei vorzeichenbeschränkte:

  $$x_i^+ \geq 0 \text{ und } x_i^- \geq 0 \;\text{ mit }\; x_i = x_i^+ - x_i^-$$

Kleines Beispiel zur Illustration: Mozart-Problem

Maximiere den Gewinn bei Produktion von Mozart-Kugeln und -Talern:

	Marzipan	Nougat	Edelherb	Gewinn
Pro Kugel:	2 Teile	1 Teil	1 Teil	3 Euro
Pro Taler:	1 Teil	1 Teil	2 Teile	2 Euro
verfügbar:	10 Teile	6 Teile	9 Teile	

Variablen:

x_K ... Anzahl Kugeln

x_T ... Anzahl Taler

$$\begin{array}{lll} \max & 3x_K + 2x_T & \\ \text{s.t.} & 2x_K + 1x_T \leq 10 & \\ & 1x_K + 1x_T \leq 6 & \Leftrightarrow \\ & 1x_K + 2x_T \leq 9 & \\ & x_K \geq 0, x_T \geq 0 & \end{array} \qquad \begin{array}{lll} -\min & [-3,-2,0,0,0]x & \\ \text{s.t.} & \begin{bmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 10 \\ 6 \\ 9 \end{bmatrix} & \\ & x \geq 0 \qquad [x \in \mathbb{R}^5_+] & \end{array}$$

Anwendung: Tschebyscheff-Approximation

Mathematisches Problem: Minimiere $\|Ax - b\|_\infty$ über $x \in \mathbb{R}^n$

$[\|\bar{b}\|_\infty := \max_{i=1,\ldots,m} |\bar{b}_i|$ ist die Unendlich-Norm]

$$\rightarrow \qquad \min s \text{ s.t. } -\mathbf{1}s \leq Ax - b \leq \mathbf{1}s \qquad \mathbf{1} = [1, 1, \ldots, 1]^T$$

Anwendung z.B. in der Filtersynthese (hier stark vereinfacht):

Eine gewünschte Impulsantwort (Funktion) $h : [0, 1] \rightarrow \mathbb{R}$ soll möglichst gut durch gewichtete Addition verfügbarer Impulsantworten $h_i : [0, 1] \rightarrow \mathbb{R}$, $i = 1, \ldots, n$ dargestellt werden.

Modellierung:
Variable $x_i \in \mathbb{R}$, $i = 1, \ldots, n$ als Gewicht von h_i,
Diskretisierung von $[0, 1]$ z.B. in Schritte $t_j = j/h$, $j = 0, \ldots, h$

$$\begin{array}{lll} \min & s & \\ \text{s.t.} & -s \leq h(t_j) - \sum_{i=1}^n x_i h_i(t_j) \leq s & j = 0, \ldots, h \\ & x \in \mathbb{R}^n, s \in \mathbb{R} & \end{array}$$

Ähnlich: 1-Norm Minimierung

Mathematisches Problem: Minimiere $\|Ax - b\|_1$ über $x \in \mathbb{R}^n$

$[\|\bar{b}\|_1 := \sum_{i=1}^{m} |\bar{b}_i|$ ist die 1-/Manhattan-Norm$]$

$$\rightarrow \qquad \min \mathbf{1}^T s \;\; \text{s.t.} \;\; -Is \leq Ax - b \leq Is$$

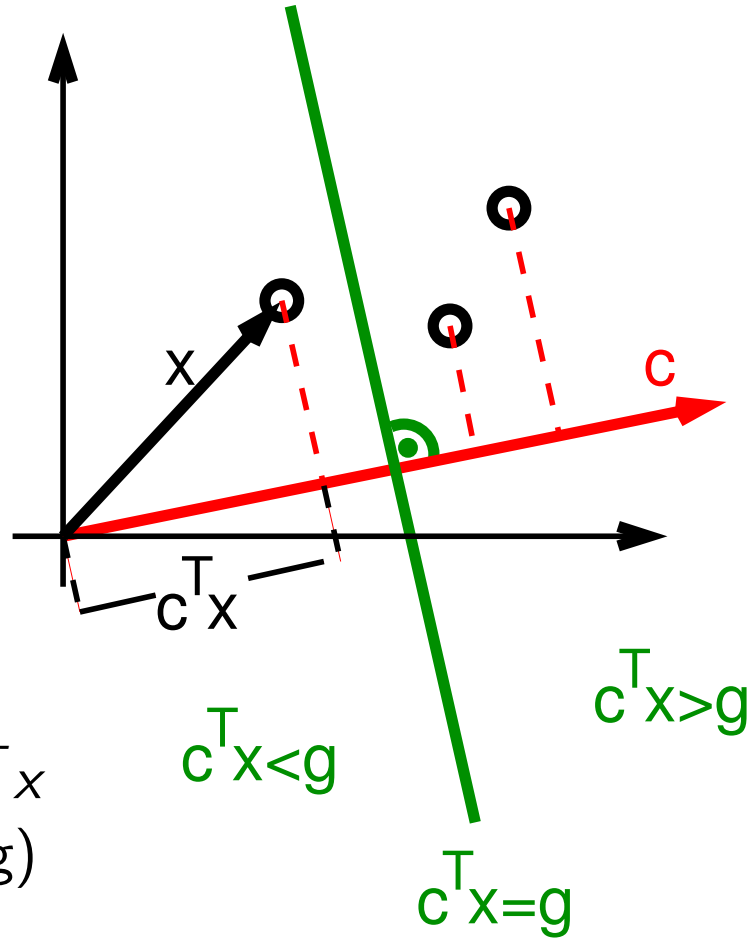
Geometrische Interpretation: WH Skalarprodukt $c^T x$

$c^T x = \|c\|\|x\| \cos\alpha$
mit α Winkel zw. c und x.

Für $\|c\| = 1$ ist (wie im Bild) $c^T x$ die Länge der Projektion von x auf die Gerade $\{\gamma c : \gamma \in \mathbb{R}\}$

$\{x : c^T x \leq \gamma\}$ ist der Halbraum aller Punkte, deren Projektion auf c kleiner gleich γ ist.

c ist Gradient der linearen Fkt. $c^T x$ (zeigt in Richtung steilster Anstieg)



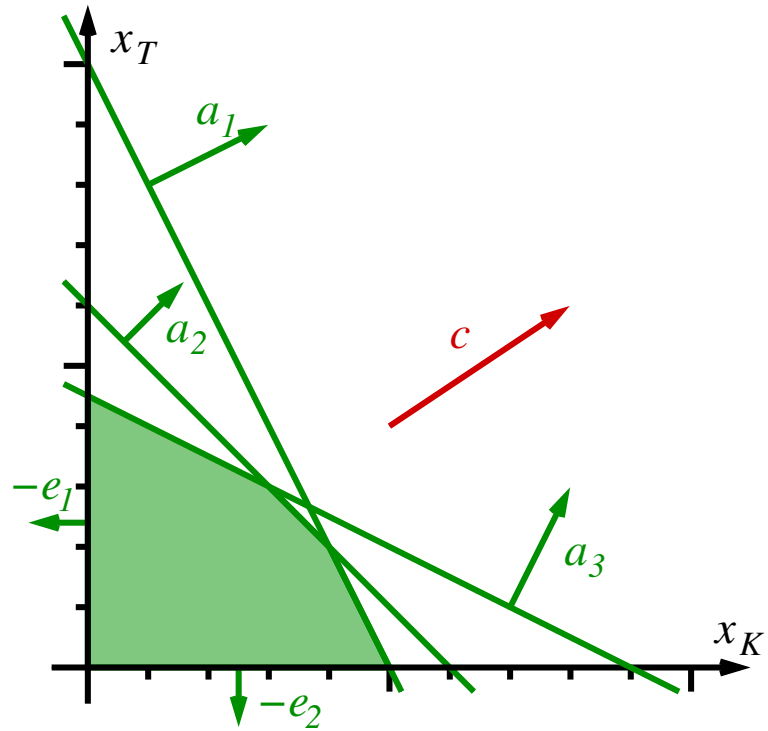
Für $A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}$ ist Zeile j von $Ax \leq b$ gerade $a_j^T x \leq b_j$

Geometrische Interpretation: am Mozart-Problem

Wegen $x_K \geq 0 \Leftrightarrow -x_K \leq 0$ ist

$$\begin{array}{ll} \max & \begin{bmatrix} 3 & 2 \end{bmatrix} x \\ \text{s.t.} & \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_K \\ x_T \end{bmatrix} \leq \begin{bmatrix} 10 \\ 6 \\ 9 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

wieder das Mozart-Problem.



Definition

Ein Polyeder ist der Schnitt endlich vieler Halbräume.

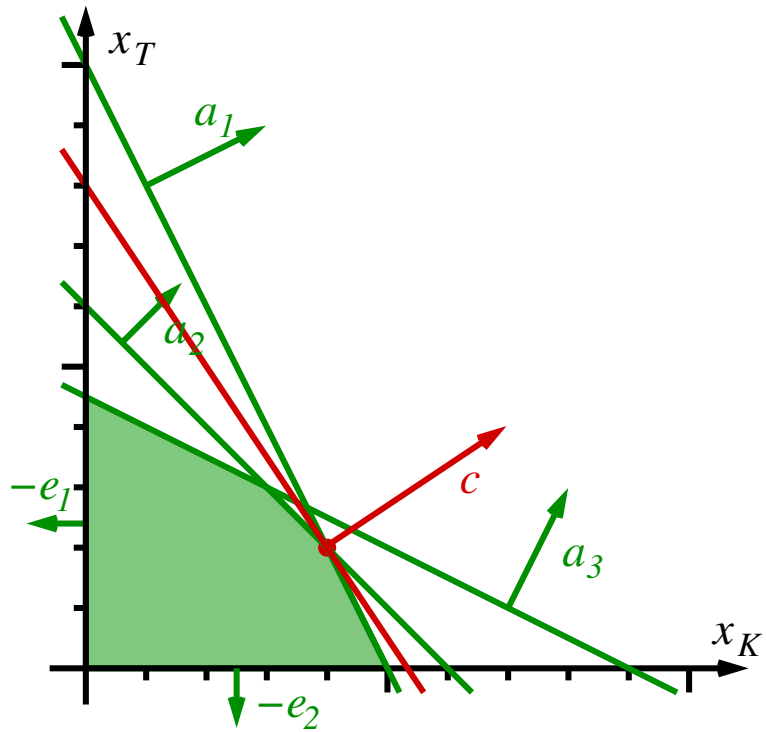
Die zulässige Menge jedes LPs ist also ein Polyeder.

Geometrische Interpretation: Optimallösung

Wegen $x_K \geq 0 \Leftrightarrow -x_K \leq 0$ ist

$$\begin{array}{ll} \max & \begin{bmatrix} 3 & 2 \end{bmatrix} x \\ \text{s.t.} & \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \\ -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_K \\ x_T \end{bmatrix} \leq \begin{bmatrix} 10 \\ 6 \\ 9 \\ 0 \\ 0 \end{bmatrix} \end{array}$$

wieder das Mozart-Problem.



„Offensichtlich“ ist eine Ecke des Polyeders eine Optimallösung,
sie erfüllt Dimension viele Ungleichungen mit Gleichheit, hier:

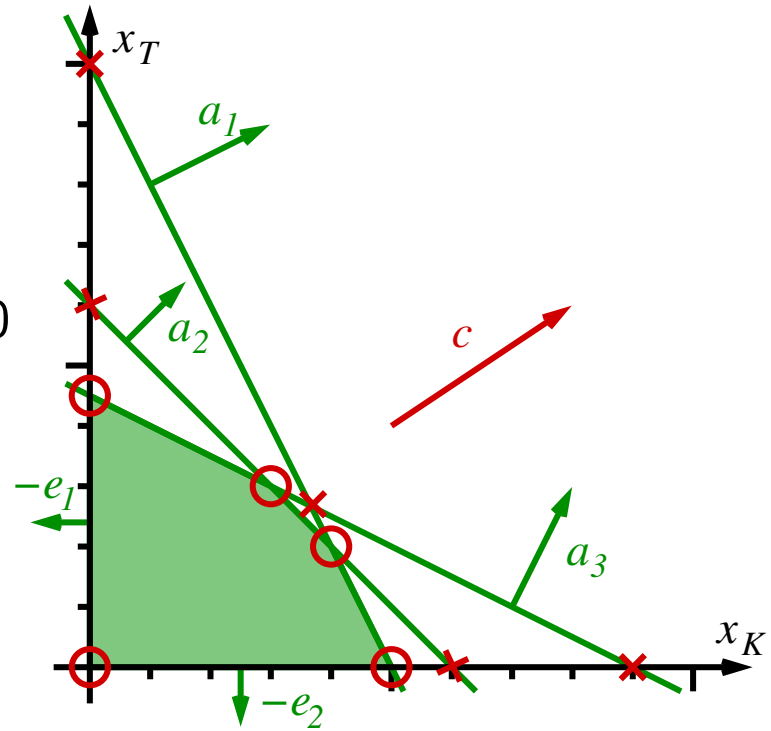
$$\begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_K \\ x_T \end{bmatrix} = \begin{bmatrix} 10 \\ 6 \end{bmatrix} \quad \Rightarrow x_K = 4, x_T = 2$$

Interpretation einer Ecke in Standardform

$$\max \begin{bmatrix} 3 & 2 & 0 & 0 & 0 \end{bmatrix} x$$

$$\text{s.t.} \begin{bmatrix} 2 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 2 & 0 & 0 & 1 \end{bmatrix} x = \begin{bmatrix} 10 \\ 6 \\ 9 \end{bmatrix}$$

$$x_K \geq 0, x_T \geq 0, s_1 \geq 0, s_2 \geq 0, s_3 \geq 0$$



In dieser Darstellung $n = 5$, $m = 3$

Jeder Ungleichung ist genau eine Schlupfvariable zugeordnet!
(x_K für $x_k \geq 0$ und x_T für $x_T \geq 0x$)

Wegen der m Gleichungen hat das Polyeder Dimension $\leq n - m$.
Für eine Ecke müssen $n - m$ weitere Ungleichungen als Gleichungen gewählt werden ($\Leftrightarrow$ Schlupfvariablen auf 0 setzen), so dass diese eindeutig einen Punkt festlegen, und für diesen der Wert der anderen Schlupfvariablen (=„Abstand“ zur Ungleichung) nichtnegativ ist.

Inhaltsübersicht

2.2 Der Simplex-Algorithmus (für Standardform)

Gegeben $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$, bestimme

$$\min c^T x \text{ s.t. } Ax = b, x \geq 0$$

Idee: Gehe von der aktuellen Ecke zu einer benachbarten besseren.

Notation: Für $B \in \{1, \ldots, n\}^m$ sei $A_B := [a_{iB(j)}]_{i,j=1,\ldots,m}$.
Wir fassen B auch als Menge von Spaltenindices auf.

Definition

Eine Spaltentupel $B \in \{1, \ldots, n\}^m$ mit A_B regulär heißt Basis und $N := \{1, \ldots, m\} \setminus B$ Nichtbasis.
Ist zusätzlich $A_B^{-1} b \geq 0$ heißt die Basis zulässig.

$$Ax = b \qquad \rightarrow \qquad A_B x_B + A_N x_N = b$$

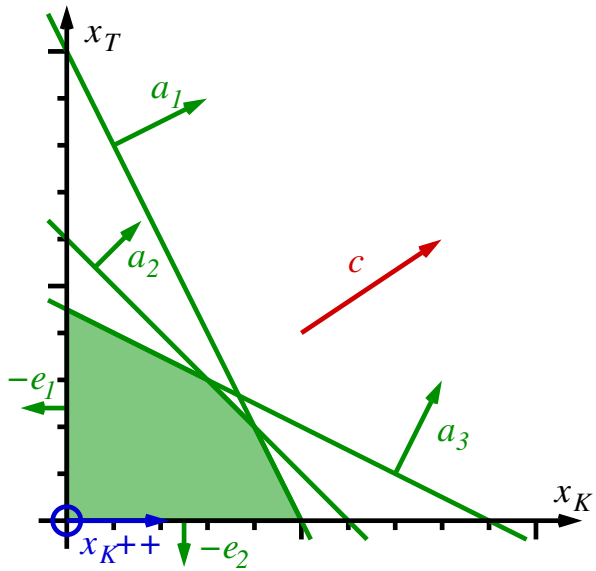
Die Nichtbasisvariablen x_i, $i \in N$, werden auf 0 gesetzt,
die Basisvariablen x_i, $i \in B$, durch $x_B = A_B^{-1} b$ bestimmt.

Eine benachbarte bessere Ecke finden

B sei nun eine zulässige Basis, also

$$0 \leq \bar{x}_B := A_B^{-1}(b - A_N \bar{x}_N) \quad \text{für} \quad \bar{x}_N := 0$$

Benachbarte Ecke: Eine Gleichung $x_i = 0$ mit $i \in N$ aufgeben und x_i vergrößern bis die nächste Ungleichung erreicht wird.



Veränderung der Zielfunktion abhängig von x_N:

$$c^T x = c_B^T x_B + c_N^T x_N = c_B^T A_B^{-1} b + \underbrace{(c_N - A_N^T A_B^{-T} c_B)}_{=:\bar{z}_N}^T x_N$$

Die reduzierten Kosten $\bar{z}_N$ zeigen die Veränderung abh. von x_N an.

Ist $\bar{z}_N \geq 0 \Rightarrow$ Ecke nicht verbesserbar, Optimallösung gefunden.

Sonst wähle im Pricing-Schritt ein $\hat{\imath} \in N$ mit $\bar{z}_{\hat{\imath}} < 0$
$\longrightarrow x_{\hat{\imath}}$ ist die (in die Basis) eintretende Variable und wird vergrößert.

Zur Zulässigkeit muss $x_B = A_B^{-1}(b - A_{\hat{\imath}} x_{\hat{\imath}}) \geq 0$ bleiben,

$$\text{mit } \bar{w} := A_B^{-1} A_{\hat{\imath}} \quad \text{muss} \quad w x_{\hat{\imath}} \leq \bar{x}_B \quad \text{erfüllt sein.}$$

Ist $\bar{w} \leq 0$ darf $x_{\hat{\imath}}$ beliebig groß werden $\Rightarrow$ Problem unbeschränkt.
Sonst wähle im Ratio-Test

$$\hat{\jmath} \in \operatorname*{Argmin}_{j=1,\ldots,m} \left\{ \frac{\bar{x}_{B(j)}}{\bar{w}_j} : \bar{w}_j > 0 \right\}$$

$x_{B(\hat{\jmath})}$ ist die (aus der Basis) austretende Variable.
Sie wird nun im Basisaustauschschritt durch $x_{\hat{\imath}}$ ersetzt,

$$N \leftarrow N \setminus \{\hat{\imath}\} \cup \{B(\hat{\jmath})\}, \; B(\hat{\jmath}) \leftarrow \hat{\imath},$$

und der Algorithmus wird von dieser neuen Ecke aus fortgesetzt.

Das (primale revidierte) Simplex-Verfahren

Input: zulässige Basis B, $\bar{x}_B = A_B^{-1} b$

1. BTRAN: Bestimme $\bar{y} := A_B^{-T} c_B$ durch Lösen von $A_B^T y = c_B$.
2. PRICE: $\bar{z}_N := c_N - A_N^T \bar{y}$, ist $z_N \geq 0$, STOP (Optimallösung), sonst wähle $\hat{\imath} \in N$ mit $z_{\hat{\imath}} < 0$.
3. FTRAN: Bestimme $\bar{w} := A_B^{-1} A_{\hat{\imath}}$ durch Lösen von $A_B w = A_{\hat{\imath}}$.
4. RATIO: Ist $\bar{w} \leq 0$, STOP (Problem unbeschränkt), sonst wähle $\hat{\jmath} \in \operatorname{Argmin}_{j=1,\ldots,m} \left\{ \frac{\bar{x}_{B(j)}}{\bar{w}_j} : \bar{w}_j > 0 \right\}$, $\xi := \frac{\bar{x}_{B(\hat{\jmath})}}{\bar{w}_{\hat{\jmath}}}$
5. Update: $\bar{x}_B := \bar{x}_B - \xi \bar{w}$, $\bar{x}_{\hat{\imath}} := \xi$, $N := (N \setminus \{\hat{\imath}\}) \cup \{B(\hat{\jmath})\}$, $B(\hat{\jmath}) := \hat{\imath}$, GOTO 1.

(Lösung der Glgssysteme nutzt sparsity etc., nicht invertieren!)

Das Paar $(\hat{\jmath}, \hat{\imath})$ wird auch Pivot-Element genannt.

Endlichkeit und Kreisen

In jeder Iteration mit $\xi > 0$ wird der Zielfunktionswert strikt besser, und der Algorithmus besucht diese Ecke nie wieder.

Ist $\xi = 0$, wechselt man nur zu einer anderen Basisdarstellung derselben Ecke (sie liegt auf mehr als $n - m$ Ungleichungen). Bei ungünstiger Wahl in Pricing und Ratio-Test wird die gleiche Basis später wieder besucht $\longrightarrow$ der Algorithmus kreist.

Mit den Auswahlregeln von Bland (wähle aus den erlaubten Variablen jeweils die mit kleinstem Index) wird jede Basis höchstens einmal besucht und der Algorithmus endet nach endlich vielen Iterationen. (In der Praxis nimmt man andere Auswahlregeln und nutzt Bland nur, wenn Kreisen beobachtet wird.)

Hat ein LP zwei unterschiedliche Basen, die den gleichen Punkt beschreiben, nennt man das LP und derartige Basen degeneriert oder entartet. Für entartete LPs kann der Simplex-Algorithmus sehr langsam sein, dann sind Innere-Punkte-Verfahren (s. dort) meist besser.

Finden einer zulässigen Ausgangsbasis

Die 2-Phasen-Methode

Löse in Phase 1 das Hilfsproblem (o.B.d.A. $b \geq 0$)

$$\begin{array}{ll} \min & \mathbf{1}^T s \\ \text{s.t.} & Ax + Is = b \\ & x \geq 0, s \geq 0 \end{array}$$

zul. Basis: die Spalten von s

Ist Optimalwert=0 $\rightarrow$ zul. Basis des Originalproblems gefunden, löse diese in Phase 2.

Liefert überprüfbaren Nachweis (Zertifikat) für (Un-)Zulässigkeit!

Die Big-M Methode (o.B.d.A. $b \geq 0$)

$$\begin{array}{ll} \min & c^T x + M\mathbf{1}^T s \\ \text{s.t.} & Ax + Is = b \\ & x \geq 0, s \geq 0 \end{array}$$

zul. Basis: die Spalten von s

Ist $M > 0$ groß genug, werden die $s_i = 0$

Vorteil: sucht gleich gute Basis
Nachteil: Wie groß muss M sein?

In Standard-Software muss man sich darum nicht kümmern!

Mit dem Simplex-Algorithmus zeigt man

Satz (Hauptsatz der Linearen Optimierung)

Hat ein LP in Standardform einen endlichen Optimalwert, so wird dieser auch in einer Ecke angenommen.

Inhaltsübersicht

2.3 Dualität

Mit jedem konvexen Minimierungsproblem löst man automatisch ein verwandtes konvexes Maximierungsproblem, das als die Bestimmung einer besten Schranke für den Optimalwert des Originalproblems interpretiert werden kann.

Die duale Optimallösung (wenn sie existiert) gibt wichtige Zusatzinformation über das Originalproblem.

Für Lineare Optimierung ist das besonders einfach und anschaulich.

Untere Schranken, algebraisch

Eine Ungleichung $a^T x \geq \alpha$ heißt gültig für ein Optimierungsproblem mit zulässiger Menge $\mathcal{X}$, wenn sie für alle $x \in \mathcal{X}$ erfüllt ist.

Betrachte $\mathcal{X} := \{x : Ax \geq b\}$ mit $A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}$.

Nichtneg. Linearkombination der Zeilen $Ax \geq b$ und r.S. vergrößern

$$\begin{array}{rrll} a_1^T x & \geq & b_1 & \mid \cdot y_1 (\geq 0) \\ & \vdots & & \\ a_m^T x & \geq & b_m & \mid \cdot y_m (\geq 0) \\ 0^T x & \geq & -1 & \mid \cdot \eta (\geq 0) \\ \hline \oplus: \quad (y^T A)x & \geq & y^T b - \eta & \end{array}$$

liefert eine gültige Ungleichung für $\mathcal{X}$ (und man erhält alle so!).

Untere Schranken, algebraisch

Eine Ungleichung $a^T x \geq \alpha$ heißt gültig für ein Optimierungsproblem mit zulässiger Menge $\mathcal{X}$, wenn sie für alle $x \in \mathcal{X}$ erfüllt ist.

Betrachte $\mathcal{X} := \{x : Ax \geq b\}$ mit $A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}$.

Nichtneg. Linearkombination der Zeilen $Ax \geq b$ und r.S. vergrößern

$$\begin{array}{rrll} a_1^T x & \geq & b_1 & \mid \cdot y_1 (\geq 0) \\ & \vdots & & \\ a_m^T x & \geq & b_m & \mid \cdot y_m (\geq 0) \\ 0^T x & \geq & -1 & \mid \cdot \eta (\geq 0) \\ \hline \oplus: \quad (y^T A)x & \geq & y^T b - \eta & \end{array}$$

liefert eine gültige Ungleichung für $\mathcal{X}$ (und man erhält alle so!).

Für ein $\bar{y} \geq 0$ mit $A^T \bar{y} = c$ folgt $\min\{c^T x : Ax \geq b\} \geq b^T \bar{y}$.
Die größte untere Schranke?

Untere Schranken, algebraisch

Eine Ungleichung $a^T x \geq \alpha$ heißt gültig für ein Optimierungsproblem mit zulässiger Menge $\mathcal{X}$, wenn sie für alle $x \in \mathcal{X}$ erfüllt ist.

Betrachte $\mathcal{X} := \{x : Ax \geq b\}$ mit $A = \begin{bmatrix} a_1^T \\ \vdots \\ a_m^T \end{bmatrix}$.

Nichtneg. Linearkombination der Zeilen $Ax \geq b$ und r.S. vergrößern

$$\begin{array}{rrll} a_1^T x & \geq & b_1 & \mid \cdot y_1 (\geq 0) \\ & \vdots & & \\ a_m^T x & \geq & b_m & \mid \cdot y_m (\geq 0) \\ 0^T x & \geq & -1 & \mid \cdot \eta (\geq 0) \\ \hline \oplus: \quad (y^T A)x & \geq & y^T b - \eta & \end{array}$$

liefert eine gültige Ungleichung für $\mathcal{X}$ (und man erhält alle so!).

Für ein $\bar{y} \geq 0$ mit $A^T \bar{y} = c$ folgt $\quad \min\{c^T x : Ax \geq b\} \geq b^T \bar{y}$.
Die größte untere Schranke? $\quad \max\{b^T y : A^T y = c, y \geq 0\}$

Das Duale in Normalform

Betrachte $\min c^T x$ s.t. $Ax = b,\ x \geq 0.$

Erlaubte Linearkombination der Zeilen $Ax = b$ und $Ix \geq 0$

$$\begin{array}{rrcll} & a_1^T x & = & b_1 & \mid \cdot y_1 (\in \mathbb{R}) \\ & \vdots & & & \\ & a_m^T x & = & b_m & \mid \cdot y_m (\in \mathbb{R}) \\ & e_1^T x & \geq & 0 & \mid \cdot z_1 (\geq 0) \\ & \vdots & & & \\ & e_n^T x & \geq & 0 & \mid \cdot z_n (\geq 0) \\ \hline \oplus: & (y^T A + z^T I) x & \geq & y^T b & \end{array}$$

Das Duale in Normalform

Betrachte $\min c^T x$ s.t. $Ax = b,\ x \geq 0.$

Erlaubte Linearkombination der Zeilen $Ax = b$ und $Ix \geq 0$

$$\begin{array}{rrcll} & a_1^T x & = & b_1 & \mid \cdot y_1 (\in \mathbb{R}) \\ & \vdots & & & \\ & a_m^T x & = & b_m & \mid \cdot y_m (\in \mathbb{R}) \\ & e_1^T x & \geq & 0 & \mid \cdot z_1 (\geq 0) \\ & \vdots & & & \\ & e_n^T x & \geq & 0 & \mid \cdot z_n (\geq 0) \\ \hline \oplus: & (y^T A + z^T I)x & \geq & y^T b & \end{array}$$

Die beste untere Schranke liefert das Duale LP in Normalform:

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array} \qquad \left[\begin{array}{ll} \Leftrightarrow & \max \quad b^T y \\ & \text{s.t.} \quad A^T y \geq 0 \\ & \qquad y \in \mathbb{R}^m, \end{array} \right]$$

y und z sind die Dualvariablen.

Dualisierungsregeln

Nach dem gleichen Muster ergibt sich

$$
\begin{array}{ccc}
\min c^T x & \leftrightarrow & \max b^T y \\
Ax \geq b & \leftrightarrow & y \geq 0 \\
Ax \leq b & \leftrightarrow & y \leq 0 \\
Ax = b & \leftrightarrow & y \text{ frei} \\
x \geq 0 & \leftrightarrow & A^T y \leq c \\
x \leq 0 & \leftrightarrow & A^T y \geq c \\
x \text{ frei} & \leftrightarrow & A^T y = c
\end{array}
$$

Insbesondere ist das Duale des Dualen LPs wieder das Primale LP.

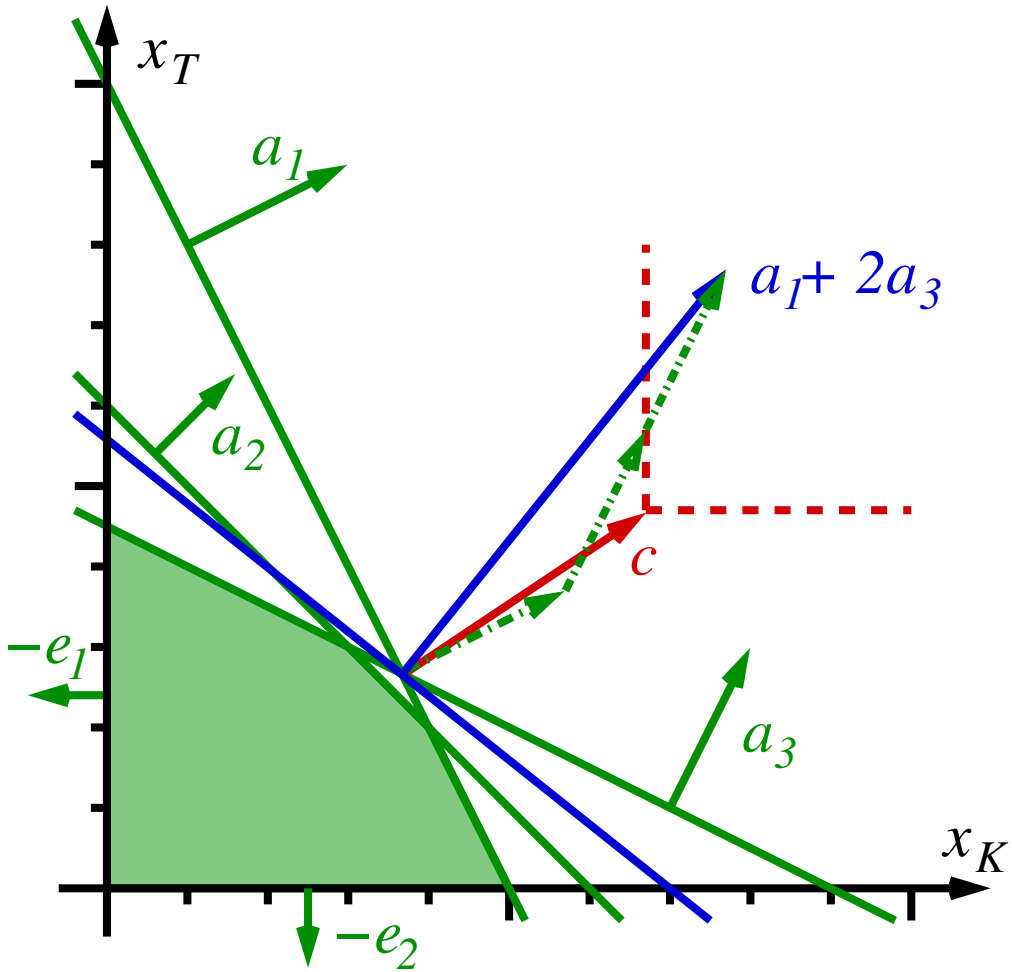
Mozart: das Duale geometrisch

$$(P_M) \quad \begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \end{array}$$

$$(D_M) \quad \begin{array}{ll} \min & b^T y \\ \text{s.t.} & A^T y \geq c \\ & y \geq 0 \end{array}$$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \; b = \begin{bmatrix} 10 \\ 6 \\ 9 \end{bmatrix},$$

$$c = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$



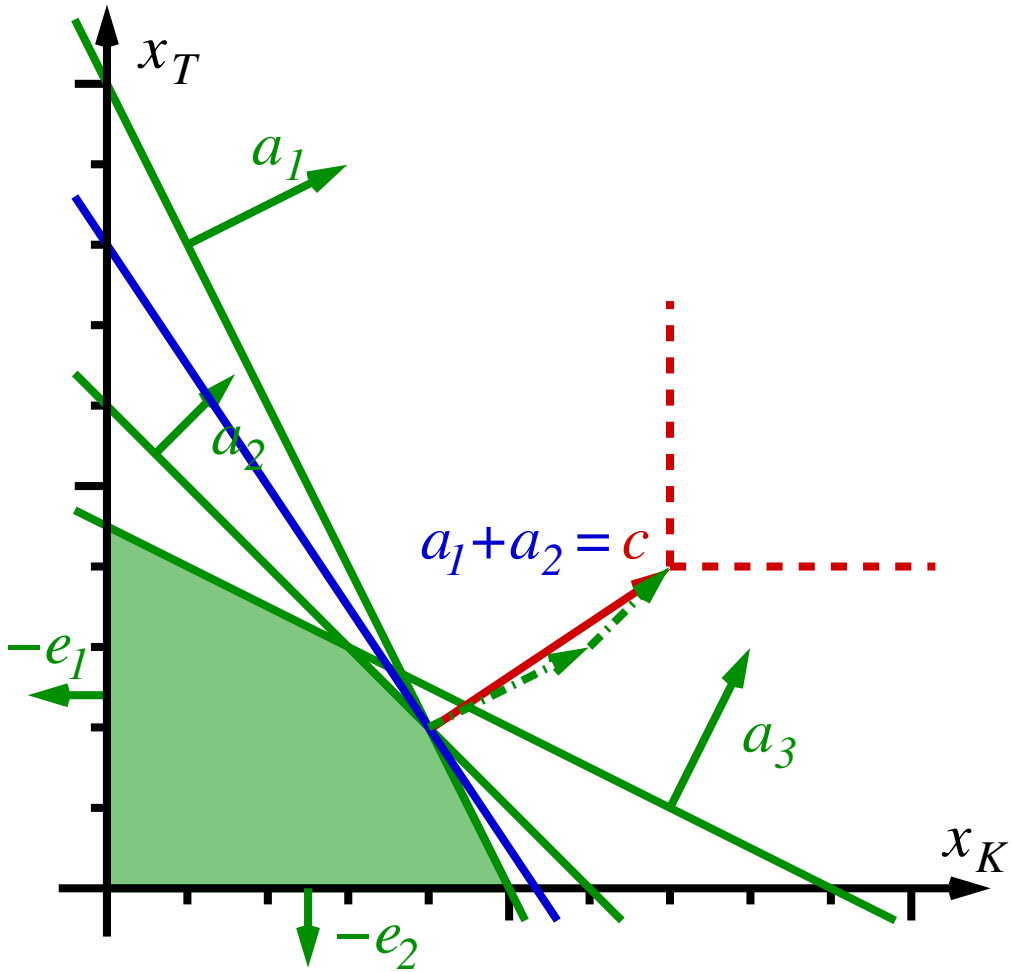
$$\bar{y} = \begin{bmatrix} 1 \\ 0 \\ 2 \end{bmatrix} \rightarrow A^T \bar{y} = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \geq c, \; b^T \bar{y} = 28$$

Mozart: das Duale optimal

$$(P_M) \quad \begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax \leq b \\ & x \geq 0 \end{array}$$

$$(D_M) \quad \begin{array}{ll} \min & b^T y \\ \text{s.t.} & A^T y \geq c \\ & y \geq 0 \end{array}$$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix}, \; b = \begin{bmatrix} 10 \\ 6 \\ 9 \end{bmatrix}, \; c = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$



$$\bar{y}^* = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \rightarrow A^T \bar{y}^* = \begin{bmatrix} 3 \\ 2 \end{bmatrix} = c, \; b^T \bar{y}^* = 16, \qquad \bar{x}^* = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

Wo ist das Duale im Simplex-Algorithmus?

Input: zulässige Basis B, $\bar{x}_B = A_B^{-1} b$

1. BTRAN: Bestimme $\bar{y} := A_B^{-T} c_B$ durch Lösen von $A_B^T y = c_B$.
2. PRICE: $\bar{z}_N := c_N - A_N^T \bar{y}$, ist $\bar{z}_N \geq 0$, STOP (Optimallösung), sonst wähle $\hat{\imath} \in N$ mit $\bar{z}_{\hat{\imath}} < 0$.

$\vdots$

$$(P) \quad \begin{array}{l} \min c^T x \\ \text{s.t. } [A_B, A_N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b \\ x \geq 0 \end{array} \qquad (D) \quad \begin{array}{l} \max b^T y \\ \text{s.t. } \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} y + \begin{bmatrix} z_B \\ z_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \\ y \in \mathbb{R}^m, z \geq 0 \end{array}$$

1. Schritt: Setze $\bar{z}_B = 0$ und löse 1. duale Block-Zeile $\rightarrow \bar{y}$
2. Schritt: Bestimme $\bar{z}_N$ aus der 2. dualen Block-Zeile

Wo ist das Duale im Simplex-Algorithmus?

Input: zulässige Basis B, $\bar{x}_B = A_B^{-1} b$

1. BTRAN: Bestimme $\bar{y} := A_B^{-T} c_B$ durch Lösen von $A_B^T y = c_B$.
2. PRICE: $\bar{z}_N := c_N - A_N^T \bar{y}$, ist $\bar{z}_N \geq 0$, STOP (Optimallösung), sonst wähle $\hat{\imath} \in N$ mit $\bar{z}_{\hat{\imath}} < 0$.

$\vdots$

$$(P) \quad \begin{array}{l} \min c^T x \\ \text{s.t. } [A_B, A_N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b \\ x \geq 0 \end{array} \qquad (D) \quad \begin{array}{l} \max b^T y \\ \text{s.t. } \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} y + \begin{bmatrix} z_B \\ z_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \\ y \in \mathbb{R}^m, z \geq 0 \end{array}$$

1. Schritt: Setze $\bar{z}_B = 0$ und löse 1. duale Block-Zeile $\rightarrow \bar{y}$
2. Schritt: Bestimme $\bar{z}_N$ aus der 2. dualen Block-Zeile

Ist die duale Lösung zulässig ($\bar{z}_N \geq 0$), ergibt das die Schranke

$$\min_{x \in \mathcal{X}} c^T x \;\geq\; \bar{y}^T b \;=\; c_B^T A_B^{-1} \;=\; c_B \bar{x}_B \;=\; c^T \bar{x} \qquad [\Rightarrow \text{OL}]$$

Wo ist das Duale im Simplex-Algorithmus?

Input: zulässige Basis B, $\bar{x}_B = A_B^{-1} b$

1. BTRAN: Bestimme $\bar{y} := A_B^{-T} c_B$ durch Lösen von $A_B^T y = c_B$.
2. PRICE: $\bar{z}_N := c_N - A_N^T \bar{y}$, ist $\bar{z}_N \geq 0$, STOP (Optimallösung), sonst wähle $\hat{\imath} \in N$ mit $\bar{z}_{\hat{\imath}} < 0$.

$\vdots$

$$(P) \quad \begin{array}{l} \min c^T x \\ \text{s.t. } [A_B, A_N] \begin{bmatrix} x_B \\ x_N \end{bmatrix} = b \\ x \geq 0 \end{array} \qquad (D) \quad \begin{array}{l} \max b^T y \\ \text{s.t. } \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} y + \begin{bmatrix} z_B \\ z_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix} \\ y \in \mathbb{R}^m, z \geq 0 \end{array}$$

1. Schritt: Setze $\bar{z}_B = 0$ und löse 1. duale Block-Zeile $\rightarrow \bar{y}$
2. Schritt: Bestimme $\bar{z}_N$ aus der 2. dualen Block-Zeile

Ist die duale Lösung zulässig ($\bar{z}_N \geq 0$), ergibt das die Schranke

$$\min_{x \in \mathcal{X}} c^T x \;\geq\; \bar{y}^T b \;=\; c_B^T A_B^{-1} \;=\; c_B \bar{x}_B \;=\; c^T \bar{x} \qquad [\Rightarrow \text{OL}]$$

Beachte: PRICE sucht einfach ein dual unzulässiges z_i mit $i \in N$!

Das duale Simplex-Verfahren

$$(D)\ \max\ b^T y \text{ s.t. } \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} y + \begin{bmatrix} z_B \\ z_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix},\ y \in \mathbb{R}^m, z \geq 0$$

Dual zulässige Basis: $z_B = 0$, $y = A_B^{-T}(c_B - z_B)$, $z_N = c_N - A_N^T y \geq 0$

Kostenänderung in Abh. von z_B: $y^T b = c_B^T c_B - z_B^T \underbrace{A_B^{-1} b}_{=:\bar{x}_B}$

Das duale Simplex-Verfahren

$$(D)\ \max\ b^T y \text{ s.t. } \begin{bmatrix} A_B^T \\ A_N^T \end{bmatrix} y + \begin{bmatrix} z_B \\ z_N \end{bmatrix} = \begin{bmatrix} c_B \\ c_N \end{bmatrix},\ y \in \mathbb{R}^m, z \geq 0$$

Dual zulässige Basis: $z_B = 0$, $y = A_B^{-T}(c_B - z_B)$, $z_N = c_N - A_N^T y \geq 0$
Kostenänderung in Abh. von z_B: $y^T b = c_B^T c_B - z_B^T \underbrace{A_B^{-1} b}_{=:\bar{x}_B}$

Input: dual zulässige Basis B, $\bar{y} = A_B^{-T} c_b$, $\bar{z}_N = c_N - A_N^T \bar{y} \geq 0$

1. FTRAN: Bestimme $\bar{x}_B := A_B^{-1} b$ durch Lösen von $A_B x_B = b$.
2. PRICE: Ist $\bar{x}_B \geq 0$, STOP (Optimallösung),
   sonst wähle $\hat{\jmath} \in \{1, \ldots, m\}$ mit $\bar{x}_{B(\hat{\jmath})} < 0$.
3. BTRAN: Bestimme $\bar{w} := A_B^{-T} e_{\hat{\jmath}}$ und berechne $\bar{\omega}_N := -A_N^T \bar{w}$
4. RATIO: Ist $\bar{\omega}_N \leq 0$ STOP (Duales unbeschränkt), sonst wähle
   $\hat{\imath} \in \operatorname{Argmin}_{i \in N}\{\frac{\bar{z}_i}{\bar{\omega}_i} : \bar{\omega}_i > 0\}$, $\zeta := \frac{\bar{z}_{\hat{\imath}}}{\bar{\omega}_{\hat{\imath}}}$
5. Update: $\bar{y} := \bar{y} - \zeta \bar{w}$, $\bar{z}_N := \bar{z}_N - \zeta \bar{\omega}$, $\bar{z}_{B(\hat{\jmath})} := \zeta$
   $N := (N \setminus \{\hat{\imath}\}) \cup \{B(\hat{\jmath})\}$, $B(\hat{\jmath}) := \hat{\imath}$, GOTO 1.

Schwache und Starke Dualität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Ist x primal zulässig und (y, z) dual zulässig, dann gilt

$$c^T x - b^T y = (A^T y + z)^T x - b^T y = (Ax - b)^T y + z^T x = z^T x \geq 0$$

Schwache und Starke Dualität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Ist x primal zulässig und (y, z) dual zulässig, dann gilt

$$c^T x - b^T y = (A^T y + z)^T x - b^T y = (Ax - b)^T y + z^T x = z^T x \geq 0$$

Schwache Dualität: Der Wert jeder zulässigen Lösung beschränkt den Optimalwert des jeweils Dualen. [gilt immer!]

$\Rightarrow$ *Ist ein LP unbeschränkt, ist dessen Duales unzulässig.*

Schwache und Starke Dualität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Ist x primal zulässig und (y, z) dual zulässig, dann gilt

$$c^T x - b^T y = (A^T y + z)^T x - b^T y = (Ax - b)^T y + z^T x = z^T x \geq 0$$

Schwache Dualität: Der Wert jeder zulässigen Lösung beschränkt den Optimalwert des jeweils Dualen. [gilt immer!]

$\Rightarrow$ *Ist ein LP unbeschränkt, ist dessen Duales unzulässig.*

Dualitätslücke: Differenz aus primalem und dualem Optimalwert.

Schwache und Starke Dualität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Ist x primal zulässig und (y, z) dual zulässig, dann gilt

$$c^T x - b^T y = (A^T y + z)^T x - b^T y = (Ax - b)^T y + z^T x = z^T x \geq 0$$

Schwache Dualität: Der Wert jeder zulässigen Lösung beschränkt den Optimalwert des jeweils Dualen. [gilt immer!]

$\Rightarrow$ *Ist ein LP unbeschränkt, ist dessen Duales unzulässig.*

Dualitätslücke: Differenz aus primalem und dualem Optimalwert.

Starke Dualität: Primaler und dualer Optimalwert sind gleich, (beide werden angenommen). [gilt i.A. nicht für Konv. Opt.]

Schwache und Starke Dualität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Ist x primal zulässig und (y, z) dual zulässig, dann gilt

$$c^T x - b^T y = (A^T y + z)^T x - b^T y = (Ax - b)^T y + z^T x = z^T x \geq 0$$

Schwache Dualität: Der Wert jeder zulässigen Lösung beschränkt den Optimalwert des jeweils Dualen. [gilt immer!]
$\Rightarrow$ *Ist ein LP unbeschränkt, ist dessen Duales unzulässig.*
Dualitätslücke: Differenz aus primalem und dualem Optimalwert.
Starke Dualität: Primaler und dualer Optimalwert sind gleich, (beide werden angenommen). [gilt i.A. nicht für Konv. Opt.]

Satz (Starke Dualität in der Linearen Optimierung)

Hat ein LP einen endlichen Optimalwert, so wird dieser sowohl primal als auch dual angenommen.

Beweis: Simplex-Alg. mit Auswahlregeln von Bland. □

Es können beide unzulässig sein!

$$(P_u) \quad \begin{array}{ll} \max & x_1 \\ \text{s.t.} & x_1 - x_2 \leq -1 \\ & -x_1 + x_2 \leq 0 \\ & x_1 \geq 0, x_2 \geq 0 \end{array} \qquad (D_u) \quad \begin{array}{ll} \max & -y_1 \\ \text{s.t.} & y_1 - y_2 \geq 1 \\ & -y_1 + y_2 \geq 0 \\ & y_1 \geq 0, y_2 \geq 0 \end{array}$$

(P_u) Es kann nicht $x_2 \geq 1 + x_1$ und $x_2 \leq x_1$ sein.

(D_u) Es kann nicht $y_1 \geq 1 + y_2$ und $y_1 \leq y_2$ sein.

Anwendung: Spieltheorie

2-Personen Matrix-Spiel: Für eine gegebene Auszahlungsmatrix $M \in \mathbb{R}^{m \times n}$ wählt Spieler 1 (S1) eine Zeile i und Spieler 2 (S2) eine Spalte j. Spieler 1 zahlt an Spieler 2 den Betrag M_{ij}.

Anwendung: Spieltheorie

2-Personen Matrix-Spiel: Für eine gegebene Auszahlungsmatrix $M \in \mathbb{R}^{m \times n}$ wählt Spieler 1 (S1) eine Zeile i und Spieler 2 (S2) eine Spalte j. Spieler 1 zahlt an Spieler 2 den Betrag M_{ij}.

Wenn S2 weiß, dass S1 Zeile i wählt, wählt S2 $\max_{j=1,\ldots,n}(A^T e_i)_j$.

Anwendung: Spieltheorie

2-Personen Matrix-Spiel: Für eine gegebene Auszahlungsmatrix $M \in \mathbb{R}^{m \times n}$ wählt Spieler 1 (S1) eine Zeile i und Spieler 2 (S2) eine Spalte j. Spieler 1 zahlt an Spieler 2 den Betrag M_{ij}.

Wenn S2 weiß, dass S1 Zeile i wählt, wählt S2 $\max_{j=1,\ldots,n}(A^T e_i)_j$.

S1 bestimmt eine optimale Wahrscheinlichkeits-Verteilung für eine zufällige Zeilenwahl, um nicht so leicht voraussagbar zu sein:

$$\begin{array}{ll} \min & \max_{j=1,\ldots,n}(M^T x)_j \\ \text{s.t.} & \mathbf{1}^T x = 1 \\ & x \geq 0 \end{array}$$

Anwendung: Spieltheorie

2-Personen Matrix-Spiel: Für eine gegebene Auszahlungsmatrix $M \in \mathbb{R}^{m \times n}$ wählt Spieler 1 (S1) eine Zeile i und Spieler 2 (S2) eine Spalte j. Spieler 1 zahlt an Spieler 2 den Betrag M_{ij}.

Wenn S2 weiß, dass S1 Zeile i wählt, wählt S2 $\max_{j=1,\dots,n}(A^T e_i)_j$.

S1 bestimmt eine optimale Wahrscheinlichkeits-Verteilung für eine zufällige Zeilenwahl, um nicht so leicht voraussagbar zu sein:

$$\begin{array}{ll} \min & \max_{j=1,\dots,n}(M^T x)_j \\ \text{s.t.} & \mathbf{1}^T x = 1 \\ & x \geq 0 \end{array} \quad \Leftrightarrow (P_1) \quad \begin{array}{ll} \min & s \\ \text{s.t.} & s \geq (M^T x)_j \quad j = 1, \dots, n \\ & \mathbf{1}^T x = 1 \\ & x \geq 0, s \in \mathbb{R} \end{array}$$

Anwendung: Spieltheorie

2-Personen Matrix-Spiel: Für eine gegebene Auszahlungsmatrix $M \in \mathbb{R}^{m \times n}$ wählt Spieler 1 (S1) eine Zeile i und Spieler 2 (S2) eine Spalte j. Spieler 1 zahlt an Spieler 2 den Betrag M_{ij}.

Wenn S2 weiß, dass S1 Zeile i wählt, wählt S2 $\max_{j=1,\dots,n}(A^T e_i)_j$.

S1 bestimmt eine optimale Wahrscheinlichkeits-Verteilung für eine zufällige Zeilenwahl, um nicht so leicht voraussagbar zu sein:

$$\begin{array}{ll} \min & \max_{j=1,\dots,n}(M^T x)_j \\ \text{s.t.} & \mathbf{1}^T x = 1 \\ & x \geq 0 \end{array} \quad \Leftrightarrow (P_1) \quad \begin{array}{ll} \min & s \\ \text{s.t.} & s \geq (M^T x)_j \quad j = 1, \dots, n \\ & \mathbf{1}^T x = 1 \\ & x \geq 0, s \in \mathbb{R} \end{array}$$

S2 tut das gleiche für sich

$$\begin{array}{ll} \max & \min_{i=1,\dots,m}(My)_i \\ \text{s.t.} & \mathbf{1}^T y = 1 \\ & y \geq 0 \end{array} \quad \Leftrightarrow (P_2) \quad \begin{array}{ll} \max & t \\ \text{s.t.} & t \leq (My)_j \quad i = 1, \dots, m \\ & \mathbf{1}^T y = 1 \\ & y \geq 0, t \in \mathbb{R} \end{array}$$

(P_2) ist das Duale zu (P_1), beide sind zul., also ist $v(P_1) = v(P_2)$.
Man muss die Strategie des Gegners gar nicht kennen!

Inhaltsübersicht

Komplementarität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

$c^T x - b^T y = z^T x \geq 0$ gilt für bel. primal und dual zul. Lösungen.

Satz (vom komplementären Schlupf)

Primal und dual zulässige Lösungen $\bar{x}$ und $(\bar{y}, \bar{z})$ sind genau dann optimal, wenn $\bar{x}_i \bar{z}_i = 0$ für $i = 1, \ldots, n$.

Komplementarität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

$c^T x - b^T y = z^T x \geq 0$ gilt für bel. primal und dual zul. Lösungen.

Satz (vom komplementären Schlupf)

Primal und dual zulässige Lösungen $\bar{x}$ und $(\bar{y}, \bar{z})$ sind genau dann optimal, wenn $\bar{x}_i \bar{z}_i = 0$ für $i = 1, \ldots, n$.

Eine Ungleichung heißt aktiv in einem Punkt, wenn sie in dem Punkt mit Gleichheit erfüllt ist, sonst nicht aktiv oder inaktiv.

Komplementarität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

$c^T x - b^T y = z^T x \geq 0$ gilt für bel. primal und dual zul. Lösungen.

Satz (vom komplementären Schlupf)

Primal und dual zulässige Lösungen $\bar{x}$ und $(\bar{y}, \bar{z})$ sind genau dann optimal, wenn $\bar{x}_i \bar{z}_i = 0$ für $i = 1, \ldots, n$.

Eine Ungleichung heißt aktiv in einem Punkt, wenn sie in dem Punkt mit Gleichheit erfüllt ist, sonst nicht aktiv oder inaktiv.
Folgerungen:

- Ist in einer OL eine Variable in x^* oder $z^* > 0$, ist deren duale Ungleichung in jeder OL des jeweiligen dualen aktiv!

Komplementarität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

$c^T x - b^T y = z^T x \geq 0$ gilt für bel. primal und dual zul. Lösungen.

Satz (vom komplementären Schlupf)

Primal und dual zulässige Lösungen $\bar{x}$ und $(\bar{y}, \bar{z})$ sind genau dann optimal, wenn $\bar{x}_i \bar{z}_i = 0$ für $i = 1, \ldots, n$.

Eine Ungleichung heißt aktiv in einem Punkt, wenn sie in dem Punkt mit Gleichheit erfüllt ist, sonst nicht aktiv oder inaktiv.
Folgerungen:

- Ist in einer OL eine Variable in x^* oder $z^* > 0$, ist deren duale Ungleichung in jeder OL des jeweiligen dualen aktiv!
- Ist in einer OL eine Unglg. des LPs inaktiv, ist deren Dualvariable in jeder OL des jeweiligen dualen gleich Null!

Komplementarität

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

$c^T x - b^T y = z^T x \geq 0$ gilt für bel. primal und dual zul. Lösungen.

Satz (vom komplementären Schlupf)

Primal und dual zulässige Lösungen $\bar{x}$ und $(\bar{y}, \bar{z})$ sind genau dann optimal, wenn $\bar{x}_i \bar{z}_i = 0$ für $i = 1, \ldots, n$.

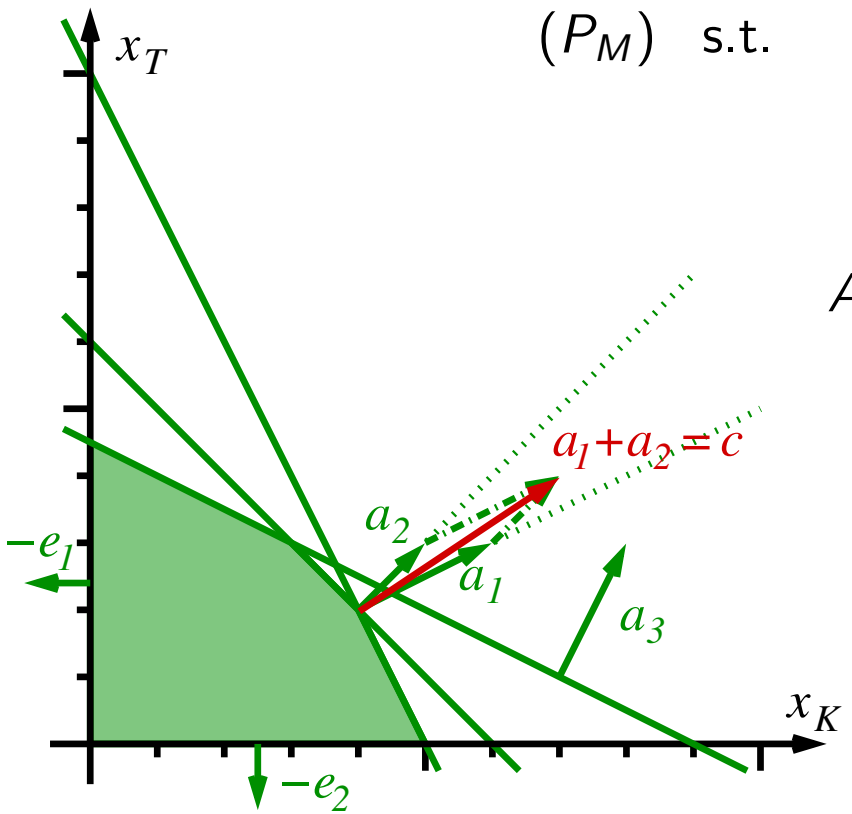
Eine Ungleichung heißt aktiv in einem Punkt, wenn sie in dem Punkt mit Gleichheit erfüllt ist, sonst nicht aktiv oder inaktiv.
Folgerungen:

- Ist in einer OL eine Variable in x^* oder $z^* > 0$, ist deren duale Ungleichung in jeder OL des jeweiligen dualen aktiv!
- Ist in einer OL eine Unglg. des LPs inaktiv, ist deren Dualvariable in jeder OL des jeweiligen dualen gleich Null!

Tatsächlich gibt die Größe der Dualvariablen wesentliche Information über den Einfluss der Unglg. auf den Optimalwert ...

Geometrische Interpretation der Komplementarität

$$(P_M)\quad \begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \qquad (D_M)\quad \begin{array}{ll} \min & b^T y \\ \text{s.t.} & A^T y \ge c \\ & y \ge 0 \end{array}$$



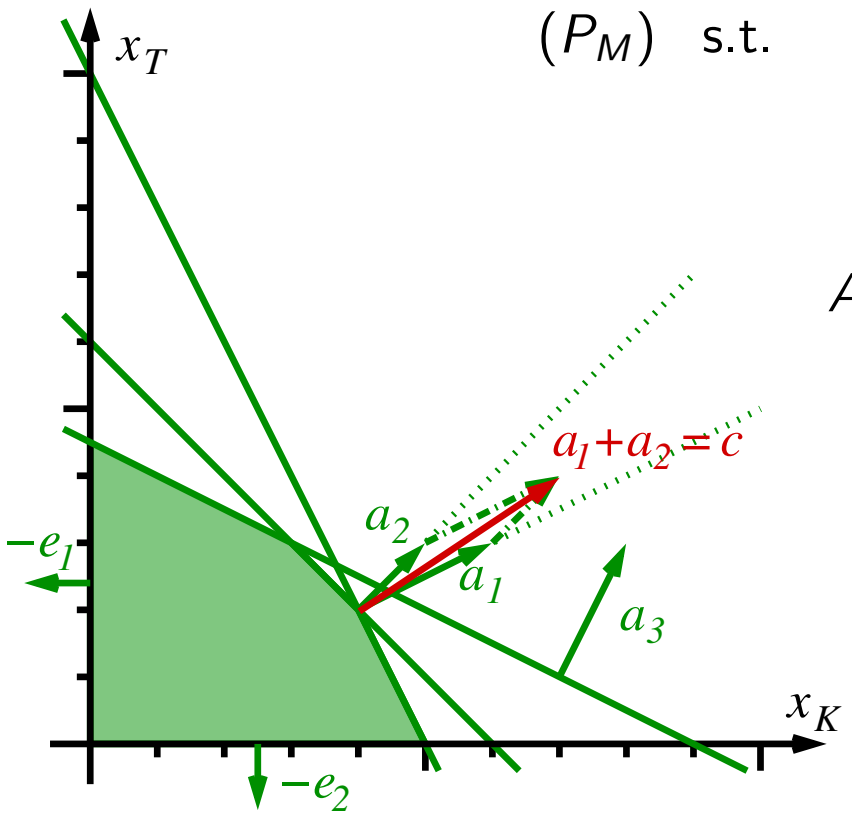
$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix},\ b = \begin{bmatrix} 10 \\ 6 \\ 9 \end{bmatrix},\ c = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Inaktive Ungleichungen können nicht zur besten Schranke beitragen, die aktiven Restr. „halten“ die OL. (Gleichungen sind immer aktiv!)

$y_j \|a_j\|$ (bzw. z_i) ist „Kraftanteil“ von Restriktion i im „Kräftegleichgewicht“.

Geometrische Interpretation der Komplementarität

$$(P_M)\quad \begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax \le b \\ & x \ge 0 \end{array} \qquad (D_M)\quad \begin{array}{ll} \min & b^T y \\ \text{s.t.} & A^T y \ge c \\ & y \ge 0 \end{array}$$



$$A = \begin{bmatrix} 2 & 1 \\ 1 & 1 \\ 1 & 2 \end{bmatrix},\ b = \begin{bmatrix} 10 \\ 6 \\ 9 \end{bmatrix},\ c = \begin{bmatrix} 3 \\ 2 \end{bmatrix}$$

Inaktive Ungleichungen können nicht zur besten Schranke beitragen, die aktiven Restr. „halten“ die OL. (Gleichungen sind immer aktiv!)

$y_j \|a_j\|$ (bzw. z_i) ist „Kraftanteil“ von Restriktion i im „Kräftegleichgewicht“.

„Der Gradient der Zielfunktion liegt im Kegel der Gradienten der aktiven Restriktionen.“

(Die Vorzeichen sind je nach Richtungen anzupassen!)

Sensitivitätsanalyse

Wie sensibel reagiert die OL auf Änderungen in welchen Daten?

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Wie weit ist b veränderbar bei gleicher optimaler Basis B?

$$x_B^* = A_B^{-1} b \geq 0, \quad y^* = A_B^{-T} c_B, \quad z_N^* = c_N - A_N^T y^* \geq 0$$

Sensitivitätsanalyse

Wie sensibel reagiert die OL auf Änderungen in welchen Daten?

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Wie weit ist b veränderbar bei gleicher optimaler Basis B?

$$x_B^* = A_B^{-1} b \geq 0, \quad y^* = A_B^{-T} c_B, \quad z_N^* = c_N - A_N^T y^* \geq 0$$

$b(t) := b + t\Delta b$ erhält duale Zul. für $t \in \mathbb{R}$, primale aber nur wenn

$$x_B(t) = x_B^* + t \underbrace{A_B^{-1} \Delta b}_{=:\Delta x_B} \geq 0 \quad \Leftrightarrow \quad \max_{[\Delta x_B]_i > 0} -\frac{[x_B^*]_i}{[\Delta x_B]_i} \leq t \leq \max_{[\Delta x_B]_i < 0} -\frac{[x_B^*]_i}{[\Delta x_B]_i}$$

Sensitivitätsanalyse

Wie sensibel reagiert die OL auf Änderungen in welchen Daten?

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Wie weit ist b veränderbar bei gleicher optimaler Basis B?

$$x_B^* = A_B^{-1} b \geq 0, \quad y^* = A_B^{-T} c_B, \quad z_N^* = c_N - A_N^T y^* \geq 0$$

$b(t) := b + t\Delta b$ erhält duale Zul. für $t \in \mathbb{R}$, primale aber nur wenn

$$x_B(t) = x_B^* + t \underbrace{A_B^{-1} \Delta b}_{=:\Delta x_B} \geq 0 \quad \Leftrightarrow \quad \max_{[\Delta x_B]_i > 0} -\frac{[x_B^*]_i}{[\Delta x_B]_i} \leq t \leq \max_{[\Delta x_B]_i < 0} -\frac{[x_B^*]_i}{[\Delta x_B]_i}$$

Für diese t ist der Optimalwert $= b^T y^* + t\Delta b^T y^*$ (Kompl.!),
sonst ist dieser Wert auf jeden Fall eine untere Schranke.

Sensitivitätsanalyse

Wie sensibel reagiert die OL auf Änderungen in welchen Daten?

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array}$$

Wie weit ist b veränderbar bei gleicher optimaler Basis B?

$$x_B^* = A_B^{-1} b \geq 0, \quad y^* = A_B^{-T} c_B, \quad z_N^* = c_N - A_N^T y^* \geq 0$$

$b(t) := b + t\Delta b$ erhält duale Zul. für $t \in \mathbb{R}$, primale aber nur wenn

$$x_B(t) = x_B^* + t \underbrace{A_B^{-1} \Delta b}_{=:\Delta x_B} \geq 0 \quad \Leftrightarrow \quad \max_{[\Delta x_B]_i > 0} -\frac{[x_B^*]_i}{[\Delta x_B]_i} \leq t \leq \max_{[\Delta x_B]_i < 0} -\frac{[x_B^*]_i}{[\Delta x_B]_i}$$

Für diese t ist der Optimalwert $= b^T y^* + t\Delta b^T y^*$ (Kompl.!), sonst ist dieser Wert auf jeden Fall eine untere Schranke.

Für $\Delta b = e_j$ beziffert y_j^* die marginalen Kosten der Veränderung, $\rightarrow$ bei Ressourcennebenbed. heißen die y_j^* oft Schattenpreise.

Sensitivitätsanalyse für die Kostenkoeffizienten

Wie weit ist c veränderbar bei gleicher optimaler Basis B?

$$x_B^* = A_B^{-1} b \geq 0, \quad y^* = A_B^{-T} c_B, \quad z_N^* = c_N - A_N^T y^* \geq 0$$

$c(t) := c + t\Delta c$ erhält primale Zul. für $t \in \mathbb{R}$, duale aber nur wenn

$$z_N(t) = z_N^* + t \underbrace{(\Delta c_N - A_N^T A_B^{-T} \Delta c_B)}_{=:\Delta z_N} \geq 0$$

$$\Leftrightarrow \quad \max_{[\Delta z_N]_i > 0} -\frac{[z_N^*]_i}{[\Delta z_N]_i} \leq t \leq \max_{[\Delta z_N]_i < 0} -\frac{[z_N^*]_i}{[\Delta z_N]_i}$$

Für diese t ist der Optimalwert $= c^T x^* + t\Delta c^T x^*$ (Kompl.), sonst ist dieser Wert eine obere Schranke.

Inhaltsübersicht

2.5 Spaltengenerierung

Was tun, wenn es zu viele Variablen gibt um das LP aufzustellen?

Das Simpex-Verfahren benötigt je Iteration nur die Basis-Variablen ($x_N = 0!$), solange eine verbessernde Nichtbasis-Spalte in PRICE erzeugt werden kann. Das nutzt die Spaltengenerierung.

2.5 Spaltengenerierung

Was tun, wenn es zu viele Variablen gibt um das LP aufzustellen?

Das Simpex-Verfahren benötigt je Iteration nur die Basis-Variablen ($x_N = 0!$), solange eine verbessernde Nichtbasis-Spalte in PRICE erzeugt werden kann. Das nutzt die Spaltengenerierung.

Am Beispiel:

Zuschnittproblem: Aus möglichst wenigen Metern von Standardblechrollen der Breite ($b_0 = 1000\text{mm}$) sollen kleinere Breiten b_j (in mm) mit Gesamtlänge h_j, $j = 1, \ldots, m$ geschnitten werden.

Modellierung des Zuschnittproblems

Daten: Länge h_j der Breite $b_j \in \mathbb{N}$ $(j = 1, \ldots, m)$ aus Breite b_0

Modell: Schnittmuster $s \in \mathbb{N}_0^m$ enthält die Breite b_j genau s_j mal.

Menge der zul. Schnittmuster: $S = \left\{ s \in \mathbb{N}_0^m : \sum_{j=0}^{m} s_j b_j \leq b_0 \right\}$

Modellierung des Zuschnittproblems

Daten: Länge h_j der Breite $b_j \in \mathbb{N}$ $(j = 1, \ldots, m)$ aus Breite b_0
Modell: Schnittmuster $s \in \mathbb{N}_0^m$ enthält die Breite b_j genau s_j mal.

Menge der zul. Schnittmuster: $S = \left\{ s \in \mathbb{N}_0^m : \sum_{j=0}^{m} s_j b_j \leq b_0 \right\}$

Variable x_s gibt die Länge an, mit der $s \in S$ geschnitten wird.
$[sx_s]_j$ ist der Beitrag von Schnittmuster s zum Bedarf l_j.

Modellierung des Zuschnittproblems

Daten: Länge h_j der Breite $b_j \in \mathbb{N}$ $(j = 1, \ldots, m)$ aus Breite b_0
Modell: Schnittmuster $s \in \mathbb{N}_0^m$ enthält die Breite b_j genau s_j mal.

Menge der zul. Schnittmuster: $S = \left\{ s \in \mathbb{N}_0^m : \sum_{j=0}^{m} s_j b_j \leq b_0 \right\}$

Variable x_s gibt die Länge an, mit der $s \in S$ geschnitten wird.
$[sx_s]_j$ ist der Beitrag von Schnittmuster s zum Bedarf l_j.

$$
\begin{array}{llll}
 & \min & \sum_{s \in S} x_s & \\
(P_S) & \text{s.t.} & \sum_{s \in S} s x_s \geq h & \\
 & & x \geq 0 &
\end{array}
\qquad
\begin{array}{llll}
 & \max & h^T y & \\
(D_S) & \text{s.t.} & s^T y \leq 1 & s \in S \\
 & & y \geq 0 &
\end{array}
$$

sucht die minimal notwendige Gesamtlänge um alle l_j zu erfüllen.
Schwierigkeit: $|S|$ ist riesig!

Modellierung des Zuschnittproblems

Daten: Länge h_j der Breite $b_j \in \mathbb{N}$ $(j = 1, \ldots, m)$ aus Breite b_0
Modell: Schnittmuster $s \in \mathbb{N}_0^m$ enthält die Breite b_j genau s_j mal.

Menge der zul. Schnittmuster: $S = \left\{ s \in \mathbb{N}_0^m : \sum_{j=0}^m s_j b_j \leq b_0 \right\}$

Variable x_s gibt die Länge an, mit der $s \in S$ geschnitten wird.
$[sx_s]_j$ ist der Beitrag von Schnittmuster s zum Bedarf l_j.

$$(P_S) \quad \begin{array}{ll} \min & \sum_{s \in S} x_s \\ \text{s.t.} & \sum_{s \in S} s x_s \geq h \\ & x \geq 0 \end{array} \qquad (D_S) \quad \begin{array}{ll} \max & h^T y \\ \text{s.t.} & s^T y \leq 1 \quad s \in S \\ & y \geq 0 \end{array}$$

sucht die minimal notwendige Gesamtlänge um alle l_j zu erfüllen.
Schwierigkeit: $|S|$ ist riesig!
Idee: Starte mit $\hat{S} = \{e_j : j = 1, \ldots, m\}$, löse $(P_{\hat{S}}) \to \hat{y}^*$,
bestimme, wie in PRICE, $s \in S$ mit kleinsten red. Kosten
$\underbrace{\bar{s} \in Argmin_{s \in S} \hat{z}_s}_{\text{Pricingproblem}} := 1 - s^T \hat{y}^* \to$ setze $\hat{S} := \hat{S} \cup \{\bar{s}\}$, iteriere.

Das Pricingproblem des Zuschnittproblems

Für $\hat{y} \geq 0$, $b_j \in \mathbb{N}$ $(j = 0, \ldots, m)$, bestimme $\bar{s} \in \operatorname*{Argmin}_{s \in S} \{1 - s^T \hat{y}\}$

$$\max \ \hat{y}^T s \quad \text{s.t.} \quad b^T s \leq b_0, \ s \in \mathbb{N}_0^m$$

Dieses ganzzahlige Optimierungsproblem heißt Rucksackproblem.

Das Pricingproblem des Zuschnittproblems

Für $\hat{y} \geq 0$, $b_j \in \mathbb{N}$ $(j = 0, \ldots, m)$, bestimme $\bar{s} \in \operatorname{Argmin}_{s \in S}\{1 - s^T \hat{y}\}$

$$\max\ \hat{y}^T s \quad \text{s.t.} \quad b^T s \leq b_0,\ s \in \mathbb{N}_0^m$$

Dieses ganzzahlige Optimierungsproblem heißt Rucksackproblem.

Falls b_0 nicht zu groß, gut rekursiv lösbar (dynamic programming).
Die beste Füllung bis Kapazität $\bar{b}$ mit Objekten $1, \ldots, \bar{k}$ hat den Wert

$$F(\bar{b}, \bar{k}) := \max \left\{ \sum_{j=1}^{\bar{k}} \hat{y}_j s_j : \sum_{j=1}^{\bar{k}} b_j s_j \leq \bar{b}, s_j \in \mathbb{N}_0 \right\} \text{ für } \begin{array}{l} \bar{b} = 0, \ldots, b_0, \\ \bar{k} = 0, \ldots, m, \end{array}$$

wobei $F(0,0) := 0$ und $F(\bar{b}, \bar{k}) := -\infty$ für $(-\bar{b}) \in \mathbb{N},\ \bar{k} = 0, \ldots, m$.

Das Pricingproblem des Zuschnittproblems

Für $\hat{y} \geq 0$, $b_j \in \mathbb{N}$ $(j = 0, \ldots, m)$, bestimme $\bar{s} \in \operatorname{Argmin}_{s \in S}\{1 - s^T \hat{y}\}$

$$\max\ \hat{y}^T s \quad \text{s.t.} \quad b^T s \leq b_0,\ s \in \mathbb{N}_0^m$$

Dieses ganzzahlige Optimierungsproblem heißt Rucksackproblem.

Falls b_0 nicht zu groß, gut rekursiv lösbar (dynamic programming).
Die beste Füllung bis Kapazität $\bar{b}$ mit Objekten $1, \ldots, \bar{k}$ hat den Wert

$$F(\bar{b}, \bar{k}) := \max\left\{\sum_{j=1}^{\bar{k}} \hat{y}_j s_j : \sum_{j=1}^{\bar{k}} b_j s_j \leq \bar{b}, s_j \in \mathbb{N}_0\right\} \text{ für } \begin{array}{l}\bar{b} = 0, \ldots, b_0,\\ \bar{k} = 0, \ldots, m,\end{array}$$

wobei $F(0,0) := 0$ und $F(\bar{b}, \bar{k}) := -\infty$ für $(-\bar{b}) \in \mathbb{N},\ \bar{k} = 0, \ldots, m$.
Nun gilt

$$F(\bar{b}, \bar{k}) = \max\left\{F(\bar{b}, \bar{k} - 1), F(\bar{b} - b_{\bar{k}}, \bar{k}) + \hat{y}_{\bar{k}}\right\} \text{ für } \begin{array}{l}\bar{b} = 1, \ldots, b_0,\\ \bar{k} = 1, \ldots, m.\end{array}$$

(Entweder $\bar{k}$ nicht verwenden oder $\bar{k}$ einmal verwenden und Rest bestmöglich füllen)

Beispiel Rucksackproblem

$b_0 = 10,\ b_1 = 3,\ b_2 = 5,\ b_3 = 6,\ \hat{y}_1 = 2,\ \hat{y}_2 = 4,\ \hat{y}_3 = 5$

$F(\bar{b}, \bar{k}) = \max\left\{F(\bar{b}, \bar{k}-1), F(\bar{b} - b_{\bar{k}}, \bar{k}) + \hat{y}_{\bar{k}}\right\}$ für $\bar{b} = 1, \ldots, b_0,\ \bar{k} = 1, \ldots, m.$

$\bar{b}$	$F(\bar{b}, 0)$	$F(\bar{b}, 1)$	$F(\bar{b}, 2)$	$F(\bar{b}, 3)$
10	0	6(1)	8 (2)	8 (2)
9	0	6(1)	6 (1)	7 (3)
8	0	4(1)	6 (2)	6 (2)
7	0	4(1)	4 (1)	5 (3)
6	0	4(1)	4 (1)	5 (3)
5	0	2(1)	4 (2)	4 (2)
4	0	2(1)	2 (1)	2 (1)
3	0	2(1)	2 (1)	2 (1)
2	0	0	0	0
1	0	0	0	0
0	0	0	0	0
-1	$-\infty$	$-\infty$	$-\infty$	$-\infty$

Rekonstruktion der Lösung durch Objektindices.

Zusammenfassung Spaltengenerierung Zuschnittproblem

Daten: Länge h_j der Breite b_j $(j = 1, \ldots, m)$ aus Breite b_0

0. Schritt: $\hat{S} = \{e_j : j = 1, \ldots, m\}$ (pro j: Breite j einmal)

1. Schritt Ermittle OLen $\hat{x}^*$, $\hat{y}^*$ von

$$(P_{\hat{S}}) \quad \begin{array}{ll} \min & \sum_{s\in\hat{S}} x_s \\ \text{s.t.} & \sum_{s\in\hat{S}} s x_s \geq h \\ & x \geq 0 \end{array} \qquad (D_{\hat{S}}) \quad \begin{array}{ll} \max & h^T y \\ \text{s.t.} & s^T y \leq 1 \quad s \in \hat{S} \\ & y \geq 0 \end{array}$$

2. Schritt: Löse das Pricingprobelm als Rucksack

$$\max (\hat{y}^*)^T s \quad \text{s.t.} \quad b^T s \leq b_0,\ s \in \mathbb{N}_0^m \qquad \rightarrow \bar{s}$$

3. Schritt: Ist $1 - \bar{s}^T \hat{y}^* \geq 0$ (oder beinahe) STOP (gut genug)
sonst $\hat{S} := \hat{S} \cup \{\bar{s}\}$, GOTO 1.
(auch Entfernen ungebrauchter Muster möglich)

Zusammenfassung Spaltengenerierung Zuschnittproblem

Daten: Länge h_j der Breite b_j $(j = 1, \ldots, m)$ aus Breite b_0

0. Schritt: $\hat{S} = \{e_j : j = 1, \ldots, m\}$ (pro j: Breite j einmal)

1. Schritt Ermittle OLen $\hat{x}^*$, $\hat{y}^*$ von

$$(P_{\hat{S}}) \quad \begin{array}{ll} \min & \sum_{s\in\hat{S}} x_s \\ \text{s.t.} & \sum_{s\in\hat{S}} s x_s \geq h \\ & x \geq 0 \end{array} \qquad (D_{\hat{S}}) \quad \begin{array}{ll} \max & h^T y \\ \text{s.t.} & s^T y \leq 1 \quad s \in \hat{S} \\ & y \geq 0 \end{array}$$

2. Schritt: Löse das Pricingprobelm als Rucksack

$$\max\ (\hat{y}^*)^T s \quad \text{s.t.} \quad b^T s \leq b_0,\ s \in \mathbb{N}_0^m \qquad \rightarrow \bar{s}$$

3. Schritt: Ist $1 - \bar{s}^T \hat{y}^* \geq 0$ (oder beinahe) STOP (gut genug)
sonst $\hat{S} := \hat{S} \cup \{\bar{s}\}$, GOTO 1.
(auch Entfernen ungebrauchter Muster möglich)

Bemerkungen zur Praxis:

- Anfangsmenge $\hat{S}$ muss Zul. und endl. OL garantieren
- Verwendet m verschiedene Schnittmuster (# Basisvar.), oft zuviel
- Längen $\hat{x}_S$ nicht ganzzahlig, oft wenige lange, einige sehr kurze
- Anfangs gute Verbesserung, dann SEHR langsam (tailing off effect)

Inhaltsübersicht

2.6 Schnittebenenverfahren

Gleiche Idee, wenn es zu viele Ungleichungen $a_j^T x \geq b_j$, $j \in \mathcal{I}$ gibt:

0. Schritt: Wähle $\hat{\mathcal{I}} \subset \mathcal{I}$, die endliche OL garantiert

1. Schritt Ermittle OLen $\hat{x}^*$, $\hat{y}^*$ von

$$(P_{\hat{\mathcal{I}}}) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_j^T x \geq b_j \quad j \in \hat{\mathcal{I}} \end{array} \qquad (D_{\hat{\mathcal{I}}}) \quad \begin{array}{ll} \max & \sum_{j\in\hat{\mathcal{I}}} b_j y_j \\ \text{s.t.} & \sum_{j\in\hat{\mathcal{I}}} a_j y_j \leq c \\ & y \geq 0 \end{array}$$

2. Schritt: Löse das Separierungsproblem: Finde eine (maximal) verletzte Ungleichung $\bar{j} \in \operatorname{Argmax}_{j\in\mathcal{I}}\{b_j - a_j^T \hat{x}^*\}$

3. Schritt: Ist $b_j - a_j^T \hat{x}^* \leq 0$ (oder beinahe) STOP (gut genug)
sonst $\hat{\mathcal{I}} := \hat{\mathcal{I}} \cup \{\bar{j}\}$, GOTO 1.
(auch Entfernen inaktiver Ungleichungen möglich)

2.6 Schnittebenenverfahren

Gleiche Idee, wenn es zu viele Ungleichungen $a_j^T x \geq b_j$, $j \in \mathcal{I}$ gibt:

0. Schritt: Wähle $\hat{\mathcal{I}} \subset \mathcal{I}$, die endliche OL garantiert

1. Schritt Ermittle OLen $\hat{x}^*$, $\hat{y}^*$ von

$$(P_{\hat{\mathcal{I}}}) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & a_j^T x \geq b_j \quad j \in \hat{\mathcal{I}} \end{array} \qquad (D_{\hat{\mathcal{I}}}) \quad \begin{array}{ll} \max & \sum_{j\in\hat{\mathcal{I}}} b_j y_j \\ \text{s.t.} & \sum_{j\in\hat{\mathcal{I}}} a_j y_j \leq c \\ & y \geq 0 \end{array}$$

2. Schritt: Löse das Separierungsproblem: Finde eine (maximal) verletzte Ungleichung $\bar{j} \in \operatorname{Argmax}_{j\in\mathcal{I}}\{b_j - a_j^T \hat{x}^*\}$

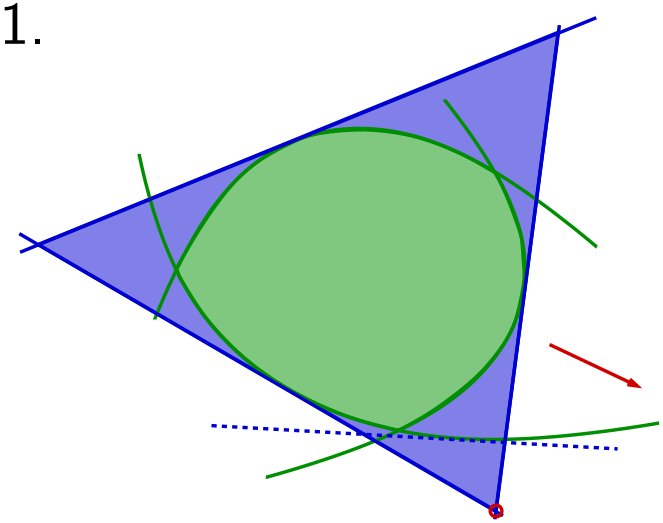
3. Schritt: Ist $b_j - a_j^T \hat{x}^* \leq 0$ (oder beinahe) STOP (gut genug)
sonst $\hat{\mathcal{I}} := \hat{\mathcal{I}} \cup \{\bar{j}\}$, GOTO 1.
(auch Entfernen inaktiver Ungleichungen möglich)

Bemerkungen:

- Schnittebenenvefahren = Spaltengenerierung im Dualen
- gute Erfolge in kombinatorischer/ganzz. Optimierung
- Linearisierung konvexer Nebenbedingungen (oft gibt es besseres!)
- Je nach separierter Unglg.-Klasse starker tailing off effect

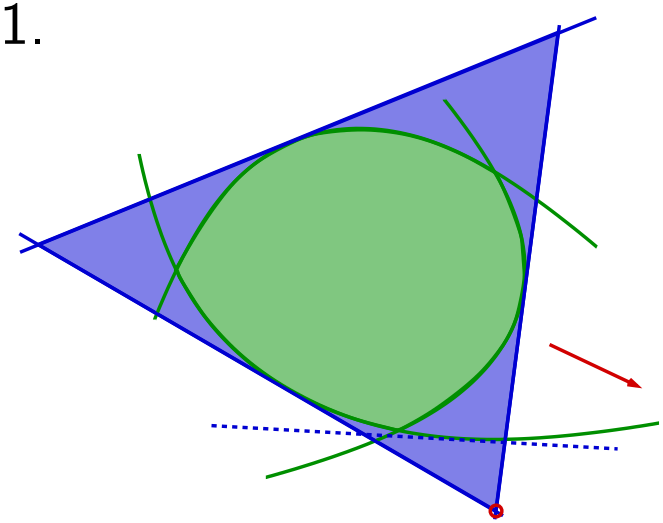
Konvexe Nebenbedingungen mit Schnittebenen

1.

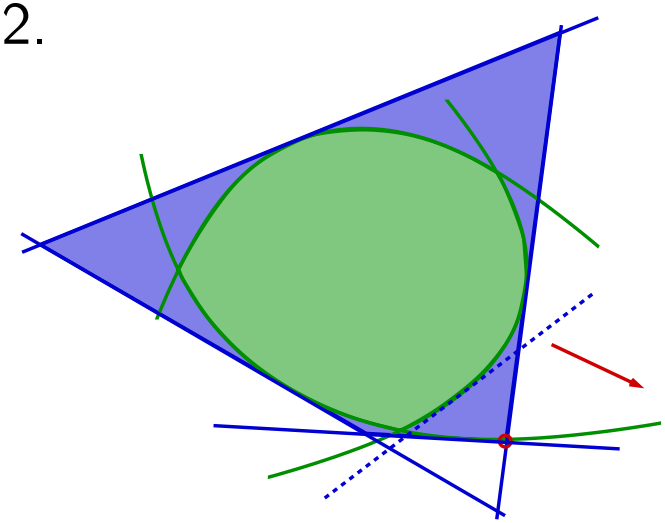


Konvexe Nebenbedingungen mit Schnittebenen

1.

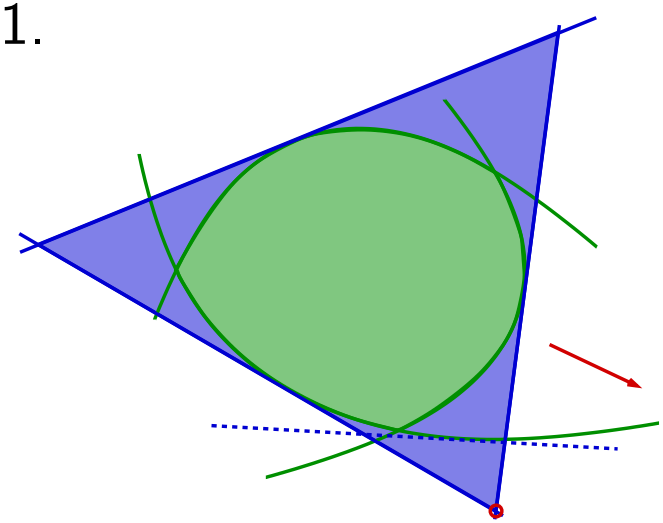


2.

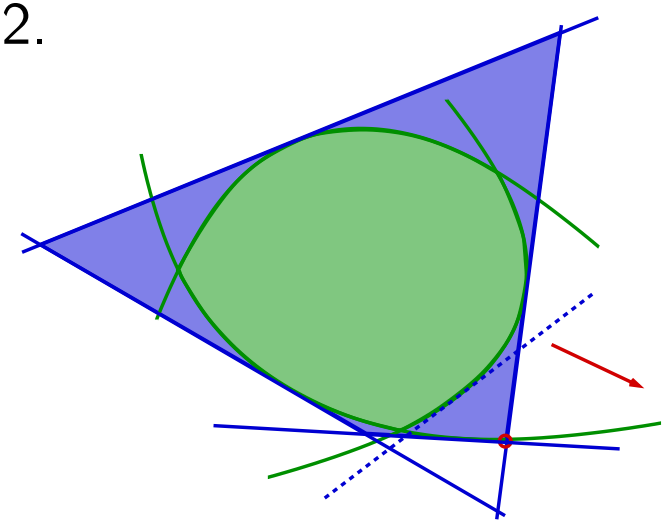


Konvexe Nebenbedingungen mit Schnittebenen

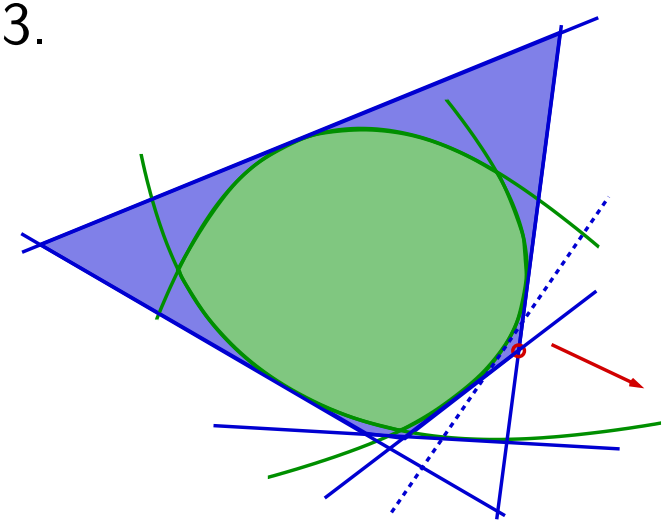
1.



2.

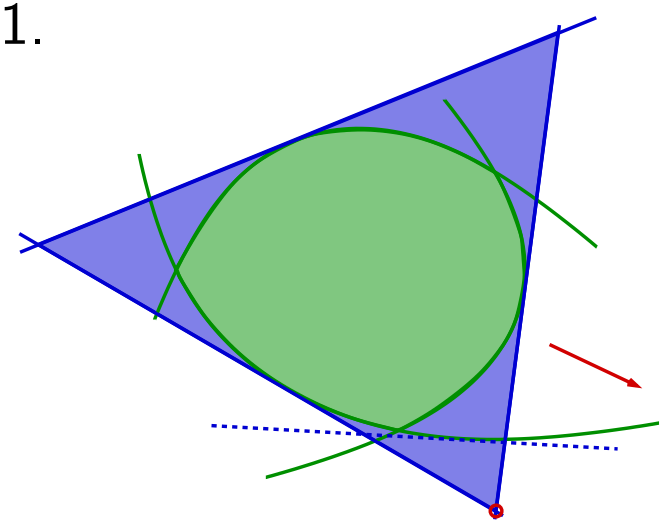


3.

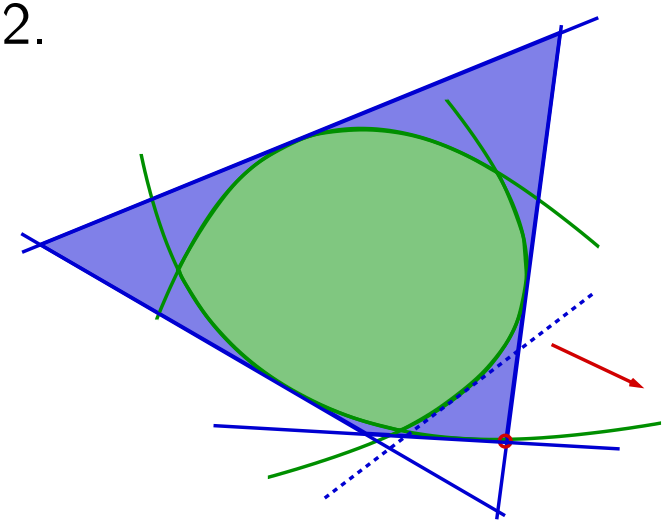


Konvexe Nebenbedingungen mit Schnittebenen

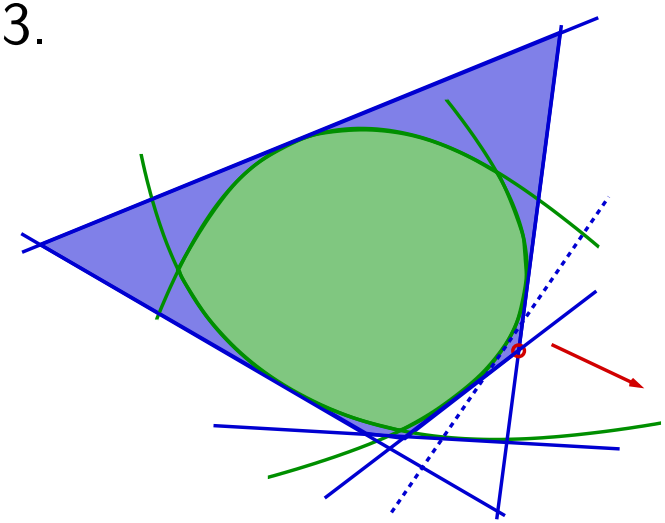
1.



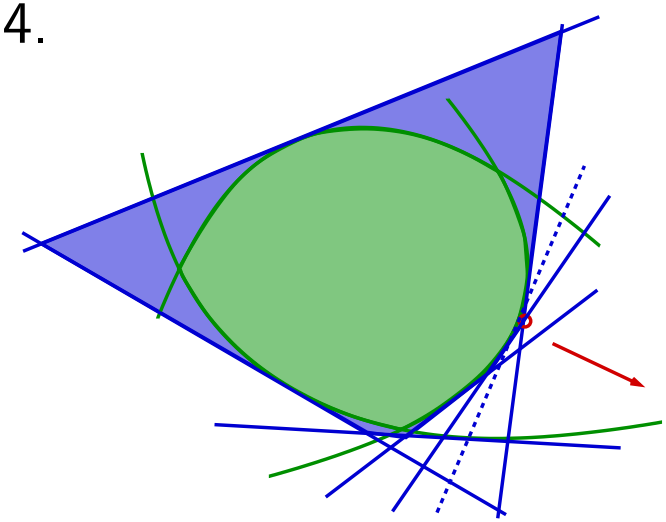
2.



3.



4.



Problem: Schleifende Schnitte numerisch ungünstig, ungenau

Inhaltsübersicht

2.7 Wann primalen, wann dualen Simplex verwenden?

Für ein festes LP:

Für viele LPs aus der Praxis scheint der duale Simplex etwas schneller zu sein (vermutlich weil viele LPs nach ähnlichen Denkmustern erstellt werden).

Innere-Punkte-Verfahren (s. später) sind bei großen Instanzen oft schneller und weniger anfällig bzgl. Degeneriertheit, aber ein Test lohnt sich.

2.7 Wann primalen, wann dualen Simplex verwenden?

Für ein festes LP:

Für viele LPs aus der Praxis scheint der duale Simplex etwas schneller zu sein (vermutlich weil viele LPs nach ähnlichen Denkmustern erstellt werden).
Innere-Punkte-Verfahren (s. später) sind bei großen Instanzen oft schneller und weniger anfällig bzgl. Degeneriertheit, aber ein Test lohnt sich.

Spaltengenerierung: Nach Einfügen von Variablen bleibt die primale Basis zulässig (die alte Duale Lösung nicht), also setzt man mit dem primalen Simplex direkt fort!
(Innere-Punkte-Verfahren sind dafür ungeeignet)

2.7 Wann primalen, wann dualen Simplex verwenden?

Für ein festes LP:
Für viele LPs aus der Praxis scheint der duale Simplex etwas schneller zu sein (vermutlich weil viele LPs nach ähnlichen Denkmustern erstellt werden).
Innere-Punkte-Verfahren (s. später) sind bei großen Instanzen oft schneller und weniger anfällig bzgl. Degeneriertheit, aber ein Test lohnt sich.

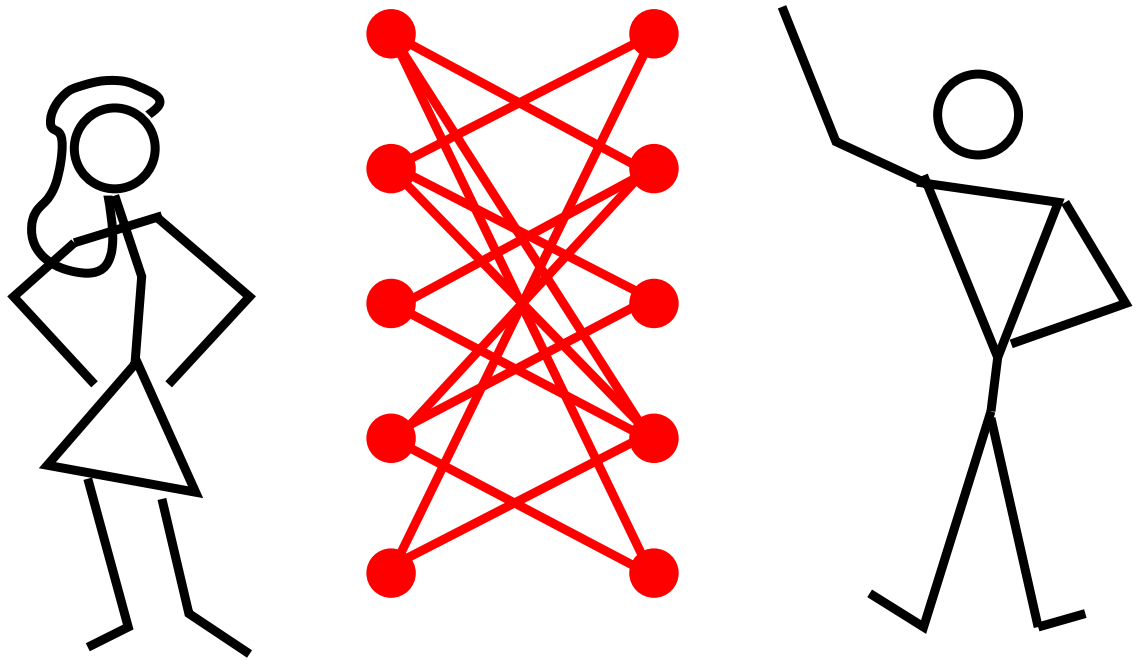
Spaltengenerierung: Nach Einfügen von Variablen bleibt die primale Basis zulässig (die alte Duale Lösung nicht), also setzt man mit dem primalen Simplex direkt fort!
(Innere-Punkte-Verfahren sind dafür ungeeignet)

Schnittebenen-Verfahren:
Nach Einfügen einer Schnittebene bleibt die duale Basis zulässig (die alte primale Lösung nicht), also setzt man mit dem dualen Simplex direkt fort!
(Innere-Punkte-Verfahren sind dafür ungeeignet)

Inhaltsübersicht

Inhaltsübersicht

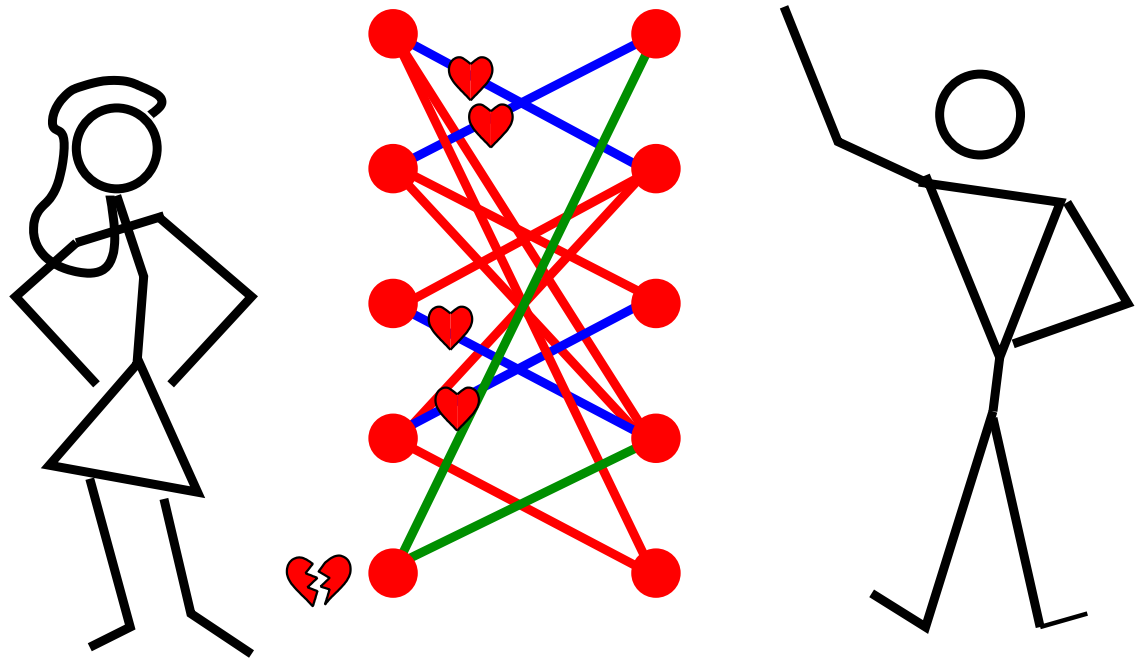
3.1 Anwendung: Das Heiratsproblem (bipartites Matching)



Bilde möglichst viele Paare!

Männer $\leftrightarrow$ Frauen, Arbeiter $\leftrightarrow$ Maschinen, Studierende $\leftrightarrow$ Studienplätze, ...
(geht auch gewichtet)

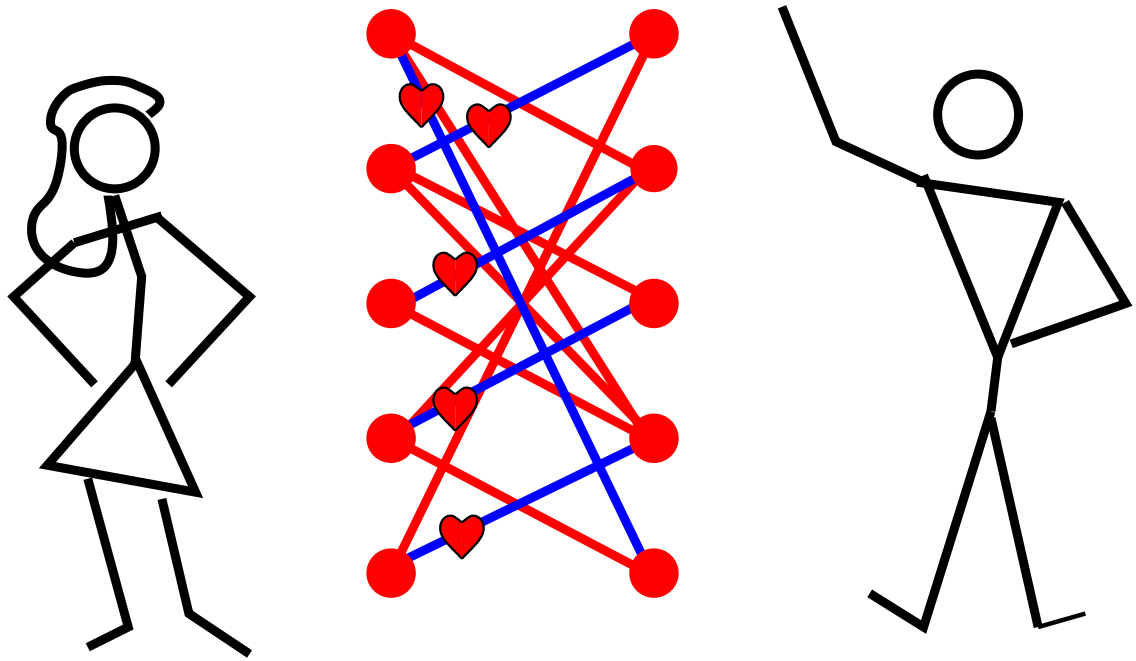
3.1 Anwendung: Das Heiratsproblem (bipartites Matching)



Bilde möglichst viele Paare!
Maximal (nicht vergrößerbar), aber kein Kardinalitätsmaximum

Männer $\leftrightarrow$ Frauen, Arbeiter $\leftrightarrow$ Maschinen, Studierende $\leftrightarrow$ Studienplätze, ...
(geht auch gewichtet)

3.1 Anwendung: Das Heiratsproblem (bipartites Matching)

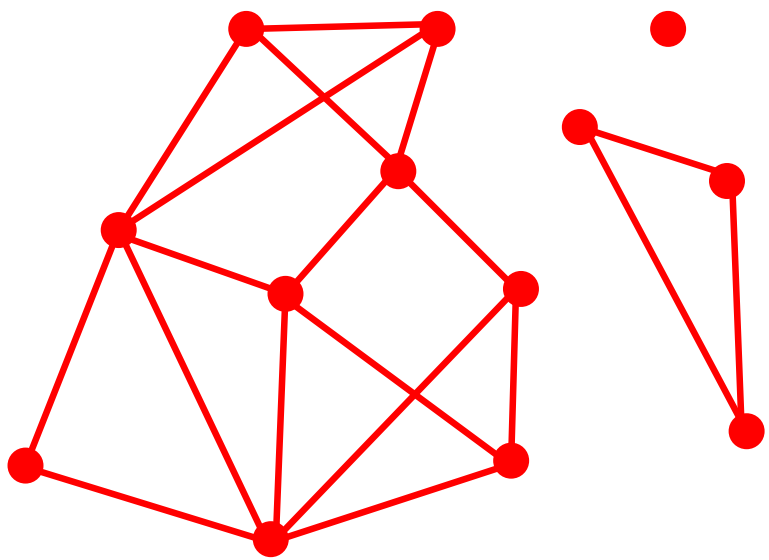


Bilde möglichst viele Paare!
Maximum Cardinality Matching (sogar perfekt)

Männer $\leftrightarrow$ Frauen, Arbeiter $\leftrightarrow$ Maschinen, Studierende $\leftrightarrow$ Studienplätze, ...
(geht auch gewichtet)

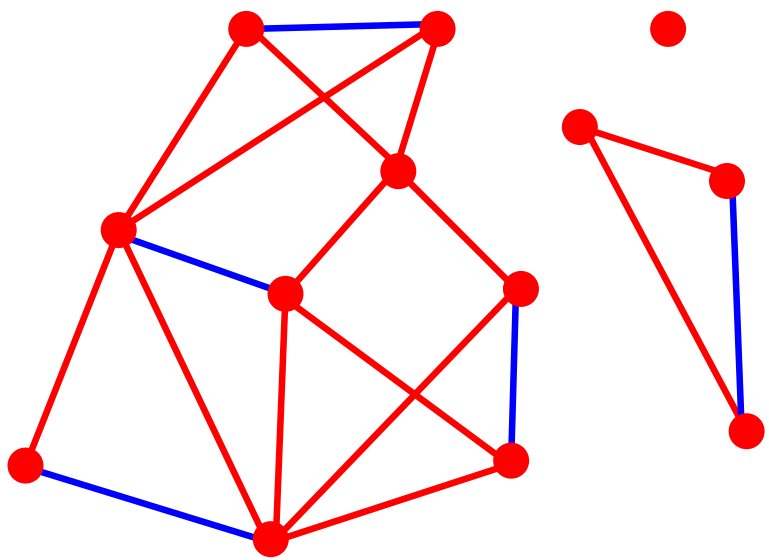
Bipartites Matching

- Ein (ungerichteter) Graph $G = (V, E)$ ist ein Paar bestehend aus Knotenmenge V und Kantenmenge $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$.
- Zwei Knoten $u, v \in V$ heißen adjazent/benachbart, falls $\{u, v\} \in E$.
- Ein Knoten $v \in V$ und eine Kante $e \in E$ heißen inzident, falls $v \in e$.
- Zwei Kanten $e, f \in V$ heißen inzident, falls $e \cap f \neq \emptyset$.



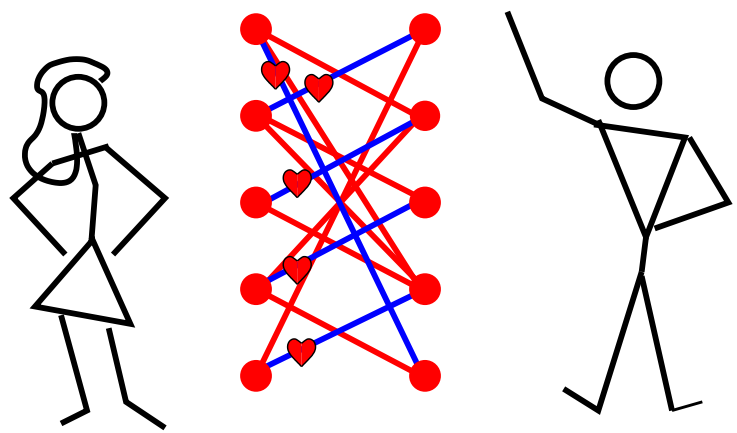
Bipartites Matching

- Ein (ungerichteter) Graph $G = (V, E)$ ist ein Paar bestehend aus Knotenmenge V und Kantenmenge $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$.
- Zwei Knoten $u, v \in V$ heißen adjazent/benachbart, falls $\{u, v\} \in E$.
- Ein Knoten $v \in V$ und eine Kante $e \in E$ heißen inzident, falls $v \in e$.
- Zwei Kanten $e, f \in V$ heißen inzident, falls $e \cap f \neq \emptyset$.
- Eine Kantenmenge $M \subseteq E$ heißt Matching/Paarung, falls für $e, f \in M$ mit $e \neq f$ stets $e \cap f = \emptyset$. Das Matching heißt perfekt, falls $|V| = 2|M|$.



Bipartites Matching

- Ein (ungerichteter) Graph $G = (V, E)$ ist ein Paar bestehend aus Knotenmenge V und Kantenmenge $E \subseteq \{\{u, v\} : u, v \in V, u \neq v\}$.
- Zwei Knoten $u, v \in V$ heißen adjazent/benachbart, falls $\{u, v\} \in E$.
- Ein Knoten $v \in V$ und eine Kante $e \in E$ heißen inzident, falls $v \in e$.
- Zwei Kanten $e, f \in V$ heißen inzident, falls $e \cap f \neq \emptyset$.
- Eine Kantenmenge $M \subseteq E$ heißt Matching/Paarung, falls für $e, f \in M$ mit $e \neq f$ stets $e \cap f = \emptyset$. Das Matching heißt perfekt, falls $|V| = 2|M|$.
- $G = (V, E)$ heißt bipartit, falls $V = V_1 \cup V_2$ mit $V_1 \cap V_2 = \emptyset$ und $E \subseteq \{\{u, v\} : u \in V_1, v \in V_2\}$.



Modellierung: kardinalitätsmaximales bipartites Matching

gegeben: $G = (V_1 \dot\cup V_2, E)$ bipartit
gesucht: Matching $M \subseteq E$ mit $|M|$ maximal

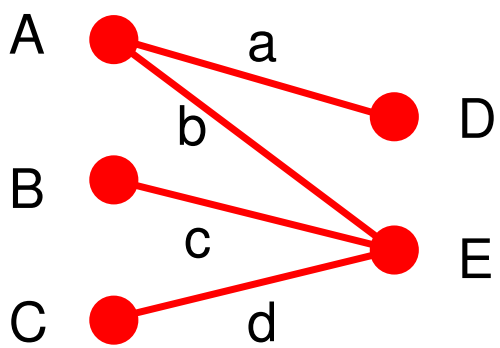
Variablen: $x \in \{0,1\}^E$ mit $x_e = \begin{cases} 1 & \text{falls } e \in M \\ 0 & \text{sonst.} \end{cases} \quad (e \in E)$

(steht für den Inzidenz-/charakteristischen Vektor von M bzgl. E)

Nebenbedingung: $Ax \leq \mathbf{1}$,

wobei $A \in \{0,1\}^{V \times E}$ Knoten-Kanten-Inzidenzmatrix zu G:

$$A_{v,e} = \begin{cases} 1 & \text{falls } v \in e \\ 0 & \text{sonst.} \end{cases} \quad (v \in V, e \in E)$$



$$A = \left[\begin{array}{c|cccc} & (a) & (b) & (c) & (d) \\ (A) & 1 & 1 & 0 & 0 \\ (B) & 0 & 0 & 1 & 0 \\ (C) & 0 & 0 & 0 & 1 \\ \hline (D) & 1 & 0 & 0 & 0 \\ (E) & 0 & 1 & 1 & 1 \end{array}\right]$$

Modellierung: kardinalitätsmaximales bipartites Matching

gegeben: $G = (V_1 \dot\cup V_2, E)$ bipartit
gesucht: Matching $M \subseteq E$ mit $|M|$ maximal

Variablen: $x \in \{0,1\}^E$ mit $x_e = \begin{cases} 1 & \text{falls } e \in M \\ 0 & \text{sonst.} \end{cases} \quad (e \in E)$

(steht für den Inzidenz-/charakteristischen Vektor von M bzgl. E)

Nebenbedingung: $Ax \leq \mathbf{1}$,

wobei $A \in \{0,1\}^{V \times E}$ Knoten-Kanten-Inzidenzmatrix zu G:

$$A_{v,e} = \begin{cases} 1 & \text{falls } v \in e \\ 0 & \text{sonst.} \end{cases} \quad (v \in V, e \in E)$$

Optimierungsproblem:

$$\begin{aligned} \max \quad & \mathbf{1}^T x \\ \text{s.t.} \quad & Ax \leq \mathbf{1} \\ & x \in \{0,1\}^E \end{aligned}$$

So kein LP! $x \in \{0,1\}^E$ zu $x \in [0,1]^E$ vergrößern $\rightarrow$ LP

Für G bipartit gilt: Simplex liefert immer eine Optimallösung $x^* \in \{0,1\}^E$!
(Für allgemeine Graphen G i.A. aber nicht!)

Inhaltsübersicht

3.2 Ganzzahlige Polyeder

Simplex liefert automatisch eine ganzzahlige Lösung, wenn alle Ecken der zulässigen Menge ganzzahlig sind.

$$\min c^T x \quad \text{s.t.} \quad x \in \mathcal{X} := \{x \geq 0 : Ax = b\}$$

Hat $\mathcal{X}$ nur ganzzahlige Ecken?

3.2 Ganzzahlige Polyeder

Simplex liefert automatisch eine ganzzahlige Lösung, wenn alle Ecken der zulässigen Menge ganzzahlig sind.

$$\min c^T x \quad \text{s.t.} \quad x \in \mathcal{X} := \{x \geq 0 : Ax = b\}$$

Hat $\mathcal{X}$ nur ganzzahlige Ecken? Fast nie!
Aber es gibt wichtige Klassen von Matrizen $A \in \mathbb{Z}^{m \times n}$, für die $\mathcal{X}$ für jedes (!) $b \in \mathbb{Z}^m$ nur ganzzahlige Ecken hat:

Eine Ecke ist ganzzahlig $\Leftrightarrow$ Basislösung $x_B = A_B^{-1} b \in \mathbb{Z}^m$
Wenn $|\det(A_B)| = 1$, folgt mit der Cramer'schen Regel $x_B \in \mathbb{Z}^m$.

3.2 Ganzzahlige Polyeder

Simplex liefert automatisch eine ganzzahlige Lösung, wenn alle Ecken der zulässigen Menge ganzzahlig sind.

$$\min c^T x \quad \text{s.t.} \quad x \in \mathcal{X} := \{x \geq 0 : Ax = b\}$$

Hat $\mathcal{X}$ nur ganzzahlige Ecken? Fast nie!
Aber es gibt wichtige Klassen von Matrizen $A \in \mathbb{Z}^{m \times n}$, für die $\mathcal{X}$ für jedes (!) $b \in \mathbb{Z}^m$ nur ganzzahlige Ecken hat:

Eine Ecke ist ganzzahlig $\Leftrightarrow$ Basislösung $x_B = A_B^{-1} b \in \mathbb{Z}^m$
Wenn $|\det(A_B)| = 1$, folgt mit der Cramer'schen Regel $x_B \in \mathbb{Z}^m$.
Eine Matrix $A \in \mathbb{Z}^{m \times n}$ mit vollem Zeilenrang heißt unimodular, falls $|\det(A_B)| = 1$ für jede Basis B erfüllt ist.

Satz

$A \in \mathbb{Z}^{m \times n}$ ist genau dann unimodular, wenn für jedes $b \in \mathbb{Z}^m$ alle Ecken des Polyeders $\mathcal{X} := \{x \geq 0 : Ax = b\}$ ganzzahlig sind.

Gilt das auch für $\mathcal{X} := \{x \geq 0 : Ax \geq b\}$?

Total unimodulare Matrizen

$$\{x \geq 0 : Ax \geq b\} \quad \rightarrow \quad \left\{ \begin{bmatrix} x \\ s \end{bmatrix} \geq 0 : [A, I] \begin{bmatrix} x \\ s \end{bmatrix} = b \right\}$$

Sicher ganzzahlig, falls $\bar{A} = [A, I]$ unimodular ist.
Determinantenentwicklung nach Laplace für jede Basis $B \rightarrow$

Eine Matrix A heißt total unimodular, falls für jede quadratische Untermatrix von A die Determinante den Wert 0, 1 oder -1 hat.
(geht nur, wenn $A \in \{0, 1, -1\}^{m \times n}$)

Satz (Hoffmann und Kruskal)

$A \in \mathbb{Z}^{m \times n}$ ist genau dann total unimodular, wenn für jedes $b \in \mathbb{Z}^m$ alle Ecken des Polyeders $\mathcal{X} := \{x \geq 0 : Ax \geq b\}$ ganzzahlig sind.

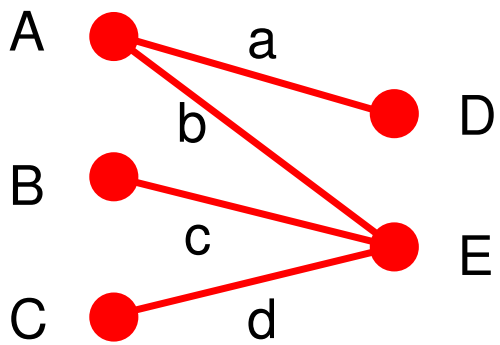
Beachte: A tot. unimod. $\Leftrightarrow$ A^T bzw. $[A, -A, I, -I]$ tot. unimod.
Konsequenz: duales LP, Gleichungsvarianten, etc. sind ganzzahlig

Total unimodulare Matrizen erkennen

Satz (Heller und Tompkins)

$A \in \{0, 1, -1\}^{m \times n}$ habe höchstens zwei Einträge pro Spalte.
A ist total unimodular $\Leftrightarrow$ Die Zeilen von A können in zwei Klassen eingeteilt werden, sodass
(i) Zeilen mit einem $+1$ und einem -1 Eintrag in derselben Spalte in dieselbe Klasse,
(ii) Zeilen mit zwei vorzeichengleichen Einträgen in der gleichen Spalte in unterschiedliche Klassen kommen.

Beispiel 1: die Knoten-Kanten-Inzidenzmatrix eines bipartiten Graphen



$$A = \left[\begin{array}{c|cccc} & (a) & (b) & (c) & (d) \\ (A) & 1 & 1 & 0 & 0 \\ (B) & 0 & 0 & 1 & 0 \\ (C) & 0 & 0 & 0 & 1 \\ \hline (D) & 1 & 0 & 0 & 0 \\ (E) & 0 & 1 & 1 & 1 \end{array}\right]$$

Beispiel 1: bipartite Graphen

A ... Knoten-Kanten-Inzidenzmatrix von $G = (V_1 \dot\cup V_2, E)$ bipartit

---

Bipartites Matching maximaler Kardinalität:

$$\max \mathbf{1}^T x \quad \text{s.t.} \quad Ax \leq \mathbf{1},\ x \geq 0$$

Wegen A tot. unimod. hat die zul. Menge nur ganzzahlige Ecken
$\Rightarrow$ Simplex liefert Optimallösung $x^* \in \{0,1\}^E$.

Das Duale ist auch ganzzahlig, wegen A^T tot. unimod.:

$$\min \mathbf{1}^T y \quad \text{s.t.} \quad A^T y \geq \mathbf{1},\ y \geq 0$$

Interpretation: $y^* \in \{0,1\}^V$ ist Inzidenzvektor einer kleinsten Knotenmenge $V' \subseteq V$, sodass $\forall e \in E : e \cap V' \neq \emptyset$ (**Minimum Vertex Cover**)

Das **Zuweisungsproblem**: $|V_1| = |V_2| = n$, vollständig bipartit:
$E = \{\{u,v\} : u \in V_1, v \in V_2\}$; Kantengewichte $c \in \mathbb{R}^E$
Suche ein perfektes Matching minimalen Gesamtgewichts:

$$\min c^T x \quad \text{s.t.} \quad Ax = \mathbf{1},\ x \geq 0$$

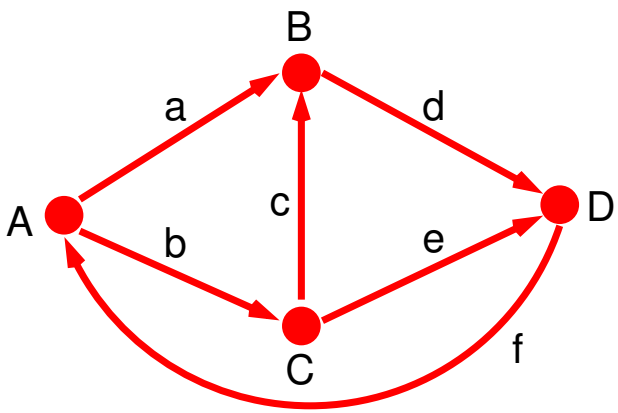
ist auch ganzzahlig, weil $[A; -A]$ tot. unimod.

Beispiel 2: Knoten-Kanten-Inzidenzmatrix von Digraphen

- Ein Digraph/gerichteter Graph $D = (V, E)$ ist ein Paar bestehend aus Knotenmenge V und einer (Multi-)Menge gerichteter Kanten/Pfeile $E \subseteq \{(u, v) : u, v \in V, u \neq v\}$. [Mehrfachkanten sind erlaubt!]
- Für $e = (u, v) \in E$ ist u der Schaft und v die Spitze von e.
- Die Knoten-Kanten-Inzidenzmatrix $A \in \{0, 1, -1\}^{V \times E}$ von D hat Einträge

$$A_{v,e} = \begin{cases} -1 & v \text{ ist Schaft von } e \\ 1 & v \text{ ist Spitze von } e \\ 0 & \text{sonst} \end{cases} \qquad (v \in V, e \in E).$$

Die Knoten-Kanten-Inzidenzmatrix eines Digraphen ist total unimodular.



$$A = \begin{array}{c} \\ (A) \\ (B) \\ (C) \\ (D) \end{array} \begin{array}{c} \begin{array}{cccccc} (a) & (b) & (c) & (d) & (e) & (f) \end{array} \\ \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix} \end{array}$$

Inhaltsübersicht

3.3 Anwendung: Netzwerkflüsse

Modellierungswerkzeug: Transportprobleme, Evakuierungspläne, Auftrags-, Verkehrsplanung, ...

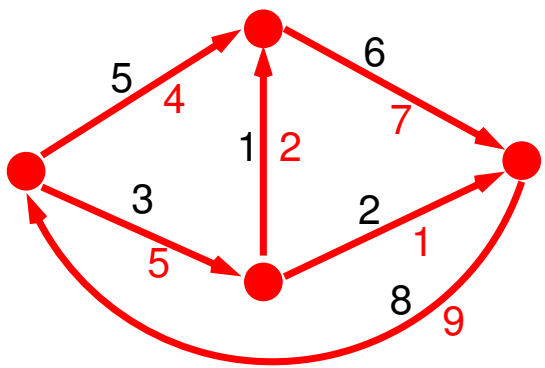
- Ein Netzwerk (D, w) besteht aus einem Digraphen $D = (V, E)$ und (Kanten-)Kapazitäten $w \in \mathbb{R}_+^E$.
- Ein Vektor $x \in \mathbb{R}^E$ heißt Fluss auf (D, w), falls er die Flusserhaltungsgleichungen

$$\sum_{e=(u,v)\in E} x_e = \sum_{e=(v,u)\in E} x_e \quad (v \in V)$$

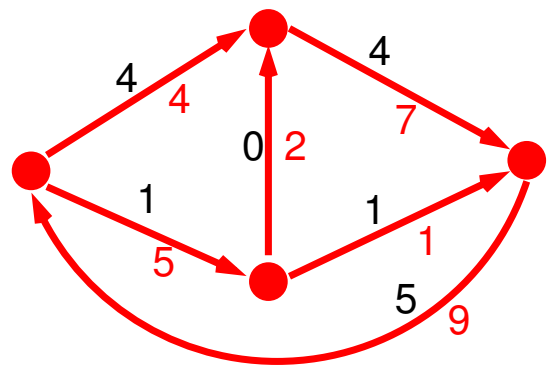
erfüllt. [$\Leftrightarrow Ax = 0$ für Knoten-Kanten-Inzidenzmatrix A]

- Ein Fluss $x \in \mathbb{R}^E$ auf (D,w) heißt zulässig, falls $0 \leq x \leq w$

[auch untere Schranken machbar]



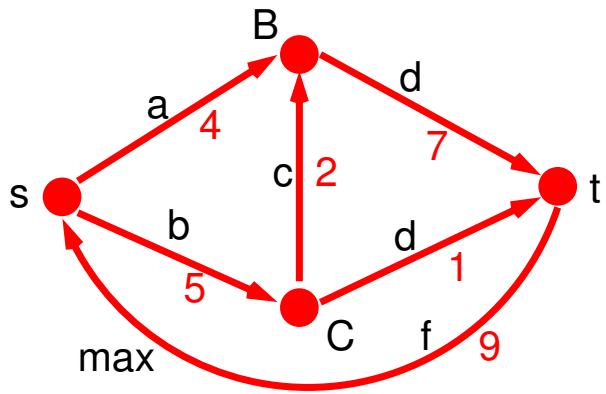
Fluss



zulässiger Fluss

Maximale s-t-Flüsse, Minimale s-t-Schnitte

Gegeben Quelle $s \in V$ und Senke $t \in V$ mit $(t,s) \in E$, finde einen zulässigen Fluss $x \in \mathbb{R}^E$ auf (D,w) mit maximalem Flusswert $x_{(t,s)}$.



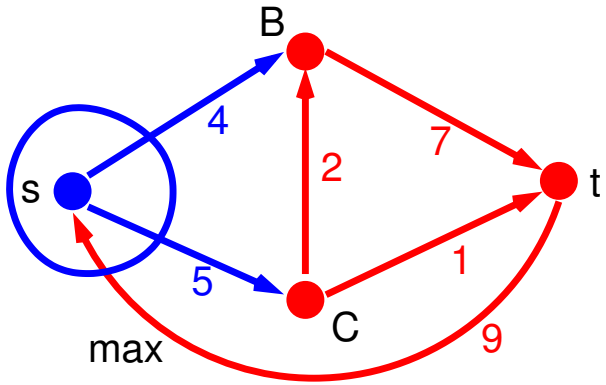
$$A = \begin{bmatrix} & (a) & (b) & (c) & (d) & (e) & (f) \\ (s) & -1 & -1 & 0 & 0 & 0 & 1 \\ (B) & 1 & 0 & 1 & -1 & 0 & 0 \\ (C) & 0 & 1 & -1 & 0 & -1 & 0 \\ (t) & 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix}$$

LP: $\max\ x_{(t,s)}$ s.t. $Ax = 0,\ 0 \le x \le w,$

Falls $w \in \mathbb{Z}^E$, ist Simplex-OL $x^* \in \mathbb{Z}^E$, weil $[A; -A; I]$ tot. unimod.

Maximale s-t-Flüsse, Minimale s-t-Schnitte

Gegeben Quelle $s \in V$ und Senke $t \in V$ mit $(t, s) \in E$, finde einen zulässigen Fluss $x \in \mathbb{R}^E$ auf (D, w) mit maximalem Flusswert $x_{(t,s)}$.



$S = \{s\}$,
$\delta^+(S) = \{(s, B), (s, C)\}$,
$w(\delta^+(S)) = 4 + 5 = 9$.

LP: max $x_{(t,s)}$ s.t. $Ax = 0,\ 0 \leq x \leq w$,

Falls $w \in \mathbb{Z}^E$, ist Simplex-OL $x^* \in \mathbb{Z}^E$, weil $[A; -A; I]$ tot. unimod.

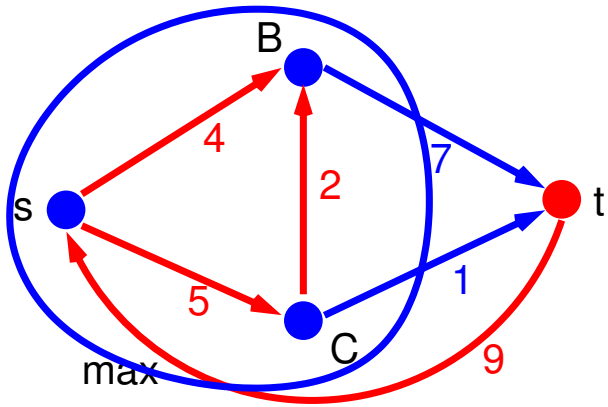
Jedes $S \subseteq V$ mit $s \in S$ und $t \notin S$ definiert einen s-t-Schnitt

$$\delta^+(S) := \{(u, v) \in E : u \in S, v \notin S\},$$

darüber fließt höchstens $w(\delta^+(S)) := \sum_{e \in \delta^+(S)} w_e$, der Wert des Schnitts.

Maximale s-t-Flüsse, Minimale s-t-Schnitte

Gegeben Quelle $s \in V$ und Senke $t \in V$ mit $(t,s) \in E$, finde einen zulässigen Fluss $x \in \mathbb{R}^E$ auf (D,w) mit maximalem Flusswert $x_{(t,s)}$.



$S = \{s, B, C\}$,
$\delta^+(S) = \{(B,t),(C,t)\}$,
$w(\delta^+(S)) = 7 + 1 = 8$.

LP: max $x_{(t,s)}$ s.t. $Ax = 0,\ 0 \le x \le w$,

Falls $w \in \mathbb{Z}^E$, ist Simplex-OL $x^* \in \mathbb{Z}^E$, weil $[A; -A; I]$ tot. unimod.

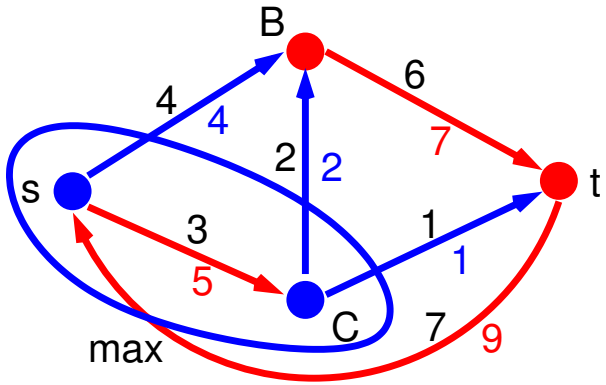
Jedes $S \subseteq V$ mit $s \in S$ und $t \notin S$ definiert einen s-t-Schnitt

$$\delta^+(S) := \{(u,v) \in E : u \in S, v \notin S\},$$

darüber fließt höchstens $w(\delta^+(S)) := \sum_{e \in \delta^+(S)} w_e$, der Wert des Schnitts.

Maximale s-t-Flüsse, Minimale s-t-Schnitte

Gegeben Quelle $s \in V$ und Senke $t \in V$ mit $(t,s) \in E$, finde einen zulässigen Fluss $x \in \mathbb{R}^E$ auf (D,w) mit maximalem Flusswert $x_{(t,s)}$.



$S = \{s, C\}$,
$\delta^+(S) = \{(s,B),(C,B),(C,t)\}$,
$w(\delta^+(S)) = 4+2+1 = 7 = x^*_{(t,s)}$

LP: max $x_{(t,s)}$ s.t. $Ax = 0,\ 0 \le x \le w$,

Falls $w \in \mathbb{Z}^E$, ist Simplex-OL $x^* \in \mathbb{Z}^E$, weil $[A; -A; I]$ tot. unimod.

Jedes $S \subseteq V$ mit $s \in S$ und $t \notin S$ definiert einen s-t-Schnitt

$$\delta^+(S) := \{(u,v) \in E : u \in S, v \notin S\},$$

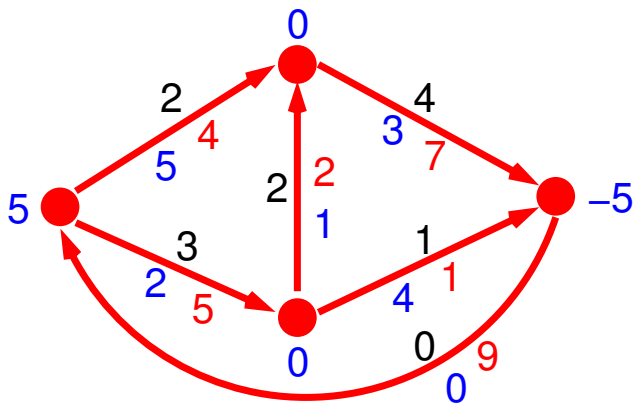
darüber fließt höchstens $w(\delta^+(S)) := \sum_{e \in \delta^+(S)} w_e$, der Wert des Schnitts.

Satz (Max-Flow Min-Cut Theorem von Ford und Fulkerson)

Der maximale Wert eines s-t-Flusses ist gleich dem minimalen Werte eines s-t-Schnitts. [beide durch Simplex bestimmbar]

Minimale-Kosten-Flüsse (Min-Cost-Flow)

Die Flussmenge wird durch Balancen $b \in \mathbb{R}^V$ $(\mathbf{1}^T b = 0)$ auf den Knoten vorgegeben, pro Flusseinheit fallen Kantenkosten $c \in \mathbb{R}^E$ an.
Finde den günstigsten Fluss.



$$\begin{array}{ll} \min & \begin{bmatrix} 5 & 2 & 1 & 3 & 4 & 0 \end{bmatrix} x \\ \text{s.t.} & \begin{bmatrix} -1 & -1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 1 & -1 \end{bmatrix} x = \begin{bmatrix} 5 \\ 0 \\ 0 \\ -5 \end{bmatrix} \\ & 0 \le x \le w \end{array}$$

LP: $\min\ c^T x \quad \text{s.t.} \quad Ax = b,\ 0 \le x \le w,$

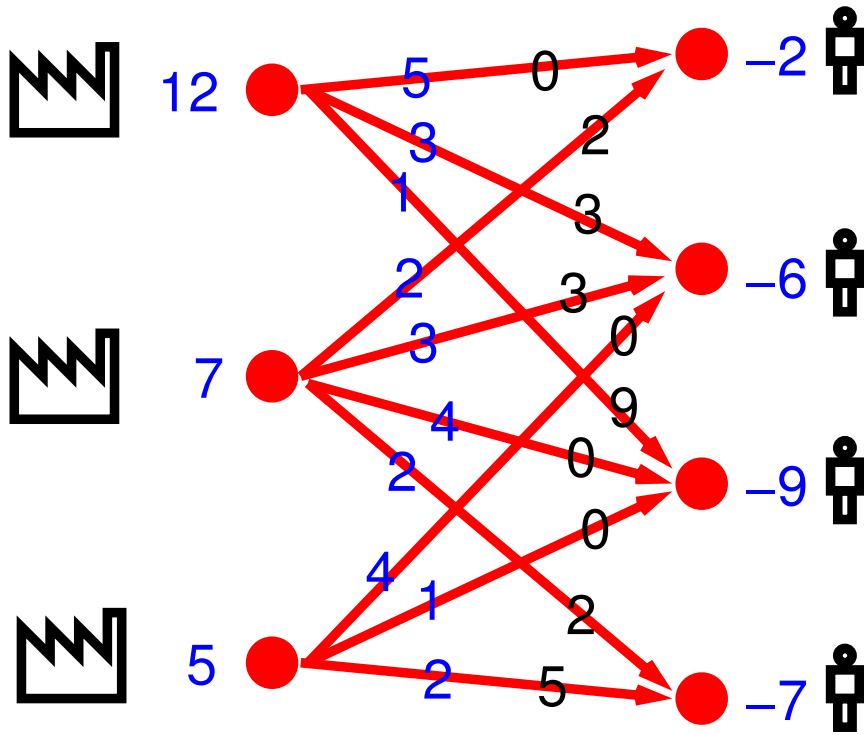
Für b, c und w ganzz. ist Simplex-OL $x^* \in \mathbb{Z}^E$, weil $[A; -A; I]$ tot. unimod.
[geht auch mit unteren Schranken auf den Kanten: $u \le x \le w$!]

Für LPs $\min\ c^T x \quad \text{s.t.} \quad Ax = b,\ u \le x \le w,$ A Knoten-Kanten-Inz. gibt es eine besonders effiziente Simplex-Variante, den Netzwerksimplex, dieser braucht nur Addition, Subtraktion und Vergleiche!

Extrem breite Anwendungsmöglichkeiten, beliebtes Modellierungswerkzeug

Beispiel: Transportproblem

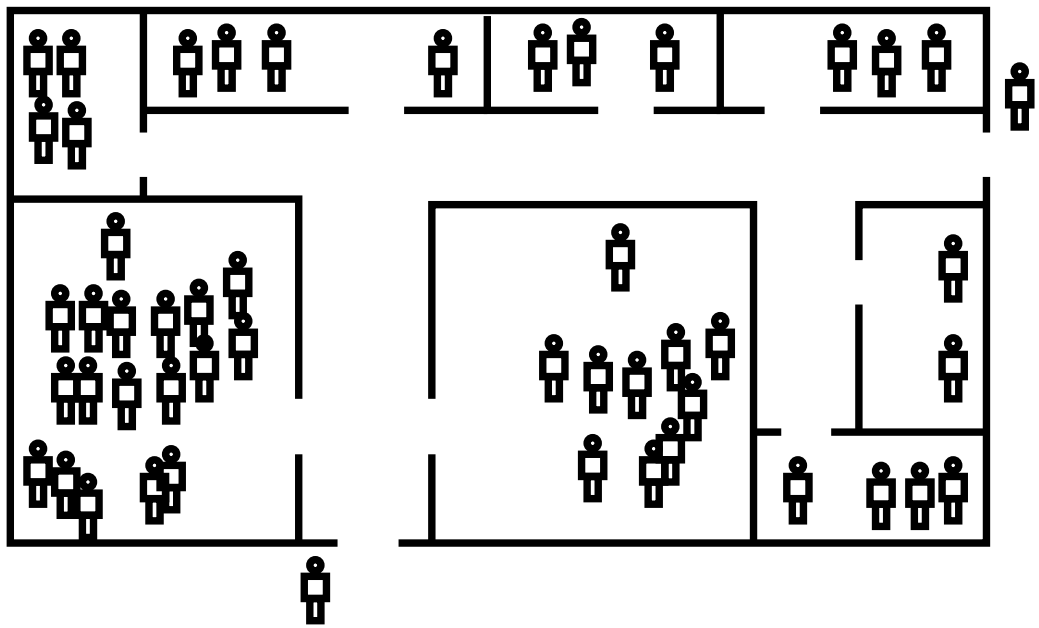
Eine Firma mit mehreren Produktionsstandorten hat mehrere Kunden zu beliefern. Wie geschieht dies am günstigsten unter Berücksichtigung der Transportkosten?



Beachte: Nur ein Produkttyp!

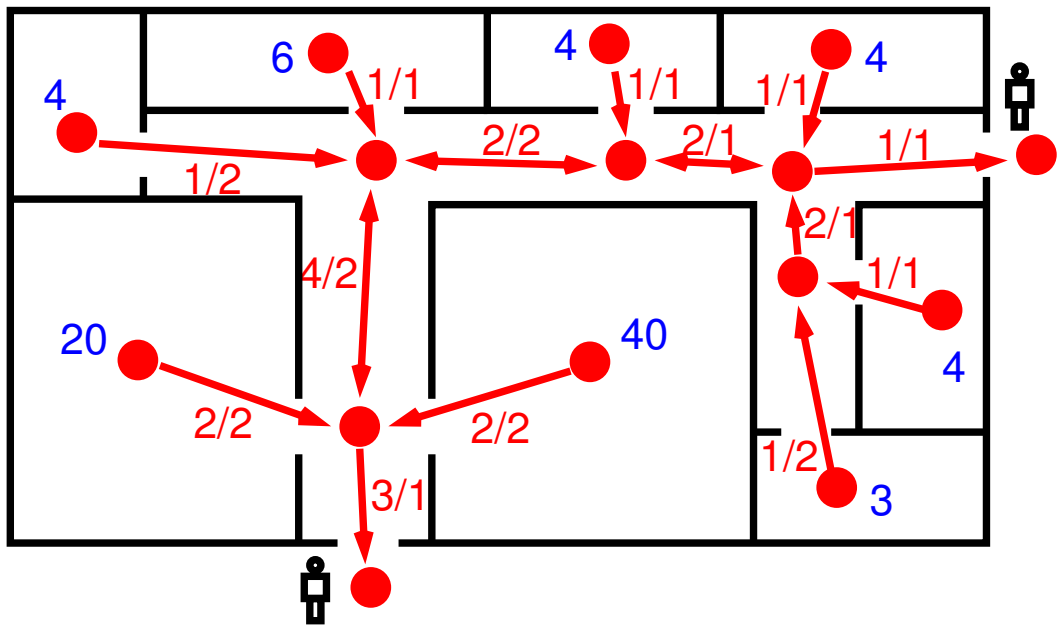
Beispiel: Evakuierungsplanung

Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Beispiel: Evakuierungsplanung

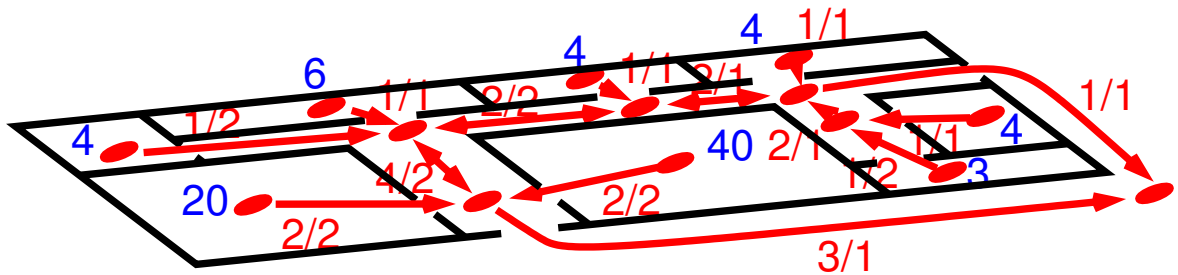
Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Kanten mit Kapazität und Durchquerungszeit

Beispiel: Evakuierungsplanung

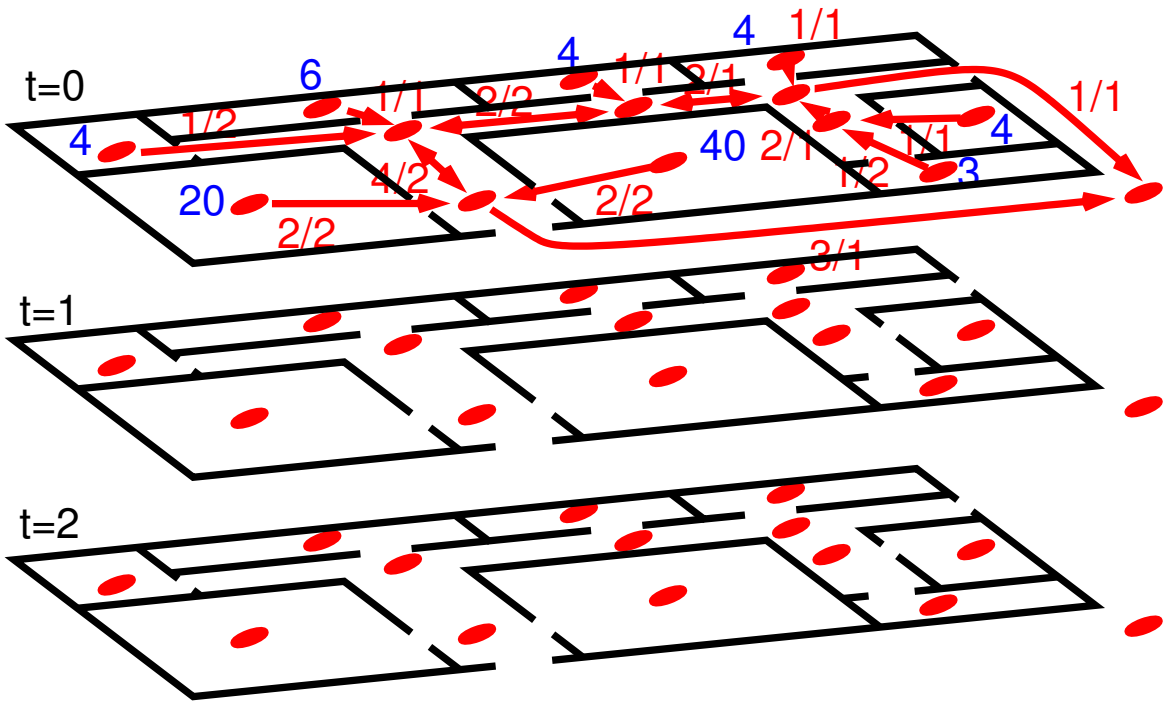
Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Geometrie nicht wichtig, vereinfachbar

Beispiel: Evakuierungsplanung

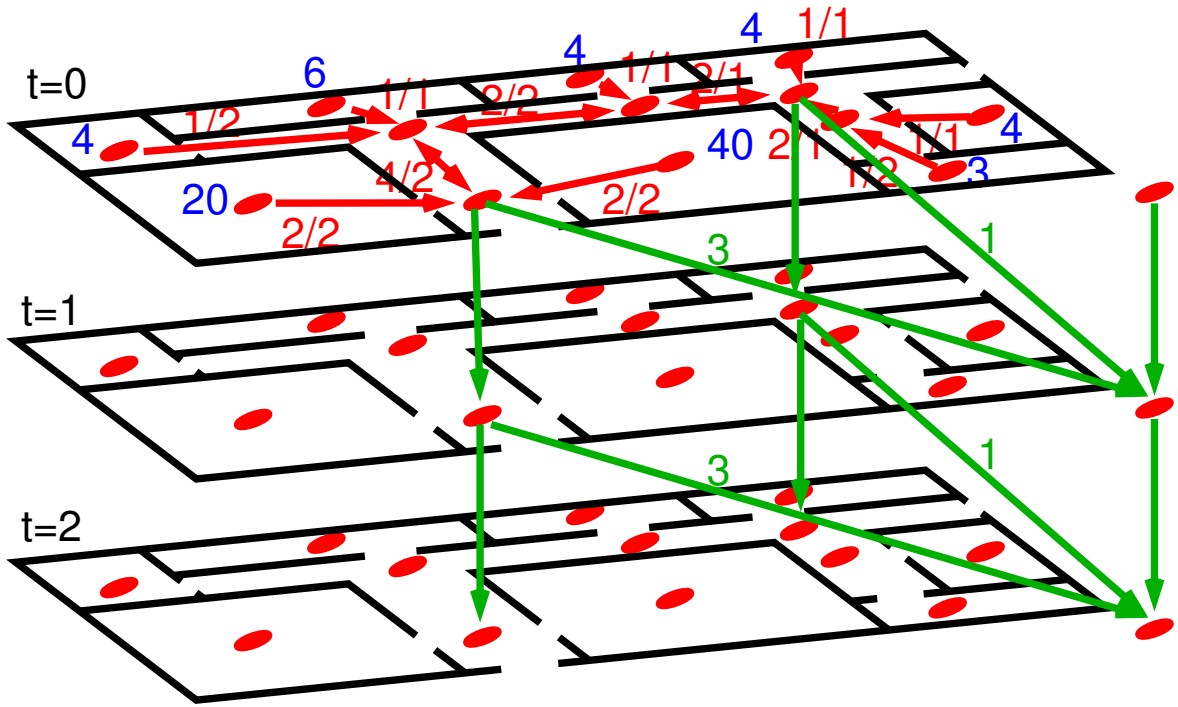
Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Diskretisierung der Zeit, ein Niveau pro Zeiteinheit

Beispiel: Evakuierungsplanung

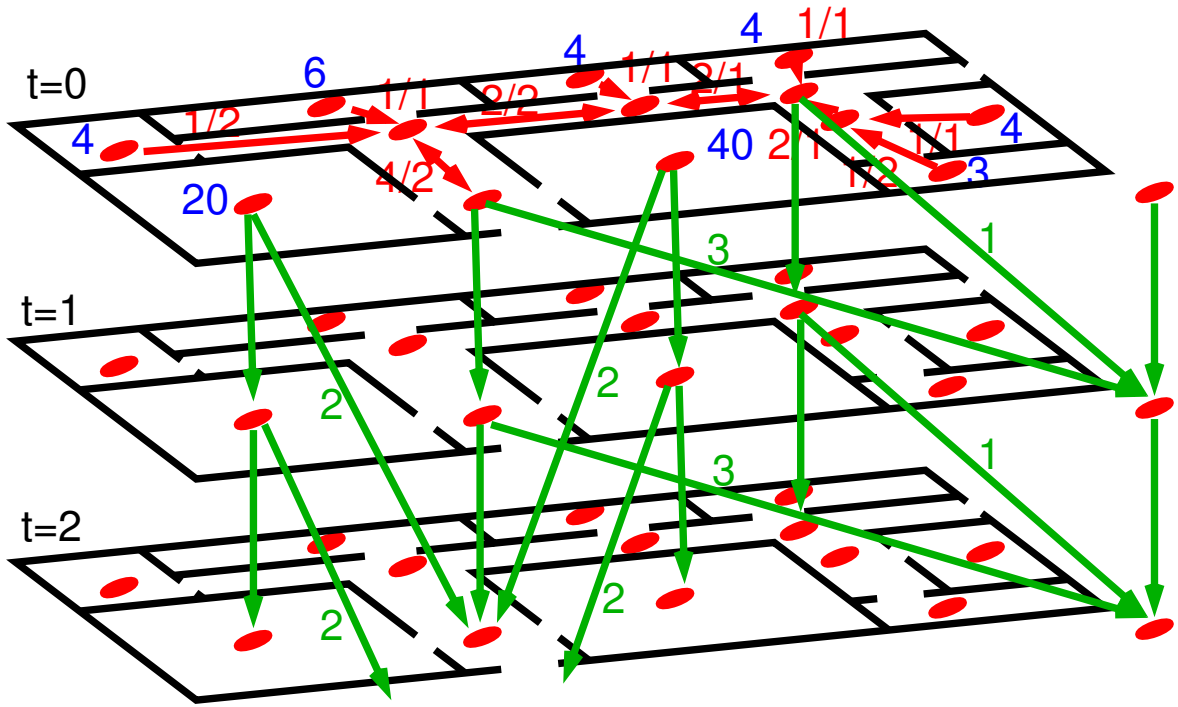
Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Durchquerungszeit verbindet Niveaus, Kapazität bleibt

Beispiel: Evakuierungsplanung

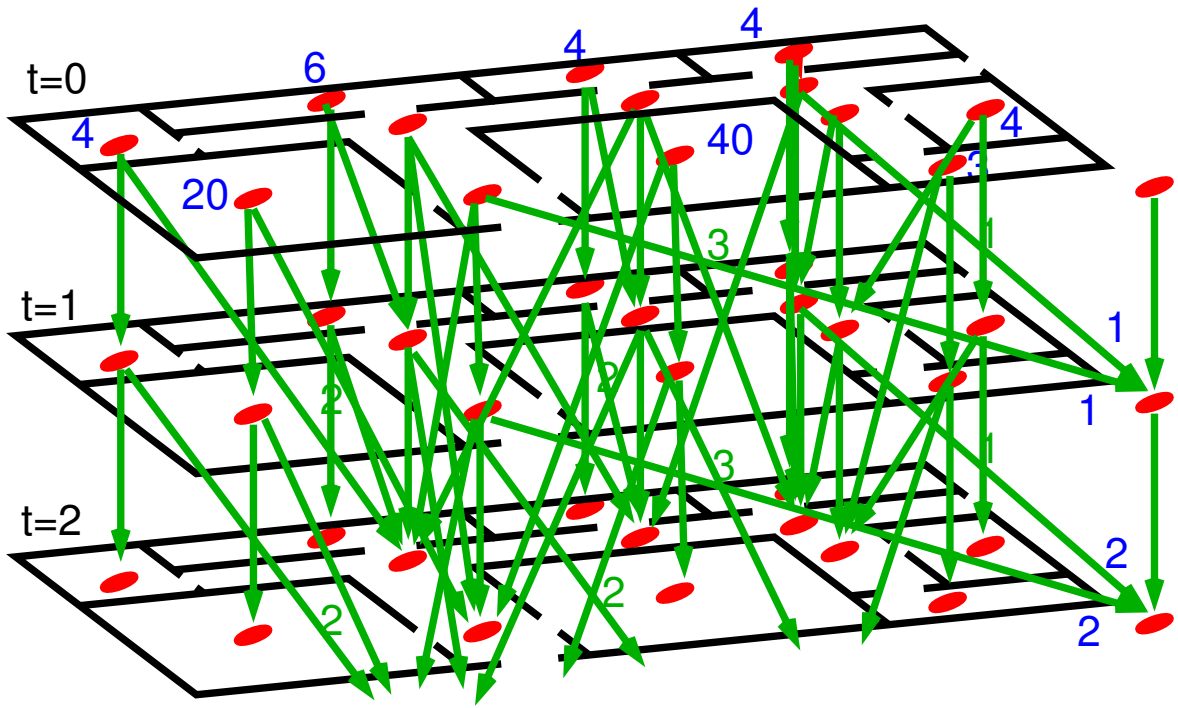
Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Durchquerungszeit verbindet Niveaus, Kapazität bleibt

Beispiel: Evakuierungsplanung

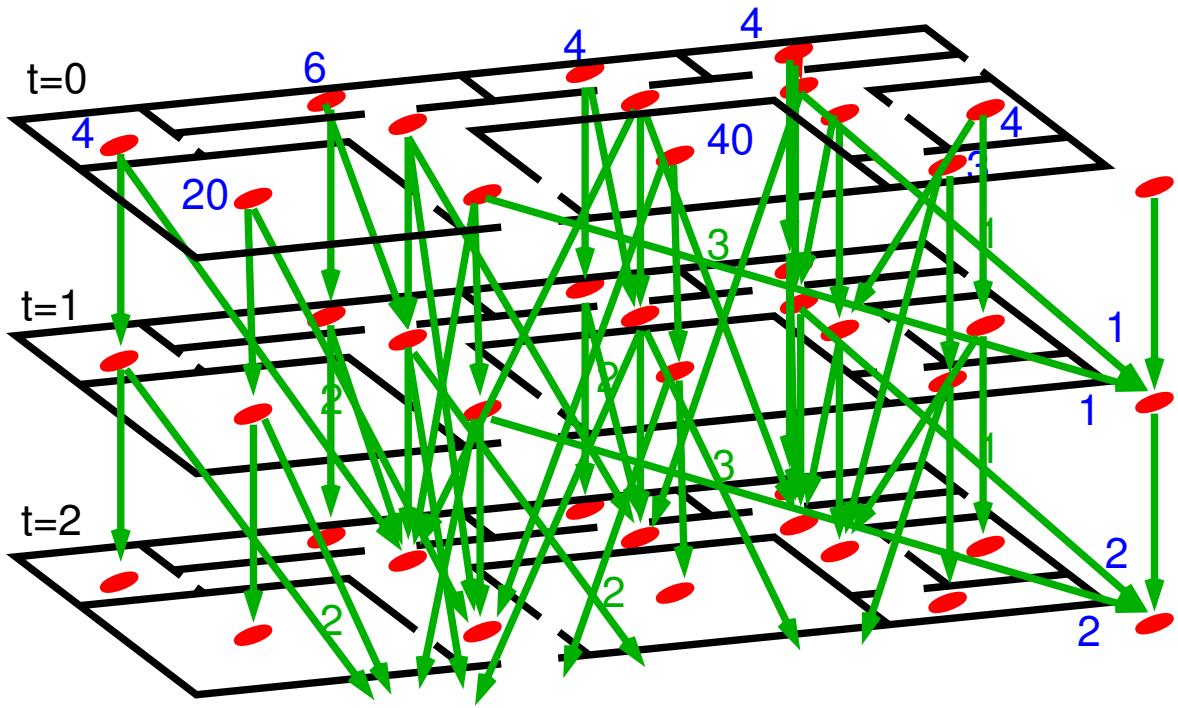
Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



Steigende Kosten auf den Ausgangskanten für rasches Verlassen

Beispiel: Evakuierungsplanung

Bestimme für jeden Raum den Fluchtweg, sodass das Gebäude schnellstmöglich geräumt ist. Pro Abschnitt sind Kapazität pro Zeiteinheit und Durchquerungszeit bekannt.



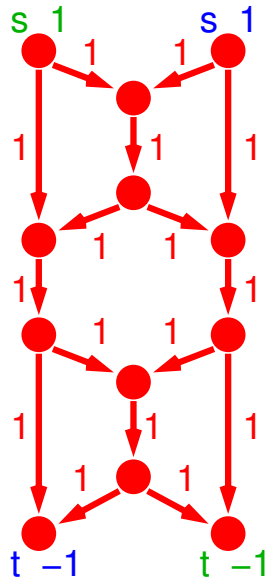
Steigende Kosten auf den Ausgangskanten für rasches Verlassen

Ansatz nur ok, wenn Personen nicht unterschieden werden müssen!

Inhaltsübersicht

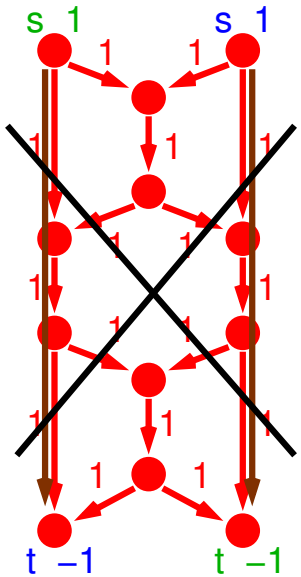
3.4 Mehrgüterflussprobleme (Multicommodity Flow)

In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



3.4 Mehrgüterflussprobleme (Multicommodity Flow)

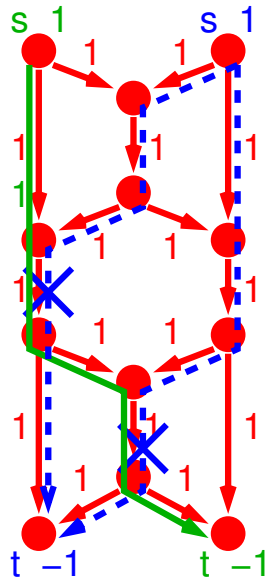
In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



Mischen verboten!

3.4 Mehrgüterflussprobleme (Multicommodity Flow)

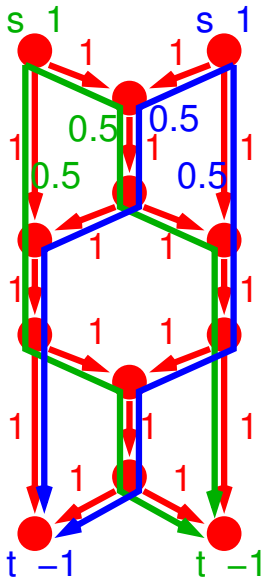
In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



ganzz. geht nicht

3.4 Mehrgüterflussprobleme (Multicommodity Flow)

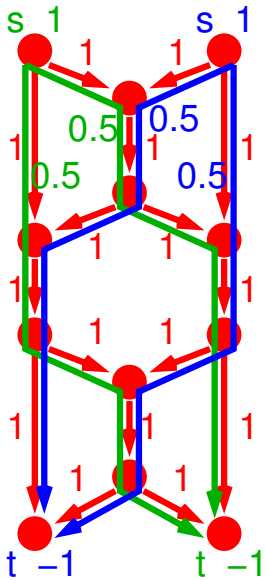
In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



gebrochen geht

3.4 Mehrgüterflussprobleme (Multicommodity Flow)

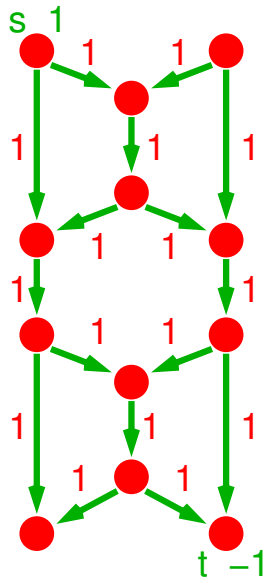
In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



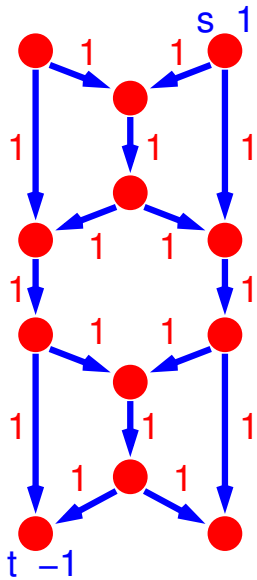
gebrochen geht

Umsetzung

$\longrightarrow$



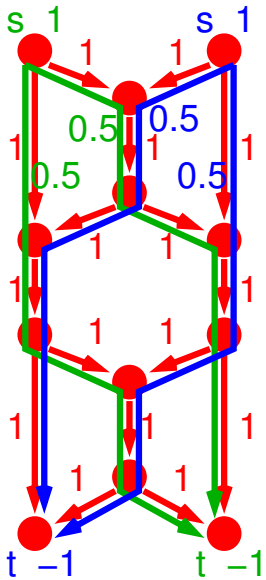
Kopie 1



Kopie 2

3.4 Mehrgüterflussprobleme (Multicommodity Flow)

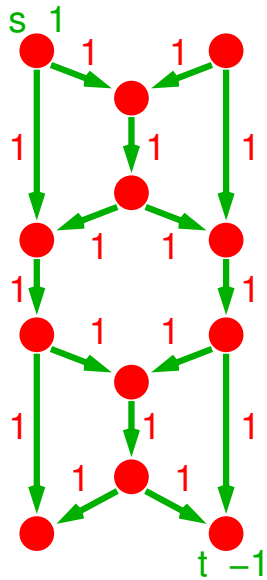
In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



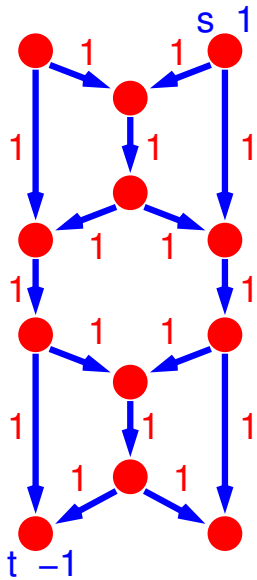
gebrochen geht

Umsetzung

$\longrightarrow$



Kopie 1

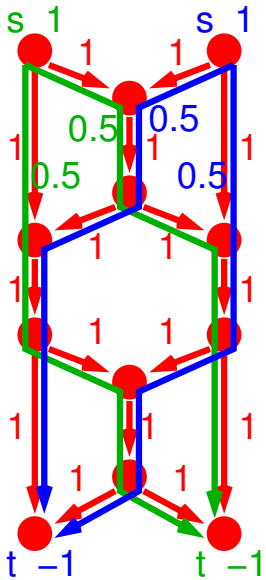


Kopie 2

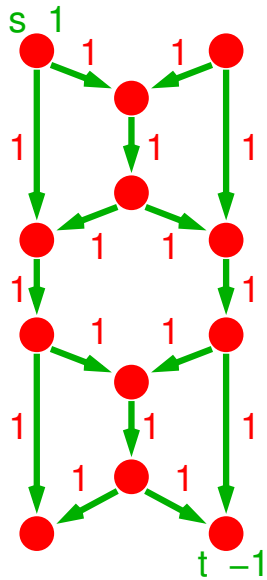
$$
\begin{array}{lrcl}
\min & c^{(1)\,T} x^{(1)} + c^{(2)\,T} x^{(2)} & & \\
\text{s.t.} & A x^{(1)} & = & b^{(1)} \\
 & A x^{(2)} & = & b^{(2)} \\
 & I x^{(1)} + I x^{(2)} & \leq & w \\
 & x^{(1)} \geq 0, \quad x^{(2)} \geq 0. & &
\end{array}
$$

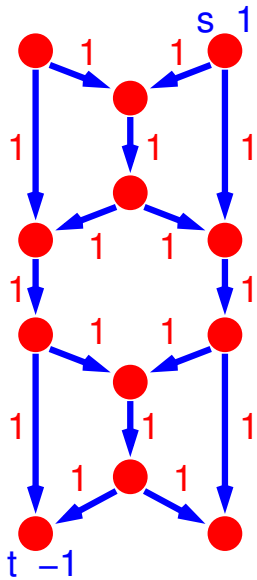
3.4 Mehrgüterflussprobleme (Multicommodity Flow)

In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.



Umsetzung $\longrightarrow$





gebrochen geht

Kopie 1

Kopie 2

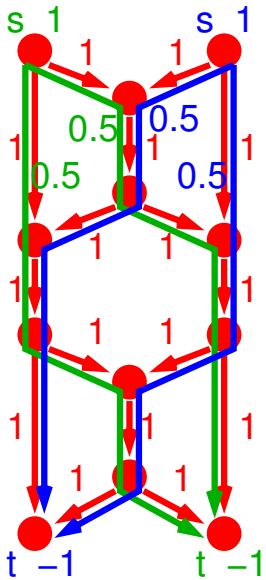
$$\begin{bmatrix} A & 0 \\ 0 & A \\ I & I \end{bmatrix}$$

i.A. nicht tot.unimod.!

$$\begin{array}{lllll} \min & c^{(1)T}x^{(1)} & + & c^{(2)T}x^{(2)} & \\ \text{s.t.} & Ax^{(1)} & & & = b^{(1)} \\ & & & Ax^{(2)} & = b^{(2)} \\ & Ix^{(1)} & + & Ix^{(2)} & \leq w \\ & x^{(1)} \geq 0, & & x^{(2)} \geq 0. & \end{array}$$

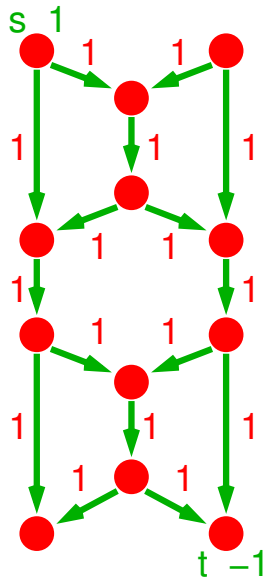
3.4 Mehrgüterflussprobleme (Multicommodity Flow)

In einem Neztwerk (D, w) sind für unterschiedliche Güter $K = \{1, \ldots, k\}$ mit Quellen/Senken (s_i, t_i) und Flusswerten f_i, $i \in K$, zulässige Flüsse $x^{(i)} \in \mathbb{R}^E$ zu finden, die in Summe die Kapazitätsbedingungen einhalten.

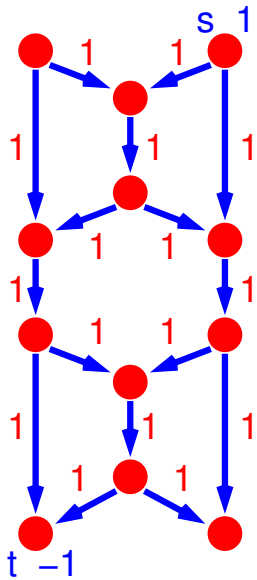


gebrochen geht

Umsetzung $\longrightarrow$



Kopie 1



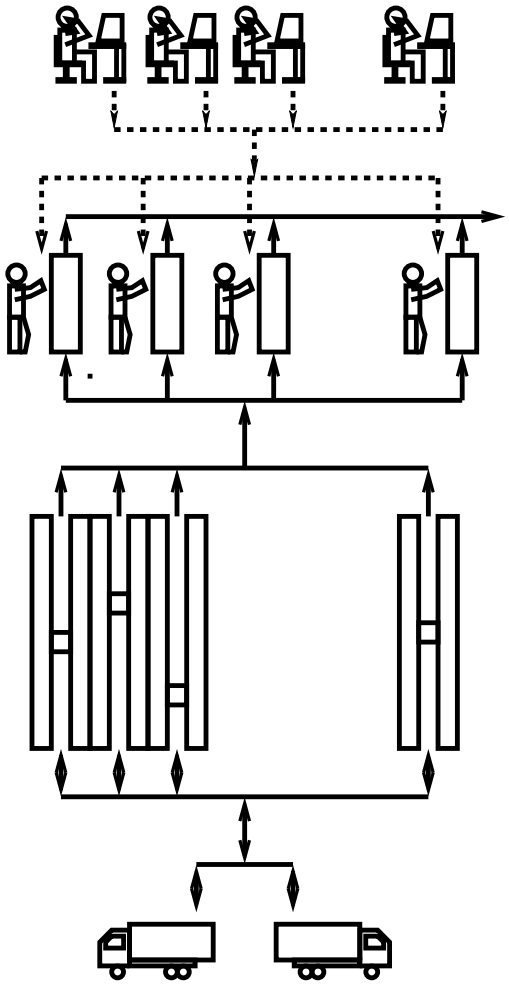
Kopie 2

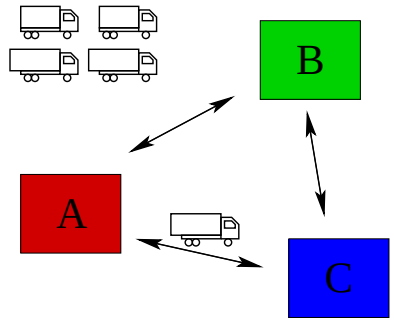
Gebrochen gut lösbar,
ganzzahlig SEHR schwer!

$$
\begin{array}{lrcl}
\min & {c^{(1)}}^T x^{(1)} & + & {c^{(2)}}^T x^{(2)} \\
\text{s.t.} & Ax^{(1)} & & = b^{(1)} \\
& & Ax^{(2)} & = b^{(2)} \\
& Ix^{(1)} & + & Ix^{(2)} \leq w \\
& x^{(1)} \geq 0, & & x^{(2)} \geq 0.
\end{array}
$$

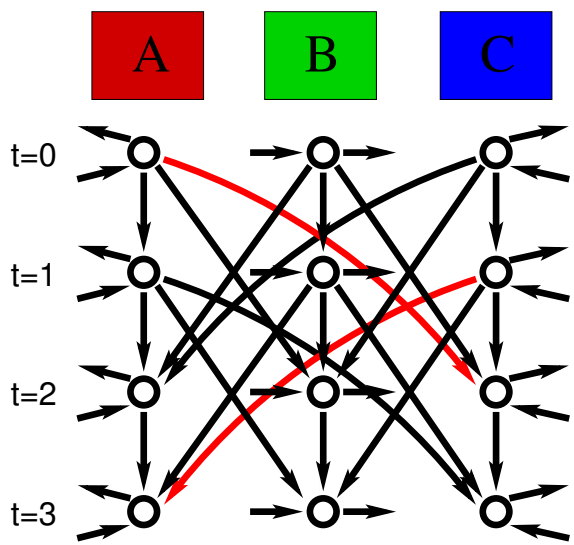
Beispiel: Logistik

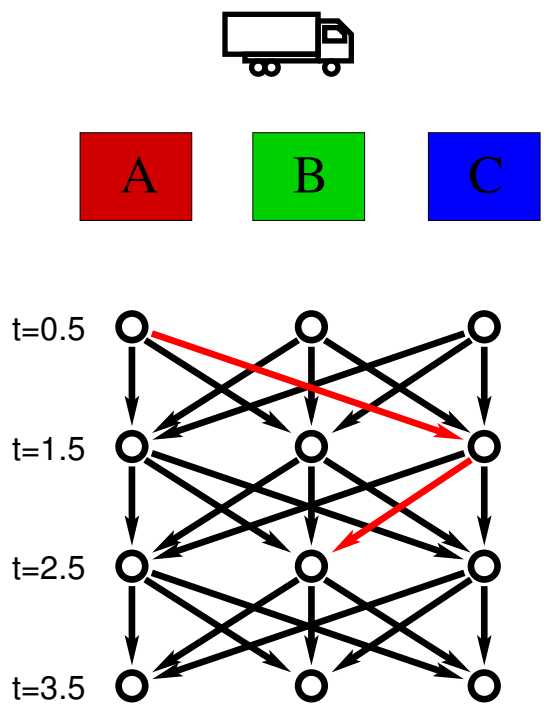
Paletten sind bedarfsgerecht mit LKWs zwischen Lagern zu verschieben





pro Artikel ein Palettengraph





Weitere Anwendungsbereiche

- gebrochen: Kapazitätsplanung
- ganzzahlig: Zeitdiskretisierte Routen- und Ablaufplanung

} für

  - Straßenverkehr
  - Schienenverkehr
  - Internet
  - Logistik (Engpassanalyse/Steuerung)
  - Produktion (Maschinenauslastung/-belegung)

Network-Design:
Auslegung soll Erfüllung möglichst aller Bedarfe auch bei Störung erlauben. [„Robuste“ Varianten sind extrem schwer!]

Mehrgüterflussprobleme werden oft als zugrundeliegendes Modell eingesetzt, das mit weiteren Bedingungen kombiniert wird.

Inhaltsübersicht

3.5 Ganzzahlige Optimierung (Integer Programming)

vorwiegend: Lineare Programme mit ausschließlich ganzzahligen Variablen
(sonst gemischt-ganzzahlige Optimierung/Mixed Integer Programming)

$$\begin{array}{ll} \max & c^T x \\ \text{s.t.} & Ax \leq b \\ & x \in \mathbb{Z}^n \end{array}$$

Enthält meist viele Binär-Variablen ($\{0,1\}$) für ja/nein Entscheidungen

---

Schwierigkeit: i.A. nicht „effizient“ lösbar, Komplexitätsklasse *NP*
$\Rightarrow$ exakte Lösung oft sehr Enumerations-lastig (system. Durchprobieren)

Exakte Lösung durch Kombination folgender Techniken:

- (obere) Schranken durch lineare/konvexe Relaxation
  verbessert durch Schnittebenenansätze
- zulässige Lösungen (untere Schranken) durch Rundungs- und Lokale-Suche-Heuristiken
- Enumeration durch Branch&Bound, Branch&Cut

Kombinatorische Optimierung

Mathematisch: Gegeben eine endliche Grundmenge Ω, eine Menge zulässiger Teilmengen $\mathcal{F} \subseteq 2^\Omega$ [Potenzmenge, Menge aller Teilmengen] und eine Zielfunktion $c : \mathcal{F} \to \mathbb{Q}$, bestimme

$$\max\{c(F) : F \in \mathcal{F}\} \quad \text{oder} \quad F \in \operatorname{Argmax}\{c(F) : F \in \mathcal{F}\}$$

Hier nur lineares c: $c(F) := \sum_{e \in F} c_e$ mit $c \in \mathbb{Q}^\Omega$.

Bsp1: Matching maximaler Kardinalität in $G = (V, E)$:
$\Omega = E$, $\mathcal{F} = \{M \subseteq E : M \text{ Matching in } G\}$, $c = \mathbf{1}$

Bsp2: Minimum Vertex Cover für $G = (V, E)$:
$\Omega = V$, $\mathcal{F} = \{V' \subseteq V : e \cap V' \neq \emptyset \text{ für } e \in E\}$, $c = -\mathbf{1}$

Formulierung über Binäre Programme:

Notation: Inzidenz-/Charakteristischer Vektor $\chi_\Omega(F) \in \{0,1\}^\Omega$ für $F \subseteq \Omega$
[kurz $\chi(F)$, erfüllt $[\chi(F)]_e = 1 \Leftrightarrow e \in F$]

Ein lineares Programm $\max\{c^T x : Ax \leq b, x \in [0,1]^\Omega\}$ heißt Formulierung des kombinatorischen Optimierungsproblems, falls

$$\{x \in \{0,1\}^\Omega : Ax \leq b\} = \{\chi(F) : F \in \mathcal{F}\}.$$

(Algorithmische) Komplexität von Problemen

Eine Instanz I eines Problems ist eine konkrete Wertebelegung der Problemdaten; ihre Größe $|I|$ ist die Kodierungslänge, also die Anzahl der Zeichen im Beschreibungsstring nach einem sinnvollen Kodierungsschema.

Die Laufzeit eines Algorithmus für eine Instanz ist die Anzahl der ausgeführten elementaren Operationen (ein Symbol lesen/schreiben, Bytes addieren/multiplizieren/vergleichen, etc.)

Ein Algorithmus löst ein Problem polynomial oder effizient, wenn er für jede Instanz die richtige Antwort innerhalb einer Laufzeit liefert, die durch ein Polynom in der Kodierungslänge beschränkt ist.

Ein Problem heißt polynomial/effizient lösbar, wenn es einen Algorithmus gibt, der es effizient löst. Die Klasse P umfasst alle Probleme, die effizient lösbar sind.

Bsp1: Lineare Optimierung ist in P, ABER Simplex löst es nicht effizient.
Bsp2: Kardinalitätsmaximales Matching in allgemeinen Graphen ist in P.
Bsp3: Minimum Vertex Cover in bipartiten Graphen ist in P.

Entscheidungsprobleme und die Klasse *NP*

In einem Entscheidungsproblem ist jede Instanz eine Frage, die entweder mit „Ja“ oder mit „Nein“ zu beantworten ist.

Ein Entscheidungsproblem ist nichtdeterministisch polynomial lösbar, wenn für jede „Ja“-Instanz I ein Lösungsstring (das Zertifikat) existiert, mit dem die Richtigkeit der „Ja“-Antwort in Laufzeit polynomial beschränkt in $|I|$ nachgewiesen werden kann. [Es geht nur um „Ja“!]

Die Klasse NP umfasst alle Entscheidungsprobleme, die nichtdeterministisch polynomial lösbar sind. Es gilt: $P \subseteq NP$

Entscheidungsprobleme und die Klasse *NP*

In einem Entscheidungsproblem ist jede Instanz eine Frage, die entweder mit „Ja“ oder mit „Nein“ zu beantworten ist.

Ein Entscheidungsproblem ist nichtdeterministisch polynomial lösbar, wenn für jede „Ja“-Instanz I ein Lösungsstring (das Zertifikat) existiert, mit dem die Richtigkeit der „Ja“-Antwort in Laufzeit polynomial beschränkt in $|I|$ nachgewiesen werden kann. [Es geht nur um „Ja“!]

Die Klasse NP umfasst alle Entscheidungsprobleme, die nichtdeterministisch polynomial lösbar sind. Es gilt: $P \subseteq NP$

Bsp: **Hamilton'scher Kreis:** In einem Graphen $G = (V, E)$ heißt eine Kantenmenge $C = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}\} \subseteq E$ ein Kreis (der Länge k), falls $v_i \neq v_j$ für $i \neq j$.
Ein Kreis C heißt hamiltonsch, falls $|C| = |V|$.

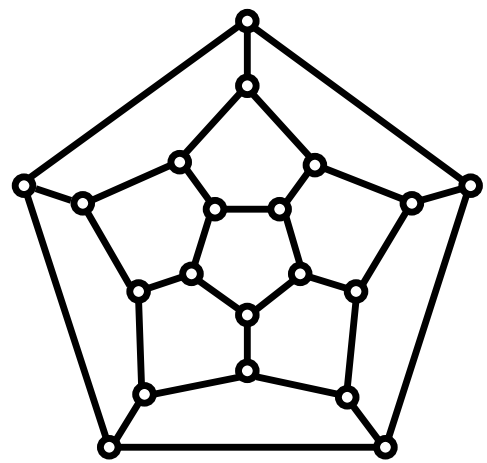
Entscheidungsprobleme und die Klasse *NP*

In einem Entscheidungsproblem ist jede Instanz eine Frage, die entweder mit „Ja“ oder mit „Nein“ zu beantworten ist.

Ein Entscheidungsproblem ist nichtdeterministisch polynomial lösbar, wenn für jede „Ja“-Instanz I ein Lösungsstring (das Zertifikat) existiert, mit dem die Richtigkeit der „Ja“-Antwort in Laufzeit polynomial beschränkt in $|I|$ nachgewiesen werden kann. [Es geht nur um „Ja“!]

Die Klasse NP umfasst alle Entscheidungsprobleme, die nichtdeterministisch polynomial lösbar sind. Es gilt: $P \subseteq NP$

Bsp: **Hamilton'scher Kreis:** In einem Graphen $G = (V, E)$ heißt eine Kantenmenge $C = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}\} \subseteq E$ ein Kreis (der Länge k), falls $v_i \neq v_j$ für $i \neq j$.
Ein Kreis C heißt hamiltonsch, falls $|C| = |V|$.



Entscheidungsproblem:
Enthält G einen Hamilton'schen Kreis?

Ja-Antwort ist effizient überprüfbar,
das Problem ist in *NP*

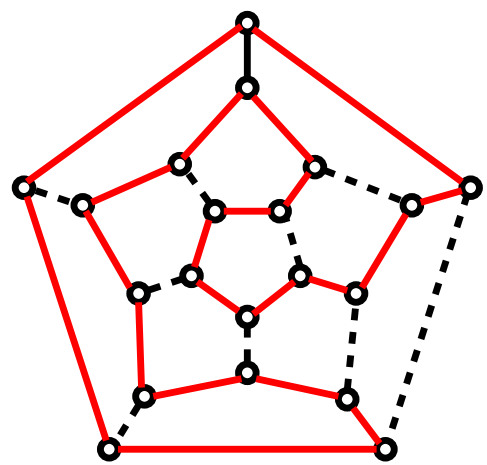
Entscheidungsprobleme und die Klasse *NP*

In einem Entscheidungsproblem ist jede Instanz eine Frage, die entweder mit „Ja“ oder mit „Nein“ zu beantworten ist.

Ein Entscheidungsproblem ist nichtdeterministisch polynomial lösbar, wenn für jede „Ja“-Instanz I ein Lösungsstring (das Zertifikat) existiert, mit dem die Richtigkeit der „Ja“-Antwort in Laufzeit polynomial beschränkt in $|I|$ nachgewiesen werden kann. [Es geht nur um „Ja“!]

Die Klasse NP umfasst alle Entscheidungsprobleme, die nichtdeterministisch polynomial lösbar sind. Es gilt: $P \subseteq NP$

Bsp: **Hamilton'scher Kreis:** In einem Graphen $G = (V, E)$ heißt eine Kantenmenge $C = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}\} \subseteq E$ ein Kreis (der Länge k), falls $v_i \neq v_j$ für $i \neq j$.
Ein Kreis C heißt hamiltonsch, falls $|C| = |V|$.



Entscheidungsproblem:
Enthält G einen Hamilton'schen Kreis?

Ja-Antwort ist effizient überprüfbar,
das Problem ist in *NP*

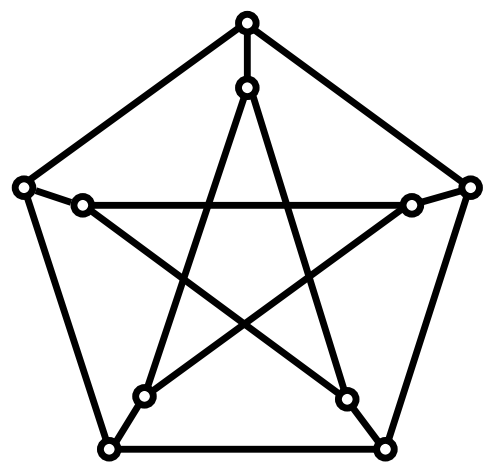
Entscheidungsprobleme und die Klasse *NP*

In einem Entscheidungsproblem ist jede Instanz eine Frage, die entweder mit „Ja“ oder mit „Nein“ zu beantworten ist.

Ein Entscheidungsproblem ist nichtdeterministisch polynomial lösbar, wenn für jede „Ja“-Instanz I ein Lösungsstring (das Zertifikat) existiert, mit dem die Richtigkeit der „Ja“-Antwort in Laufzeit polynomial beschränkt in $|I|$ nachgewiesen werden kann. [Es geht nur um „Ja“!]

Die Klasse NP umfasst alle Entscheidungsprobleme, die nichtdeterministisch polynomial lösbar sind. Es gilt: $P \subseteq NP$

Bsp: **Hamilton'scher Kreis:** In einem Graphen $G = (V, E)$ heißt eine Kantenmenge $C = \{\{v_1, v_2\}, \{v_2, v_3\}, \ldots, \{v_{k-1}, v_k\}, \{v_k, v_1\}\} \subseteq E$ ein Kreis (der Länge k), falls $v_i \neq v_j$ für $i \neq j$.
Ein Kreis C heißt hamiltonsch, falls $|C| = |V|$.



Entscheidungsproblem:
Enthält G einen Hamilton'schen Kreis?

Ja-Antwort ist effizient überprüfbar,
das Problem ist in *NP*

NP-vollständige Probleme

Ein Entscheidungsproblem P_1 ist polynomial transformierbar auf ein Entscheidungsproblem P_2, wenn es einen Algorithmus gibt, der jede Instanz I_1 von P_1 in Laufzeit polynomial in $|I_1|$ in eine Instanz I_2 von P_2 transformiert, sodass I_2 genau dann Ja-Instanz von P_2 ist, wenn I_1 Ja-Instanz von P_1 ist.

Ist P_1 polynomial auf P_2 transformierbar, dann löst ein effizienter Algorithmus für P_2 auch P_1 effizient; P_2 ist mindestens so schwer wie P_1.

Ist $\bar{P} \in NP$ und ist jedes $\hat{P} \in NP$ polynomial auf $\bar{P}$ transformierbar, so heißt $\bar{P}$ *NP*-vollständig.

Kann ein *NP*-vollständiges Problem $\bar{P}$ polynomial auf ein Problem $\hat{P} \in NP$ transformiert werden, ist auch $\hat{P}$ *NP*-vollständig; sie sind gleich schwer.

NP-vollständige Probleme

Ein Entscheidungsproblem P_1 ist polynomial transformierbar auf ein Entscheidungsproblem P_2, wenn es einen Algorithmus gibt, der jede Instanz I_1 von P_1 in Laufzeit polynomial in $|I_1|$ in eine Instanz I_2 von P_2 transformiert, sodass I_2 genau dann Ja-Instanz von P_2 ist, wenn I_1 Ja-Instanz von P_1 ist.

Ist P_1 polynomial auf P_2 transformierbar, dann löst ein effizienter Algorithmus für P_2 auch P_1 effizient; P_2 ist mindestens so schwer wie P_1.

Ist $\bar{P} \in NP$ und ist jedes $\hat{P} \in NP$ polynomial auf $\bar{P}$ transformierbar, so heißt $\bar{P}$ *NP*-vollständig.

Kann ein *NP*-vollständiges Problem $\bar{P}$ polynomial auf ein Problem $\hat{P} \in NP$ transformiert werden, ist auch $\hat{P}$ *NP*-vollständig; sie sind gleich schwer.

Es gibt dicke Sammlungen *NP*-vollständiger Probleme, dazu gehören:

- ganzzahlige Optimierung (in Entscheidungsversion)
- ganzzahliger Mehrgüterfluss
- Hamilton'scher Kreis
- Minimum Vertex Cover auf allgemeinen Graphen
- Das Rucksack-Problem (für große Zahlen)

NP-vollständige Probleme

Ein Entscheidungsproblem P_1 ist polynomial transformierbar auf ein Entscheidungsproblem P_2, wenn es einen Algorithmus gibt, der jede Instanz I_1 von P_1 in Laufzeit polynomial in $|I_1|$ in eine Instanz I_2 von P_2 transformiert, sodass I_2 genau dann Ja-Instanz von P_2 ist, wenn I_1 Ja-Instanz von P_1 ist.

Ist P_1 polynomial auf P_2 transformierbar, dann löst ein effizienter Algorithmus für P_2 auch P_1 effizient; P_2 ist mindestens so schwer wie P_1.

Ist $\bar{P} \in NP$ und ist jedes $\hat{P} \in NP$ polynomial auf $\bar{P}$ transformierbar, so heißt $\bar{P}$ *NP*-vollständig.

Kann ein *NP*-vollständiges Problem $\bar{P}$ polynomial auf ein Problem $\hat{P} \in NP$ transformiert werden, ist auch $\hat{P}$ *NP*-vollständig; sie sind gleich schwer.

Gibt es einen effizienten Algorithmus für eines der *NP*-vollständigen Probleme, sind alle effizient lösbar. Man vermutet seit Jahren: $P \neq NP$.

Will man alle Instanzen lösen können, ist wahrscheinlich eine teilweise Enumeration unvermeidbar.

Ein Problem ist *NP*-schwer, wenn damit ein *NP*-vollständiges lösbar wäre.

Inhaltsübersicht

Ganzzahlige Optimierung

3.6 Branch-and-Bound

Beim systematischen Enumerieren aller Lösungen sollen möglichst viele frühzeitig durch obere und untere Schranken ausgeschlossen werden.

Bsp: $\{0,1\}$-Rucksack: Gewichte $a \in \mathbb{N}^n$, Kapazität $b \in \mathbb{N}$, Nutzen $c \in \mathbb{N}^n$,

$$\max\ c^T x \quad \text{s.t.} \quad a^T x \leq b,\ x \in \{0,1\}^n$$

Obere Schranke: $\max\ c^T x$ s.t. $a^T x \leq b,\ x \in [0,1]^n$ [LP-Relaxation]
Untere Schranke: Sortiere nach Nutzen/Gewicht und fülle danach auf

---

Ablaufskizze (für Maximierungsprobleme):
M ... Menge offener Probleme, anfangs $M = \{\text{Ursprungsproblem}\}$
$\underline{f}$ ... Wert der besten bekannten Lösung, anfangs $\underline{f} = -\infty$

1. Falls $M = \emptyset$ STOP, sonst wähle $P \in M$, $M \leftarrow M \setminus \{P\}$
2. Berechne obere Schranke $\bar{f}(P)$.
3. Falls $\bar{f}(P) < \underline{f}$ (P enthält keine OL), gehe zu 1.
4. Berechne zulässige Lösung $\hat{f}(P)$ für P (untere Schranke).
5. Ist $\hat{f}(P) > \underline{f}$ (neue beste Lösung), setze $\underline{f} \leftarrow \hat{f}(P)$
6. Ist $\bar{f}(P) = \hat{f}(P)$ (keine bessere Lösung in P), gehe zu 1.
7. Teile P in „kleinere“ Teilprobleme P_i, $M \leftarrow M \cup \{P_1, \ldots, P_k\}$
8. Gehe zu 1.

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

$$1 \quad \leftarrow \quad x_A \quad \rightarrow \quad 0$$

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

$$1 \quad \leftarrow \quad x_A \quad \rightarrow \quad 0$$

P_2: $x_A = 1 \Rightarrow x_B = x_C = 0$
OS: $A + D + \frac{1}{3}F = 26\frac{2}{3}$
OS $< 30 \Rightarrow$ keine OL $\square$

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_A \quad \rightarrow \quad 0$

P_2: $x_A = 1 \Rightarrow x_B = x_C = 0$
OS: $A + D + \frac{1}{3}F = 26\frac{2}{3}$
OS $< 30 \Rightarrow$ keine OL $\square$

P_3: $x_A = 0$
OS: $C + D + F + \frac{1}{4}E = 31\frac{1}{2}$
US: $C + D + F = 30$

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_A \quad \rightarrow \quad 0$

P_2: $x_A = 1 \Rightarrow x_B = x_C = 0$
OS: $A + D + \frac{1}{3}F = 26\frac{2}{3}$
OS $< 30 \Rightarrow$ keine OL $\square$

P_3: $x_A = 0$
OS: $C + D + F + \frac{1}{4}E = 31\frac{1}{2}$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_E \quad \rightarrow \quad 0$

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_A \quad \rightarrow \quad 0$

P_2: $x_A = 1 \Rightarrow x_B = x_C = 0$
OS: $A + D + \frac{1}{3}F = 26\frac{2}{3}$
OS $< 30 \Rightarrow$ keine OL □

P_3: $x_A = 0$
OS: $C + D + F + \frac{1}{4}E = 31\frac{1}{2}$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_E \quad \rightarrow \quad 0$

P_4: $x_A = 0,\ x_E = 1$
OS: $E + C + D = 31$
US: $E + C + D = 31$ □

Beispiel: $\{0,1\}$-Rucksackproblem

Gegenstand	A	B	C	D	E	F	Kapazität
Gewicht(a)	9	7	6	4	4	3	14
Nutzen (c)	18	6	18	7	6	5	

Sortierung Nutzen/Gewicht: $C > A > D > F > E > B$.

Obere Schranke: max $c^T x$ s.t. $a^T x \leq 14,\ x \in [0,1]^6$ [LP-Relaxation]

Untere Schranke: Nach Sortierung möglichst lange auffüllen

P_1: Originalproblem
OS: $C + \frac{8}{9}A = 34$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_A \quad \rightarrow \quad 0$

P_2: $x_A = 1 \Rightarrow x_B = x_C = 0$
OS: $A + D + \frac{1}{3}F = 26\frac{2}{3}$
OS $< 30 \Rightarrow$ keine OL □

P_3: $x_A = 0$
OS: $C + D + F + \frac{1}{4}E = 31\frac{1}{2}$
US: $C + D + F = 30$

$1 \quad \leftarrow \quad x_E \quad \rightarrow \quad 0$

P_4: $x_A = 0,\ x_E = 1$
OS: $E + C + D = 31$
US: $E + C + D = 31$ □

P_5: $x_A = x_E = 0$
OS: $C + D + F + \frac{1}{7}B = 30\frac{6}{7}$
OS $< 31 \Rightarrow$ keine OL □

Immer dann wird der Branch&Bound Baum groß werden, wenn viele Lösungen sehr nahe an der Optimallösung sind.

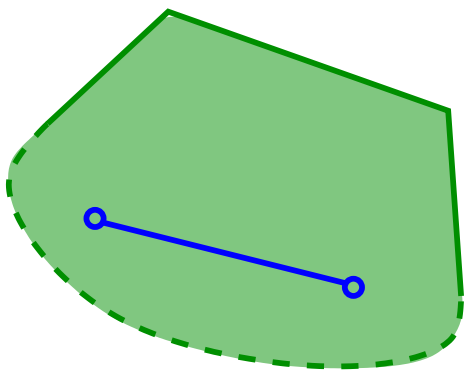
Entscheidend für den Erfolg von Branch&Bound:

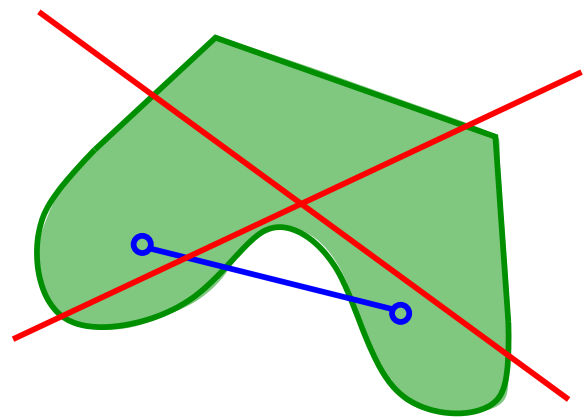
Wie kommt man zu guten oberen und unteren Schranken?

Inhaltsübersicht

3.7 Konvexe Mengen und konvexe Hülle

Eine Menge $C \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x, y \in C$ auch die Verbindungsstrecke $[x, y] := \{\alpha x + (1 - \alpha)y : \alpha \in [0, 1]\}$ in C liegt.



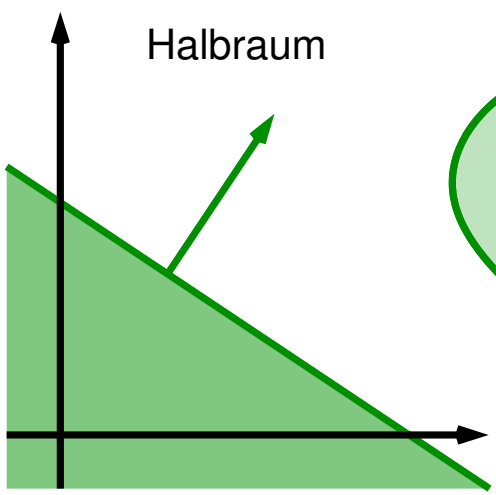


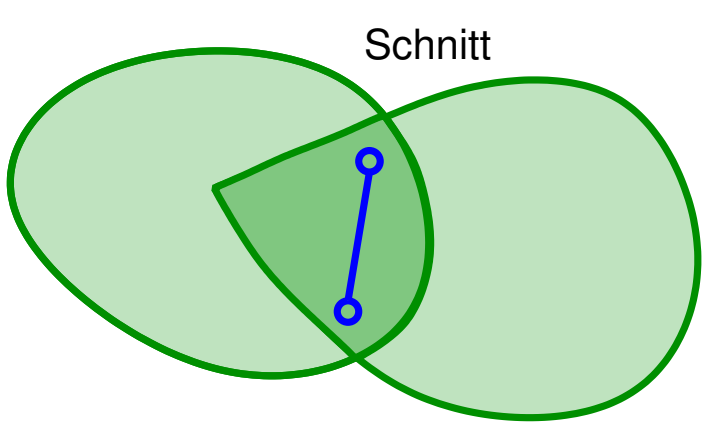
3.7 Konvexe Mengen und konvexe Hülle

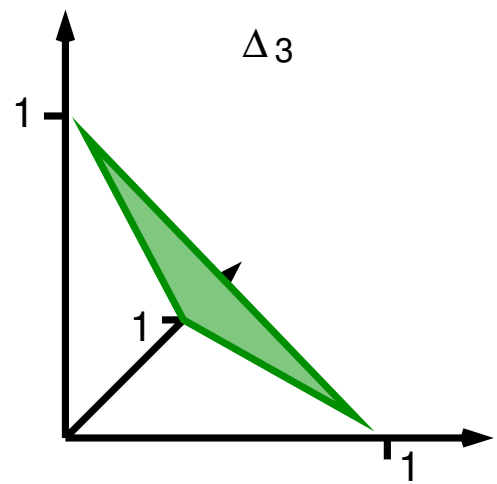
Eine Menge $C \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x, y \in C$ auch die Verbindungsstrecke $[x, y] := \{\alpha x + (1-\alpha)y : \alpha \in [0,1]\}$ in C liegt.

Bspe: $\emptyset$, $\mathbb{R}^n$, Halbräume, der Schnitt konvexer Mengen ist konvex,

Polyeder, der k-dim. Einheitssimplex $\Delta_k := \{\alpha \in \mathbb{R}^k_+ : \sum_{i=1}^k \alpha_i = 1\}$







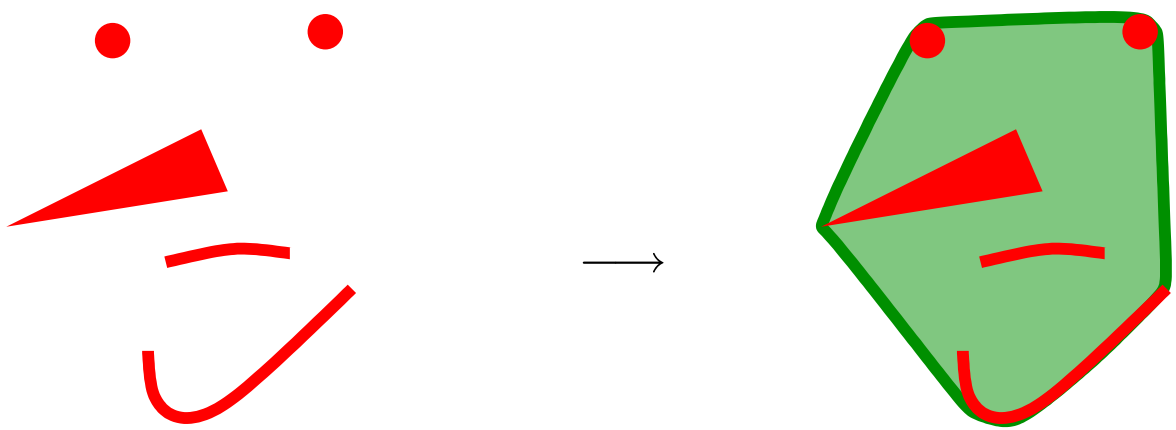
3.7 Konvexe Mengen und konvexe Hülle

Eine Menge $C \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x, y \in C$ auch die Verbindungsstrecke $[x, y] := \{\alpha x + (1-\alpha)y : \alpha \in [0,1]\}$ in C liegt.

Bspe: $\emptyset$, $\mathbb{R}^n$, Halbräume, der Schnitt konvexer Mengen ist konvex,

Polyeder, der k-dim. Einheitssimplex $\Delta_k := \{\alpha \in \mathbb{R}^k_+ : \sum_{i=1}^k \alpha_i = 1\}$

Für $S \subseteq \mathbb{R}^n$ ist die konvexe Hülle der Schnitt aller konvexen Mengen, die S enthalten, $\operatorname{conv} S := \bigcap\{C \text{ konvex} : S \subseteq C\}$.



3.7 Konvexe Mengen und konvexe Hülle

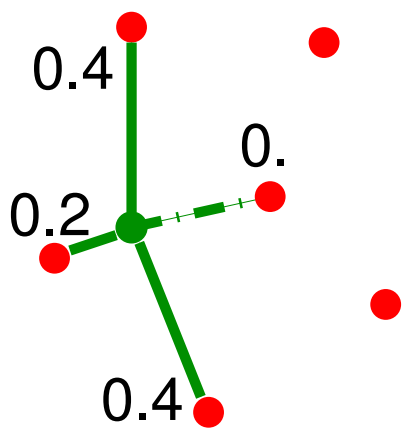
Eine Menge $C \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x, y \in C$ auch die Verbindungsstrecke $[x, y] := \{\alpha x + (1-\alpha)y : \alpha \in [0,1]\}$ in C liegt.

Bspe: $\emptyset$, $\mathbb{R}^n$, Halbräume, der Schnitt konvexer Mengen ist konvex,

Polyeder, der k-dim. Einheitssimplex $\Delta_k := \{\alpha \in \mathbb{R}^k_+ : \sum_{i=1}^k \alpha_i = 1\}$

Für $S \subseteq \mathbb{R}^n$ ist die konvexe Hülle der Schnitt aller konvexen Mengen, die S enthalten, $\operatorname{conv} S := \bigcap\{C \text{ konvex} : S \subseteq C\}$.

Für gegebene Punkte $x^{(i)} \in \mathbb{R}^n$, $i \in \{1, \ldots, k\}$ und $\alpha \in \Delta_k$ heißt $x = \sum \alpha_i x^{(i)}$ Konvexkombination der Punkte $x^{(i)}$.



3.7 Konvexe Mengen und konvexe Hülle

Eine Menge $C \subseteq \mathbb{R}^n$ heißt konvex, wenn für alle $x, y \in C$ auch die Verbindungsstrecke $[x, y] := \{\alpha x + (1-\alpha)y : \alpha \in [0,1]\}$ in C liegt.

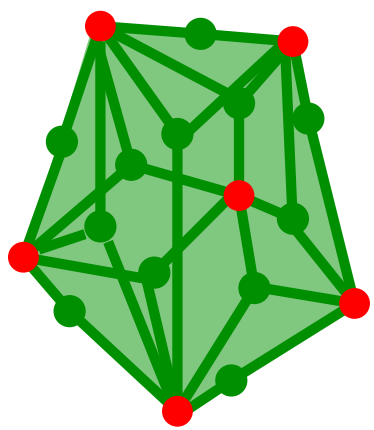
Bspe: $\emptyset$, $\mathbb{R}^n$, Halbräume, der Schnitt konvexer Mengen ist konvex,

Polyeder, der k-dim. Einheitssimplex $\Delta_k := \{\alpha \in \mathbb{R}^k_+ : \sum_{i=1}^k \alpha_i = 1\}$

Für $S \subseteq \mathbb{R}^n$ ist die konvexe Hülle der Schnitt aller konvexen Mengen, die S enthalten, $\operatorname{conv} S := \bigcap\{C \text{ konvex} : S \subseteq C\}$.

Für gegebene Punkte $x^{(i)} \in \mathbb{R}^n$, $i \in \{1, \ldots, k\}$ und $\alpha \in \Delta_k$ heißt $x = \sum \alpha_i x^{(i)}$ Konvexkombination der Punkte $x^{(i)}$.

$\operatorname{conv} S$ ist die Menge aller Konvexkombinationen endlich vieler Punkte aus S, $\operatorname{conv} S = \{\sum_{i=1}^k \alpha_i x^{(i)} : x^{(i)} \in S, i = 1, \ldots, k \in \mathbb{N}, \alpha \in \Delta_k\}$.



Konvexe Hülle und ganzz. Optimierung

Satz

Die konvexe Hülle endlich vieler Punkte ist ein (beschränktes) Polyeder.

Konvexe Hülle und ganzz. Optimierung

Satz

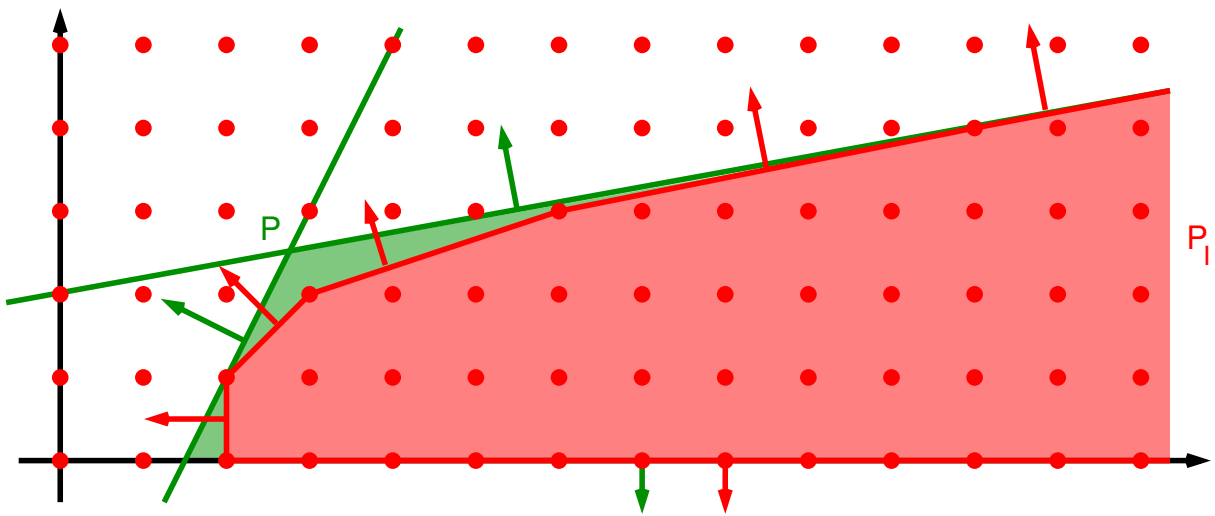
Die konvexe Hülle endlich vieler Punkte ist ein (beschränktes) Polyeder.

Die ganzzahlige Hülle eines Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ist die konvexe Hülle der in P enthaltenen ganzz. Punkte, $P_I := \text{conv}(P \cap \mathbb{Z}^n)$.

Satz

Ist $A \in \mathbb{Q}^{m \times n}$ und $b \in \mathbb{Q}^m$, dann ist die ganzz. Hülle P_I des Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ selbst ein Polyeder.

Problem: Beschreibung von P_I meist unbekannt oder extrem groß!



[Ausnahme z.B. für A tot. unimod., $b \in \mathbb{Z}^n$ ist $P = P_I$]

Konvexe Hülle und ganzz. Optimierung

Satz

Die konvexe Hülle endlich vieler Punkte ist ein (beschränktes) Polyeder.

Die ganzzahlige Hülle eines Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ ist die konvexe Hülle der in P enthaltenen ganzz. Punkte, $P_I := \text{conv}(P \cap \mathbb{Z}^n)$.

Satz

Ist $A \in \mathbb{Q}^{m \times n}$ und $b \in \mathbb{Q}^m$, dann ist die ganzz. Hülle P_I des Polyeders $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ selbst ein Polyeder.

Problem: Beschreibung von P_I meist unbekannt oder extrem groß!

Falls die ganzzahlige Hülle gut linear beschreibbar ist, kann man das ganzzahlige Optimierungsproblem mit Simplex lösen:

Satz

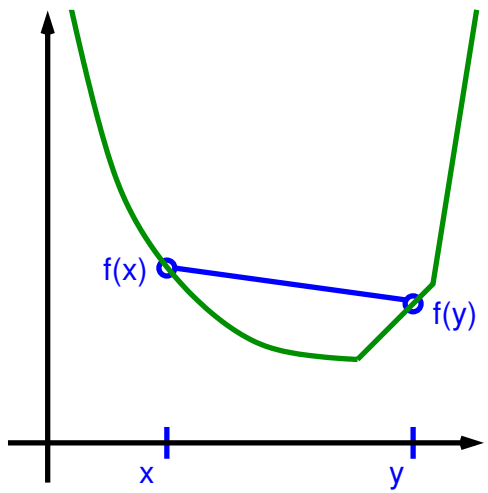
Ist für $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ die ganzzahlige Hülle durch $P_I = \{x \in \mathbb{R}^n : A_I x \leq b_I\}$ gegeben, so gilt:

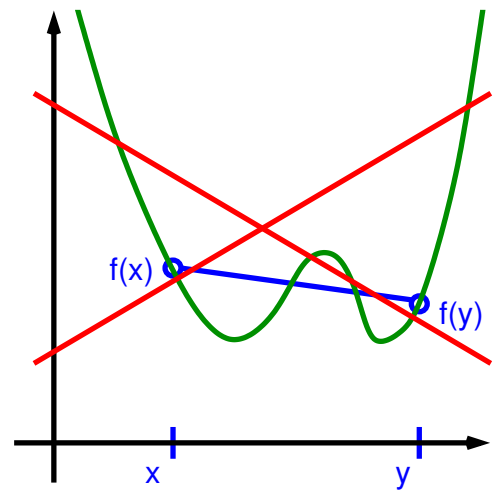
$$\begin{aligned} \sup\{c^T x : A_I x \leq b_I, x \in \mathbb{R}^n\} &= \sup\{c^T x : Ax \leq b, x \in \mathbb{Z}^n\}, \\ \text{Argmin}\{c^T x : A_I x \leq b_I, x \in \mathbb{R}^n\} &= \text{conv Argmin}\{c^T x : Ax \leq b, x \in \mathbb{Z}^n\}. \end{aligned}$$

Konvexe Funktionen

Eine Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ heißt konvex, wenn $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$ und $\alpha \in [0,1]$.

Sie heißt streng konvex, wenn $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$, $x \neq y$ und $\alpha \in (0,1)$.





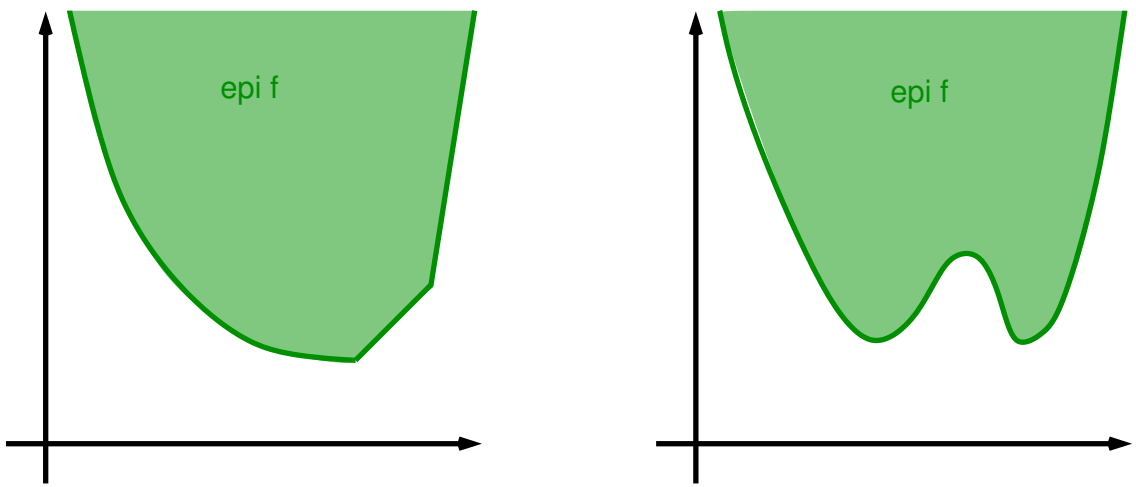
Konvexe Funktionen

Eine Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ heißt konvex, wenn $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$ und $\alpha \in [0,1]$.

Sie heißt streng konvex, wenn $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$, $x \neq y$ und $\alpha \in (0,1)$.

Der Epigraph einer Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ ist die Menge

$$\operatorname{epi} f := \left\{ \begin{pmatrix} x \\ r \end{pmatrix} : x \in \mathbb{R}^n, r \geq f(x) \right\}$$ [die Punkte „oberhalb“von $f(x)$]



Konvexe Funktionen

Eine Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ heißt konvex, wenn $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$ und $\alpha \in [0,1]$.

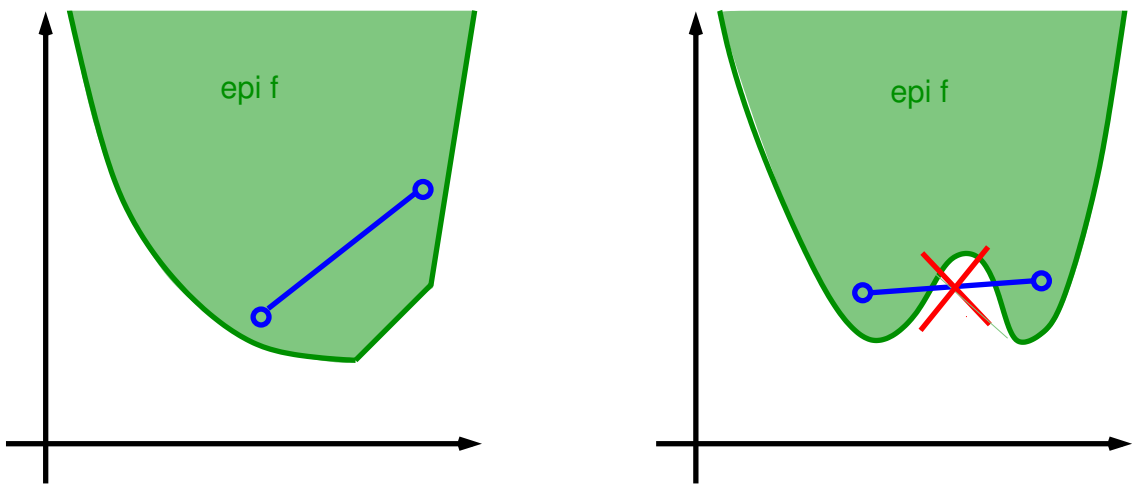
Sie heißt streng konvex, wenn $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$, $x \neq y$ und $\alpha \in (0,1)$.

Der Epigraph einer Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ ist die Menge

$$\operatorname{epi} f := \left\{ \begin{pmatrix} x \\ r \end{pmatrix} : x \in \mathbb{R}^n, r \geq f(x) \right\}$$ [die Punkte „oberhalb“von $f(x)$]

Satz

Eine Funktion ist genau dann konvex, wenn ihr Epigraph konvex ist.



Konvexe Funktionen

Eine Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ heißt konvex, wenn $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$ und $\alpha \in [0,1]$.

Sie heißt streng konvex, wenn $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$, $x \neq y$ und $\alpha \in (0,1)$.

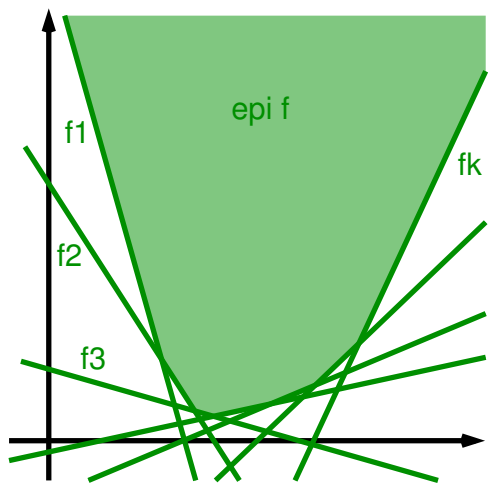
Der Epigraph einer Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ ist die Menge

$$\operatorname{epi} f := \left\{ \begin{pmatrix} x \\ r \end{pmatrix} : x \in \mathbb{R}^n, r \geq f(x) \right\}$$ [die Punkte „oberhalb“von $f(x)$]

Satz

Eine Funktion ist genau dann konvex, wenn ihr Epigraph konvex ist.

Bsp: Sind $f_i : \mathbb{R}^n \to \bar{\mathbb{R}}$ konvex, so auch f mit $f(x) := \sup_i f_i(x)$, $x \in \mathbb{R}^n$.



Konvexe Funktionen

Eine Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ heißt konvex, wenn $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$ und $\alpha \in [0,1]$.

Sie heißt streng konvex, wenn $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$, $x \neq y$ und $\alpha \in (0,1)$.

Der Epigraph einer Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ ist die Menge

$$\operatorname{epi} f := \left\{ \begin{pmatrix} x \\ r \end{pmatrix} : x \in \mathbb{R}^n, r \geq f(x) \right\}$$ [die Punkte „oberhalb“von $f(x)$]

Satz

Eine Funktion ist genau dann konvex, wenn ihr Epigraph konvex ist.

Bsp: Sind $f_i : \mathbb{R}^n \to \bar{\mathbb{R}}$ konvex, so auch f mit $f(x) := \sup_i f_i(x)$, $x \in \mathbb{R}^n$.

Satz

Jedes lokale Minimum einer konvexen Funktion ist auch globales Minimum, und für streng konvexe Funktionen ist es das einzige.

Für konvexe Funktionen gibt es gute Optimierungsverfahren.

Konvexe Funktionen

Eine Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$ heißt konvex, wenn $f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$ und $\alpha \in [0,1]$.

Sie heißt streng konvex, wenn $f(\alpha x + (1-\alpha)y) < \alpha f(x) + (1-\alpha)f(y)$ für $x, y \in \mathbb{R}^n$, $x \neq y$ und $\alpha \in (0,1)$.

Der Epigraph einer Funktion $f : \mathbb{R}^n \to \bar{\mathbb{R}}$ ist die Menge

$$\operatorname{epi} f := \left\{ \begin{pmatrix} x \\ r \end{pmatrix} : x \in \mathbb{R}^n, r \geq f(x) \right\}$$ [die Punkte „oberhalb“von $f(x)$]

Satz

Eine Funktion ist genau dann konvex, wenn ihr Epigraph konvex ist.

Bsp: Sind $f_i : \mathbb{R}^n \to \bar{\mathbb{R}}$ konvex, so auch f mit $f(x) := \sup_i f_i(x)$, $x \in \mathbb{R}^n$.

Satz

Jedes lokale Minimum einer konvexen Funktion ist auch globales Minimum, und für streng konvexe Funktionen ist es das einzige.

Für konvexe Funktionen gibt es gute Optimierungsverfahren.

Eine Funktion f heißt konkav, wenn $-f$ konvex ist.
(Jedes lokale Maximum einer konkaven Funktion ist auch globales.)

Inhaltsübersicht

3.8 Relaxation

Konzept auf beliebige Optimierungsprobleme anwendbar (hier Maximieren):

Definition

Gegeben zwei Optimierungsprobleme mit $\mathcal{X}, \mathcal{W} \subseteq \mathbb{R}^n$ *und* $f, f' : \mathbb{R}^n \to \mathbb{R}$

$$(OP) \quad \max \ f(x) \text{ s.t. } x \in \mathcal{X} \qquad \textit{bzw.} \qquad (RP) \quad \max \ f'(x) \text{ s.t. } x \in \mathcal{W},$$

(RP) *heißt Relaxation von* (OP)*, falls*

1. $\mathcal{X} \subseteq \mathcal{W}$,
2. $f(x) \leq f'(x)$ *für alle* $x \in \mathcal{X}$.

3.8 Relaxation

Konzept auf beliebige Optimierungsprobleme anwendbar (hier Maximieren):

Definition

Gegeben zwei Optimierungsprobleme mit $\mathcal{X}, \mathcal{W} \subseteq \mathbb{R}^n$ *und* $f, f' : \mathbb{R}^n \to \mathbb{R}$

$$(OP) \quad \max \ f(x) \text{ s.t. } x \in \mathcal{X} \qquad \textit{bzw.} \qquad (RP) \quad \max \ f'(x) \text{ s.t. } x \in \mathcal{W},$$

(RP) *heißt Relaxation von* (OP), *falls*

1. $\mathcal{X} \subseteq \mathcal{W}$,
2. $f(x) \leq f'(x)$ *für alle* $x \in \mathcal{X}$.

Sei (RP) eine Relaxation von (OP).

Beobachtung

1. $v(RP) \geq v(OP)$. *[(RP) liefert obere Schranke]*
2. *Ist* (RP) *unzulässig, so auch* (OP),
3. *Ist* x^* *OL von (RP) und gilt* $x^* \in \mathcal{X}$ *sowie* $f'(x^*) = f(x^*)$, *dann ist* x^* *OL von (OP).*

3.8 Relaxation

Konzept auf beliebige Optimierungsprobleme anwendbar (hier Maximieren):

Definition

Gegeben zwei Optimierungsprobleme mit $\mathcal{X}, \mathcal{W} \subseteq \mathbb{R}^n$ *und* $f, f' : \mathbb{R}^n \to \mathbb{R}$

$$(OP) \quad \max \ f(x) \text{ s.t. } x \in \mathcal{X} \qquad \textit{bzw.} \qquad (RP) \quad \max \ f'(x) \text{ s.t. } x \in \mathcal{W},$$

(RP) *heißt Relaxation von* (OP)*, falls*

1. $\mathcal{X} \subseteq \mathcal{W}$,
2. $f(x) \leq f'(x)$ *für alle* $x \in \mathcal{X}$.

Sei (RP) eine Relaxation von (OP).

Beobachtung

1. $v(RP) \geq v(OP)$. *[(RP) liefert obere Schranke]*
2. *Ist* (RP) *unzulässig, so auch* (OP),
3. *Ist* x^* *OL von (RP) und gilt* $x^* \in \mathcal{X}$ *sowie* $f'(x^*) = f(x^*)$, *dann ist* x^* *OL von (OP).*

Man sucht nun möglichst „kleines“ $\mathcal{W} \supseteq \mathcal{X}$ und $f' \geq f$ so, dass (RP) noch gut lösbar ist.

Eine Relaxation (RP) von (OP) heißt exakt, falls $v(OP) = v(RP)$ gilt.

Allgemein: Konvexe Relaxation

Wird Konvexität für $\mathcal{W}$ und (bei max) Konkavität für f' gefordert, spricht man von konvexer Relaxation. Meist (aber nicht immer!) dient die Konvexität als Garant für vernünftige Lösbarkeit der Relaxation.

Allgemein: Konvexe Relaxation

Wird Konvexität für $\mathcal{W}$ und (bei max) Konkavität für f' gefordert, spricht man von konvexer Relaxation. Meist (aber nicht immer!) dient die Konvexität als Garant für vernünftige Lösbarkeit der Relaxation.

Bsp: Für ein kombinatorisches Max.-Problem mit endl. Grundmenge Ω, zulässigen Lösungen $\mathcal{F} \subseteq 2^\Omega$ und linearer Zielfunktion $c \in \mathbb{R}^\Omega$ ist

$$\max c^T x \text{ s.t. } x \in \operatorname{conv}\{\chi_\Omega(F) : F \in \mathcal{F}\}$$

eine exakte konvexe (sogar lineare) Relaxation, aber nur dann nützlich, wenn das Polyeder $\operatorname{conv}\{\chi(F) : F \in \mathcal{F}\}$ gut durch ein nicht zu großes Ungleichungssystem $Ax \leq b$ darstellbar ist.

Allgemein: Konvexe Relaxation

Wird Konvexität für $\mathcal{W}$ und (bei max) Konkavität für f' gefordert, spricht man von konvexer Relaxation. Meist (aber nicht immer!) dient die Konvexität als Garant für vernünftige Lösbarkeit der Relaxation.

Bsp: Für ein kombinatorisches Max.-Problem mit endl. Grundmenge Ω, zulässigen Lösungen $\mathcal{F} \subseteq 2^\Omega$ und linearer Zielfunktion $c \in \mathbb{R}^\Omega$ ist

$$\max c^T x \text{ s.t. } x \in \operatorname{conv}\{\chi_\Omega(F) : F \in \mathcal{F}\}$$

eine exakte konvexe (sogar lineare) Relaxation, aber nur dann nützlich, wenn das Polyeder $\operatorname{conv}\{\chi(F) : F \in \mathcal{F}\}$ gut durch ein nicht zu großes Ungleichungssystem $Ax \leq b$ darstellbar ist.

In der globalen Optimierung werden auf Teilintervallen nichtlineare Funktionen nach unten durch konvexe Funktionen abgeschätzt.

Bsp: Betrachte (OP) $\min f(x) := \frac{1}{2}x^T Qx + q^T x$ s.t. $x \in [0,1]^n$
mit f nicht konvex, d.h., $\lambda_{\min}(Q) < 0$. [$\lambda_{\min}$ . . . minimaler Eigenwert]
Es ist $Q - \lambda_{\min}(Q)I$ positiv semidefinit und wegen $x_i^2 \leq x_i$ auf $[0,1]^n$ gilt
$f'(x) := \frac{1}{2}x^T(Q - \lambda_{\min}(Q)I)x + (q + \lambda_{\min}(Q)\mathbf{1})^T x \leq f(x) \quad \forall x \in [0,1]^n$.
Damit ist $(RP)\min f'(x)$ s.t. $x \in [0,1]^n$ eine konvexe Relaxation von (OP).

Die LP-Relaxation für Ganzzahlige Programme

Für ein ganzzahliges Programm $\max\ c^T x$ s.t. $Ax \leq b,\ x \in \mathbb{Z}^n$
entsteht die LP-Relaxation durch Weglassen der Ganzz.-Bedingung,

$$\max\ c^T x \text{ s.t. } Ax \leq b,\ x \in \mathbb{R}^n.$$

Ist Relaxation, denn

$$\mathcal{X} := \{x \in \mathbb{Z}^n : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n : Ax \leq b\} =: \mathcal{W}.$$

[wird von allen Standardlösern für gemischt-ganzz. Programme verwendet]

Die LP-Relaxation für Ganzzahlige Programme

Für ein ganzzahliges Programm $\max\ c^T x$ s.t. $Ax \leq b,\ x \in \mathbb{Z}^n$
entsteht die LP-Relaxation durch Weglassen der Ganzz.-Bedingung,

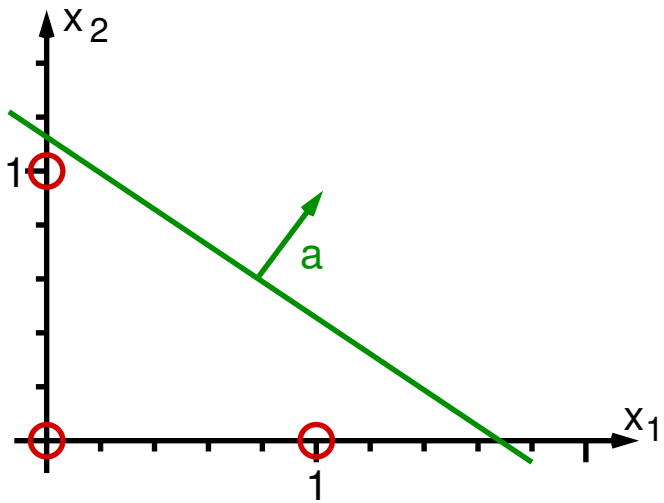
$$\max\ c^T x \text{ s.t. } Ax \leq b,\ x \in \mathbb{R}^n.$$

Ist Relaxation, denn

$$\mathcal{X} := \{x \in \mathbb{Z}^n : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n : Ax \leq b\} =: \mathcal{W}.$$

[wird von allen Standardlösern für gemischt-ganzz. Programme verwendet]

Bsp: Rucksackproblem: $n = 2$, Gewichte $a = (6, 8)^T$, Kapazität $b = 10$,
$\max\ c^T x$ s.t. $a^T x \leq b,\ x \in \mathbb{Z}^n_+ \quad \rightarrow \quad \max\ c^T x$ s.t. $a^T x \leq b,\ x \geq 0,$



zulässige ganzzahlige Punkte: ◯

Die LP-Relaxation für Ganzzahlige Programme

Für ein ganzzahliges Programm $\max\ c^T x$ s.t. $Ax \leq b,\ x \in \mathbb{Z}^n$
entsteht die LP-Relaxation durch Weglassen der Ganzz.-Bedingung,

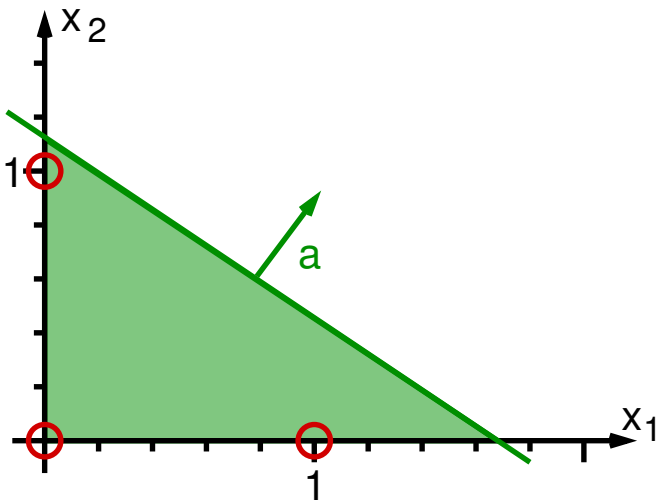
$$\max\ c^T x \text{ s.t. } Ax \leq b,\ x \in \mathbb{R}^n.$$

Ist Relaxation, denn

$$\mathcal{X} := \{x \in \mathbb{Z}^n : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n : Ax \leq b\} =: \mathcal{W}.$$

[wird von allen Standardlösern für gemischt-ganzz. Programme verwendet]

Bsp: Rucksackproblem: $n = 2$, Gewichte $a = (6, 8)^T$, Kapazität $b = 10$,
$\max\ c^T x$ s.t. $a^T x \leq b,\ x \in \mathbb{Z}^n_+ \quad \rightarrow \quad \max\ c^T x$ s.t. $a^T x \leq b,\ x \geq 0,$



zulässige ganzzahlige Punkte: ◯

LP-Relaxation: grün

Die LP-Relaxation für Ganzzahlige Programme

Für ein ganzzahliges Programm $\max\ c^T x$ s.t. $Ax \leq b,\ x \in \mathbb{Z}^n$
entsteht die LP-Relaxation durch Weglassen der Ganzz.-Bedingung,

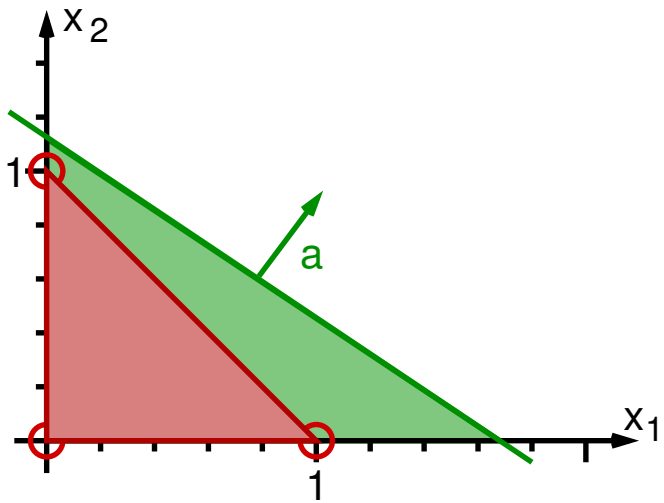
$$\max\ c^T x \text{ s.t. } Ax \leq b,\ x \in \mathbb{R}^n.$$

Ist Relaxation, denn

$$\mathcal{X} := \{x \in \mathbb{Z}^n : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n : Ax \leq b\} =: \mathcal{W}.$$

[wird von allen Standardlösern für gemischt-ganzz. Programme verwendet]

Bsp: Rucksackproblem: $n = 2$, Gewichte $a = (6, 8)^T$, Kapazität $b = 10$,
$\max\ c^T x$ s.t. $a^T x \leq b,\ x \in \mathbb{Z}^n_+ \quad \rightarrow \quad \max\ c^T x$ s.t. $a^T x \leq b,\ x \geq 0,$



zulässige ganzzahlige Punkte: ◯

LP-Relaxation: grün

beste Relaxation: die konvexe Hülle

Die LP-Relaxation für Ganzzahlige Programme

Für ein ganzzahliges Programm $\max\ c^T x$ s.t. $Ax \leq b,\ x \in \mathbb{Z}^n$
entsteht die LP-Relaxation durch Weglassen der Ganzz.-Bedingung,

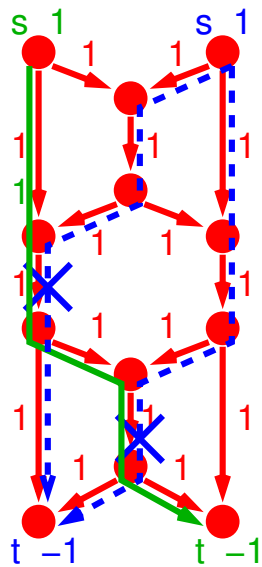
$$\max\ c^T x \text{ s.t. } Ax \leq b,\ x \in \mathbb{R}^n.$$

Ist Relaxation, denn

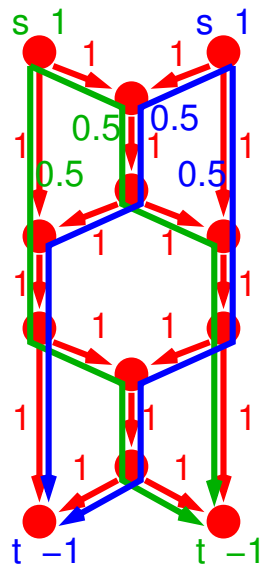
$$\mathcal{X} := \{x \in \mathbb{Z}^n : Ax \leq b\} \subseteq \{x \in \mathbb{R}^n : Ax \leq b\} =: \mathcal{W}.$$

[wird von allen Standardlösern für gemischt-ganzz. Programme verwendet]

Bsp: ganzzahliger Mehrgüterfluss → gebrochener Mehrgüterfluss



ganzz. geht nicht, $\mathcal{X} = \emptyset$



gebrochen geht, $\mathcal{W} \neq \emptyset$

auch zu groß, bräuchten die konvexe Hülle

Lagrange-Relaxation

[Allg. für restr. Opt. verwendbar, hier vorerst nur Ungls.-Nebenbed.]

Unbequeme Nebenbedingungen werden mit einem Lagrangemultiplikator, der die Verletzung bestraft, in die Zielfunktion gehoben ($g : \mathbb{R}^n \to \mathbb{R}^k$):

$$(OP) \quad \begin{array}{ll} \max & f(x) \\ \text{s.t.} & g(x) \leq 0 \\ & x \in \Omega \end{array} \quad \left| \cdot \lambda \geq 0 \right. \quad \to \quad (RP_\lambda) \quad \begin{array}{ll} \max & c^T x - \lambda^T g(x) \\ \text{s.t.} & x \in \Omega \end{array}$$

Ist Relaxation: $\mathcal{X} := \{x \in \Omega : g(x) \leq 0\} \subseteq \{x \in \Omega\} =: \mathcal{W}$ und für $x \in \mathcal{X}$, $\lambda \geq 0$ gilt $f(x) \leq f(x) - \lambda^T g(x) =: f'(x)$ wegen $g(x) \leq 0$.

Lagrange-Relaxation

[Allg. für restr. Opt. verwendbar, hier vorerst nur Unglgs.-Nebenbed.]

Unbequeme Nebenbedingungen werden mit einem Lagrangemultiplikator, der die Verletzung bestraft, in die Zielfunktion gehoben ($g : \mathbb{R}^n \to \mathbb{R}^k$):

$$(OP) \quad \begin{array}{ll} \max & f(x) \\ \text{s.t.} & g(x) \leq 0 \\ & x \in \Omega \end{array} \quad \left| \cdot \lambda \geq 0 \right. \quad \to \quad (RP_\lambda) \quad \begin{array}{ll} \max & c^T x - \lambda^T g(x) \\ \text{s.t.} & x \in \Omega \end{array}$$

Ist Relaxation: $\mathcal{X} := \{x \in \Omega : g(x) \leq 0\} \subseteq \{x \in \Omega\} =: \mathcal{W}$ und für $x \in \mathcal{X}$, $\lambda \geq 0$ gilt $f(x) \leq f(x) - \lambda^T g(x) =: f'(x)$ wegen $g(x) \leq 0$.

Definiere die duale Funktion $\varphi(\lambda) := \sup_{x \in \Omega} \left[f(x) - g(x)^T \lambda \right] = v(RP_\lambda)$
[für jedes feste x linear in λ]

- Für jedes $\lambda \geq 0$ gilt $\varphi(\lambda) \geq v(OP)$ [obere Schranke]
- $\varphi(\lambda)$ gut berechenbar, falls (RP_λ) „leicht“ lösbar
- φ ist als sup von in λ linearen Funktionen konvex
- Beste Schranke ist $\inf\{\varphi(\lambda) : \lambda \geq 0\}$ [konvexes Problem!]

gut mit Verfahren der konvexen Optimierung bestimmbar!

Beispiel: Ganzzahliger Mehrgüterfluss

A sei die Knoten-Kanten-Inz.-Matrix zu $D = (V, E)$, 2 Güter,
Lagrange-Relaxation der koppelnden Kapazitätsbedingungen mit $\lambda \geq 0$:

$$\begin{array}{lllll} \min & {c^{(1)}}^T x^{(1)} & + & {c^{(2)}}^T x^{(2)} & \\ \text{s.t.} & Ax^{(1)} & & = & b^{(1)} \\ & & Ax^{(2)} & = & b^{(2)} \\ \lambda \cdot | & x^{(1)} & + & x^{(2)} & \leq w \\ & x^{(1)} \leq w, & & x^{(2)} \leq w & \\ & x^{(1)} \in \mathbb{Z}_+^E, & & x^{(2)} \in \mathbb{Z}_+^E. & \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & (c^{(1)}+\lambda)^T x^{(1)} + (c^{(2)}+\lambda)^T x^{(2)} - \lambda^T w \\ \text{s.t.} & Ax^{(1)} & = b^{(1)} \\ & & Ax^{(2)} = b^{(2)} \\ & x^{(1)} \leq w, & x^{(2)} \leq w, \\ & x^{(1)} \in \mathbb{Z}_+^E, & x^{(2)} \in \mathbb{Z}_+^E. \end{array}$$

Relaxation zerfällt in zwei unabhängige Minimale-Kosten-Fluss-Probleme

$$(RP_\lambda^{(i)}) \quad \min (c^{(i)}+\lambda)^T x^{(i)} \quad \text{s.t.} \quad Ax^{(i)} = b^{(i)},\ w \geq x^{(i)} \in \mathbb{Z}_+^E \quad i \in \{1, 2\}$$

Diese sind effizient ganzzahlig lösbar!

Beispiel: Ganzzahliger Mehrgüterfluss

A sei die Knoten-Kanten-Inz.-Matrix zu $D = (V, E)$, 2 Güter,
Lagrange-Relaxation der koppelnden Kapazitätsbedingungen mit $\lambda \geq 0$:

$$\begin{array}{lll} \min & {c^{(1)}}^T x^{(1)} & + \; {c^{(2)}}^T x^{(2)} \\ \text{s.t.} & Ax^{(1)} & = b^{(1)} \\ & Ax^{(2)} & = b^{(2)} \\ \lambda \cdot | & x^{(1)} + x^{(2)} & \leq w \\ & x^{(1)} \leq w, & x^{(2)} \leq w \\ & x^{(1)} \in \mathbb{Z}^E_+, & x^{(2)} \in \mathbb{Z}^E_+. \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & (c^{(1)}+\lambda)^T x^{(1)} + (c^{(2)}+\lambda)^T x^{(2)} - \lambda^T w & \\ \text{s.t.} & Ax^{(1)} & = b^{(1)} \\ & Ax^{(2)} & = b^{(2)} \\ & x^{(1)} \leq w, & x^{(2)} \leq w, \\ & x^{(1)} \in \mathbb{Z}^E_+, & x^{(2)} \in \mathbb{Z}^E_+. \end{array}$$

Relaxation zerfällt in zwei unabhängige Minimale-Kosten-Fluss-Probleme

$$(RP^{(i)}_\lambda) \quad \min (c^{(i)}+\lambda)^T x^{(i)} \quad \text{s.t.} \quad Ax^{(i)} = b^{(i)},\ w \geq x^{(i)} \in \mathbb{Z}^E_+ \quad i \in \{1, 2\}$$

Diese sind effizient ganzzahlig lösbar!

Zerfällt das Problem bei Lagrange-Relaxation in unabhängige Teilprobleme, nennt man das Lagrange-Dekomposition. Damit können oft deutlich größere Probleme sehr effizient gelöst werden.

Bekommt man dadurch auch eine bessere Schranke?

Vergleich Lagrange- und LP-Relaxation

Sei $\Omega \subset \mathbb{Z}^n$ endlich und $D \in \mathbb{Q}^{k\times n}$, $d \in \mathbb{Q}^k$,

$$(OP)\quad \begin{array}{ll} \max & c^T x \\ \text{s.t.} & Dx \le d \quad | \cdot \lambda \ge 0 \\ & x \in \Omega \end{array} \quad \rightarrow \quad (RP_\lambda)\quad \begin{array}{ll} \max & c^T x + \lambda^T (d - Dx) \\ \text{s.t.} & x \in \Omega \end{array}$$

Im Bsp: $\Omega = \Omega^{(1)} \times \Omega^{(2)}$ mit $\Omega^{(i)} = \{x \in \mathbb{Z}_+^E : Ax = b^{(i)}, x \le w\}$, $i \in \{1, 2\}$.

Satz

$$\inf_{\lambda \ge 0} v(RP_\lambda) = \sup\{c^T x : Dx \le d, x \in \operatorname{conv} \Omega\}.$$

Ist $\operatorname{conv} \Omega$ gleich der zulässigen Menge der LP-Relaxation von Ω, sind die Werte der besten Lagrange- und der LP-Relaxation gleich!

Vergleich Lagrange- und LP-Relaxation

Sei $\Omega \subset \mathbb{Z}^n$ endlich und $D \in \mathbb{Q}^{k\times n}$, $d \in \mathbb{Q}^k$,

$$(OP)\quad \begin{array}{ll} \max & c^T x \\ \text{s.t.} & Dx \le d \quad | \cdot \lambda \ge 0 \\ & x \in \Omega \end{array} \quad \rightarrow \quad (RP_\lambda)\quad \begin{array}{ll} \max & c^T x + \lambda^T (d - Dx) \\ \text{s.t.} & x \in \Omega \end{array}$$

Im Bsp: $\Omega = \Omega^{(1)} \times \Omega^{(2)}$ mit $\Omega^{(i)} = \{x \in \mathbb{Z}_+^E : Ax = b^{(i)}, x \le w\}$, $i \in \{1,2\}$.

Satz

$$\inf_{\lambda \ge 0} v(RP_\lambda) = \sup\{c^T x : Dx \le d, x \in \operatorname{conv}\Omega\}.$$

Ist $\operatorname{conv}\Omega$ gleich der zulässigen Menge der LP-Relaxation von Ω, sind die Werte der besten Lagrange- und der LP-Relaxation gleich!

Im Bsp ist A total unimodular, daher gilt für $i \in \{1,2\}$ und $w \in \mathbb{Z}^E$

$$\operatorname{conv}\{x \in \mathbb{Z}_+^E : Ax = b^{(i)}, x \le w\} = \{x \ge 0 : Ax = b^{(i)}, x \le w\}.$$

Das beste λ ergibt den Wert des gebrochenen Mehrgüterflussproblems!

Vergleich Lagrange- und LP-Relaxation

Sei $\Omega \subset \mathbb{Z}^n$ endlich und $D \in \mathbb{Q}^{k\times n}$, $d \in \mathbb{Q}^k$,

$$(OP)\quad \begin{array}{ll}\max & c^T x \\ \text{s.t.} & Dx \le d \\ & x \in \Omega\end{array} \quad \left| \cdot \lambda \ge 0 \right. \quad \rightarrow \quad (RP_\lambda)\quad \begin{array}{ll}\max & c^T x + \lambda^T(d - Dx) \\ \text{s.t.} & x \in \Omega\end{array}$$

Im Bsp: $\Omega = \Omega^{(1)} \times \Omega^{(2)}$ mit $\Omega^{(i)} = \{x \in \mathbb{Z}_+^E : Ax = b^{(i)}, x \le w\}$, $i \in \{1,2\}$.

Satz

$$\inf_{\lambda \ge 0} v(RP_\lambda) = \sup\{c^T x : Dx \le d, x \in \operatorname{conv}\Omega\}.$$

Ist $\operatorname{conv}\Omega$ gleich der zulässigen Menge der LP-Relaxation von Ω, sind die Werte der besten Lagrange- und der LP-Relaxation gleich!

Im Bsp ist A total unimodular, daher gilt für $i \in \{1,2\}$ und $w \in \mathbb{Z}^E$

$$\operatorname{conv}\{x \in \mathbb{Z}_+^E : Ax = b^{(i)}, x \le w\} = \{x \ge 0 : Ax = b^{(i)}, x \le w\}.$$

Das beste λ ergibt den Wert des gebrochenen Mehrgüterflussproblems!

Allgemein: Sei $\{x \in \mathbb{Z}^n : Ax \le b\} = \Omega$ eine Formulierung von Ω. Nur falls $\{x \in \mathbb{R}^n : Ax \le b\} \ne \operatorname{conv}\Omega$, kann die Lagrange-Relaxation einen besseren Wert liefern als die LP-Relaxation.

Inhaltsübersicht

3.9 Anwendung: Rundreiseprobleme (TSP)

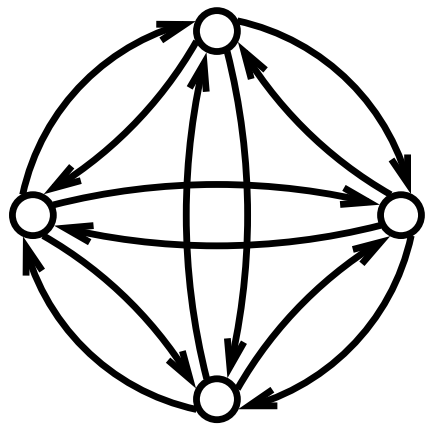
Problem des Handelsreisenden: Gegeben n Städte mit allen paarweisen Distanzen, finde eine kürzeste Rundreise, die jede genau einmal besucht.

3.9 Anwendung: Rundreiseprobleme (TSP)

Problem des Handelsreisenden: Gegeben n Städte mit allen paarweisen Distanzen, finde eine kürzeste Rundreise, die jede genau einmal besucht.

Komb. Opt.: $D = (V, E = \{(u, v) : u, v \in V, u \neq v\})$ vollst. Digraph, Kosten $c \in \mathbb{R}^E$, zul. Menge $\mathcal{F} = \{R \subset E : R \text{ (ger.) Kreis in } D, |R| = n\}$.
Finde $R \in \operatorname{Argmin}\{c(R) = \sum_{e \in R} c_e : R \in \mathcal{F}\}$

NP-vollständiges Problem
Anzahl Rundreisen?



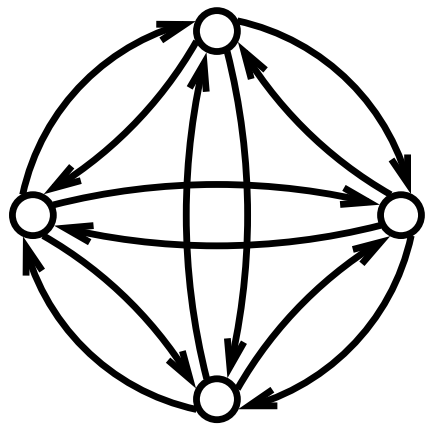
3.9 Anwendung: Rundreiseprobleme (TSP)

Problem des Handelsreisenden: Gegeben n Städte mit allen paarweisen Distanzen, finde eine kürzeste Rundreise, die jede genau einmal besucht.

Komb. Opt.: $D = (V, E = \{(u,v) : u, v \in V, u \neq v\})$ vollst. Digraph, Kosten $c \in \mathbb{R}^E$, zul. Menge $\mathcal{F} = \{R \subset E : R \text{ (ger.) Kreis in } D, |R| = n\}$.
Finde $R \in \operatorname{Argmin}\{c(R) = \sum_{e \in R} c_e : R \in \mathcal{F}\}$

NP-vollständiges Problem
Anzahl Rundreisen? $(n-1)!$

$\rightarrow$kombinatorische Explosion



$n = 3$:	2
$n = 4$:	$3 \cdot 2 = 6$
$n = 5$:	$4 \cdot 3 \cdot 2 = 24$
$n = 10$:	362880
$n = 11$:	3628800
$n = 12$:	39916800
$n = 13$:	479001600
$n = 14$:	6227020800
$n = 100$:	10^{158}

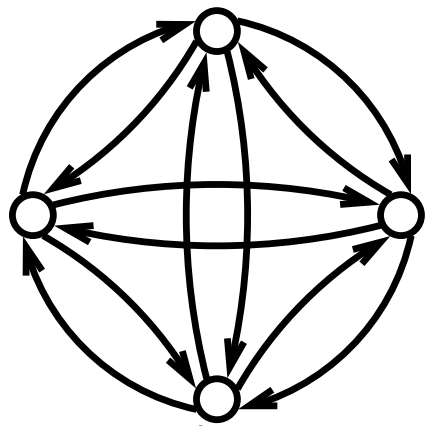
3.9 Anwendung: Rundreiseprobleme (TSP)

Problem des Handelsreisenden: Gegeben n Städte mit allen paarweisen Distanzen, finde eine kürzeste Rundreise, die jede genau einmal besucht.

Komb. Opt.: $D = (V, E = \{(u,v) : u,v \in V, u \neq v\})$ vollst. Digraph, Kosten $c \in \mathbb{R}^E$, zul. Menge $\mathcal{F} = \{R \subset E : R \text{ (ger.) Kreis in } D, |R| = n\}$.
Finde $R \in \operatorname{Argmin}\{c(R) = \sum_{e \in R} c_e : R \in \mathcal{F}\}$

NP-vollständiges Problem
Anzahl Rundreisen? $(n-1)!$

→kombinatorische Explosion



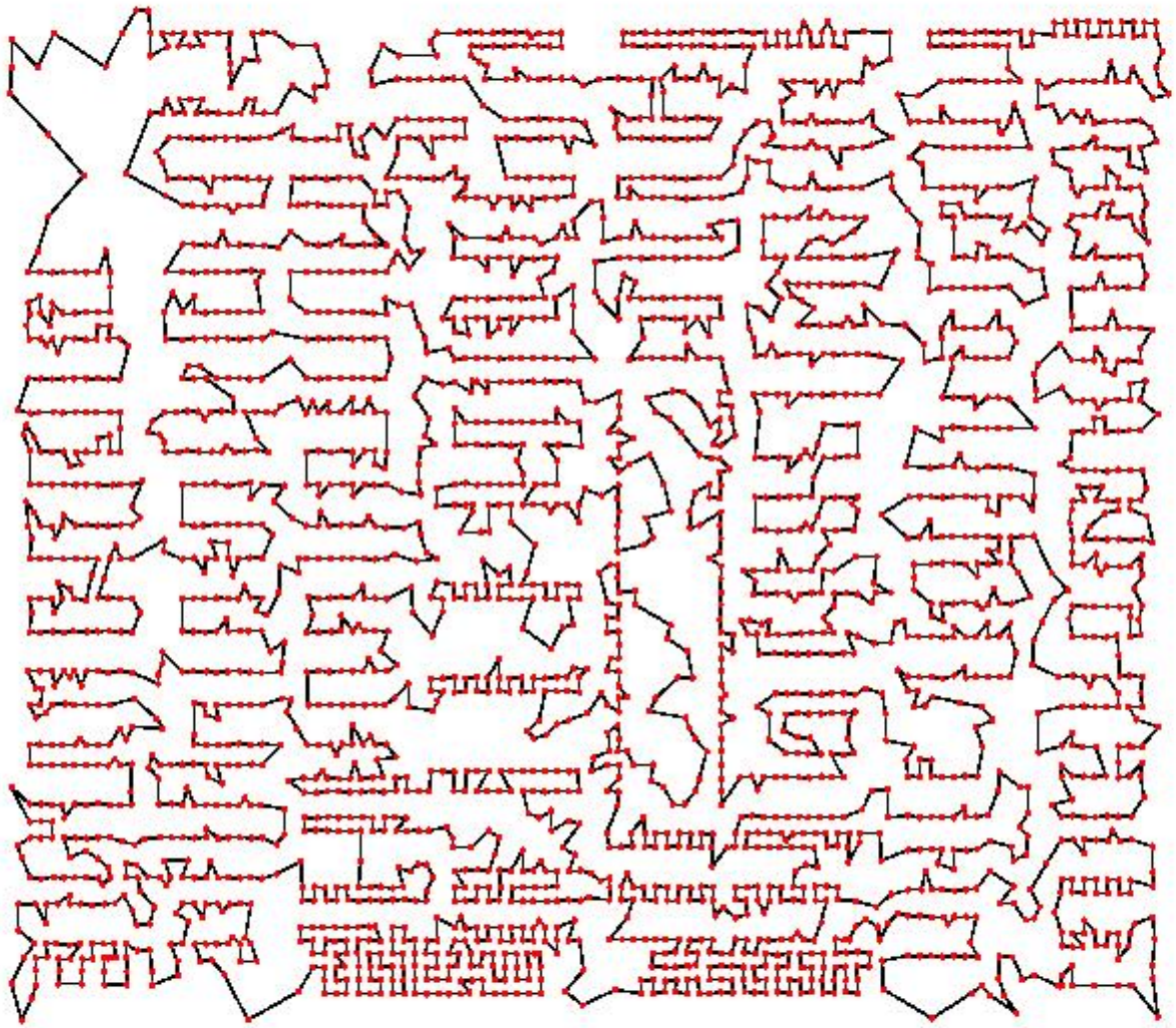
$n = 3$:	2
$n = 4$:	$3 \cdot 2 = 6$
$n = 5$:	$4 \cdot 3 \cdot 2 = 24$
$n = 10$:	362880
$n = 11$:	3628800
$n = 12$:	39916800
$n = 13$:	479001600
$n = 14$:	6227020800
$n = 100$:	10^{158}

Typisches Grundproblem in:

- Zustelldiensten (Paket-, Arznei-Transporte)
- Taxiservice, Havarie-Dienste
- Auftragsplanung mit Rüstkosten [z.B. Autos färben]
- Robotersteuerung (meist nichtlinear)

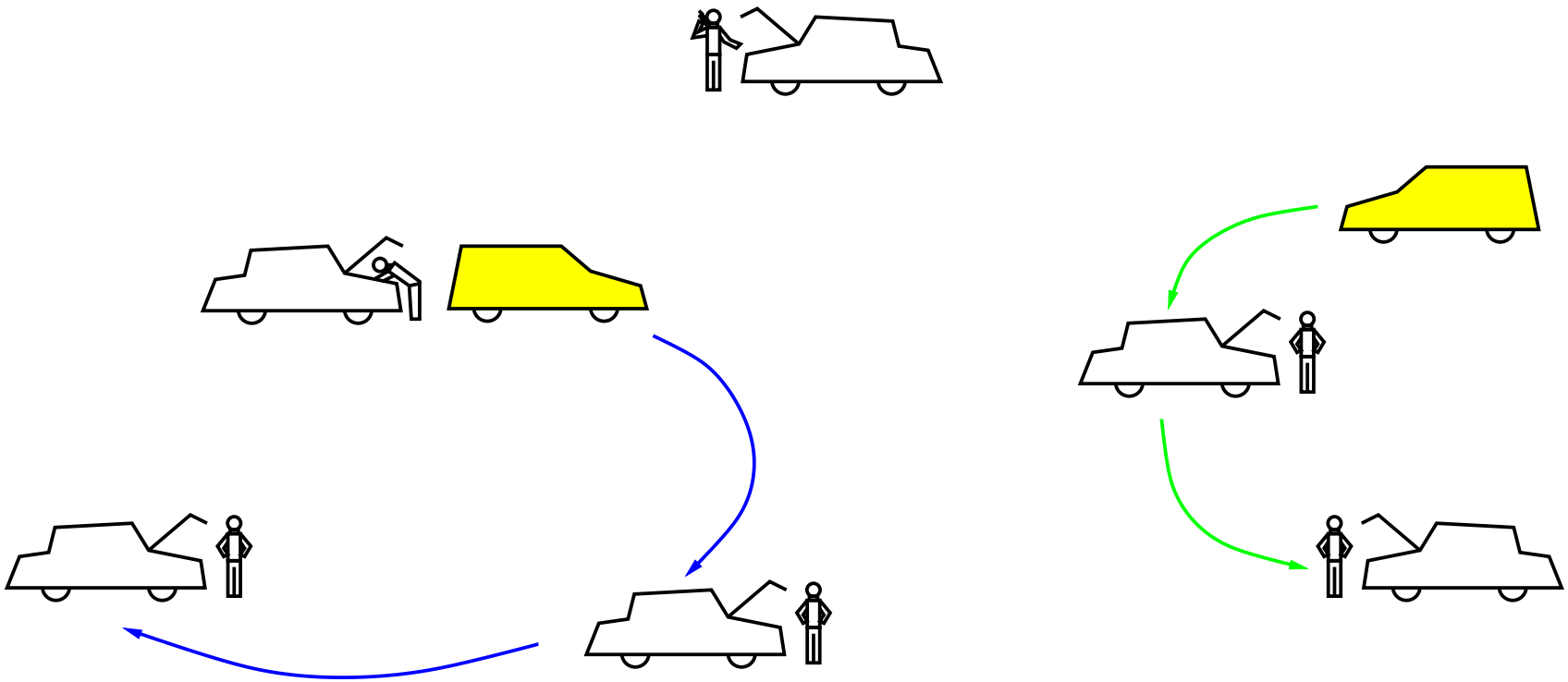
[oft mit mehreren „Fahrzeugen“ und Zeitfenstern]

Löcher in Platinen bohren



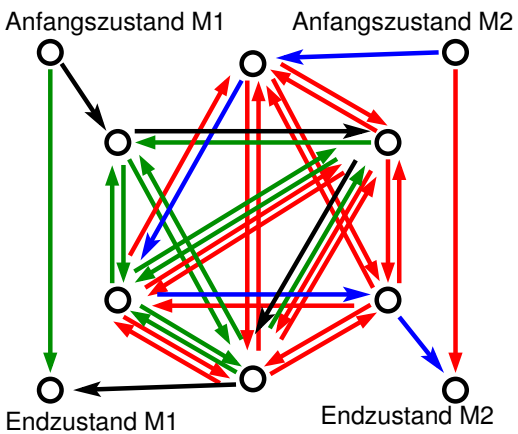
[http://www.math.princeton.edu/tsp]

Mit Online-Aspekten: Pannenhilfe

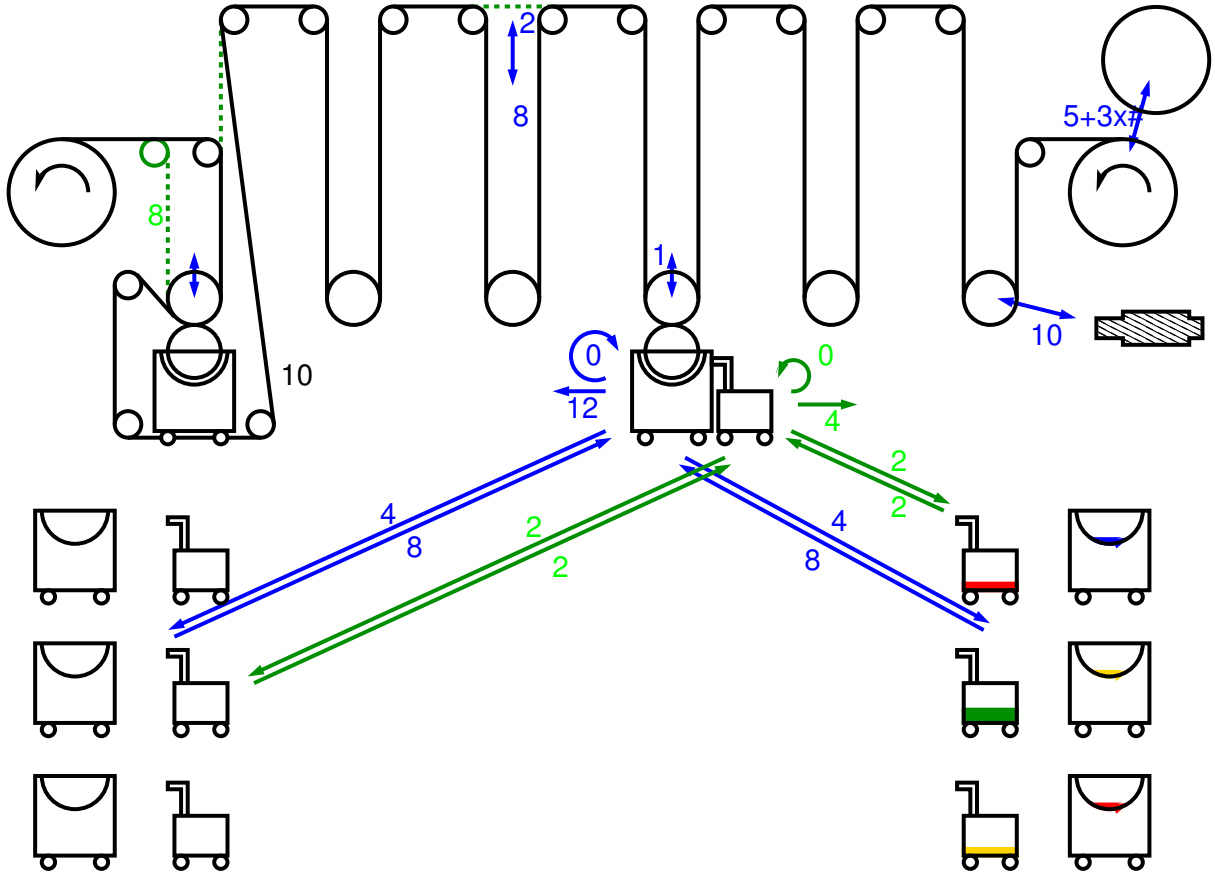


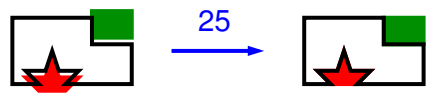
Bestimme Autozuweisung und Reihenfolge für jedes Pannenfahrzeug so, dass bereits versprochene Wartezeiten nicht überschritten werden.

Reihenfolgeoptimierung bei langen Rüstzeiten



Zwei Rotationstiefdruckmaschinen
für den Druck von Geschenkpapier



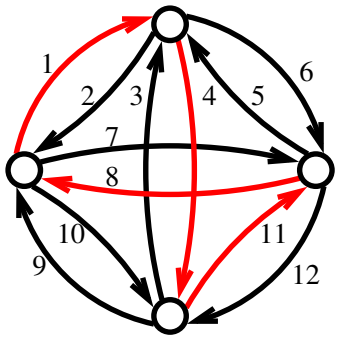


Passer

Farbanpassung neue Farben: 20

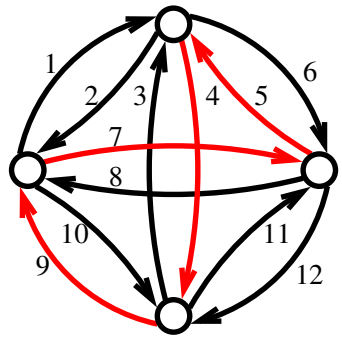
Modellierung als ganzzahliges Programm

Abstrakte Formulierung über konvexe Hülle der Inzidenzvektoren:



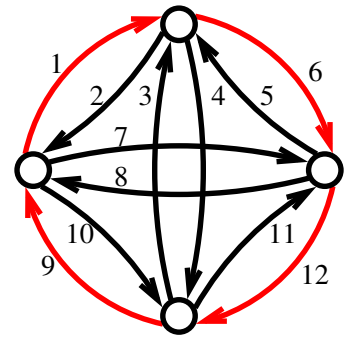
$R_1 = \{1, 4, 8, 11\}$

$$\chi(R_1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$



$R_2 = \{4, 5, 7, 9\}$

$$\chi(R_2) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$



$R_3 = \{1, 6, 9, 12\}$

$$\chi(R_3) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Für Kantenlängen $c \in \mathbb{R}^E$ ist die Länge von Rundreise R: $\sum_{e \in R} c_e = c^T \chi(R)$.

(TSP) min $c^T x$ s.t. $x \in \text{conv}\{\chi(R) : R \text{ Rundreise in } D = (V, E)\} =: P_{TSP}$

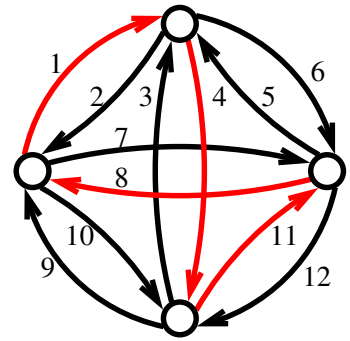
Wäre exakt, aber keine lineare Beschreibung $A_I x \leq b_I$ von P_{TSP} bekannt!

Ganzzahlige Formulierung für (TSP)

Ziel: P_{TSP} durch ein größeres Polytop $P = \{x \in \mathbb{R}^n : Ax \leq b\} \supseteq P_{TSP}$ möglichst eng einfassen, sodass wenigstens $P \cap \mathbb{Z}^E = \{\chi(R) : R \text{ Rundreise}\}$.

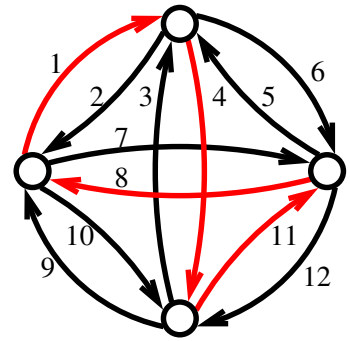
Eine Gleichung/Ungleichung heißt gültig für P_{TSP}, wenn sie für alle $x \in \{\chi(R) : R \text{ Rundreise}\}$ erfüllt ist.

Vorschläge?



Ganzzahlige Formulierung für (TSP)

Ziel: P_{TSP} durch ein größeres Polytop $P = \{x \in \mathbb{R}^n : Ax \leq b\} \supseteq P_{TSP}$ möglichst eng einfassen, sodass wenigstens $P \cap \mathbb{Z}^E = \{\chi(R) : R \text{ Rundreise}\}$.



Eine Gleichung/Ungleichung heißt gültig für P_{TSP}, wenn sie für alle $x \in \{\chi(R) : R \text{ Rundreise}\}$ erfüllt ist.

0-1 Würfel: $0 \leq x \leq 1$ ist gültig

Gradgleichungen:
aus jedem und in jeden Knoten geht genau ein Pfeil,

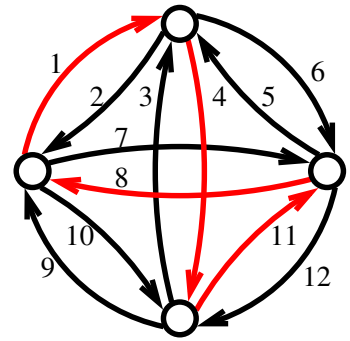
$$\text{für } v \in V: \quad \sum_{(v,u)\in E} x_{(v,u)} = 1, \quad \sum_{(u,v)\in E} x_{(u,v)} = 1$$

(pro Zeile/Spalte genau eine 1 $\leftrightarrow$ Zuweisungsproblem)

Schon Formulierung? $P \cap \mathbb{Z}^E = \{\chi(R) : R \text{ Rundreise}\}$?

Ganzzahlige Formulierung für (TSP)

Ziel: P_{TSP} durch ein größeres Polytop $P = \{x \in \mathbb{R}^n : Ax \leq b\} \supseteq P_{TSP}$ möglichst eng einfassen, sodass wenigstens $P \cap \mathbb{Z}^E = \{\chi(R) : R \text{ Rundreise}\}$.



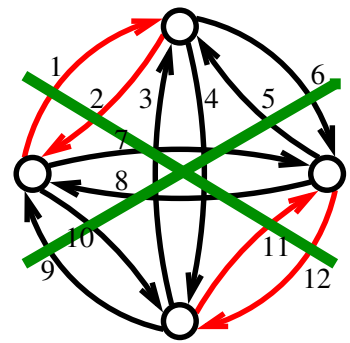
Eine Gleichung/Ungleichung heißt gültig für P_{TSP}, wenn sie für alle $x \in \{\chi(R) : R \text{ Rundreise}\}$ erfüllt ist.

0-1 Würfel: $0 \leq x \leq 1$ ist gültig

Gradgleichungen:
aus jedem und in jeden Knoten geht genau ein Pfeil,

$$\text{für } v \in V: \quad \sum_{(v,u)\in E} x_{(v,u)} = 1, \quad \sum_{(u,v)\in E} x_{(u,v)} = 1$$

(pro Zeile/Spalte genau eine 1 $\leftrightarrow$ Zuweisungsproblem)



Schon Formulierung? $P \cap \mathbb{Z}^E = \{\chi(R) : R \text{ Rundreise}\}$?

Kurzzyklusungleichungen:
Aus jeder Knoten-Teilmenge muss mindestens eine Kante hinausführen,

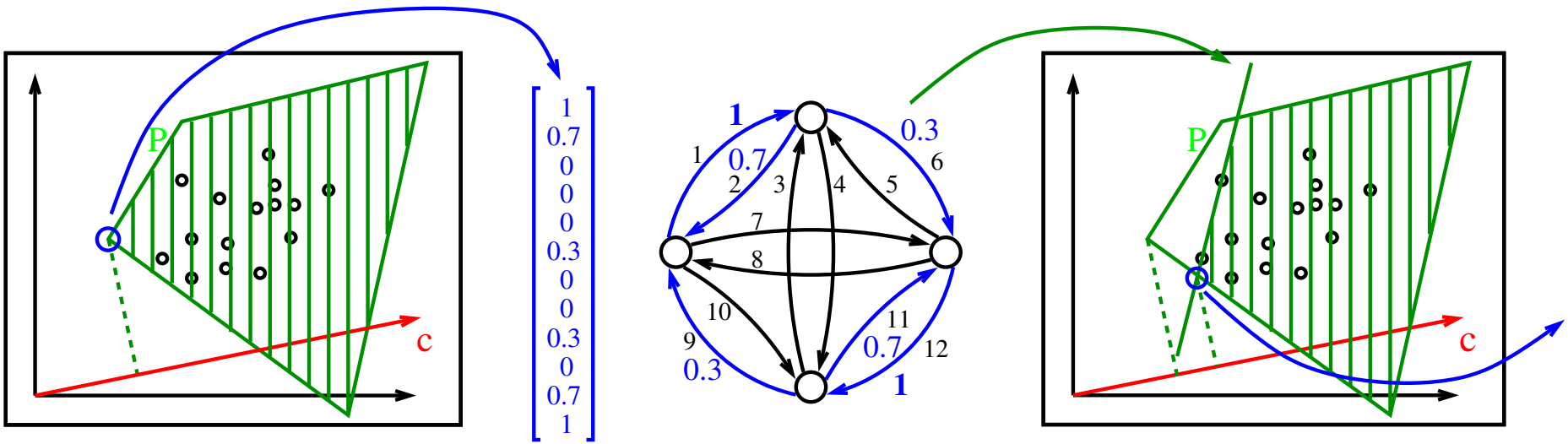
$$\text{für } S \subset V, 2 \leq |S| \leq n-2: \quad \sum_{e\in\delta^+(S)} x_e \geq 1$$

Ist nun eine Formulierung, aber mit etwa 2^n Ungleichungen!

Lösen der TSP LP-Relaxation

Geht nur mit Schnittebenenverfahren:
Die 1. Relaxation bildet das Zuweisungsproblem (Box+Gradgleichungen)
Dessen Lösung ist ganzzahlig und besteht meist aus getrennten Kreisen.

Ab jetzt die Schranke iterativ durch Kurzzyklus-Schnittebenen verbessern

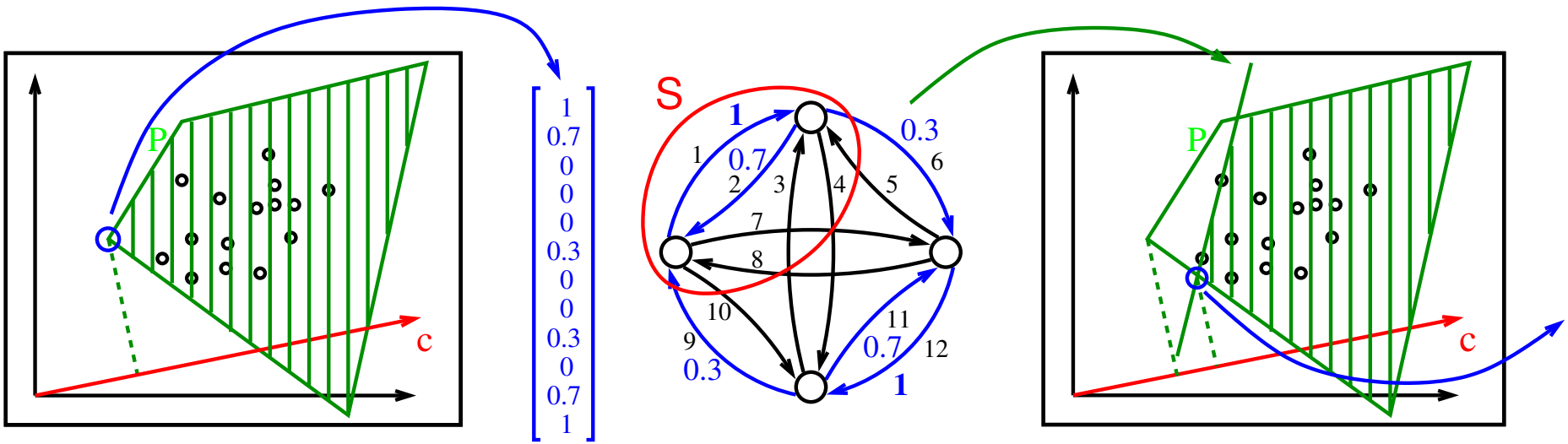


Separationsproblem: Finde ein $S \subset V, 2 \leq |S| \leq n-2$: $\sum_{e \in \delta^+(S)} x_e \not\geq 1$.

Lösen der TSP LP-Relaxation

Geht nur mit Schnittebenenverfahren:
Die 1. Relaxation bildet das Zuweisungsproblem (Box+Gradgleichungen)
Dessen Lösung ist ganzzahlig und besteht meist aus getrennten Kreisen.

Ab jetzt die Schranke iterativ durch Kurzzyklus-Schnittebenen verbessern



Separationsproblem: Finde ein $S \subset V, 2 \leq |S| \leq n-2$: $\sum_{e \in \delta^+(S)} x_e \not\geq 1$.

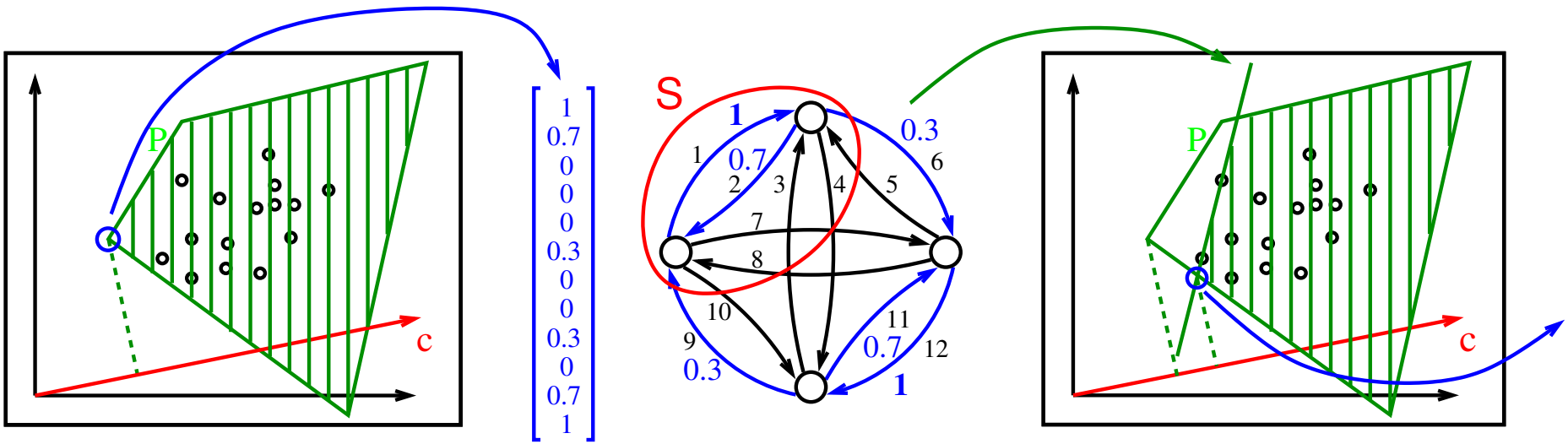
$\Leftrightarrow$ Finde im Netzwerk $D = (V, E)$ mit Kantenkap. x einen Schnitt $x(\delta^+(S)) < 1$.

$\rightarrow$ Maximale s-t-Flüsse/minimale s-t-Schnitte für $s, t \in V$, exakt lösbar!

Lösen der TSP LP-Relaxation

Geht nur mit Schnittebenenverfahren:
Die 1. Relaxation bildet das Zuweisungsproblem (Box+Gradgleichungen)
Dessen Lösung ist ganzzahlig und besteht meist aus getrennten Kreisen.

Ab jetzt die Schranke iterativ durch Kurzzyklus-Schnittebenen verbessern



Separationsproblem: Finde ein $S \subset V, 2 \leq |S| \leq n-2$: $\sum_{e \in \delta^+(S)} x_e \not\geq 1$.

$\Leftrightarrow$ Finde im Netzwerk $D = (V, E)$ mit Kantenkap. x einen Schnitt $x(\delta^+(S)) < 1$.
$\rightarrow$ Maximale s-t-Flüsse/minimale s-t-Schnitte für $s, t \in V$, exakt lösbar!

Grad+Kurzz. liefern sehr gute Schranken, aber noch lange nicht P_{TSP}!
Schranke mit weiteren Unglg. verbesserbar (Kamm-, etc.),
die Lösung der Relaxation wird aber fast nie ganzzahlig!

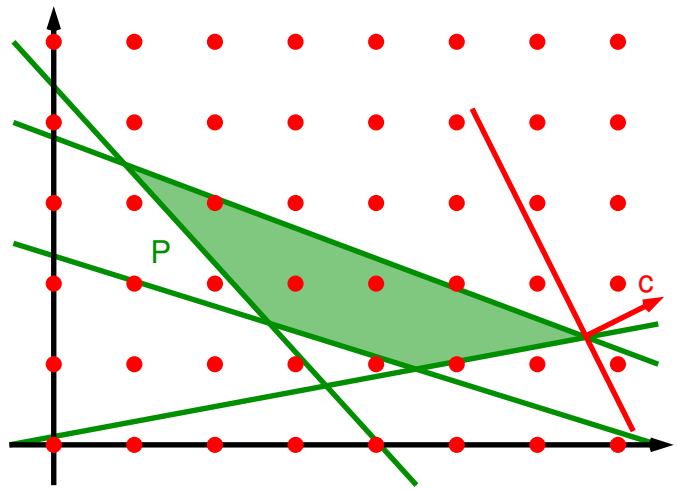
Allgemeine Schnittebenen

Es gibt auch problemübergreifende allgemeine Schnittebenen, die in state-of-the-art Lösern für ganzz. Opt. eingesetzt werden:

- Gomory-Schnitte [Abrunden der Koeffizienten durch $\lfloor \cdot \rfloor$]

  Ist $a^T x \leq \beta$ gültig für $x \in P \cap \mathbb{Z}^n_+$ dann ist wegen $\lfloor a \rfloor^T x \leq a^T x$ auch $\lfloor a \rfloor^T x \leq \lfloor \beta \rfloor$ gültig.

  So eine verletzte Unglg. ist automatisch aus nicht ganzz. OLs erzeugbar.



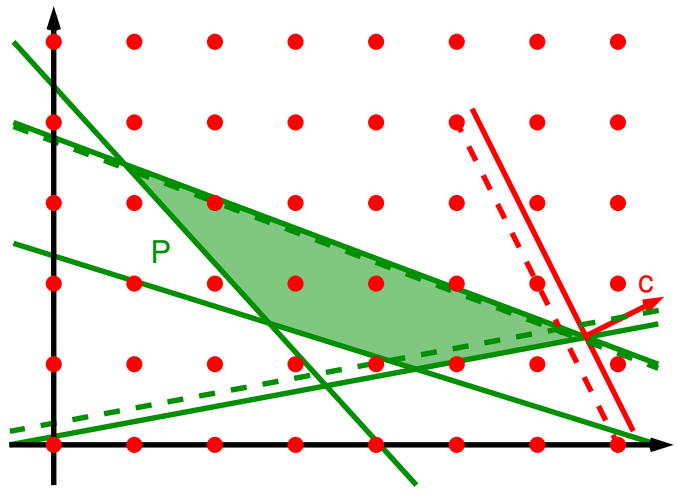
Allgemeine Schnittebenen

Es gibt auch problemübergreifende allgemeine Schnittebenen, die in state-of-the-art Lösern für ganzz. Opt. eingesetzt werden:

- Gomory-Schnitte [Abrunden der Koeffizienten durch $\lfloor \cdot \rfloor$]

  Ist $a^T x \leq \beta$ gültig für $x \in P \cap \mathbb{Z}^n_+$ dann ist wegen $\lfloor a \rfloor^T x \leq a^T x$ auch $\lfloor a \rfloor^T x \leq \lfloor \beta \rfloor$ gültig.

  So eine verletzte Unglg. ist automatisch aus nicht ganzz. OLs erzeugbar.



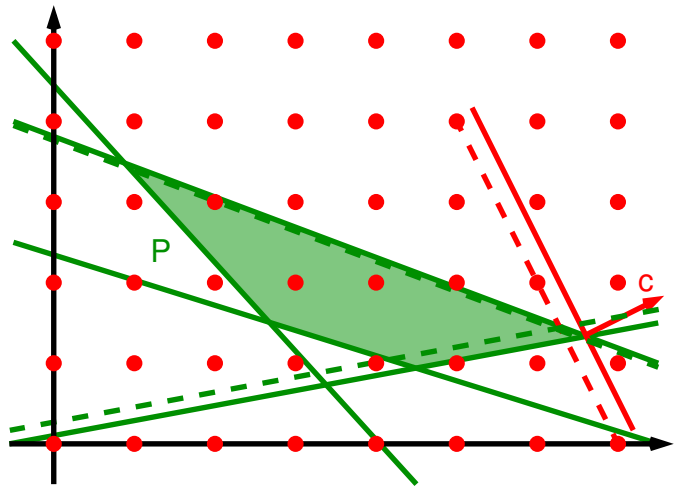
Allgemeine Schnittebenen

Es gibt auch problemübergreifende allgemeine Schnittebenen, die in state-of-the-art Lösern für ganzz. Opt. eingesetzt werden:

- Gomory-Schnitte [Abrunden der Koeffizienten durch $\lfloor \cdot \rfloor$]

  Ist $a^T x \leq \beta$ gültig für $x \in P \cap \mathbb{Z}^n_+$ dann ist wegen $\lfloor a \rfloor^T x \leq a^T x$ auch $\lfloor a \rfloor^T x \leq \lfloor \beta \rfloor$ gültig.

  So eine verletzte Unglg. ist automatisch aus nicht ganzz. OLs erzeugbar.



- Lift-and-Project Cuts,
- Clique-Ungleichungen,

etc.

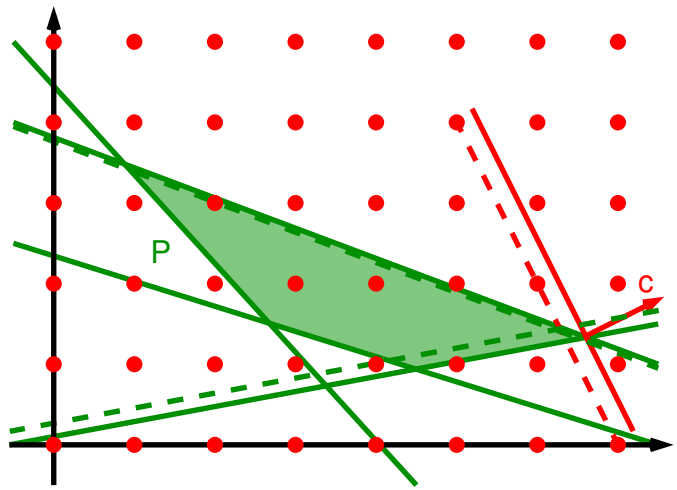
Allgemeine Schnittebenen

Es gibt auch problemübergreifende allgemeine Schnittebenen, die in state-of-the-art Lösern für ganzz. Opt. eingesetzt werden:

- Gomory-Schnitte [Abrunden der Koeffizienten durch $\lfloor \cdot \rfloor$]

  Ist $a^T x \leq \beta$ gültig für $x \in P \cap \mathbb{Z}^n_+$ dann ist wegen $\lfloor a \rfloor^T x \leq a^T x$ auch $\lfloor a \rfloor^T x \leq \lfloor \beta \rfloor$ gültig.

  So eine verletzte Unglg. ist automatisch aus nicht ganzz. OLs erzeugbar.



- Lift-and-Project Cuts,
- Clique-Ungleichungen,

etc.

LP-Relaxation mit Schnittebenen erzeugt gute Schranken (obere für Maximierungs-, untere für Minimierungsprobleme), die Lösungen der Relaxation sind (fast) nie ganzzahlig, führen aber oft in die Nähe guter ganzzahliger Lösungen.

Inhaltsübersicht

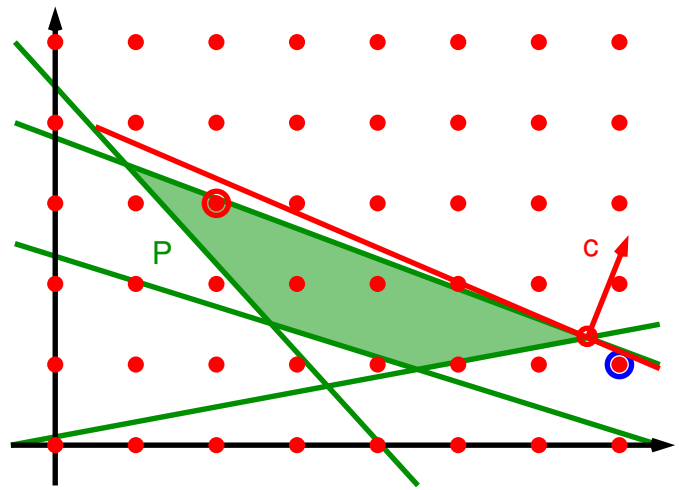
3.10 Finden „guter“ Lösungen, Heuristiken

[Heuristik hat den griechischen Wortstamm finden/erfinden]

Für „kleine“ $x \in \mathbb{Z}^n$ liefert normales Runden der LP-Lösung meist unzulässige oder schlechte Lösungen (selbst bei guter Schranke).

Es kann vorkommen, dass kein zulässiger Punkt in der Nähe der LP-Lösung ist!

In state-of-the-art Lösern gibt es aufwendige Standard-Rundeheuristiken (feasibility pump).



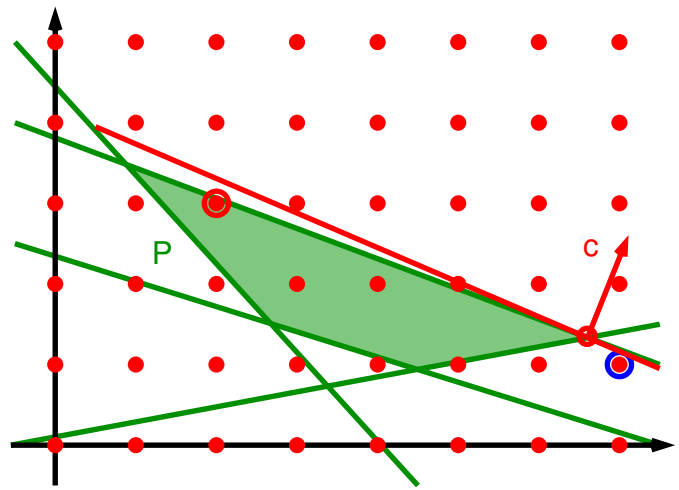
3.10 Finden „guter“ Lösungen, Heuristiken

[Heuristik hat den griechischen Wortstamm finden/erfinden]

Für „kleine“ $x \in \mathbb{Z}^n$ liefert normales Runden der LP-Lösung meist unzulässige oder schlechte Lösungen (selbst bei guter Schranke).

Es kann vorkommen, dass kein zulässiger Punkt in der Nähe der LP-Lösung ist!

In state-of-the-art Lösern gibt es aufwendige Standard-Rundeheuristiken (feasibility pump).



Generell: Ganzzahlige Probleme sind meist *NP*-schwer, haben viele „lokale Optima“ (keine benachbarte bessere Lösung) und es ist unwahrscheinlich, dass man die Optimallösung ohne Enumeration findet. Es kann sogar schwer sein, überhaupt eine zulässige Lösung zu finden!

$\rightarrow$ In Anwendungen nützt man möglichst viel problemspezifisches Wissen!

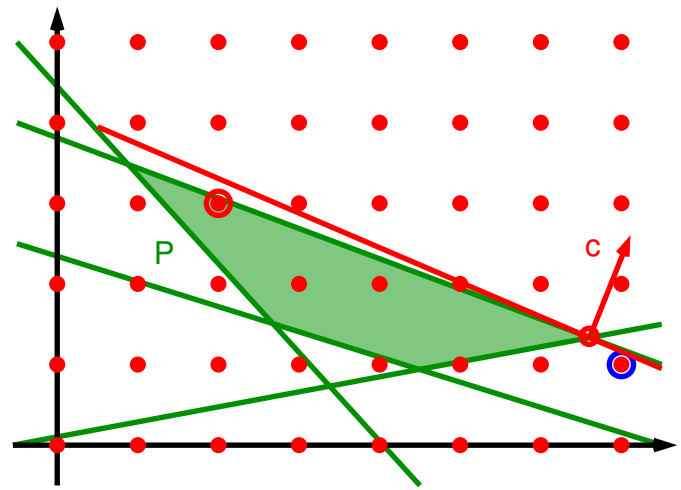
3.10 Finden „guter“ Lösungen, Heuristiken

[Heuristik hat den griechischen Wortstamm finden/erfinden]

Für „kleine“ $x \in \mathbb{Z}^n$ liefert normales Runden der LP-Lösung meist unzulässige oder schlechte Lösungen (selbst bei guter Schranke).

Es kann vorkommen, dass kein zulässiger Punkt in der Nähe der LP-Lösung ist!

In state-of-the-art Lösern gibt es aufwendige Standard-Rundeheuristiken (feasibility pump).



Generell: Ganzzahlige Probleme sind meist *NP*-schwer, haben viele „lokale Optima“ (keine benachbarte bessere Lösung) und es ist unwahrscheinlich, dass man die Optimallösung ohne Enumeration findet. Es kann sogar schwer sein, überhaupt eine zulässige Lösung zu finden!

$\longrightarrow$ In Anwendungen nützt man möglichst viel problemspezifisches Wissen!

Grober Ablauf:

- (zulässige?) Startlösung erzeugen (meist aus der LP-Lösung)
- Iterative Verbesserung der Lösung durch lokale Suche (lokal exakt, Simulated Annealing, Tabu Search, Genetische Algorithmen, etc.)

Startlösung aus der LP-Relaxation

Typische Ansätze:

- Oft darf aus mehreren $\{0,1\}$-Variablen nur eine gewählt werden:

  LP-Relaxation: $\sum_{i \in N} x_i = 1,\ x_i \in [0,1]$

  Interpretiere x_i als Wahrscheinlichkeit, dass x_i auf 1 gesetzt werden soll und erzeuge damit mehrere Lösungen zufällig, nimm die beste.

Startlösung aus der LP-Relaxation

Typische Ansätze:

- Oft darf aus mehreren $\{0,1\}$-Variablen nur eine gewählt werden:

  LP-Relaxation: $\sum_{i \in N} x_i = 1,\ x_i \in [0,1]$

  Interpretiere x_i als Wahrscheinlichkeit, dass x_i auf 1 gesetzt werden soll und erzeuge damit mehrere Lösungen zufällig, nimm die beste.
- Abweichungen von Nebenbedingungen mit großen Dualvariablen erzeugen große Verluste im Zielfunktionswert.

  $\Rightarrow$ Versuche, diese beim Runden zu erfüllen.

Startlösung aus der LP-Relaxation

Typische Ansätze:

- Oft darf aus mehreren $\{0,1\}$-Variablen nur eine gewählt werden:
  LP-Relaxation: $\sum_{i \in N} x_i = 1,\ x_i \in [0,1]$
  Interpretiere x_i als Wahrscheinlichkeit, dass x_i auf 1 gesetzt werden soll und erzeuge damit mehrere Lösungen zufällig, nimm die beste.
- Abweichungen von Nebenbedingungen mit großen Dualvariablen erzeugen große Verluste im Zielfunktionswert.
  $\Rightarrow$ Versuche, diese beim Runden zu erfüllen.
- Sukzessives Fixieren: Setze eine oder mehrere Variablen, deren Wert „fast“ ganzzahlig ist, auf den gerundeten Wert und löse das LP für die restlichen Variablen erneut (u.U. rückgängig machen, falls nun unzulässig).

Startlösung aus der LP-Relaxation

Typische Ansätze:

- Oft darf aus mehreren $\{0,1\}$-Variablen nur eine gewählt werden:
  LP-Relaxation: $\sum_{i \in N} x_i = 1,\ x_i \in [0,1]$
  Interpretiere x_i als Wahrscheinlichkeit, dass x_i auf 1 gesetzt werden soll und erzeuge damit mehrere Lösungen zufällig, nimm die beste.
- Abweichungen von Nebenbedingungen mit großen Dualvariablen erzeugen große Verluste im Zielfunktionswert.
  $\Rightarrow$ Versuche, diese beim Runden zu erfüllen.
- Sukzessives Fixieren: Setze eine oder mehrere Variablen, deren Wert „fast“ ganzzahlig ist, auf den gerundeten Wert und löse das LP für die restlichen Variablen erneut (u.U. rückgängig machen, falls nun unzulässig).

Für einige grundlegende Probleme gibt es Rundungsverfahren, die aus LP-Lösung gerundete Lösungen mit Gütegarantie erzeugen (Approximationsalgorithmen), diese sind gute Ideenlieferanten für eigene Rundungsverfahren.

Verbesserungsverfahren allgemein

Gemeinsame Grundelemente:

- **Suchumgebung/Nachbarschaft erklären:** beschreibt, welche Lösungen ausgehend von der derzeitigen untersucht werden können oder sollen (etwa alle durch gewisse Austauschschritte erreichbaren Lösungen, oder Freigeben gewisser Variablen für lokales Nachoptimieren, etc.)
  Mathematisch: Jeder Lösung $\hat{x}$ wird eine (Nachbarschafts-) Menge $\mathcal{N}(\hat{x})$ von benachbarten Lösungen zugeordnet.

Verbesserungsverfahren allgemein

Gemeinsame Grundelemente:

- **Suchumgebung/Nachbarschaft erklären:** beschreibt, welche Lösungen ausgehend von der derzeitigen untersucht werden können oder sollen (etwa alle durch gewisse Austauschschritte erreichbaren Lösungen, oder Freigeben gewisser Variablen für lokales Nachoptimieren, etc.)
  Mathematisch: Jeder Lösung $\hat{x}$ wird eine (Nachbarschafts-) Menge $\mathcal{N}(\hat{x})$ von benachbarten Lösungen zugeordnet.
- **Fortschrittsmaß definieren:** Bewertungsfunktion $f(x)$ für neu erzeugte Lösungen, die Zielfunktion und Bestrafung von Unzulässigkeiten kombiniert
  [s. Merit- und Filter-Ansatz der nichtlin. Opt.]

Verbesserungsverfahren allgemein

Gemeinsame Grundelemente:

- **Suchumgebung/Nachbarschaft erklären:** beschreibt, welche Lösungen ausgehend von der derzeitigen untersucht werden können oder sollen (etwa alle durch gewisse Austauschschritte erreichbaren Lösungen, oder Freigeben gewisser Variablen für lokales Nachoptimieren, etc.)
  Mathematisch: Jeder Lösung $\hat{x}$ wird eine (Nachbarschafts-) Menge $\mathcal{N}(\hat{x})$ von benachbarten Lösungen zugeordnet.
- **Fortschrittsmaß definieren:** Bewertungsfunktion $f(x)$ für neu erzeugte Lösungen, die Zielfunktion und Bestrafung von Unzulässigkeiten kombiniert
  [s. Merit- und Filter-Ansatz der nichtlin. Opt.]
- **Akzeptanz-Schema festlegen:** Gibt an, welche neu erzeugten Lösungen zur Fortsetzung der Suche verwendet werden sollen. Um lokale Optima verlassen zu können, werden ab und zu auch Verschlechterungen akzeptiert.

Lokal exakte Verfahren/lokales Enumerieren

Bestimme $\mathcal{N}(\cdot)$ so, dass $(P_{\hat{x}})$ $\min f(x)$ s.t. $x \in \mathcal{N}(\hat{x})$ für jedes $\hat{x}$ durch polynomiale Verfahren oder vollständige Enumeration exakt lösbar ist.

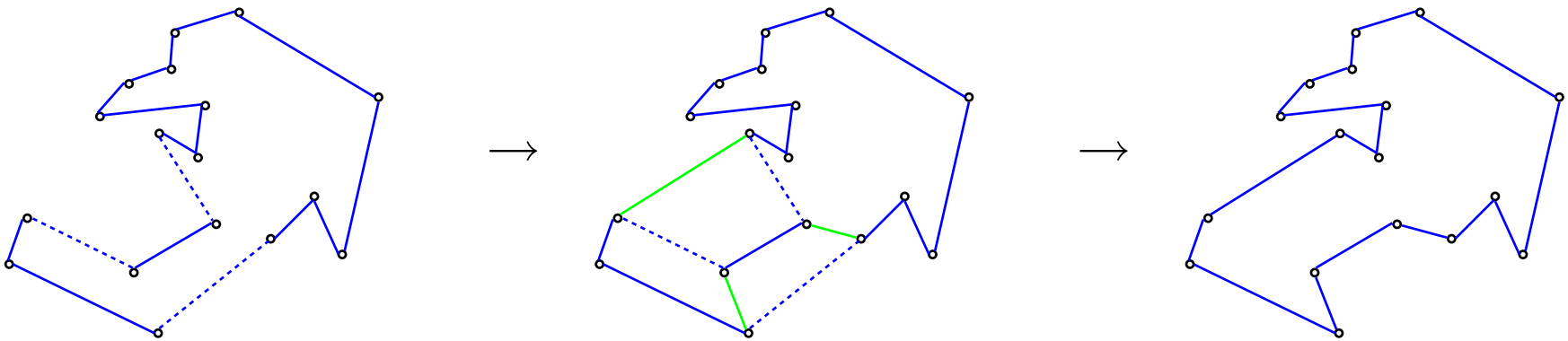
0. Bestimme eine Startlösung $\hat{x}$
1. Löse $(P_{\hat{x}}) \rightarrow \bar{x}$
2. Ist $f(\bar{x})$ besser als $f(\hat{x})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1., sonst STOP.

Lokal exakte Verfahren/lokales Enumerieren

Bestimme $\mathcal{N}(\cdot)$ so, dass $(P_{\hat{x}})$ $\min f(x)$ s.t. $x \in \mathcal{N}(\hat{x})$ für jedes $\hat{x}$ durch polynomiale Verfahren oder vollständige Enumeration exakt lösbar ist.

0. Bestimme eine Startlösung $\hat{x}$
1. Löse $(P_{\hat{x}}) \rightarrow \bar{x}$
2. Ist $f(\bar{x})$ besser als $f(\hat{x})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1., sonst STOP.

Bsp: 3-opt für TSP: 3 Kanten entfernen und neu zusammensetzen

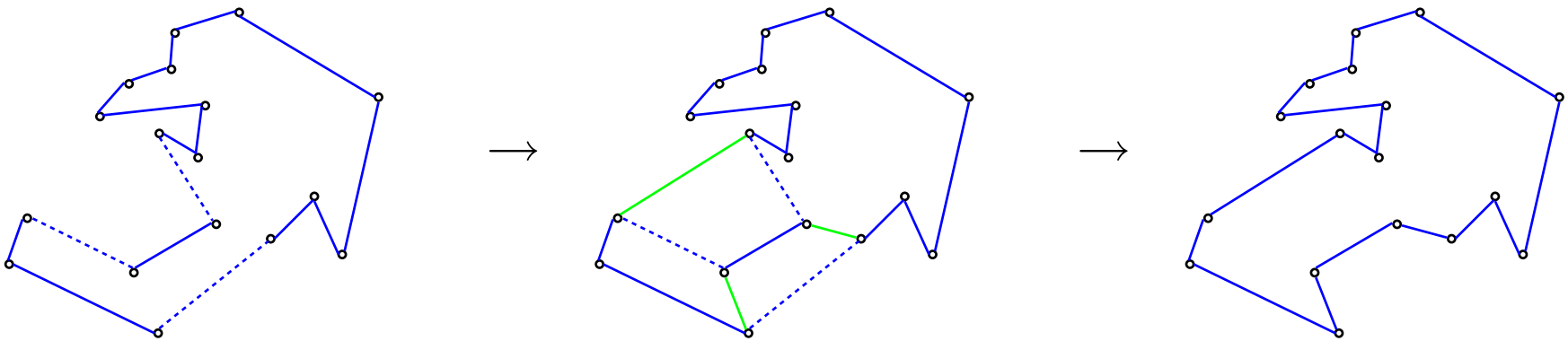


Lokal exakte Verfahren/lokales Enumerieren

Bestimme $\mathcal{N}(\cdot)$ so, dass $(P_{\hat{x}})$ $\min f(x)$ s.t. $x \in \mathcal{N}(\hat{x})$ für jedes $\hat{x}$ durch polynomiale Verfahren oder vollständige Enumeration exakt lösbar ist.

0. Bestimme eine Startlösung $\hat{x}$
1. Löse $(P_{\hat{x}}) \rightarrow \bar{x}$
2. Ist $f(\bar{x})$ besser als $f(\hat{x})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1., sonst STOP.

Bsp: 3-opt für TSP: 3 Kanten entfernen und neu zusammensetzen



---

die Kunst: Finde eine möglichst mächtige Nachbarschaft, für die $(P_{\hat{x}})$ noch polynomial lösbar ist.

Die Anzahl der Iterationen kann dennoch exponentiell sein!

Simulated Annealing (simuliertes langsames Abkühlen)

Erzeuge, in Schritt k, zufällig ein $\bar{x}$ aus $\mathcal{N}(\hat{x})$. Akzeptiere es, falls $f(\bar{x})$ besser als $f(\hat{x})$, sonst akzeptiere es nur mit Wahrscheinlichkeit

$$\exp\left(\frac{-|f(\hat{x}) - f(\bar{x})|}{T_k}\right) \qquad (0 < T_k \to 0 \text{ für } k \to \infty).$$

0. Bestimme Startlösung $\hat{x}$, Nullfolge $\{T_k > 0\}_{k\in\mathbb{N}}$, setze $k = 0$.
1. Wähle zufällig (gleichverteilt) $\bar{x} \in \mathcal{N}(\hat{x})$ setze $k \leftarrow k + 1$.
2. Ist $f(\bar{x})$ besser als $f(\hat{x})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1.
3. Ziehe gleichverteilt eine Zufallszahl $\zeta \in [0, 1]$.
   Ist $\zeta < \exp(\frac{-|f(\hat{x})-f(\bar{x})|}{c_k})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1.
4. Gehe zu 1. (ohne $\hat{x}$ zu ändern).

Wähle $\mathcal{N}(\cdot)$ so, dass jedes x über Zwischenstationen erreichbar ist.

Simulated Annealing (simuliertes langsames Abkühlen)

Erzeuge, in Schritt k, zufällig ein $\bar{x}$ aus $\mathcal{N}(\hat{x})$. Akzeptiere es, falls $f(\bar{x})$ besser als $f(\hat{x})$, sonst akzeptiere es nur mit Wahrscheinlichkeit

$$\exp\left(\frac{-|f(\hat{x}) - f(\bar{x})|}{T_k}\right) \qquad (0 < T_k \to 0 \text{ für } k \to \infty).$$

0. Bestimme Startlösung $\hat{x}$, Nullfolge $\{T_k > 0\}_{k\in\mathbb{N}}$, setze $k = 0$.
1. Wähle zufällig (gleichverteilt) $\bar{x} \in \mathcal{N}(\hat{x})$ setze $k \leftarrow k + 1$.
2. Ist $f(\bar{x})$ besser als $f(\hat{x})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1.
3. Ziehe gleichverteilt eine Zufallszahl $\zeta \in [0, 1]$.
   Ist $\zeta < \exp(\frac{-|f(\hat{x})-f(\bar{x})|}{c_k})$, setze $\hat{x} \leftarrow \bar{x}$ und gehe zu 1.
4. Gehe zu 1. (ohne $\hat{x}$ zu ändern).

Wähle $\mathcal{N}(\cdot)$ so, dass jedes x über Zwischenstationen erreichbar ist.

Geht die (Temperatur-/Abkühlungs-)Folge T_k sehr langsam gegen Null, wird jedes x mit positiver Wahrscheinlichkeit irgendwann besucht (vollständige Enumeration), also auch das Optimum. (Aber wann?)

Tabu-Search

Idee: Versuche möglichst unterschiedliche Lösungen zu erzeugen.

Beschreibe $\mathcal{N}(\cdot)$ durch Änderungsregeln $r \in \mathcal{R}$ und speichere verwendete Regeln in einer Tabuliste $\mathcal{L}$. Für ein neues $\bar{x}$ sollte wenigstens eine Regel $r \in \mathcal{R} \setminus \mathcal{L}$ verwendet werden oder es muss besseren Wert haben.

0. Bestimme Startlösung $\hat{x}$, setze $\mathcal{L} = \emptyset$.
1. Erzeuge einige $x \in \mathcal{N}(\hat{x})$ durch mehrmaliges zufälliges Anwenden von Regeln aus $\mathcal{R}$, sammle diese in einer Menge S.
2. Wähle ein $\bar{x}$ aus S nach Tabuliste $\mathcal{L}$ und $f(\cdot)$.
3. Aktualisiere die Tabuliste $\mathcal{L}$, setze $\hat{x} \leftarrow \bar{x}$, gehe zu 1.

Tabu-Search

Idee: Versuche möglichst unterschiedliche Lösungen zu erzeugen.

Beschreibe $\mathcal{N}(\cdot)$ durch Änderungsregeln $r \in \mathcal{R}$ und speichere verwendete Regeln in einer Tabuliste $\mathcal{L}$. Für ein neues $\bar{x}$ sollte wenigstens eine Regel $r \in \mathcal{R} \setminus \mathcal{L}$ verwendet werden oder es muss besseren Wert haben.

0. Bestimme Startlösung $\hat{x}$, setze $\mathcal{L} = \emptyset$.
1. Erzeuge einige $x \in \mathcal{N}(\hat{x})$ durch mehrmaliges zufälliges Anwenden von Regeln aus $\mathcal{R}$, sammle diese in einer Menge S.
2. Wähle ein $\bar{x}$ aus S nach Tabuliste $\mathcal{L}$ und $f(\cdot)$.
3. Aktualisiere die Tabuliste $\mathcal{L}$, setze $\hat{x} \leftarrow \bar{x}$, gehe zu 1.

Bsp. TSP: $\mathcal{R} = \{r_{ij} := \text{vertausche Reihenfolge von Stadt } i \text{ und } j\}$.
$\mathcal{L} = \{r_{ij} : r_{ij} \text{ wurde in den letzten } n/10 \text{ Schritten verwendet}\}$

Tabu-Search

Idee: Versuche möglichst unterschiedliche Lösungen zu erzeugen.

Beschreibe $\mathcal{N}(\cdot)$ durch Änderungsregeln $r \in \mathcal{R}$ und speichere verwendete Regeln in einer Tabuliste $\mathcal{L}$. Für ein neues $\bar{x}$ sollte wenigstens eine Regel $r \in \mathcal{R} \setminus \mathcal{L}$ verwendet werden oder es muss besseren Wert haben.

0. Bestimme Startlösung $\hat{x}$, setze $\mathcal{L} = \emptyset$.
1. Erzeuge einige $x \in \mathcal{N}(\hat{x})$ durch mehrmaliges zufälliges Anwenden von Regeln aus $\mathcal{R}$, sammle diese in einer Menge S.
2. Wähle ein $\bar{x}$ aus S nach Tabuliste $\mathcal{L}$ und $f(\cdot)$.
3. Aktualisiere die Tabuliste $\mathcal{L}$, setze $\hat{x} \leftarrow \bar{x}$, gehe zu 1.

Bsp. TSP: $\mathcal{R} = \{r_{ij} := \text{vertausche Reihenfolge von Stadt } i \text{ und } j\}$.
$\mathcal{L} = \{r_{ij} : r_{ij} \text{ wurde in den letzten } n/10 \text{ Schritten verwendet}\}$

Über die Regeln $\mathcal{R}$ sollte jede Lösung erreichbar sein.
Allgemeine theoretische Resultate oder Qualitätsgarantien gibt es nicht.

Genetische Algorithmen

Idee: Lasse die Evolution für Dich arbeiten (und warte derweilen).

Erzeuge aus einer Population von Lösungen durch Selektion (Auswahl der nächsten Eltern), Rekombination/Vermehrung (Austausch von Teillösungen), Mutation (zufälliges Verändern) die nächste Population.

0. Wähle $k \in \mathbb{N}$ und bestimme eine Startpopulation $\mathcal{P}$, $|\mathcal{P}| \geq 2k$.
1. Bestimme die durchschnittliche Fitness $\bar{f} = \sum_{x \in \mathcal{P}} f(x)/|\mathcal{P}|$
2. Lösche x aus $\mathcal{P}$ mit Wahrscheinlichkeit prop. zu $\frac{f(x)}{\bar{f}}$, bis $|P| = 2k$.
3. Bilde zufällig k Paare aus $\mathcal{P}$, erzeuge für jedes Paar einige Nachkommen durch Rekombination und Mutation $\longrightarrow \bar{\mathcal{P}}$
4. Setze $\mathcal{P} \leftarrow \bar{\mathcal{P}}$, gehe zu 1.

Genetische Algorithmen

Idee: Lasse die Evolution für Dich arbeiten (und warte derweilen).

Erzeuge aus einer Population von Lösungen durch Selektion (Auswahl der nächsten Eltern), Rekombination/Vermehrung (Austausch von Teillösungen), Mutation (zufälliges Verändern) die nächste Population.

0. Wähle $k \in \mathbb{N}$ und bestimme eine Startpopulation $\mathcal{P}$, $|\mathcal{P}| \geq 2k$.
1. Bestimme die durchschnittliche Fitness $\bar{f} = \sum_{x \in \mathcal{P}} f(x)/|\mathcal{P}|$
2. Lösche x aus $\mathcal{P}$ mit Wahrscheinlichkeit prop. zu $\frac{f(x)}{\bar{f}}$, bis $|P| = 2k$.
3. Bilde zufällig k Paare aus $\mathcal{P}$, erzeuge für jedes Paar einige Nachkommen durch Rekombination und Mutation $\longrightarrow \bar{\mathcal{P}}$
4. Setze $\mathcal{P} \leftarrow \bar{\mathcal{P}}$, gehe zu 1.

- Viele Experimente mit Population der Größe 1 (!!!)
- Theorie besagt, dass Simulated Annealing eher Optima findet.

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).
- Meta-Heuristik-Industrie betrachtet beliebige Kombinationen, jede Menge „neue“ Verfahren (Ant-Colony-, Particle-swarm-, etc.)

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).
- Meta-Heuristik-Industrie betrachtet beliebige Kombinationen, jede Menge „neue“ Verfahren (Ant-Colony-, Particle-swarm-, etc.)
- Fast alle hoffen, mit (leicht gesteuertem) Zufall eine Optimallösung zu finden (passt zu *NP*!).

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).
- Meta-Heuristik-Industrie betrachtet beliebige Kombinationen, jede Menge „neue“ Verfahren (Ant-Colony-, Particle-swarm-, etc.)
- Fast alle hoffen, mit (leicht gesteuertem) Zufall eine Optimallösung zu finden (passt zu *NP*!).

Vorteile:

- Auch bei wenig Problemverständnis ist rasch etwas implementiert, erste Lösungen sind schnell erzeugt, Regeln schnell geändert.

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).
- Meta-Heuristik-Industrie betrachtet beliebige Kombinationen, jede Menge „neue“ Verfahren (Ant-Colony-, Particle-swarm-, etc.)
- Fast alle hoffen, mit (leicht gesteuertem) Zufall eine Optimallösung zu finden (passt zu *NP*!).

Vorteile:

- Auch bei wenig Problemverständnis ist rasch etwas implementiert, erste Lösungen sind schnell erzeugt, Regeln schnell geändert.
- Man kann viele Parameter einstellen.

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).
- Meta-Heuristik-Industrie betrachtet beliebige Kombinationen, jede Menge „neue“ Verfahren (Ant-Colony-, Particle-swarm-, etc.)
- Fast alle hoffen, mit (leicht gesteuertem) Zufall eine Optimallösung zu finden (passt zu *NP*!).

Vorteile:

- Auch bei wenig Problemverständnis ist rasch etwas implementiert, erste Lösungen sind schnell erzeugt, Regeln schnell geändert.
- Man kann viele Parameter einstellen.

Nachteile:

- Man muss viele Parameter einstellen, ohne gute Richtschnur!

Bemerkungen

- SA, TS, GA sind Meta-Heuristiken (Schema ohne Problembezug).
- Meta-Heuristik-Industrie betrachtet beliebige Kombinationen, jede Menge „neue“ Verfahren (Ant-Colony-, Particle-swarm-, etc.)
- Fast alle hoffen, mit (leicht gesteuertem) Zufall eine Optimallösung zu finden (passt zu *NP*!).

Vorteile:

- Auch bei wenig Problemverständnis ist rasch etwas implementiert, erste Lösungen sind schnell erzeugt, Regeln schnell geändert.
- Man kann viele Parameter einstellen.

Nachteile:

- Man muss viele Parameter einstellen, ohne gute Richtschnur!
- Konvergenz der Verfahren sagt nichts über die Qualität der Lösung. Ohne dazupassende Relaxation hat man keine Ahnung, wie weit man vom Optimum entfernt ist (manchmal sehr weit).

Inhaltsübersicht

3.11 Gemischt-ganzzahlige Optimierung (MIP)

Für eine Teilmenge der Indices $G \subseteq \{1, \ldots, n\}$ soll x ganzzahlig sein.

$$\max\ c^T x \text{ s.t. } Ax \geq b,\ x \in \mathbb{R}^n, x_G \in \mathbb{Z}^G$$

Enthält ganzz. Optimierung als Spezialfall, ist aber deutlich breiter.

3.11 Gemischt-ganzzahlige Optimierung (MIP)

Für eine Teilmenge der Indices $G \subseteq \{1, \ldots, n\}$ soll x ganzzahlig sein.

$$\max\ c^T x \text{ s.t. } Ax \geq b,\ x \in \mathbb{R}^n, x_G \in \mathbb{Z}^G$$

Enthält ganzz. Optimierung als Spezialfall, ist aber deutlich breiter.

Anwendungsbeispiel: Standortoptimierung mit Fixkosten
Gegeben eine Menge K an Kunden mit Bedarfen b_k und eine Menge M möglicher Standorte für Versandlager, jeweils mit Mietkosten c_m, Kapazität b_m und Transportkosten c_{km} pro Einheit für $k \in K, m \in M$. Welche Standorte sollen geöffnet werden?

Variablen:

3.11 Gemischt-ganzzahlige Optimierung (MIP)

Für eine Teilmenge der Indices $G \subseteq \{1, \ldots, n\}$ soll x ganzzahlig sein.

$$\max\ c^T x \text{ s.t. } Ax \geq b,\ x \in \mathbb{R}^n, x_G \in \mathbb{Z}^G$$

Enthält ganzz. Optimierung als Spezialfall, ist aber deutlich breiter.

Anwendungsbeispiel: Standortoptimierung mit Fixkosten
Gegeben eine Menge K an Kunden mit Bedarfen b_k und eine Menge M möglicher Standorte für Versandlager, jeweils mit Mietkosten c_m, Kapazität b_m und Transportkosten c_{km} pro Einheit für $k \in K, m \in M$.
Welche Standorte sollen geöffnet werden?

Variablen:
$x_m \in \{0, 1\}$, $m \in M$ . . . Versandlager wird in m errichtet.
$x_{km} \in \mathbb{R}_+$, $k \in K, m \in M$ . . . Menge, die Kunde k von Standort m erhält.

Nebenbedingungen:

3.11 Gemischt-ganzzahlige Optimierung (MIP)

Für eine Teilmenge der Indices $G \subseteq \{1, \ldots, n\}$ soll x ganzzahlig sein.

$$\max\ c^T x \text{ s.t. } Ax \geq b,\ x \in \mathbb{R}^n, x_G \in \mathbb{Z}^G$$

Enthält ganzz. Optimierung als Spezialfall, ist aber deutlich breiter.

Anwendungsbeispiel: Standortoptimierung mit Fixkosten

Gegeben eine Menge K an Kunden mit Bedarfen b_k und eine Menge M möglicher Standorte für Versandlager, jeweils mit Mietkosten c_m, Kapazität b_m und Transportkosten c_{km} pro Einheit für $k \in K, m \in M$. Welche Standorte sollen geöffnet werden?

Variablen:

$x_m \in \{0, 1\}$, $m \in M$ ... Versandlager wird in m errichtet.

$x_{km} \in \mathbb{R}_+$, $k \in K, m \in M$ ... Menge, die Kunde k von Standort m erhält.

Nebenbedingungen:

$\sum_{m \in M} x_{km} = b_k$, $k \in K$ ... Kunde k erhält seine Bestellung

$\sum_{k \in K} x_{km} \leq b_m x_m$, $m \in M$ ... Standort m verteilt maximal b_m.

Zielfunktion:

3.11 Gemischt-ganzzahlige Optimierung (MIP)

Für eine Teilmenge der Indices $G \subseteq \{1, \ldots, n\}$ soll x ganzzahlig sein.

$$\max\ c^T x \text{ s.t. } Ax \geq b,\ x \in \mathbb{R}^n, x_G \in \mathbb{Z}^G$$

Enthält ganzz. Optimierung als Spezialfall, ist aber deutlich breiter.

Anwendungsbeispiel: Standortoptimierung mit Fixkosten

Gegeben eine Menge K an Kunden mit Bedarfen b_k und eine Menge M möglicher Standorte für Versandlager, jeweils mit Mietkosten c_m, Kapazität b_m und Transportkosten c_{km} pro Einheit für $k \in K, m \in M$. Welche Standorte sollen geöffnet werden?

Variablen:
$x_m \in \{0, 1\}$, $m \in M$ ... Versandlager wird in m errichtet.
$x_{km} \in \mathbb{R}_+$, $k \in K, m \in M$ ... Menge, die Kunde k von Standort m erhält.

Nebenbedingungen:
$\sum_{m \in M} x_{km} = b_k$, $k \in K$ ... Kunde k erhält seine Bestellung
$\sum_{k \in K} x_{km} \leq b_m x_m$, $m \in M$ ... Standort m verteilt maximal b_m.

Zielfunktion:
$\min \sum_{k \in K, m \in M} c_{km} x_{km} + \sum_{m \in M} c_m x_m = c^T x$

MIP-Modellierungstechniken:

Bedingte Ungleichungen: Ungleichungen, die nur bei bestimmten Entscheidungen beachtet werden müssen, können bei entgegengesetzter Entscheidung durch einen big M-Term erfüllt werden.

MIP-Modellierungstechniken:

Bedingte Ungleichungen: Ungleichungen, die nur bei bestimmten Entscheidungen beachtet werden müssen, können bei entgegengesetzter Entscheidung durch einen big M-Term erfüllt werden.

Bsp: Fährt Zug A mit Abfahrtszeit t_A vor Zug B mit Abfahrtszeit t_B, ist eine Mindestzugfolgezeit $t_{AB} > 0$ einzuhalten, also $t_B \geq t_A + t_{AB}$. Für Zug B vor Zug A muss wiederum $t_A \geq t_B + t_{BA}$ erfüllt sein. Die nachfolgenden Züge dürfen natürlich auch viel später fahren.

Variablen

MIP-Modellierungstechniken:

Bedingte Ungleichungen: Ungleichungen, die nur bei bestimmten Entscheidungen beachtet werden müssen, können bei entgegengesetzter Entscheidung durch einen big M-Term erfüllt werden.

Bsp: Fährt Zug A mit Abfahrtszeit t_A vor Zug B mit Abfahrtszeit t_B, ist eine Mindestzugfolgezeit $t_{AB} > 0$ einzuhalten, also $t_B \geq t_A + t_{AB}$. Für Zug B vor Zug A muss wiederum $t_A \geq t_B + t_{BA}$ erfüllt sein. Die nachfolgenden Züge dürfen natürlich auch viel später fahren.

Variablen ($M \gg 0$, größer als späteste Abfahrtszeit von t_A und t_B):
$x_{AB} \in \{0, 1\}$ . . . 1 falls A vor B fährt, sonst 0
$t_A, t_B \in [0, M]$. . . Abfahrtszeit

Nebenbedingungen:
$t_A + t_{AB} \leq t_B + M(1 - x_{AB})$ . . . nur wichtig, wenn $x_{AB} = 1$
$t_B + t_{BA} \leq t_A + M x_{AB}$ . . . nur wichtig, wenn $x_{AB} = 0$.

MIP-Modellierungstechniken:

Bedingte Ungleichungen: Ungleichungen, die nur bei bestimmten Entscheidungen beachtet werden müssen, können bei entgegengesetzter Entscheidung durch einen big M-Term erfüllt werden.

Bsp: Fährt Zug A mit Abfahrtszeit t_A vor Zug B mit Abfahrtszeit t_B, ist eine Mindestzugfolgezeit $t_{AB} > 0$ einzuhalten, also $t_B \geq t_A + t_{AB}$. Für Zug B vor Zug A muss wiederum $t_A \geq t_B + t_{BA}$ erfüllt sein. Die nachfolgenden Züge dürfen natürlich auch viel später fahren.

Variablen ($M \gg 0$, größer als späteste Abfahrtszeit von t_A und t_B):
$x_{AB} \in \{0, 1\}$ . . . 1 falls A vor B fährt, sonst 0
$t_A, t_B \in [0, M]$. . . Abfahrtszeit

Nebenbedingungen:
$t_A + t_{AB} \leq t_B + M(1 - x_{AB})$ . . . nur wichtig, wenn $x_{AB} = 1$
$t_B + t_{BA} \leq t_A + Mx_{AB}$ . . . nur wichtig, wenn $x_{AB} = 0$.

- M zu groß $\rightarrow$ Unglg. in LP-Relaxation wirkungslos $\rightarrow$ schlechte Schranke
- gut, falls Verletzungsspielraum der Unglg. gut abschätzbar (siehe das Standortplanungsbeispiel)
- in Branch&Bound hilfreich, wenn auf der Variable gebrancht wird

Modellierung logischer Bedingungen

Für ein $x_i \in \{0, 1\}$ steht $x_i = 1$ oft für „Aussage i ist wahr“.
Logische Verknüpfungen sind dann so erzeugbar:

Log. Bedingung	Formulierung
$x_2 = (\text{nicht } x_1)$	

Modellierung logischer Bedingungen

Für ein $x_i \in \{0, 1\}$ steht $x_i = 1$ oft für „Aussage i ist wahr“.
Logische Verknüpfungen sind dann so erzeugbar:

Log. Bedingung	Formulierung
$x_2 = (\text{nicht } x_1)$	$x_2 = 1 - x_1$
$x_3 = (x_1 \text{ oder } x_2)$	

Modellierung logischer Bedingungen

Für ein $x_i \in \{0,1\}$ steht $x_i = 1$ oft für „Aussage i ist wahr“.
Logische Verknüpfungen sind dann so erzeugbar:

Log. Bedingung	Formulierung
$x_2 = (\text{nicht } x_1)$	$x_2 = 1 - x_1$
$x_3 = (x_1 \text{ oder } x_2)$	$x_3 \geq x_1,\ x_3 \geq x_2,\ x_3 \leq x_1 + x_2$
$x_3 = (x_1 \text{ und } x_2)$	$x_3 \leq x_1,\ x_3 \leq x_2,\ x_3 \geq x_1 + x_2 - 1$
$x_1 \Rightarrow x_2$	

Modellierung logischer Bedingungen

Für ein $x_i \in \{0,1\}$ steht $x_i = 1$ oft für „Aussage i ist wahr“.
Logische Verknüpfungen sind dann so erzeugbar:

Log. Bedingung	Formulierung
$x_2 = (\text{nicht } x_1)$	$x_2 = 1 - x_1$
$x_3 = (x_1 \text{ oder } x_2)$	$x_3 \geq x_1,\ x_3 \geq x_2,\ x_3 \leq x_1 + x_2$
$x_3 = (x_1 \text{ und } x_2)$	$x_3 \leq x_1,\ x_3 \leq x_2,\ x_3 \geq x_1 + x_2 - 1$
$x_1 \Rightarrow x_2$	$x_1 \leq x_2$
$x_1 \Leftrightarrow x_2$	

Modellierung logischer Bedingungen

Für ein $x_i \in \{0,1\}$ steht $x_i = 1$ oft für „Aussage i ist wahr“.
Logische Verknüpfungen sind dann so erzeugbar:

Log. Bedingung	Formulierung
$x_2 = (\text{nicht } x_1)$	$x_2 = 1 - x_1$
$x_3 = (x_1 \text{ oder } x_2)$	$x_3 \geq x_1,\ x_3 \geq x_2,\ x_3 \leq x_1 + x_2$
$x_3 = (x_1 \text{ und } x_2)$	$x_3 \leq x_1,\ x_3 \leq x_2,\ x_3 \geq x_1 + x_2 - 1$
$x_1 \Rightarrow x_2$	$x_1 \leq x_2$
$x_1 \Leftrightarrow x_2$	$x_1 = x_2$

Bemerkung: Zusammen mit $0 \leq x_i \leq 1$ beschreiben die Nebenbedinungen

$$\operatorname{conv}\left\{ \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \in \{0,1\}^3 : \text{die } x_i \text{ erfüllen die logische Bedingung} \right\}.$$

Mit dieser Technik sind Formulierungen weiterer Beziehungen ableitbar.
Übung: $x_3 = (x_1 \text{ xor } x_2)$

Schnittebenen für MIP

Wie in der ganzz. Optimierung ist „conv“ die beste lineare Relaxation,

$$P_G := \mathrm{conv}\{x \in \mathbb{R}^n : Ax \leq b, x_G \in \mathbb{Z}^G\}.$$

$Ax \leq b$ ist die LP-Relaxation, P_G wird durch Schnittebenen angenähert.

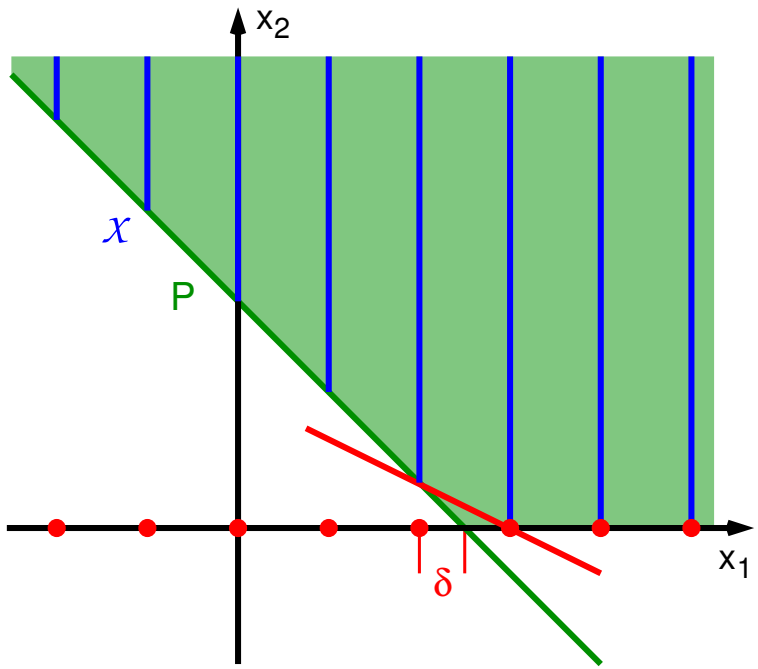
Schnittebenen für MIP

Wie in der ganzz. Optimierung ist „conv“ die beste lineare Relaxation,

$$P_G := \mathrm{conv}\{x \in \mathbb{R}^n : Ax \le b, x_G \in \mathbb{Z}^G\}.$$

$Ax \le b$ ist die LP-Relaxation, P_G wird durch Schnittebenen angenähert.

Bsp: Mixed Integer Rounding Ungleichung (MIR)
Einfachste Form: $\mathcal{X} = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{R}_+ : x_1 + x_2 \ge \beta\}$, $\beta \in \mathbb{R}$



Setze $\delta := \beta - \lfloor \beta \rfloor$,
dann ist die Unglg.

$$x_1 + \frac{1}{\delta} x_2 \ge \lceil \beta \rceil$$

gültig für $\mathcal{X}$.

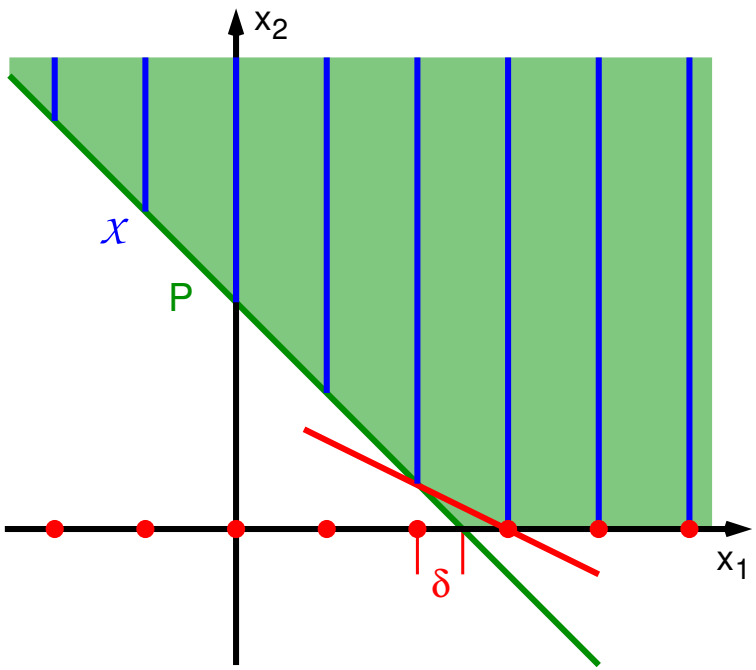
Schnittebenen für MIP

Wie in der ganzz. Optimierung ist „conv“ die beste lineare Relaxation,

$$P_G := \mathrm{conv}\{x \in \mathbb{R}^n : Ax \leq b, x_G \in \mathbb{Z}^G\}.$$

$Ax \leq b$ ist die LP-Relaxation, P_G wird durch Schnittebenen angenähert.

Bsp: Mixed Integer Rounding Ungleichung (MIR)
Einfachste Form: $\mathcal{X} = \{(x_1, x_2) \in \mathbb{Z} \times \mathbb{R}_+ : x_1 + x_2 \geq \beta\}$, $\beta \in \mathbb{R}$



Setze $\delta := \beta - \lfloor \beta \rfloor$,
dann ist die Unglg.

$$x_1 + \frac{1}{\delta} x_2 \geq \lceil \beta \rceil$$

gültig für $\mathcal{X}$.

In state-of-the-art Paketen sind viele weitere enthalten
(flow cover, cliques, etc.)

Branch-and-Cut Frameworks

Wird in einem Branch&Bound-Verfahren für jedes Teilproblem die LP-Relaxation durch Schnittebenen verschärft, spricht man von Branch&Cut-Verfahren. Dies sind derzeit die besten Verfahren für allgemeine gemischt-ganzzahlige Optimierungsprobleme.

Branch-and-Cut Frameworks

Wird in einem Branch&Bound-Verfahren für jedes Teilproblem die LP-Relaxation durch Schnittebenen verschärft, spricht man von Branch&Cut-Verfahren. Dies sind derzeit die besten Verfahren für allgemeine gemischt-ganzzahlige Optimierungsprobleme.

Eine effiziente Branch&Cut Implementierung ist sehr schwer:

- Auswahl des nächsten Teilproblems
- Auswahl der Verzweigungsvariable, Fixieren von Variablen
- Teilprobleme effizient inkrementell speichern
- Schnittebenen über mehrere Teilprobleme verwalten
- effiziente Heuristiken für zulässige Lösungen

etc.

Branch-and-Cut Frameworks

Wird in einem Branch&Bound-Verfahren für jedes Teilproblem die LP-Relaxation durch Schnittebenen verschärft, spricht man von Branch&Cut-Verfahren. Dies sind derzeit die besten Verfahren für allgemeine gemischt-ganzzahlige Optimierungsprobleme.

Eine effiziente Branch&Cut Implementierung ist sehr schwer:

- Auswahl des nächsten Teilproblems
- Auswahl der Verzweigungsvariable, Fixieren von Variablen
- Teilprobleme effizient inkrementell speichern
- Schnittebenen über mehrere Teilprobleme verwalten
- effiziente Heuristiken für zulässige Lösungen

etc.

Es gibt Software-Pakete, die alle Verwaltungsaspekte übernehmen und einem den Einbau weiterer Schnittebenen und Heuristiken ermöglichen.

z.B.: SCIP, Cplex, Gurobi, Abacus ...

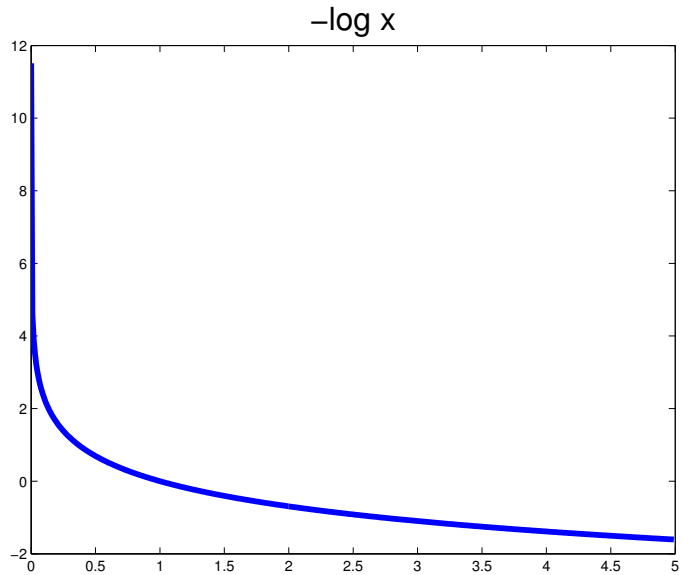
Inhaltsübersicht

Inhaltsübersicht

4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).



4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

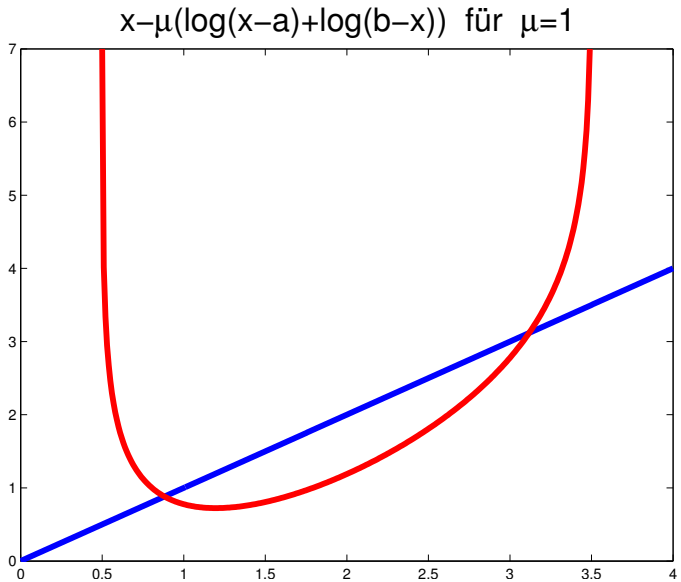
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

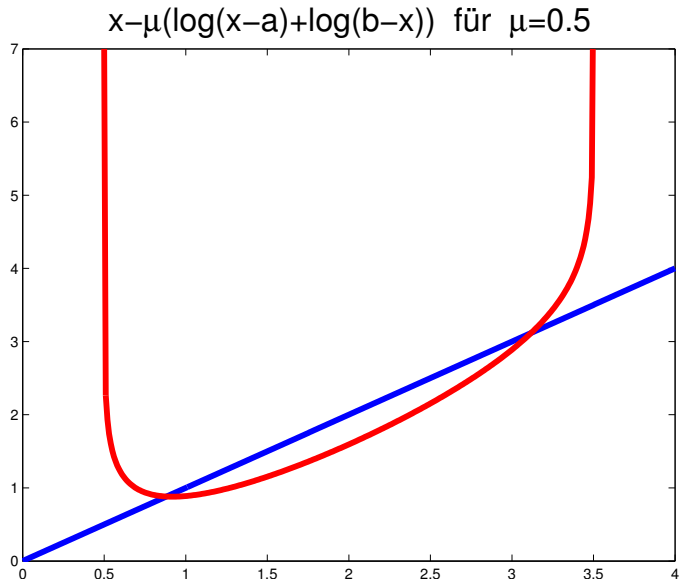
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

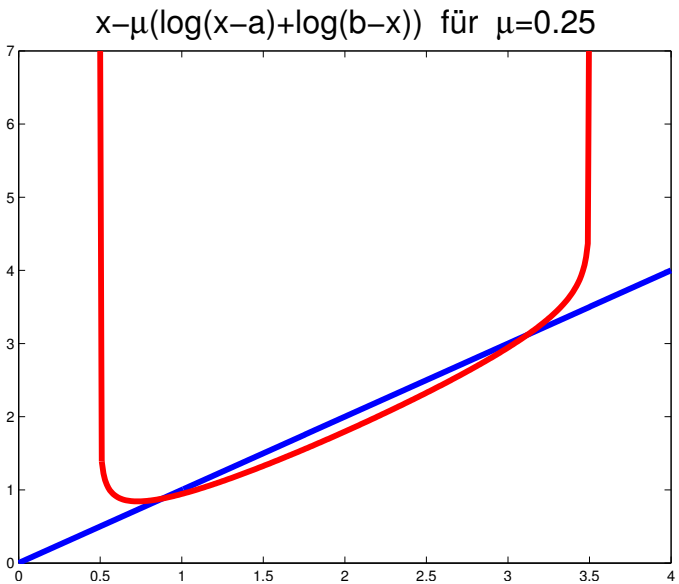
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

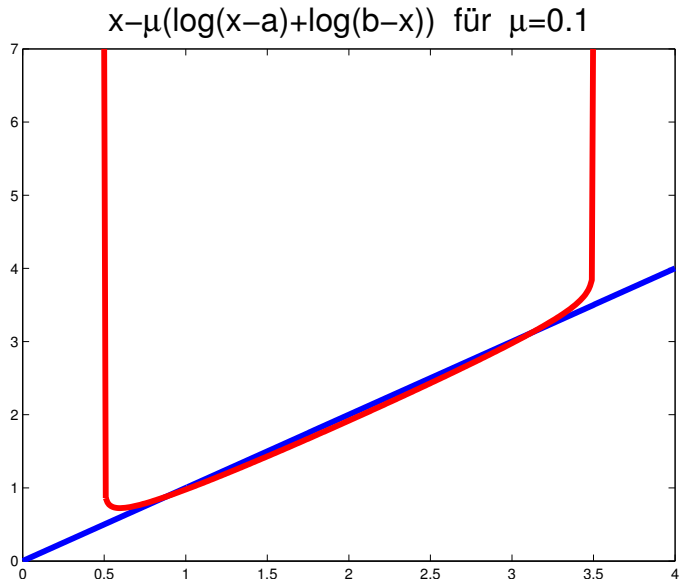
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

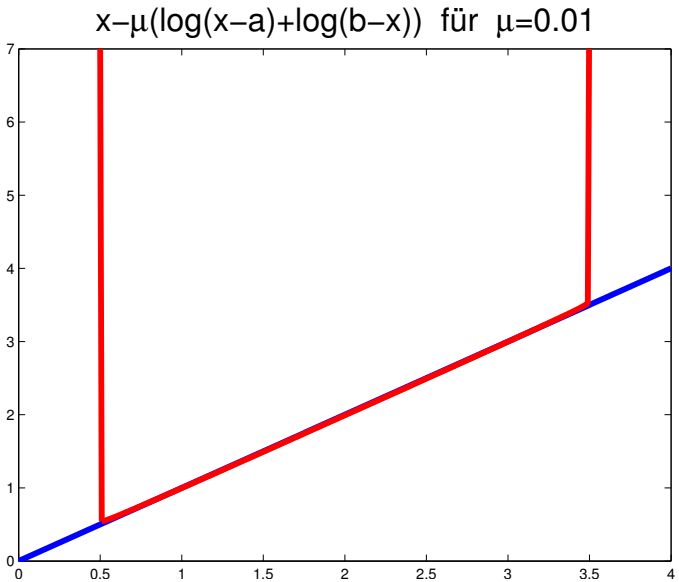
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

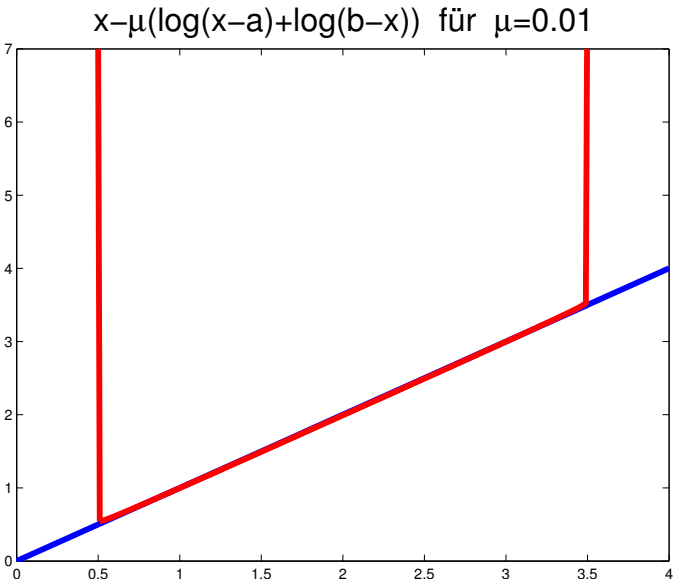
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



Primales/duales Problem: [für $x \in \mathbb{R}^n$: $\log x = \sum_{i=1}^{n} \log x_i$]

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & z \geq 0 \end{array}$$

4.1 Innere-Punkte-Verfahren für Lineare Optimierung

Nachweislich polynomiales Verfahren! [Simplex *worst case*: exponentiell]

Nutzt Idee der Barriere-Verfahren, um durch das Innere des Polyeders zu gehen: Ein mit einem Barriere-Parameter $\mu > 0$ kontrollierter Barriere-Term der Zielfunktion vehindert das Verlassen des Inneren.
Eine günstige Barriere für $x \geq 0$ ist $-\log x$ (streng konvex).

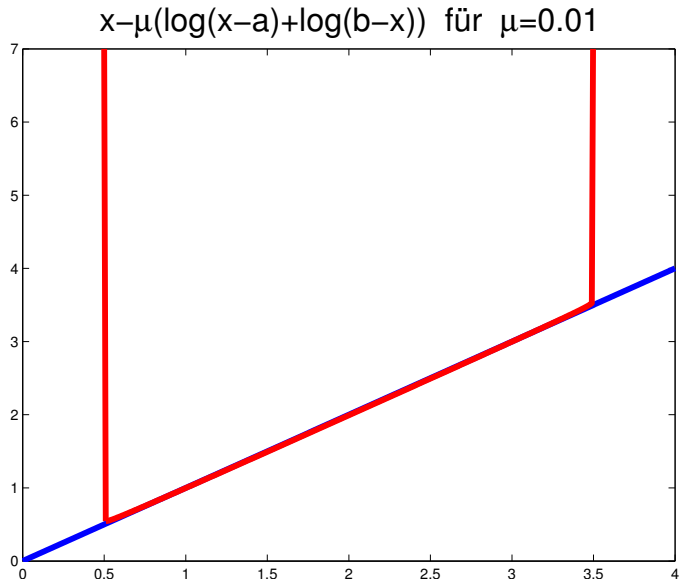
Bsp: min x s.t. $x \in [a = 0.5, b = 3.5]$

Barriere für $x \geq a$: $-\log(x - a)$
Barriere für $x \leq b$: $-\log(b - x)$

Barriereproblem:

$\min f(x) := x - \mu(\log(x - a) + \log(b - x))$

Starte mit $x \in (a, b)$, suche Minimum, verkleinere μ, suche Minimum, etc.



Primales/duales Barriere-Problem: [für $x \in \mathbb{R}^n$: $\log x = \sum_{i=1}^{n} \log x_i$]

$$(P_\mu) \quad \begin{array}{ll} \min & c^T x - \mu \log x \\ \text{s.t.} & Ax = b \\ & [x > 0] \end{array}$$

$$(D_\mu) \quad \begin{array}{ll} \max & b^T y + \mu \log z \\ \text{s.t.} & A^T y + z = c \\ & [z > 0] \end{array}$$

Das Primal-Duale KKT-System

Der Name KKT bezieht sich auf die Optimalitätsbedingungen für nichtlineare restringierte Optimierung (näheres dazu später).

$$(P_\mu)\quad \begin{array}{ll} \min & c^T x - \mu \log x \\ \text{s.t.} & Ax = b \\ & [x > 0] \end{array} \qquad (D_\mu)\quad \begin{array}{ll} \max & b^T y + \mu \log z \\ \text{s.t.} & A^T y + z = c \\ & [z > 0] \end{array}$$

Für die Lagrange-Funktion zum primalen Barriere-Problem (P_μ)

$$L_\mu(x, y) := c^T x + y^T (b - Ax) - \mu \log x$$

sucht man die Nullstellen der Ableitung nach x und y

$$\begin{array}{lll} \nabla_x L_\mu = 0 & \rightarrow & c - A^T y - \mu x^{-1} = 0 \qquad [x^{-1} := [x_1^{-1}, \ldots, x_n^{-1}]^T] \\ \nabla_y L_\mu = 0 & \rightarrow & b - Ax = 0 \end{array}$$

Das Primal-Duale KKT-System

Der Name KKT bezieht sich auf die Optimalitätsbedingungen für nichtlineare restringierte Optimierung (näheres dazu später).

$$(P_\mu)\quad \begin{array}{ll} \min & c^T x - \mu \log x \\ \text{s.t.} & Ax = b \\ & [x > 0] \end{array} \qquad (D_\mu)\quad \begin{array}{ll} \max & b^T y + \mu \log z \\ \text{s.t.} & A^T y + z = c \\ & [z > 0] \end{array}$$

Für die Lagrange-Funktion zum primalen Barriere-Problem (P_μ)

$$L_\mu(x, y) := c^T x + y^T (b - Ax) - \mu \log x$$

sucht man die Nullstellen der Ableitung nach x und y

$$\nabla_x L_\mu = 0 \;\rightarrow\; c - A^T y - \mu x^{-1} = 0 \qquad [x^{-1} := [x_1^{-1}, \ldots, x_n^{-1}]^T]$$
$$\nabla_y L_\mu = 0 \;\rightarrow\; b - Ax = 0$$

Setzt man $z = \mu x^{-1}$ erhält man das Primal-Duale KKT-System

$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array}} \qquad \begin{array}{ll} \text{duale Zulässigkeit} & [z > 0] \\ \text{primale Zulässigkeit} & [x > 0] \\ \text{perturbierte Komplementarität} & [x \circ z := [x_i z_i]] \end{array}$$

Für $\mu = 0$: $x \circ z = 0 \Leftrightarrow x^T z = 0$, mit $x, z \in \mathbb{R}^n_+$ Opt.-Bed. für (P) und (D)!

Das Primal-Duale KKT-System

Der Name KKT bezieht sich auf die Optimalitätsbedingungen für nichtlineare restringierte Optimierung (näheres dazu später).

$$(P_\mu)\quad \begin{array}{ll}\min & c^T x - \mu \log x \\ \text{s.t.} & Ax = b \\ & [x > 0]\end{array} \qquad (D_\mu)\quad \begin{array}{ll}\max & b^T y + \mu \log z \\ \text{s.t.} & A^T y + z = c \\ & [z > 0]\end{array}$$

Für die Lagrange-Funktion zum primalen Barriere-Problem (P_μ)

$$L_\mu(x, y) := c^T x + y^T(b - Ax) - \mu \log x$$

sucht man die Nullstellen der Ableitung nach x und y

$$\nabla_x L_\mu = 0 \;\rightarrow\; c - A^T y - \mu x^{-1} = 0 \qquad [x^{-1} := [x_1^{-1}, \ldots, x_n^{-1}]^T]$$
$$\nabla_y L_\mu = 0 \;\rightarrow\; b - Ax = 0$$

Setzt man $z = \mu x^{-1}$ erhält man das Primal-Duale KKT-System

$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1}\end{array}} \qquad \begin{array}{ll} \text{duale Zulässigkeit} & [z > 0] \\ \text{primale Zulässigkeit} & [x > 0] \\ \text{perturbierte Komplementarität} & [x \circ z := [x_i z_i]]\end{array}$$

Für $\mu = 0$: $x \circ z = 0 \Leftrightarrow x^T z = 0$, mit $x, z \in \mathbb{R}^n_+$ Opt.-Bed. für (P) und (D)!
Sind (P) und (D) streng zulässig (und hat A vollen Zeilenrang), gibt es für jedes $\mu > 0$ eine eindeutige Lösung $(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$, es gilt

x_μ ist OL von (P_μ) und (y_μ, z_μ) ist OL von (D_μ).

Der zentrale Pfad, strenge Komplementarität

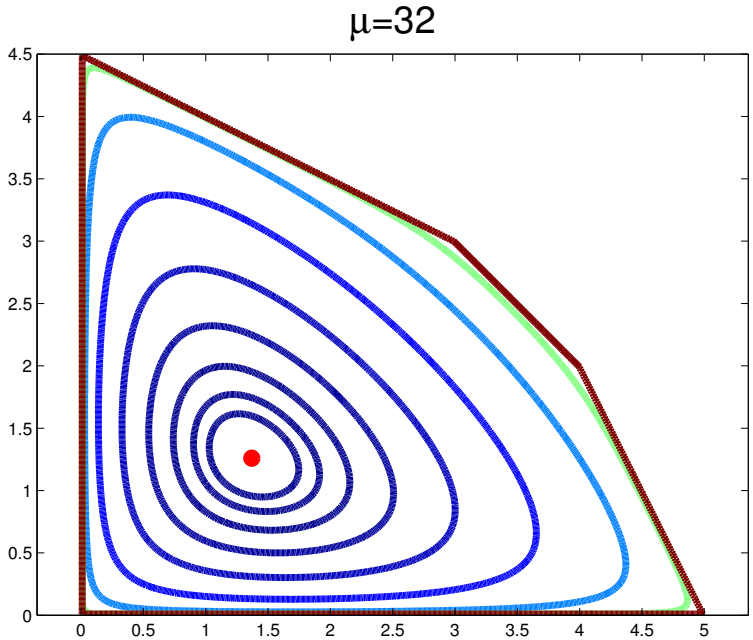
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

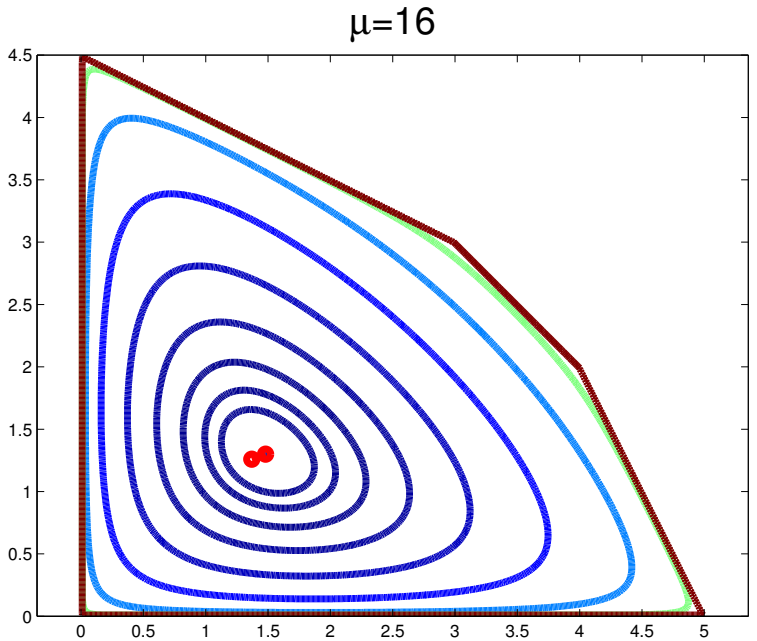
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

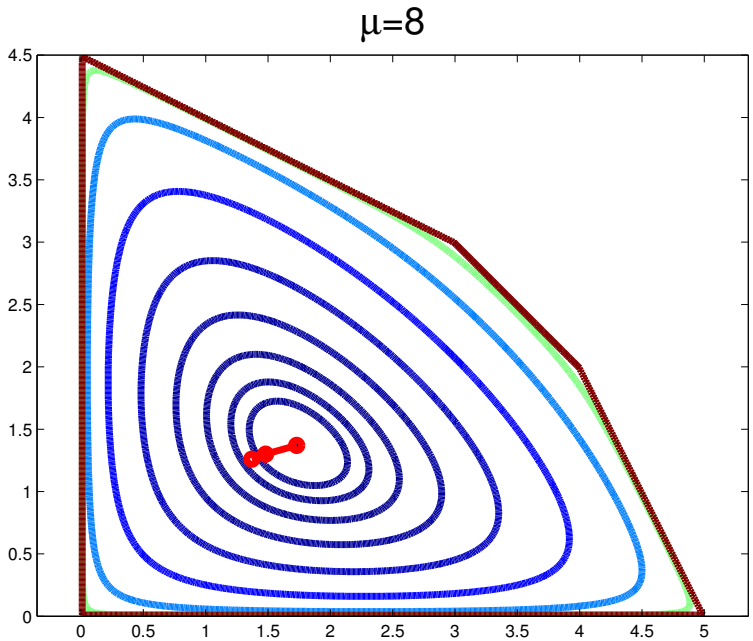
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

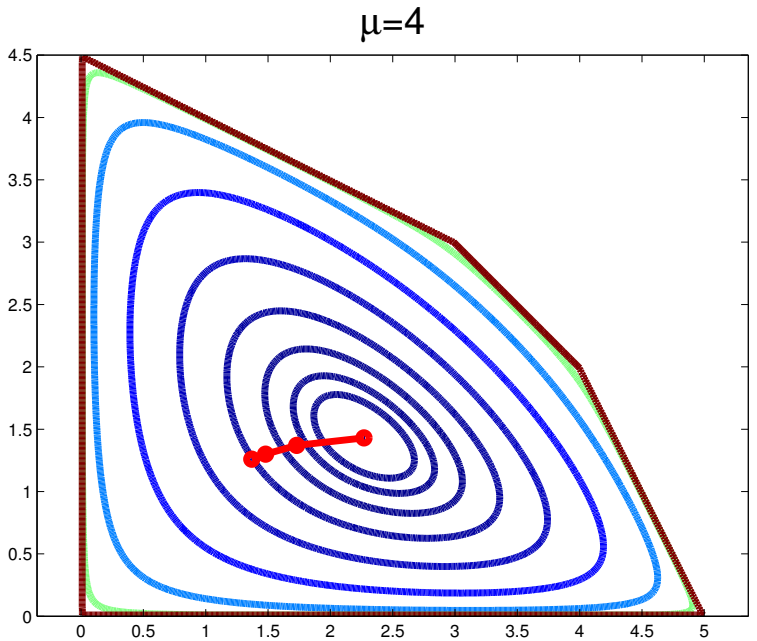
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul., } \mathrm{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

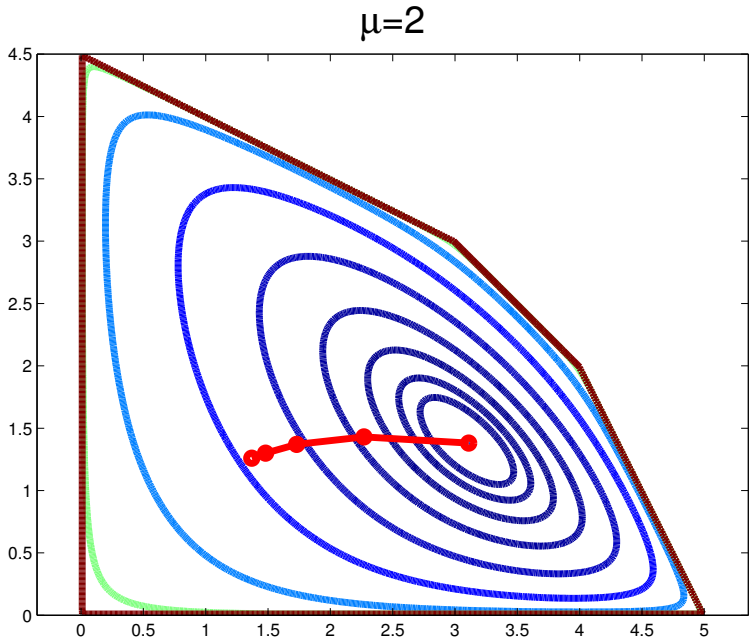
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

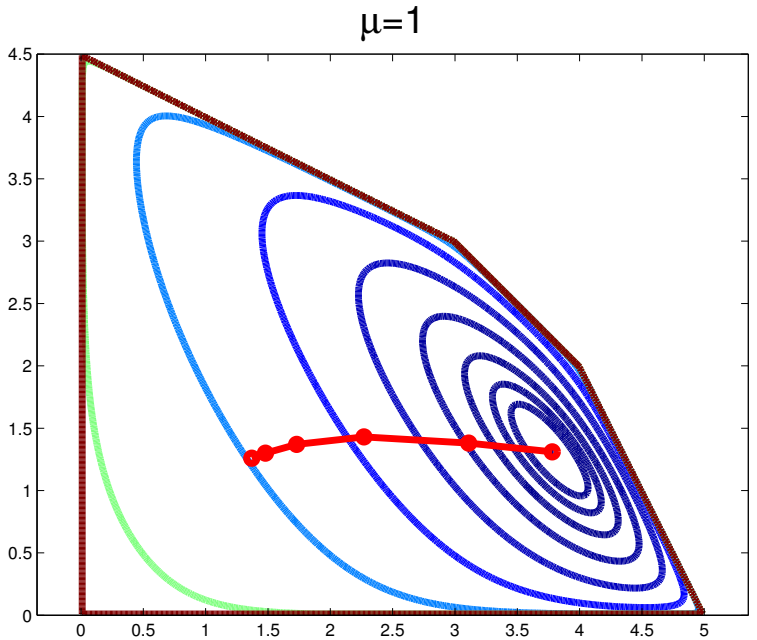
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul., } \mathrm{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

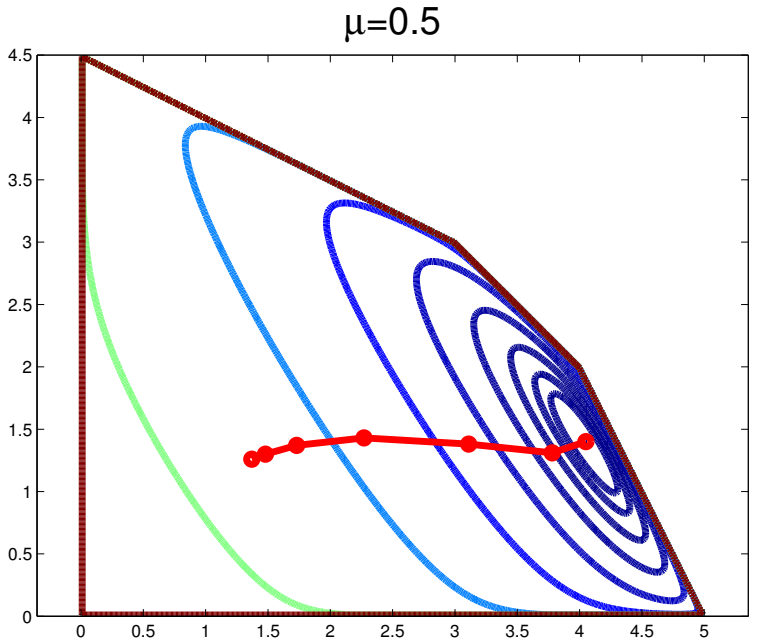
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul., } \mathrm{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

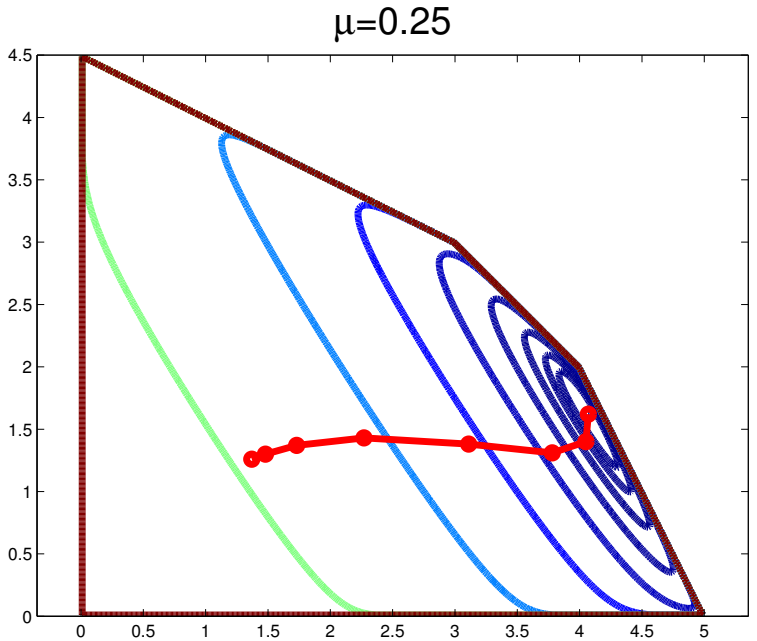
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

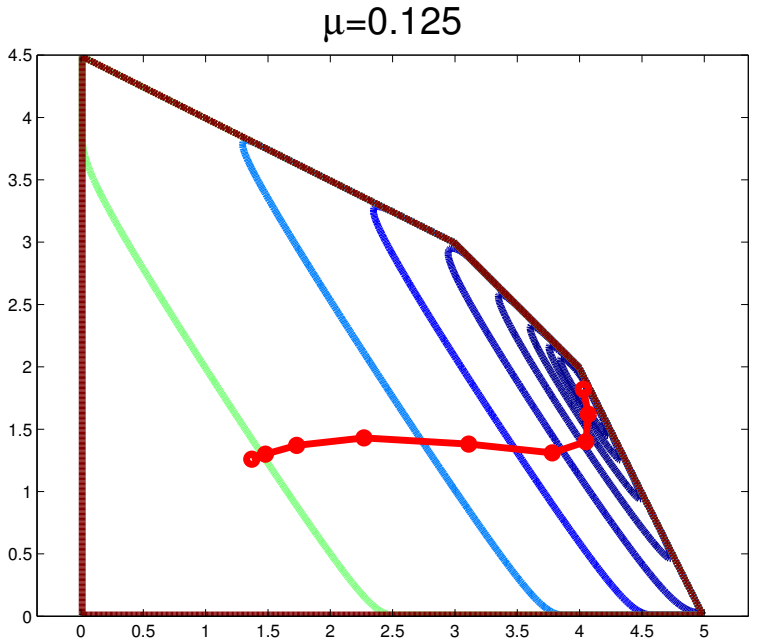
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

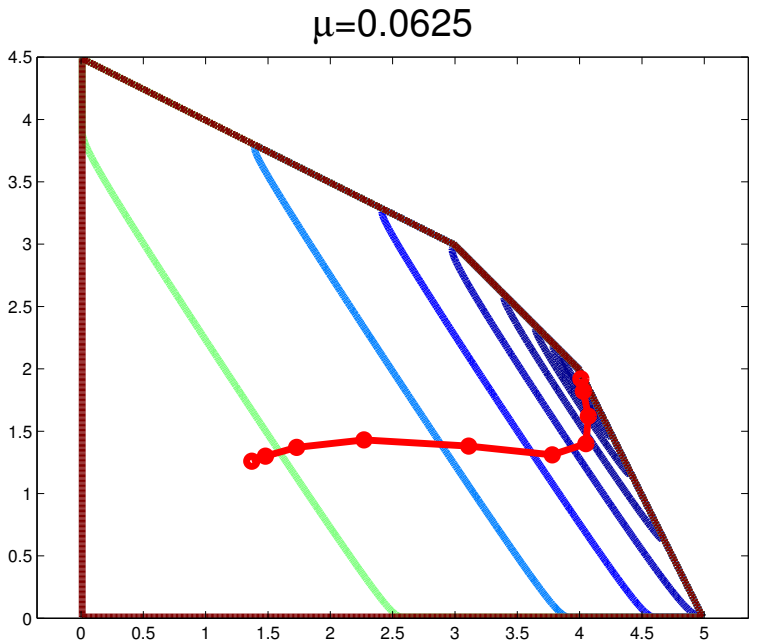
$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul.}, \operatorname{Rang}(A) = m]$$

Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

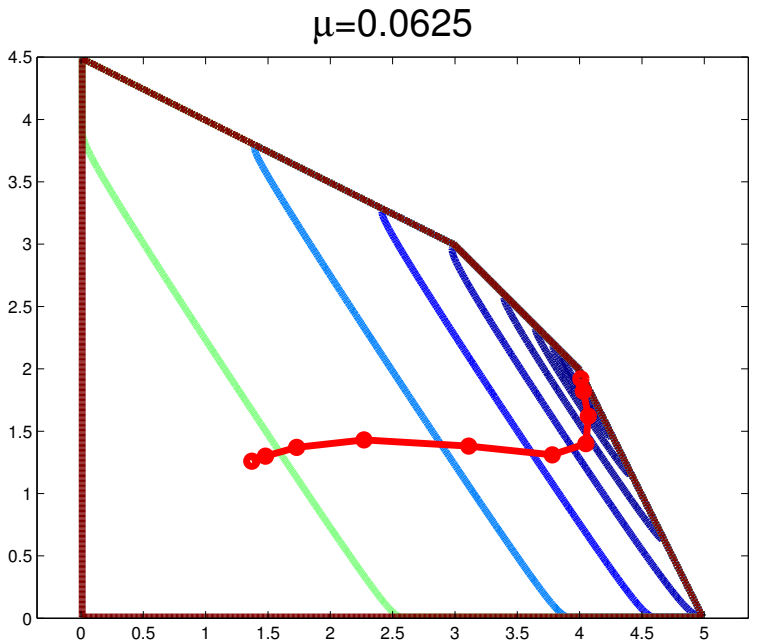
$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).



Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu)\quad \boxed{\begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array}} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul., } \mathrm{Rang}(A) = m]$$



Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$

$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+$ löst (PD_μ).

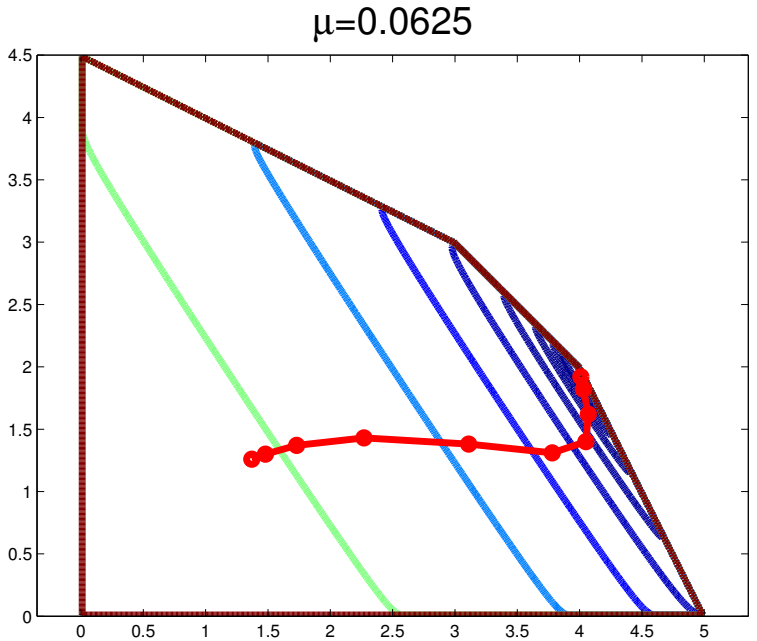
Er liegt innerhalb des primalen/dualen Polyeders und konvergiert für $\mu \to 0$ gegen das analytische Zentrum der Opt.-Lösungen

$$(x^*, y^*, z^*) := \lim_{\mu \to 0} (x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+.$$

Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu)\quad \begin{array}{rcl} A^T y + z &=& c \\ Ax &=& b \\ x \circ z &=& \mu \mathbf{1} \end{array} \qquad \begin{array}{l} [z > 0] \\ [x > 0] \end{array}$$

$$[(P), (D) \text{ str. zul., } \mathrm{Rang}(A) = m]$$



Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$
$$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+ \text{ löst } (PD_\mu).$$

Er liegt innerhalb des primalen/dualen Polyeders und konvergiert für $\mu \to 0$ gegen das analytische Zentrum der Opt.-Lösungen

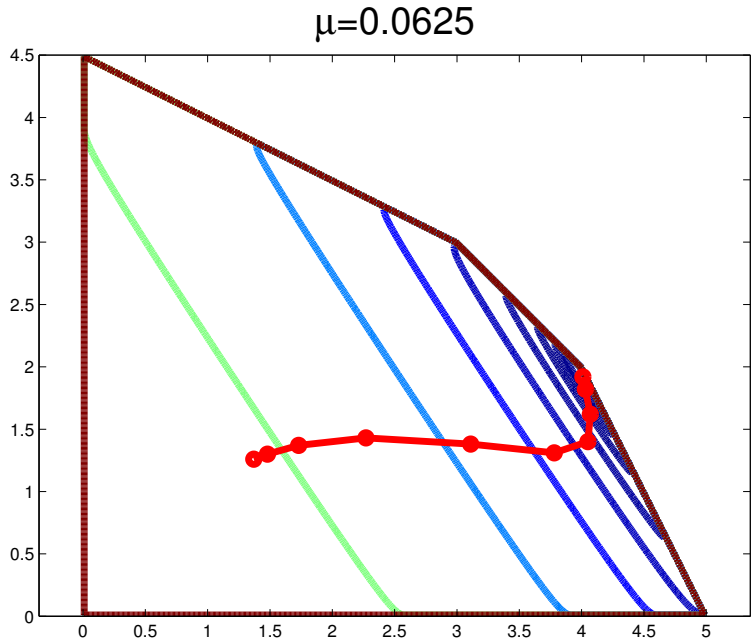
$$(x^*, y^*, z^*) := \lim_{\mu \to 0} (x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+.$$

x^* und (y^*, z^*) sind OL von (P) und (D) und streng komplementär, d.h., sie erfüllen $x_i z_i = 0$ mit entweder $x_i > 0$ oder $z_i > 0$.

Der zentrale Pfad, strenge Komplementarität

$$(PD_\mu)\quad \begin{array}{rcll} A^T y + z & = & c & [z > 0] \\ Ax & = & b & [x > 0] \\ x \circ z & = & \mu \mathbf{1} & \end{array}$$

$$[(P), (D) \text{ str. zul., } \mathrm{Rang}(A) = m]$$



Der zentrale Pfad ist die glatte Kurve

$$\mathcal{Z}(\mu) = (x_\mu, y_\mu, z_\mu), \quad \mu > 0,$$
$$(x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+ \text{ löst } (PD_\mu).$$

Er liegt innerhalb des primalen/dualen Polyeders und konvergiert für $\mu \to 0$ gegen das analytische Zentrum der Opt.-Lösungen

$$(x^*, y^*, z^*) := \lim_{\mu \to 0} (x_\mu, y_\mu, z_\mu) \in \mathbb{R}^n_+ \times \mathbb{R}^m \times \mathbb{R}^n_+.$$

x^* und (y^*, z^*) sind OL von (P) und (D) und streng komplementär, d.h., sie erfüllen $x_i z_i = 0$ mit entweder $x_i > 0$ oder $z_i > 0$.

Nur falls die Optimallösung von (P) bzw. (D) eindeutig ist, ist x^* bzw. (y^*, z^*) auch eine Ecke des zulässigen Polyeders von (P) bzw. (D)!

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \longrightarrow \quad \begin{array}{ll} \text{I.} & A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\ \text{II.} & A(\bar{x} + \Delta x) = b \\ \text{III.} & \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1} \end{array}$$

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \longrightarrow \quad \begin{array}{ll} \text{I.} & A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\ \text{II.} & A(\bar{x} + \Delta x) = b \\ \text{III.} & \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1} \end{array}$$

---

1. aus I.: $\Delta z = c - A^T \bar{y} - \bar{z} - A^T \Delta y$
2. aus III.: $\Delta x = \mu \bar{z}^{-1} - \bar{x} - \bar{x} \circ \bar{z}^{-1} \circ \Delta z \overset{(1.)}{=} \bar{x} \circ \bar{z}^{-1} \circ A^T \Delta y + \ldots$
3. in II.: $A \operatorname{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \, \Delta y = b - A\bar{x} + \ldots$

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \longrightarrow \quad \begin{array}{ll} \text{I.} & A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\ \text{II.} & A(\bar{x} + \Delta x) = b \\ \text{III.} & \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1} \end{array}$$

---

1. aus I.: $\Delta z = c - A^T \bar{y} - \bar{z} - A^T \Delta y$
2. aus III.: $\Delta x = \mu \bar{z}^{-1} - \bar{x} - \bar{x} \circ \bar{z}^{-1} \circ \Delta z \overset{(1.)}{=} \bar{x} \circ \bar{z}^{-1} \circ A^T \Delta y + \ldots$
3. in II.: $A \operatorname{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \, \Delta y = b - A\bar{x} + \ldots$

Wegen $x > 0, z > 0$ ist die Matrix $M := A \operatorname{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \in \mathbb{R}^{m \times m}$ symmetrisch positiv semidefinit und positiv definit, falls $\operatorname{Rang}(A) = m$.

$\Rightarrow$ Das Gleichungssystem $M \Delta y = \ldots$ kann in etwa $m^3/3$ arithmetischen Operationen mit dem Cholesky-Vefahren gelöst werden.
Dieses „faktorisiert“ $M = LL^T$ mit L untere Dreiecksmatrix.

Lösung des Primal-Dualen Systems

$\mathcal{Z}(\mu)$ nähert man iterativ mit dem Newton-Verfahren für nichtlineare Gleichungen an (s. später): Bestimme für gegebenes $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$ den nächsten Punkt $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ aus der Linearisierung:

$$\begin{array}{rcl} A^T y + z & = & c \\ Ax & = & b \\ x \circ z & = & \mu \mathbf{1} \end{array} \quad \longrightarrow \quad \begin{array}{ll} \text{I.} & A^T(\bar{y} + \Delta y) + \bar{z} + \Delta z = c \\ \text{II.} & A(\bar{x} + \Delta x) = b \\ \text{III.} & \bar{x} \circ \bar{z} + \Delta x \circ \bar{z} + \bar{x} \circ \Delta z = \mu \mathbf{1} \end{array}$$

---

1. aus I.: $\Delta z = c - A^T \bar{y} - \bar{z} - A^T \Delta y$
2. aus III.: $\Delta x = \mu \bar{z}^{-1} - \bar{x} - \bar{x} \circ \bar{z}^{-1} \circ \Delta z \overset{(1.)}{=} \bar{x} \circ \bar{z}^{-1} \circ A^T \Delta y + \ldots$
3. in II.: $A \operatorname{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \, \Delta y = b - A\bar{x} + \ldots$

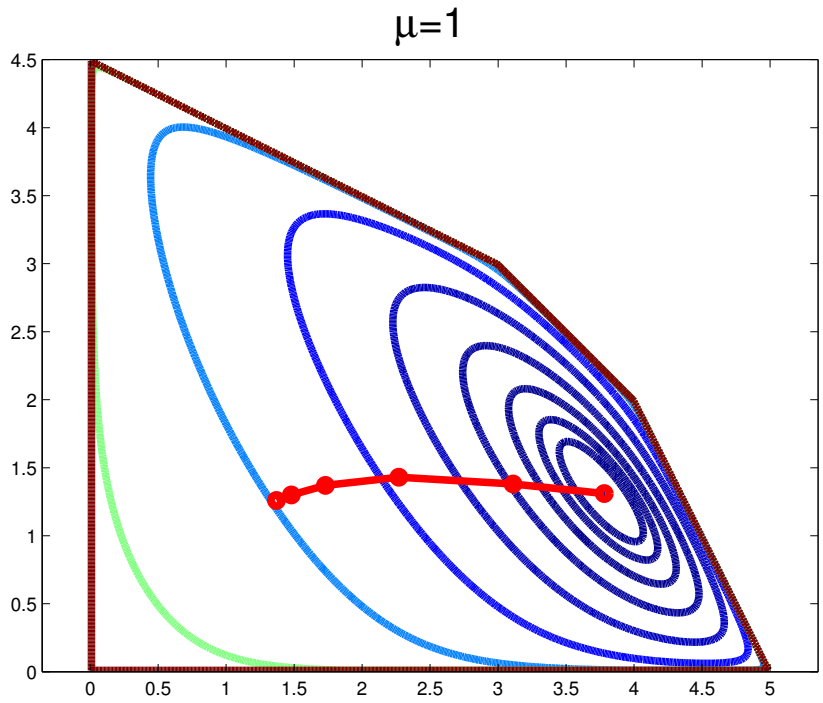
Wegen $x > 0, z > 0$ ist die Matrix $M := A \operatorname{Diag}(\bar{x} \circ \bar{z}^{-1}) A^T \in \mathbb{R}^{m \times m}$ symmetrisch positiv semidefinit und positiv definit, falls $\operatorname{Rang}(A) = m$.

$\Rightarrow$ Das Gleichungssystem $M \Delta y = \ldots$ kann in etwa $m^3/3$ arithmetischen Operationen mit dem Cholesky-Vefahren gelöst werden.
Dieses „faktorisiert“ $M = LL^T$ mit L untere Dreiecksmatrix.

Wichtiger Unterschied zum Simplex für die Praxis:
A ist meist nur sehr dünn besetzt, für Simplex in $A_B x_B = b$ gut nutzbar!
Trotz vieler Tricks ist L meist sehr viel dichter besetzt als A

$\rightarrow$ braucht viel mehr Speicher, jede Iteration sehr langsam, dafür wenige!

Grobe Näherung für $\mathcal{Z}(\mu)$ genügt



Für $\bar{\mu} > \mu > 0$ mit $\bar{\mu} - \mu$ klein, ist $\|\mathcal{Z}(\bar{\mu}) - \mathcal{Z}(\mu)\|$ klein. Ist $(\bar{x}, \bar{y}, \bar{z})$ gute Näherung für $\mathcal{Z}(\bar{\mu})$ mit $\bar{\mu} = \bar{x}^T\bar{z}/n$ und $\mu = \sigma\bar{\mu}$ mit $\sigma < 1$ nahe bei 1, dann genügt ein Newton-Schritt, damit $(\bar{x} + \Delta x, \bar{y} + \Delta y, \bar{z} + \Delta z)$ eine gute Näherung für $\mathcal{Z}(\sigma\bar{\mu})$ ist mit $\sigma\bar{\mu} = (\bar{x} + \Delta x)^T(\bar{z} + \Delta z)/n$.

Algorithmisches Schema für Innere-Punkte-Verfahren

Idee: Folge dem zentralen Pfad $\mathcal{Z}$ mit Newton-Schritten für $\mu \longrightarrow 0$.

Input: A, b, c, $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$, $\varepsilon > 0$, $\sigma \in (0,1)$

1. Setze $\mu \leftarrow \sigma \frac{\bar{x}^T \bar{z}}{n}$.
2. Bestimme $(\Delta x, \Delta y, \Delta z)$ als Newton-Schritt für (PD_μ).
3. Bestimme großes $\alpha \in (0,1]$ mit $\bar{x} + \alpha \Delta x > 0$, $\bar{z} + \alpha \Delta z > 0$
4. setze $(\bar{x}, \bar{y}, \bar{z}) \leftarrow (\bar{x}, \bar{y}, \bar{z}) + \alpha(\Delta x, \Delta y, \Delta z)$
5. Ist $\|b - A\bar{x}\| < \varepsilon$ und $\|c - A^T \bar{y} - \bar{z}\| < \varepsilon$ und $\bar{x}^T \bar{z} < \varepsilon$ STOP, sonst gehe zu 1.

Algorithmisches Schema für Innere-Punkte-Verfahren

Idee: Folge dem zentralen Pfad $\mathcal{Z}$ mit Newton-Schritten für $\mu \longrightarrow 0$.

Input: A, b, c, $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$, $\varepsilon > 0$, $\sigma \in (0,1)$

1. Setze $\mu \leftarrow \sigma \frac{\bar{x}^T \bar{z}}{n}$.
2. Bestimme $(\Delta x, \Delta y, \Delta z)$ als Newton-Schritt für (PD_μ).
3. Bestimme großes $\alpha \in (0,1]$ mit $\bar{x} + \alpha \Delta x > 0$, $\bar{z} + \alpha \Delta z > 0$
4. setze $(\bar{x}, \bar{y}, \bar{z}) \leftarrow (\bar{x}, \bar{y}, \bar{z}) + \alpha(\Delta x, \Delta y, \Delta z)$
5. Ist $\|b - A\bar{x}\| < \varepsilon$ und $\|c - A^T \bar{y} - \bar{z}\| < \varepsilon$ und $\bar{x}^T \bar{z} < \varepsilon$ STOP, sonst gehe zu 1.

Sind (P) und (D) zulässig und $\mathrm{Rang}(A) = m$, kann man für ein leicht umformuliertes Problem einen zulässigen Startpunkt $\bar{x}, \bar{y}, \bar{z}$ und ein $\sigma < 1$ angeben, sodass der Algorithmus nach höchstens $\gamma \sqrt{n \log \frac{\bar{x}^T \bar{z}}{\varepsilon}}$ Iterationen (für eine problemunabhängige Konstante $\gamma > 0$) endet.

Algorithmisches Schema für Innere-Punkte-Verfahren

Idee: Folge dem zentralen Pfad $\mathcal{Z}$ mit Newton-Schritten für $\mu \to 0$.

Input: A, b, c, $\bar{x} > 0$, $\bar{y}$, $\bar{z} > 0$, $\varepsilon > 0$, $\sigma \in (0,1)$

1. Setze $\mu \leftarrow \sigma \frac{\bar{x}^T \bar{z}}{n}$.
2. Bestimme $(\Delta x, \Delta y, \Delta z)$ als Newton-Schritt für (PD_μ).
3. Bestimme großes $\alpha \in (0,1]$ mit $\bar{x} + \alpha \Delta x > 0$, $\bar{z} + \alpha \Delta z > 0$
4. setze $(\bar{x}, \bar{y}, \bar{z}) \leftarrow (\bar{x}, \bar{y}, \bar{z}) + \alpha(\Delta x, \Delta y, \Delta z)$
5. Ist $\|b - A\bar{x}\| < \varepsilon$ und $\|c - A^T \bar{y} - \bar{z}\| < \varepsilon$ und $\bar{x}^T \bar{z} < \varepsilon$ STOP, sonst gehe zu 1.

Sind (P) und (D) zulässig und $\mathrm{Rang}(A) = m$, kann man für ein leicht umformuliertes Problem einen zulässigen Startpunkt $\bar{x}, \bar{y}, \bar{z}$ und ein $\sigma < 1$ angeben, sodass der Algorithmus nach höchstens $\gamma \sqrt{n \log \frac{\bar{x}^T \bar{z}}{\varepsilon}}$ Iterationen (für eine problemunabhängige Konstante $\gamma > 0$) endet.

- Das Ergebnis ist eine Näherung und keine exakte Lösung!
- Für ε klein genug kann für diese in einem „cross-over“-Schritt eine optimale Basis bestimmt werden (benötigt oft viele Simplex-Iterationen).
- Erkennen von Unzulässigkeit/Unbeschränktheit ist deutlich schwerer.

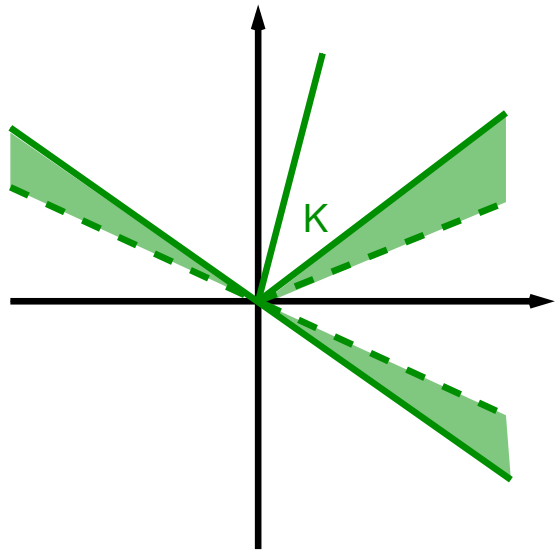
Zusammenfassung der Eigenschaften

- Innere-Punkte-Verfahren haben polynomiale Laufzeit.
- Für große Probleme sind sie oft schneller als Simplex, Degeneriertheit der Polyeder ist kaum ein Problem.
- Erkennen von Unzulässigkeit/Unbeschränktheit ist schwerer.
- Da Schnittebenen den zentralen Pfad großräumig verändern, ist kein guter Warmstart von der alten OL aus möglich.
  $\longrightarrow$ erst cross-over, dann dualer Simplex
- Der gleiche Ansatz funktioniert sehr gut für lineare Programme über gutartigen nichtlinearen Kegeln!

Inhaltsübersicht

4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

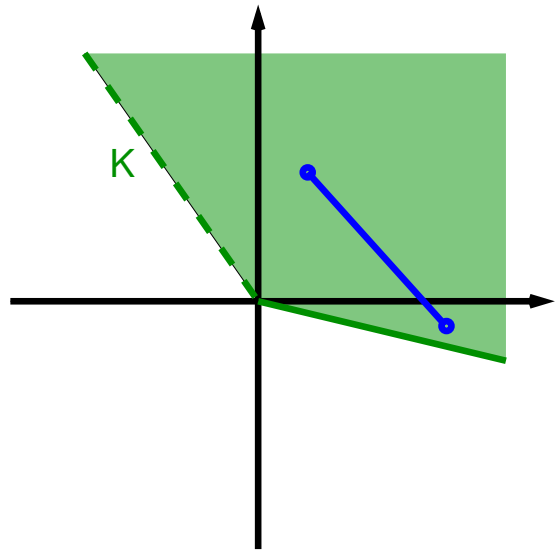
Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.



4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

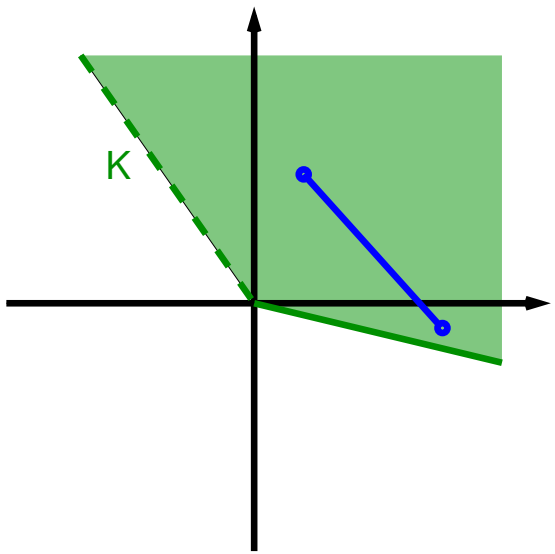


4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

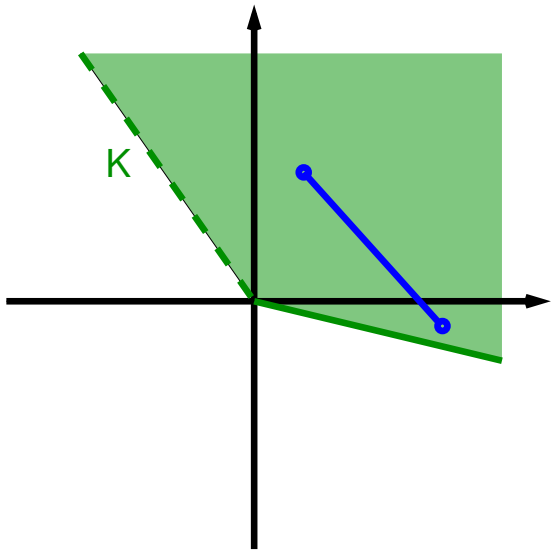
Bspe: $\emptyset$, $\{0\}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.



4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.

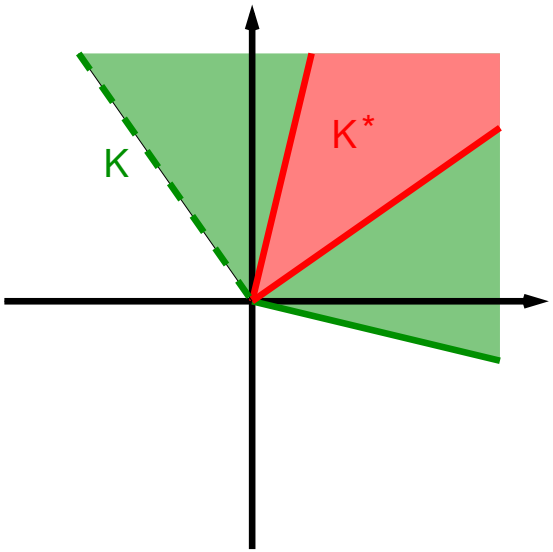


Bspe: $\emptyset$, $\{0\}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.



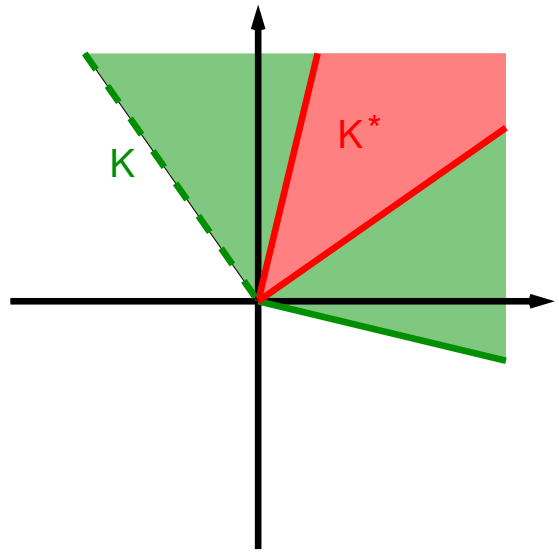
Bspe: $\emptyset$, $\{0\}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegeln, nennt man
$K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \ \forall x \in K\}$ den zu K dualen Kegel.

4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x+y) \in K$ für $\alpha \geq 0$.



Bspe: $\emptyset$, $\{0\}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

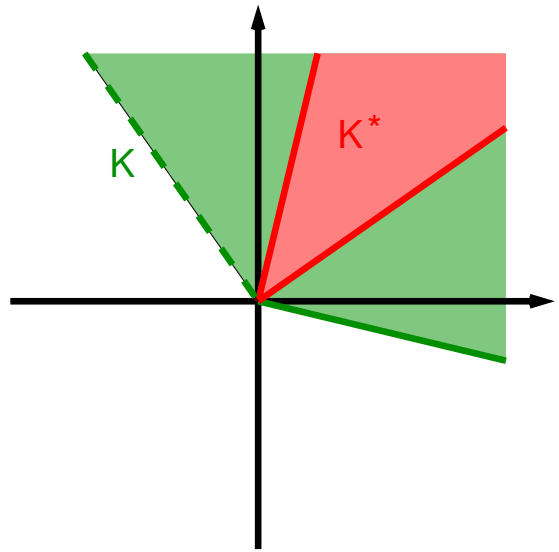
Ist $K \subseteq \mathbb{R}^n$ konvexer Kegeln, nennt man
$K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \ \forall x \in K\}$ den zu K dualen Kegel.

Bspe: $(\mathbb{R}^n_+)^* = \mathbb{R}^n_+$, $\{0\}^* = \mathbb{R}^n$, $(\mathbb{R}^n)^* = \{0\}$, $(K_1 \times K_2)^* = K_1^* \times K_2^*$.

4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.



Bspe: $\emptyset$, $\{0\}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegeln, nennt man
$K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \ \forall x \in K\}$ den zu K dualen Kegel.

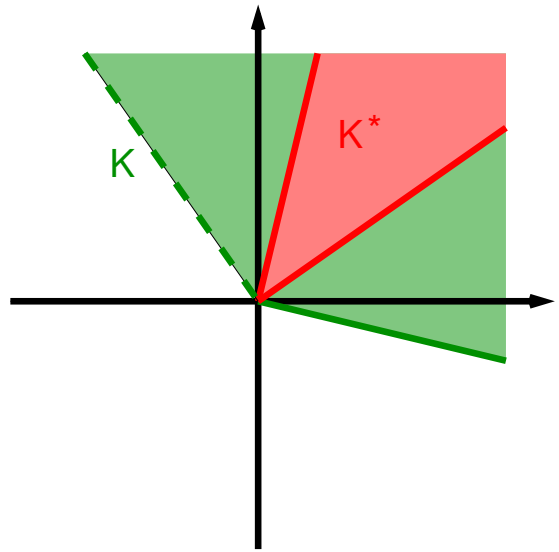
Bspe: $(\mathbb{R}^n_+)^* = \mathbb{R}^n_+$, $\{0\}^* = \mathbb{R}^n$, $(\mathbb{R}^n)^* = \{0\}$, $(K_1 \times K_2)^* = K_1^* \times K_2^*$.
Für die Optimierung wichtig ist die Eigenschaft

$$\inf_{x \in K} z^T x = \begin{cases} 0 & \Leftrightarrow z \in K^*, \\ -\infty & \textit{sonst}. \end{cases}$$

4.2 Lineare Optimierung über Kegeln: Konvexe Kegel, dualer Kegel

Eine Menge $K \subseteq \mathbb{R}^n$ heißt Kegel,
wenn mit $x \in K$ auch $\alpha x \in K$ für $\alpha \geq 0$.

Eine Menge $K \subseteq \mathbb{R}^n$ heißt konvexer Kegel,
wenn mit $x, y \in K$ auch $\alpha(x + y) \in K$ für $\alpha \geq 0$.



Bspe: $\emptyset$, $\{0\}$, $\mathbb{R}^n$, $\mathbb{R}^n_+$,
lineare Unterräume $\{x \in \mathbb{R}^n : Ax = 0\}$.
Sind $K_1 \subseteq \mathbb{R}^n$, $K_2 \subseteq \mathbb{R}^m$ konvexe Kegel,
so auch $K_1 \times K_2 = \left\{ \begin{bmatrix} x \\ y \end{bmatrix} : x \in K_1, y \in K_2 \right\}$.

Ist $K \subseteq \mathbb{R}^n$ konvexer Kegeln, nennt man
$K^* := \{z \in \mathbb{R}^n : z^T x \geq 0 \ \forall x \in K\}$ den zu K dualen Kegel.

Bspe: $(\mathbb{R}^n_+)^* = \mathbb{R}^n_+$, $\{0\}^* = \mathbb{R}^n$, $(\mathbb{R}^n)^* = \{0\}$, $(K_1 \times K_2)^* = K_1^* \times K_2^*$.
Für die Optimierung wichtig ist die Eigenschaft

$$\inf_{x \in K} z^T x = \begin{cases} 0 & \Leftrightarrow z \in K^*, \\ -\infty & \textit{sonst}. \end{cases}$$

Vorsicht! $(K^*)^* = K$ gilt NUR, wenn K ein abgeschlossener konvexer Kegel ist!

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung $\mathbb{R}^n_+$ durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \geq 0 & \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in \mathbb{R}^n_+ & \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in K & \end{array}$$

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung $\mathbb{R}^n_+$ durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \geq 0 & \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in \mathbb{R}^n_+ & \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in K & \end{array}$$

Definiere die Lagrange-Funktion

$$L(x, y) := c^T x + y^T (b - Ax) \quad \text{für } (x, y) \in K \times \mathbb{R}^m.$$

Für $y \in \mathbb{R}^m$ und $Ax = b$ ist $(b - Ax)^T y = 0$, daher gilt

$$\text{für alle } y \in \mathbb{R}^m : \quad \inf_{x \in K} L(x, y) \leq \inf\{c^T x : Ax = b, x \in K\}.$$

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung $\mathbb{R}^n_+$ durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \geq 0 & \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in \mathbb{R}^n_+ & \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in K & \end{array}$$

Definiere die Lagrange-Funktion

$$L(x, y) := c^T x + y^T (b - Ax) \quad \text{für } (x, y) \in K \times \mathbb{R}^m.$$

Für $y \in \mathbb{R}^m$ und $Ax = b$ ist $(b - Ax)^T y = 0$, daher gilt

$$\text{für alle } y \in \mathbb{R}^m : \quad \inf_{x \in K} L(x, y) \leq \inf\{c^T x : Ax = b, x \in K\}.$$

Die beste untere Schranke (Lagrange-Relaxation) liefert

$$\sup_{y \in \mathbb{R}^m} \inf_{x \in K} L(x, y) = \sup_{y \in \mathbb{R}^m} [b^T y + \inf_{x \in K} x^T (c - A^T y)]$$

Primale/Duale lineare Programme über Kegeln

Ersetze in der linearen Optimierung $\mathbb{R}^n_+$ durch konvexen Kegel $K \subseteq \mathbb{R}^n$:

$$\begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \geq 0 & \end{array} \quad \Leftrightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in \mathbb{R}^n_+ & \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & c^T x & \\ \text{s.t.} & Ax = b & \\ & x \in K & \end{array}$$

Definiere die Lagrange-Funktion

$$L(x, y) := c^T x + y^T (b - Ax) \quad \text{für } (x, y) \in K \times \mathbb{R}^m.$$

Für $y \in \mathbb{R}^m$ und $Ax = b$ ist $(b - Ax)^T y = 0$, daher gilt

$$\text{für alle } y \in \mathbb{R}^m : \quad \inf_{x \in K} L(x, y) \leq \inf\{c^T x : Ax = b, x \in K\}.$$

Die beste untere Schranke (Lagrange-Relaxation) liefert

$$\sup_{y \in \mathbb{R}^m} \inf_{x \in K} L(x, y) = \sup_{y \in \mathbb{R}^m} [b^T y + \inf_{x \in K} x^T (c - A^T y)]$$

Das inf ist nur für $z = c - A^T y \in K^*$ endlich, das duale Programm lautet

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \in K^* \end{array}$$

Schwache und starke Dualität

Sei $K \subseteq \mathbb{R}^n$ ein konvexer Kegel.

$$
\begin{array}{llll}
 & \min & c^T x & \\
(P) & \text{s.t.} & Ax = b & \\
 & & x \in K &
\end{array}
\qquad
\begin{array}{llll}
 & \max & b^T y & \\
(D) & \text{s.t.} & A^T y + z = c & \\
 & & y \in \mathbb{R}^m, z \in K^* &
\end{array}
$$

Nach Konstruktion gilt immer schwache Dualität, $v(P) \geq v(D)$,
Gleichheit gilt keineswegs immer (Beispiel später)!

Für starke Dualität braucht es Zusatzbedingungen:

Schwache und starke Dualität

Sei $K \subseteq \mathbb{R}^n$ ein konvexer Kegel.

$$
\begin{array}{llll}
 & \min & c^T x & \\
(P) & \text{s.t.} & Ax = b & \\
 & & x \in K &
\end{array}
\qquad
\begin{array}{llll}
 & \max & b^T y & \\
(D) & \text{s.t.} & A^T y + z = c & \\
 & & y \in \mathbb{R}^m, z \in K^* &
\end{array}
$$

Nach Konstruktion gilt immer schwache Dualität, $v(P) \geq v(D)$, Gleichheit gilt keineswegs immer (Beispiel später)!

Für starke Dualität braucht es Zusatzbedingungen:
Ein primal zulässiges $\bar{x}$ heißt streng zulässig für (P), wenn $\bar{x}$ im Inneren von K liegt ($\exists \rho > 0 : B_\rho(\bar{x}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \rho\} \subseteq K$). Gibt es so ein $\bar{x}$, heißt auch (P) streng zulässig.

Schwache und starke Dualität

Sei $K \subseteq \mathbb{R}^n$ ein konvexer Kegel.

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \in K \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \in K^* \end{array}$$

Nach Konstruktion gilt immer schwache Dualität, $v(P) \geq v(D)$, Gleichheit gilt keineswegs immer (Beispiel später)!

Für starke Dualität braucht es Zusatzbedingungen:
Ein primal zulässiges $\bar{x}$ heißt streng zulässig für (P), wenn $\bar{x}$ im Inneren von K liegt ($\exists \rho > 0 : B_\rho(\bar{x}) := \{x \in \mathbb{R}^n : \|x - \bar{x}\| \leq \rho\} \subseteq K$). Gibt es so ein $\bar{x}$, heißt auch (P) streng zulässig.
Ein dual zulässiges $(\bar{y}, \bar{z})$ heißt streng zulässig für (D), wenn $\bar{z}$ im Inneren von K^* liegt ($\exists \rho > 0 : B_\rho(\bar{z}) \subseteq K^*$). Gibt es so ein $(\bar{y}, \bar{z})$, heißt auch (D) streng zulässig.

Satz (Starke Dualität)

Ist (P) streng zulässig, wird das duale Optimum $v(D)$ angenommen.
Ist (D) streng zulässig, wird das primale Optimum $v(P)$ angenommen.
In beiden Fällen gilt $v(P) = v(D)$.

Selbstduale Kegel, Innere-Punkte-Verfahren

Hier werden nur drei spezielle Kegelarten K verwendet:

- $K = \mathbb{R}^n_+$, der nichtnegative Orthant
- $K = \mathcal{Q}^n$, der quadratische Kegel
- $K = S^n_+$ der Kegel der positiv semidefiniten Matrizen

Die genauen Definitionen von $\mathcal{Q}^n$ und S^n_+ folgen demnächst.

Selbstduale Kegel, Innere-Punkte-Verfahren

Hier werden nur drei spezielle Kegelarten K verwendet:

- $K = \mathbb{R}^n_+$, der nichtnegative Orthant
- $K = \mathcal{Q}^n$, der quadratische Kegel
- $K = S^n_+$ der Kegel der positiv semidefiniten Matrizen

Die genauen Definitionen von $\mathcal{Q}^n$ und S^n_+ folgen demnächst.
Die wichtigsten Eigenschaften dieser drei sind:

- Sie sind selbstdual, d.h., $K = K^*$.
- Es sind geeignete Barriere-Funktionen für sie bekannt.
- Gute Innere-Punkte-Codes, die die Verwendung der drei gleichzeitig erlauben, sind für MATLAB verfügbar (SeDuMi, SDPT3).
- Sie ermöglichen viele wichtige Anwendungen zu modellieren.

Selbstduale Kegel, Innere-Punkte-Verfahren

Hier werden nur drei spezielle Kegelarten K verwendet:

- $K = \mathbb{R}^n_+$, der nichtnegative Orthant
- $K = \mathcal{Q}^n$, der quadratische Kegel
- $K = S^n_+$ der Kegel der positiv semidefiniten Matrizen

Die genauen Definitionen von $\mathcal{Q}^n$ und S^n_+ folgen demnächst.
Die wichtigsten Eigenschaften dieser drei sind:

- Sie sind selbstdual, d.h., $K = K^*$.
- Es sind geeignete Barriere-Funktionen für sie bekannt.
- Gute Innere-Punkte-Codes, die die Verwendung der drei gleichzeitig erlauben, sind für MATLAB verfügbar (SeDuMi, SDPT3).
- Sie ermöglichen viele wichtige Anwendungen zu modellieren.

In den Anwendung ist K oft aus mehreren Einzelkegeln zusammengesetzt,

$$K = \mathbb{R}^n_+ \times \mathcal{Q}^{m_1} \times \cdots \times \mathcal{Q}^{m_k} \times S^{n_1}_+ \times \cdots \times S^{n_h}_+$$

Das wird sich aber natürlich ergeben und $K = K^*$ gilt für diese Kombinationen immer.

Inhaltsübersicht

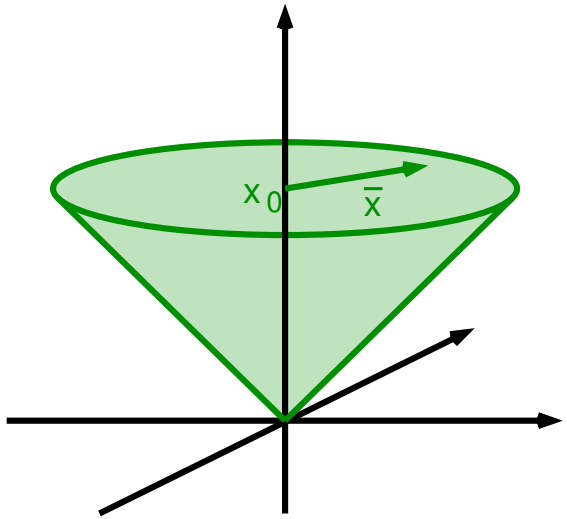
4.3 Der quadratische Kegel (Second Order Cone)

Der quadratische Kegel
oder Second-Order-Cone (SOC)

$$\mathcal{Q}^n = \left\{ \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^{n+1} : x_0 \geq \|\bar{x}\| \right\}$$

ist ein konvexer Kegel,
denn für $x, y \in \mathcal{Q}^n$, $\alpha \geq 0$ ist
$\|\alpha(\bar{x} + \bar{y})\| \leq \alpha\|\bar{x}\| + \alpha\|\bar{y}\| \leq \alpha(x_0 + y_0)$.

$\mathcal{Q}^n$ ist selbstdual, $(\mathcal{Q}^n)^* = \mathcal{Q}^n$.



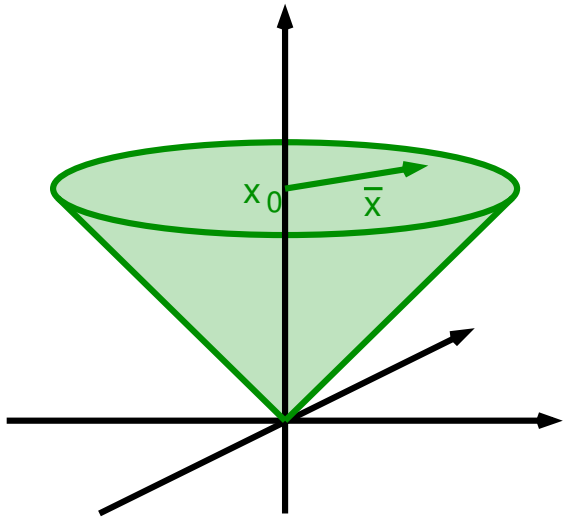
4.3 Der quadratische Kegel (Second Order Cone)

Der quadratische Kegel
oder Second-Order-Cone (SOC)

$$\mathcal{Q}^n = \left\{ \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^{n+1} : x_0 \geq \|\bar{x}\| \right\}$$

ist ein konvexer Kegel,
denn für $x, y \in \mathcal{Q}^n$, $\alpha \geq 0$ ist
$\|\alpha(\bar{x} + \bar{y})\| \leq \alpha\|\bar{x}\| + \alpha\|\bar{y}\| \leq \alpha(x_0 + y_0)$.

$\mathcal{Q}^n$ ist selbstdual, $(\mathcal{Q}^n)^* = \mathcal{Q}^n$.



Statt $x \in \mathcal{Q}^n$ schreiben wir auch $x \geq_{\mathcal{Q}} 0$. Für $a, b \in \mathbb{R}^{n+1}$ ist $a \geq_{\mathcal{Q}} b$ gleichbedeutend mit $a - b \geq_{\mathcal{Q}} 0$, also $a - b \in \mathcal{Q}^n$.

Innere-Punkte-Verfahren verwenden $-\log(x_0^2 - \bar{x}^T\bar{x})$ als Barriere.

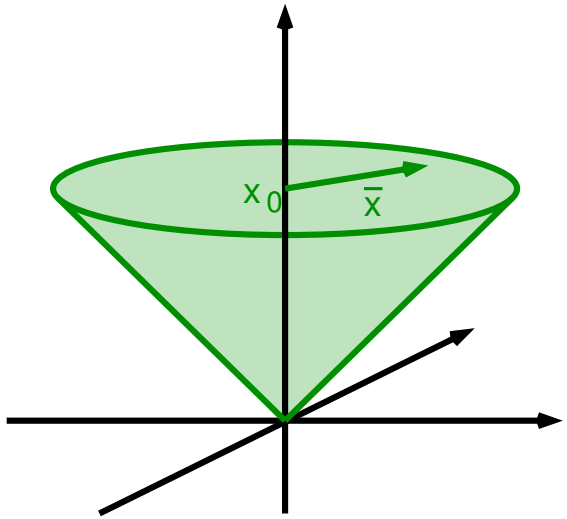
4.3 Der quadratische Kegel (Second Order Cone)

Der quadratische Kegel
oder Second-Order-Cone (SOC)

$$\mathcal{Q}^n = \left\{ \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \in \mathbb{R}^{n+1} : x_0 \geq \|\bar{x}\| \right\}$$

ist ein konvexer Kegel,
denn für $x, y \in \mathcal{Q}^n$, $\alpha \geq 0$ ist
$\|\alpha(\bar{x} + \bar{y})\| \leq \alpha\|\bar{x}\| + \alpha\|\bar{y}\| \leq \alpha(x_0 + y_0)$.

$\mathcal{Q}^n$ ist selbstdual, $(\mathcal{Q}^n)^* = \mathcal{Q}^n$.



Statt $x \in \mathcal{Q}^n$ schreiben wir auch $x \geq_{\mathcal{Q}} 0$. Für $a, b \in \mathbb{R}^{n+1}$ ist $a \geq_{\mathcal{Q}} b$ gleichbedeutend mit $a - b \geq_{\mathcal{Q}} 0$, also $a - b \in \mathcal{Q}^n$.

Innere-Punkte-Verfahren verwenden $-\log(x_0^2 - \bar{x}^T\bar{x})$ als Barriere.

Lineare Programme, die als Kegel nur $\mathbb{R}^n_+$ und mindestens ein $\mathcal{Q}^n$ einsetzen, heißen Second-Order-Cone Programme (kurz SOCP).
Ein SOCP mit nur einem $\mathcal{Q}^n$ liest sich

$$(P) \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq_{\mathcal{Q}} 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq_{\mathcal{Q}} 0 \end{array}$$

[Für nur einen einzelnen SOC ist es sogar explizit lösbar.]

Ein SOCP-Beispiel mit Dualitätslücke

$$(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array}$$

$$(D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}$$

Ein SOCP-Beispiel mit Dualitätslücke

$$(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array} \qquad (D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.

Ein SOCP-Beispiel mit Dualitätslücke

$$(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array} \qquad (D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.
In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Ein SOCP-Beispiel mit Dualitätslücke

$$
(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array}
\qquad
(D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}
$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.
In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Beachte: $v(P) \neq v(D)$ geht nur, wenn beide, Primales und Duales, keinen streng zulässigen Punkt haben.

Ein SOCP-Beispiel mit Dualitätslücke

$$(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array} \qquad (D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.
In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Beachte: $v(P) \neq v(D)$ geht nur, wenn beide, Primales und Duales, keinen streng zulässigen Punkt haben.

In Anwendungen hat eine SOC-Nebenbedingung meist die Struktur

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & g^T x + d \geq \|Ax - b\| \\ & x \in \mathbb{R}^n \end{array}$$

Ein SOCP-Beispiel mit Dualitätslücke

$$(P) \quad \begin{array}{ll} \min & -x_2 \\ \text{s.t.} & x_0 - x_1 = 0 \\ & x_0 \geq \sqrt{x_1^2 + x_2^2} \end{array} \qquad (D) \quad \begin{array}{ll} \max & 0 \\ \text{s.t.} & y_1 + z_0 = 0 \\ & -y_1 + z_1 = 0 \\ & z_2 = -1 \\ & z_0 \geq \sqrt{z_1^2 + z_2^2} \end{array}$$

In (P) folgt aus $x_0 = x_1$ und $x \in \mathcal{Q}^2$, dass $x_2 = 0$, also $v(P) = 0$.
In (D) folgt aus $z_0 = -y_1 = -z_1$ und $z \in \mathcal{Q}^2$, dass $z_2 = 0$,
in Widerspruch zu $z_2 = -1$, also $v(D) = -\infty$.

Beachte: $v(P) \neq v(D)$ geht nur, wenn beide, Primales und Duales, keinen streng zulässigen Punkt haben.

In Anwendungen hat eine SOC-Nebenbedingung meist die Struktur

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & g^T x + d \geq \|Ax - b\| \\ & x \in \mathbb{R}^n \end{array} \quad \rightarrow \quad \begin{array}{ll} \min & c^T x \\ \text{s.t.} & x_0 = g^T x + d \\ & \bar{x} = Ax - b \\ & \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \geq_{\mathcal{Q}} 0, x \in \mathbb{R}^n \end{array}$$

4.3.1 SOC Anwendung: Regularisierte Approximation

Ein Datenvektor $b \in \mathbb{R}^m$ soll einerseits möglichst gut durch eine Linearkombination $\sum_i a^{(i)} x_i$ von Grundfunktionen $a^{(i)} \in \mathbb{R}^m$ dargestellt werden ($\rightarrow \|Ax - b\|$), andererseits möchte man die „Größe“ von x möglichst klein halten oder Eigenschaften von Ax kontrollieren($\rightarrow f(x)$).

4.3.1 SOC Anwendung: Regularisierte Approximation

Ein Datenvektor $b \in \mathbb{R}^m$ soll einerseits möglichst gut durch eine Linearkombination $\sum_i a^{(i)} x_i$ von Grundfunktionen $a^{(i)} \in \mathbb{R}^m$ dargestellt werden ($\rightarrow \|Ax - b\|$), andererseits möchte man die „Größe“ von x möglichst klein halten oder Eigenschaften von Ax kontrollieren($\rightarrow f(x)$).

Dieses bikriterielle Problem wird mit Parameter $\gamma > 0$ skalarisiert zu

$$\min_{x \in \mathbb{R}^n_+} \|Ax - b\| + \gamma f(x)$$

4.3.1 SOC Anwendung: Regularisierte Approximation

Ein Datenvektor $b \in \mathbb{R}^m$ soll einerseits möglichst gut durch eine Linearkombination $\sum_i a^{(i)} x_i$ von Grundfunktionen $a^{(i)} \in \mathbb{R}^m$ dargestellt werden ($\rightarrow \|Ax - b\|$), andererseits möchte man die „Größe“ von x möglichst klein halten oder Eigenschaften von Ax kontrollieren($\rightarrow f(x)$).

Dieses bikriterielle Problem wird mit Parameter $\gamma > 0$ skalarisiert zu

$$\min_{x \in \mathbb{R}^n_+} \|Ax - b\| + \gamma f(x)$$

Für Anwendungen in der Signalrekonstruktion/-glättung/-entstörung ist
- $b_j = h(t_j)$ das empfangene Signal zum Zeitpunkt t_j
- $a_j^{(i)} = h_i(t_j)$ der Wert der Basisfunktion h_i zum Zeitpunkt t_j
- x_i der zu ermittelnde Anteil der Basisfunktion h_i am Signal

Die Wahl von f unterscheidet sich je nach Zielen, hier:
- Glättung bei Sprunganteilen
- Reduktion der Zahl verwendeter Basisfunktionen

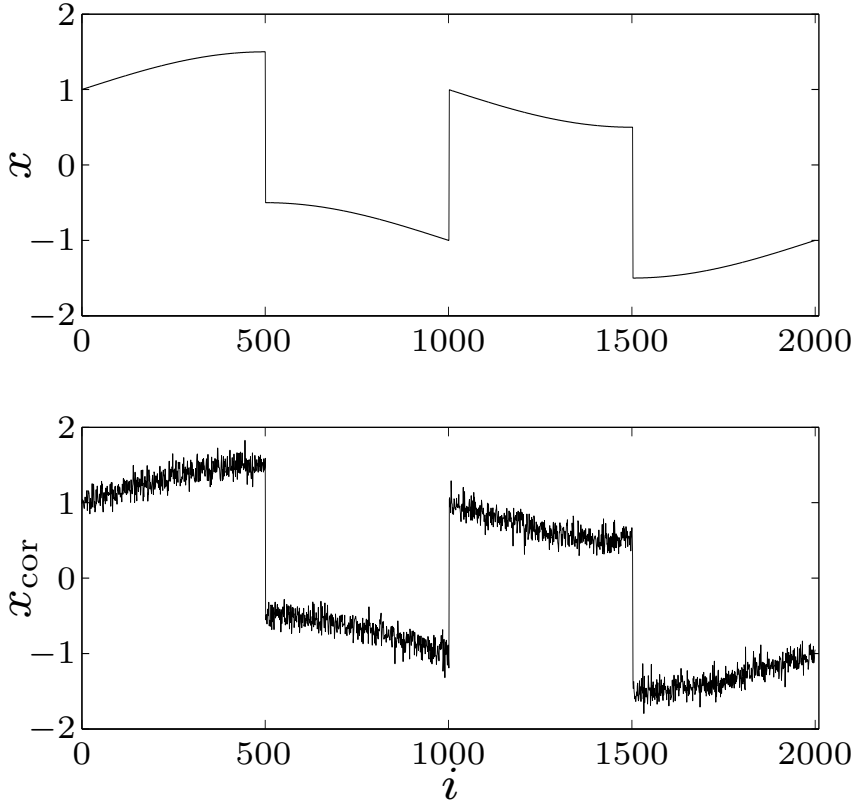
Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax, bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1}([Ax]_{j+1} - [Ax]_j)^2$, aber dadurch werden Sprünge „herausgeglättet“.

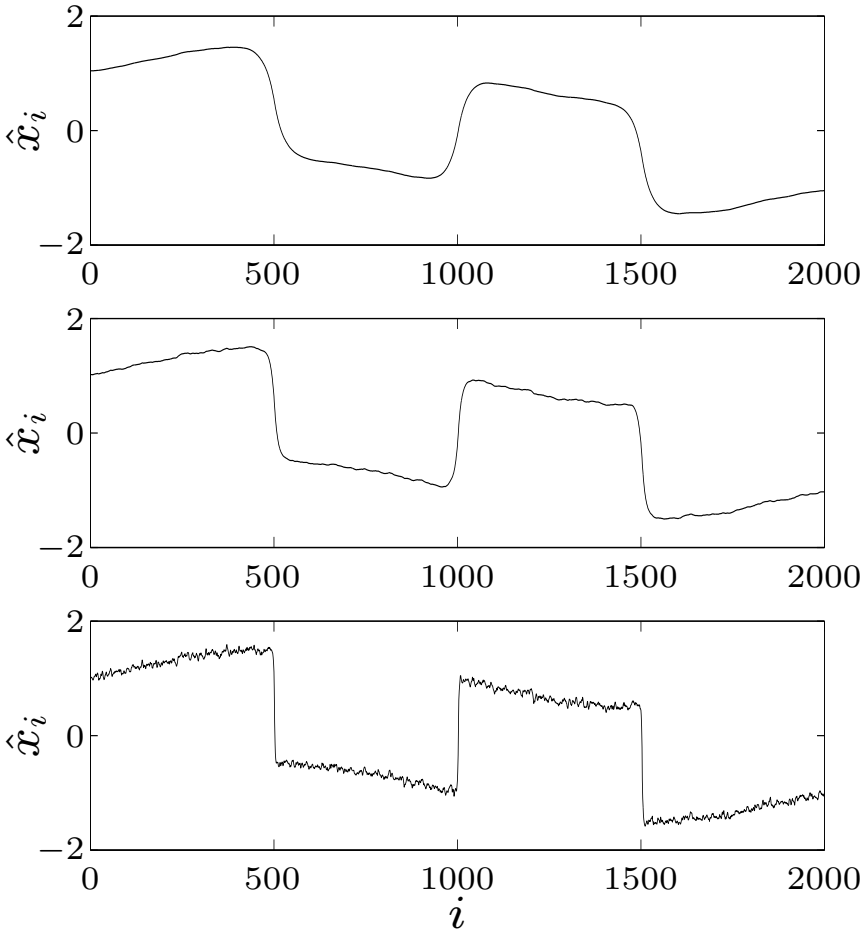
[Folie von Boyd, http://www.stanford.edu/∼boyd/cvxbook/, 25.10.2009]

Andere Notation!

total variation reconstruction example



original signal x and noisy signal x_{cor}



three solutions on trade-off curve $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{quad}}(\hat{x})$

150: 356∈[356,356]

Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax,
bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1}([Ax]_{j+1} - [Ax]_j)^2$,
aber dadurch werden Sprünge „herausgeglättet“.
Sollen Sprunganteile erhalten bleiben, verwendet man besser die 1-Norm,

$$f(x) = \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j| \,,$$

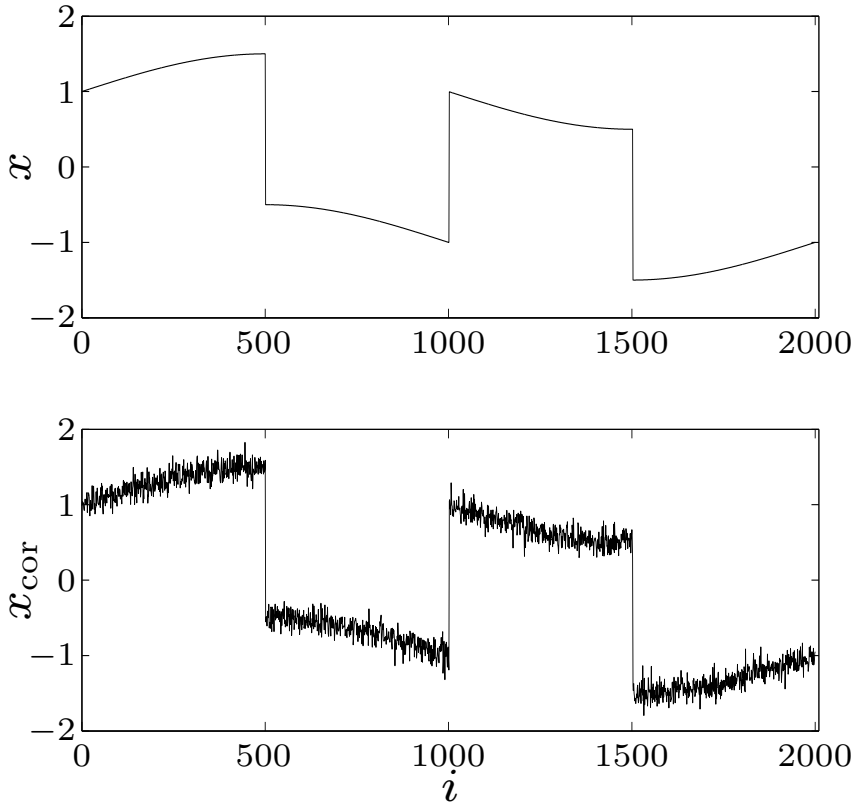
für diese dürfen gerne große Terme an wenigen Stellen auftreten, solange dadurch die Summe insgesamt nicht größer wird.

$$\min_{x \in \mathbb{R}^n_+} \|Ax - b\| + \gamma \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|$$

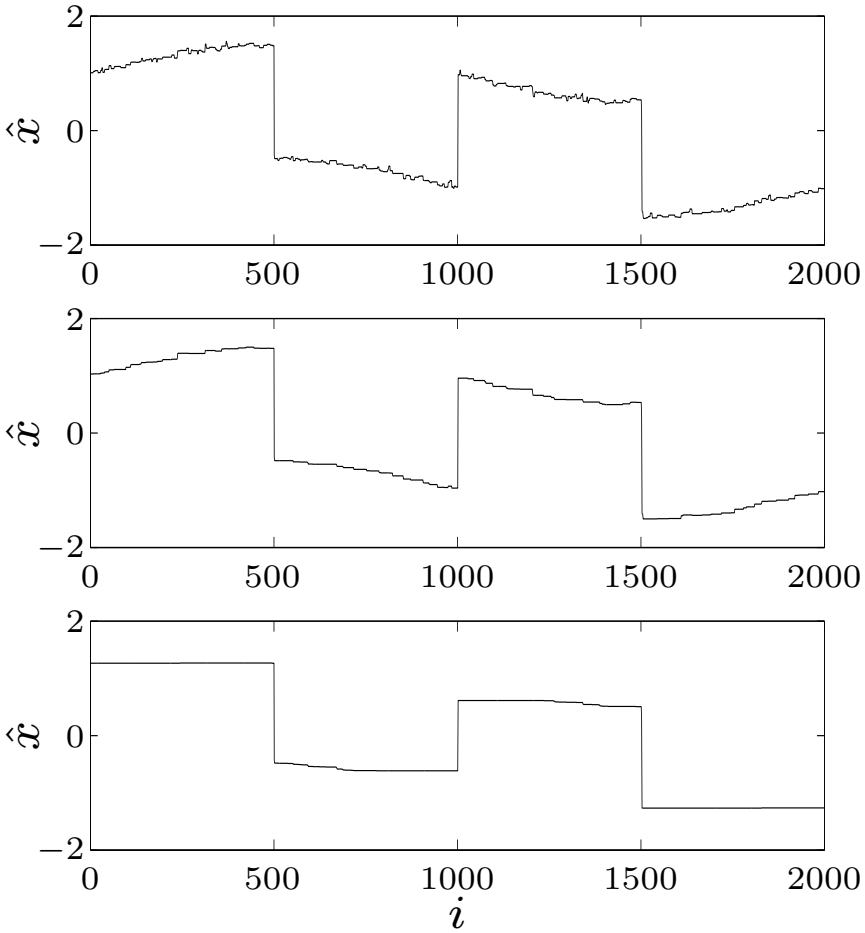
ist ein SOCP:

[Folie von Boyd, http://www.stanford.edu/∼boyd/cvxbook/, 25.10.2009]

Andere Notation!



original signal x and noisy signal x_{cor}



three solutions on trade-off curve $\|\hat{x} - x_{\text{cor}}\|_2$ versus $\phi_{\text{tv}}(\hat{x})$

Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax,
bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1}([Ax]_{j+1} - [Ax]_j)^2$,
aber dadurch werden Sprünge „herausgeglättet“.
Sollen Sprunganteile erhalten bleiben, verwendet man besser die 1-Norm,

$$f(x) = \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j| \,,$$

für diese dürfen gerne große Terme an wenigen Stellen auftreten, solange dadurch die Summe insgesamt nicht größer wird.

$$\min_{x \in \mathbb{R}^n_+} \|Ax - b\| + \gamma \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|$$

ist ein SOCP:

Glättung bei Sprunganteilen

Bei Glättung bestraft f normalerweise starke Änderungen in Ax,
bei quadratischer Glättung wäre $f(x) = \sum_{j=1}^{m-1}([Ax]_{j+1} - [Ax]_j)^2$,
aber dadurch werden Sprünge „herausgeglättet“.
Sollen Sprunganteile erhalten bleiben, verwendet man besser die 1-Norm,

$$f(x) = \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j| \,,$$

für diese dürfen gerne große Terme an wenigen Stellen auftreten, solange dadurch die Summe insgesamt nicht größer wird.

$$\min_{x \in \mathbb{R}^n_+} \|Ax - b\| + \gamma \sum_{j=1}^{m-1} |[Ax]_{j+1} - [Ax]_j|$$

ist ein SOCP: [Notation: $A_{j,\bullet}$ ... Zeile j von A]

$$\begin{array}{ll} \min & \xi_0 + \gamma \sum_{j=1}^{m-1} s_j \\ \text{s.t.} & \bar{\xi} - Ax = -b \\ & -s_j \le (A_{j+1,\bullet} - A_{j,\bullet})x \le s_j \qquad j = 1, \ldots, m-1 \\ & \begin{bmatrix} \xi_0 \\ \bar{\xi} \end{bmatrix} \ge_{\mathcal{Q}} 0, x \ge 0, s \ge 0 \end{array}$$

Reduktion der Zahl verwendeter Basisfunktionen

Wenn man weiß, dass sich das Signal nur aus wenigen Basisfunktionen zusammensetzt oder zur Datenkompression nur wenige verwenden will, möchte man ein gemischt-ganzzahliges nichtlineares Problem lösen:

$$\min_{x \in \mathbb{R}^+} \|Ax - b\| + \gamma |\{i = 1, \ldots, n : x_i > 0\}|.$$

Reduktion der Zahl verwendeter Basisfunktionen

Wenn man weiß, dass sich das Signal nur aus wenigen Basisfunktionen zusammensetzt oder zur Datenkompression nur wenige verwenden will, möchte man ein gemischt-ganzzahliges nichtlineares Problem lösen:

$$\min_{x\in\mathbb{R}^+} \|Ax-b\| + \gamma|\{i=1,\ldots,n : x_i > 0\}|.$$

Dies ist zu aufwendig, daher gibt man sich mit der 1-Norm zufrieden,

$$\min_{x\in\mathbb{R}^+} \|Ax-b\| + \gamma \sum_{i=1}^{n} |x_i|,$$

und hofft, dass diese den Wert auf nur wenige Koordinaten konzentriert.

Übung: Erstelle das SOCP zur 1-Norm Variante.

Reduktion der Zahl verwendeter Basisfunktionen

Wenn man weiß, dass sich das Signal nur aus wenigen Basisfunktionen zusammensetzt oder zur Datenkompression nur wenige verwenden will, möchte man ein gemischt-ganzzahliges nichtlineares Problem lösen:

$$\min_{x\in\mathbb{R}^+} \|Ax-b\| + \gamma|\{i=1,\ldots,n : x_i > 0\}|.$$

Dies ist zu aufwendig, daher gibt man sich mit der 1-Norm zufrieden,

$$\min_{x\in\mathbb{R}^+} \|Ax-b\| + \gamma \sum_{i=1}^{n} |x_i|,$$

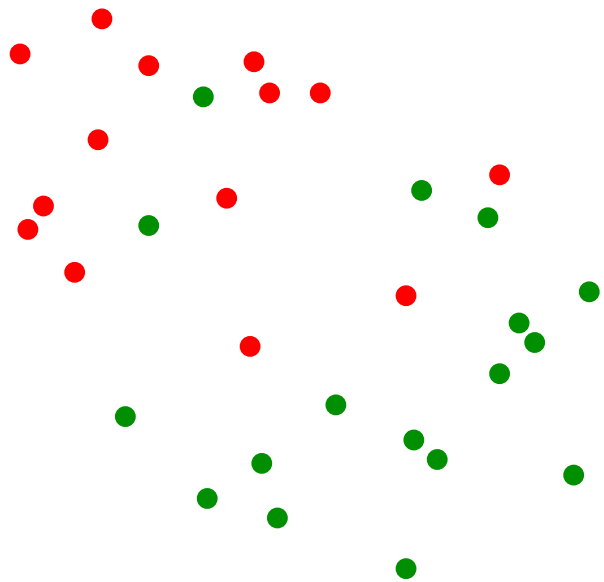
und hofft, dass diese den Wert auf nur wenige Koordinaten konzentriert.

Übung: Erstelle das SOCP zur 1-Norm Variante.

Bei bikriteriellen Aufgaben bietet es sich an, das Problem für mehrere Parameterwerte $\gamma > 0$ zu lösen und die jeweils erzeugten Funktionswertpaare in einer Graphik miteinander zu vergleichen.

4.3.2 Klassifizierung, Support-Vektor

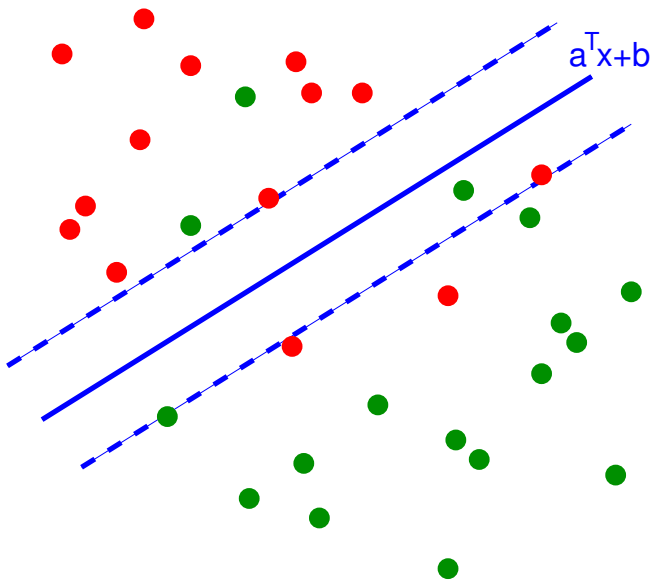
Für Datenpunkte im $\mathbb{R}^n$, die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).



4.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im $\mathbb{R}^n$, die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).

Gegeben zwei disjunkte endliche Mengen $G, R \subset \mathbb{R}^n$, finde $a^T x + b$ (mit Variablen a und b) mit „möglichst“ $a^T x + b \geq 1$ für $x \in G$ und $a^T x + b \leq -1$ für $x \in R$.



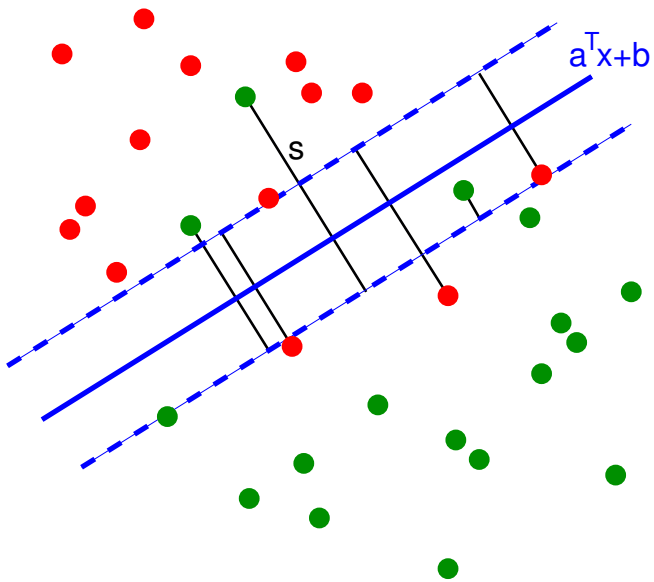
Schwierigkeiten:

- Für gute Trennung soll $\|a\|$ klein sein.
- Was tun, wenn Fehlklassifizierungen unvermeidbar sind?

4.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im $\mathbb{R}^n$, die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).

Gegeben zwei disjunkte endliche Mengen $G, R \subset \mathbb{R}^n$, finde $a^T x + b$ (mit Variablen a und b) mit „möglichst“ $a^T x + b \geq 1$ für $x \in G$ und $a^T x + b \leq -1$ für $x \in R$.



Schwierigkeiten:

- Für gute Trennung soll $\|a\|$ klein sein.
- Was tun, wenn Fehlklassifizierungen unvermeidbar sind?

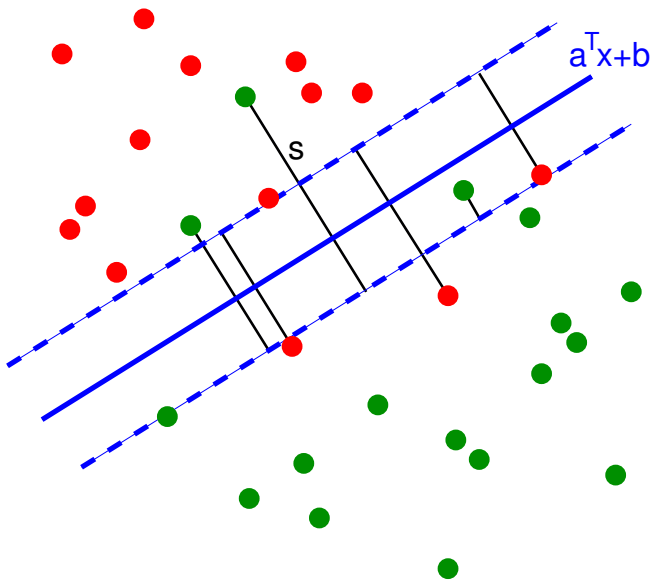
Ein Ansatz: Minimiere gleichzeitig $\|a\|$ und die Summe der Verletzungen der Ungleichungsbedingungen, skalarisiert mit Parameter $\gamma > 0$,

$$
\begin{array}{lll}
\min & \|a\| + \gamma \sum_{x \in G \cup R} s_x & \\
\text{s.t.} & x^T a - b \geq 1 - s_x & x \in G \\
& x^T a - b \leq s_x - 1 & x \in R \\
& a \in \mathbb{R}^n, b \in \mathbb{R}, s \in \mathbb{R}_+^{G \cup R} &
\end{array}
$$

4.3.2 Klassifizierung, Support-Vektor

Für Datenpunkte im $\mathbb{R}^n$, die eine bestimmte Eigenschaft haben bzw. nicht haben, soll eine Ebene gefunden werden, die die Punkttypen „möglichst gut“ voneinander trennt (Ziel: neue Punkte klassifizieren).

Gegeben zwei disjunkte endliche Mengen $G, R \subset \mathbb{R}^n$, finde $a^T x + b$ (mit Variablen a und b) mit „möglichst“ $a^T x + b \geq 1$ für $x \in G$ und $a^T x + b \leq -1$ für $x \in R$.



Schwierigkeiten:

- Für gute Trennung soll $\|a\|$ klein sein.
- Was tun, wenn Fehlklassifizierungen unvermeidbar sind?

Ein Ansatz: Minimiere gleichzeitig $\|a\|$ und die Summe der Verletzungen der Ungleichungsbedingungen, skalarisiert mit Parameter $\gamma > 0$,

$$\begin{array}{lll} \min & \|a\| + \gamma \sum_{x \in G \cup R} s_x & \\ \text{s.t.} & x^T a - b \geq 1 - s_x & x \in G \\ & x^T a - b \leq s_x - 1 & x \in R \\ & a \in \mathbb{R}^n, b \in \mathbb{R}, s \in \mathbb{R}_+^{G \cup R} & \end{array} \quad \rightarrow \quad \begin{array}{lll} \min & a_0 + \gamma \sum_{x \in G \cup R} s_x & \\ \text{s.t.} & x^T a - b \geq 1 - s_x & x \in G \\ & x^T a - b \leq s_x - 1 & x \in R \\ & \begin{bmatrix} a_0 \\ a \end{bmatrix} \geq_{\mathcal{Q}} 0, b \in \mathbb{R}, s \geq 0 & \end{array}$$

4.3.3 Das Markowitz Modell

Im Markowitz-Modell der Portfolio-Optimierung soll gegebenes Kapital so mit einem erwarteten Mindestgewinn investiert werden, dass das Risiko minimiert wird.

4.3.3 Das Markowitz Modell

Im Markowitz-Modell der Portfolio-Optimierung soll gegebenes Kapital so mit einem erwarteten Mindestgewinn investiert werden, dass das Risiko minimiert wird.
$x \in \mathbb{R}^n_+$ mit $\mathbf{1}^T x = 1$ gibt den Anteil des Budgets an, der in Aktien $1, \ldots, n$ investiert wird. Der Gewinn g pro Anteil ist eine Zufallsvariable mit Erwartungswert $\bar{g} \in \mathbb{R}^n$ und Kovarianzmatrix $G \in S^n_+$ ($n \times n$, positiv semidefinit), $s \in \mathbb{R}$ sei ein gegebener Schwellwert. Als Maß für das Risiko sieht das Markowitz-Modell $x^T G x$ vor. [Andere Maße möglich!]

$$\begin{array}{ll} \min & x^T G x \\ \text{s.t.} & \bar{g}^T x \geq s \\ & \mathbf{1}^T x = 1 \\ & x \in \mathbb{R}^n_+ \end{array}$$

4.3.3 Das Markowitz Modell

Im Markowitz-Modell der Portfolio-Optimierung soll gegebenes Kapital so mit einem erwarteten Mindestgewinn investiert werden, dass das Risiko minimiert wird.
$x \in \mathbb{R}^n_+$ mit $\mathbf{1}^T x = 1$ gibt den Anteil des Budgets an, der in Aktien $1, \ldots, n$ investiert wird. Der Gewinn g pro Anteil ist eine Zufallsvariable mit Erwartungswert $\bar{g} \in \mathbb{R}^n$ und Kovarianzmatrix $G \in S^n_+$ ($n \times n$, positiv semidefinit), $s \in \mathbb{R}$ sei ein gegebener Schwellwert. Als Maß für das Risiko sieht das Markowitz-Modell $x^T G x$ vor. [Andere Maße möglich!]

$$\begin{array}{ll} \min & x^T G x \\ \text{s.t.} & \bar{g}^T x \geq s \\ & \mathbf{1}^T x = 1 \\ & x \in \mathbb{R}^n_+ \end{array}$$

Wegen G positiv semidefinit ist das ein konvex quadratisches Problem.
[Das bikriterielle Ziel Gewinn gegen Risiko wird hier durch eine Nebenbedingung an eines der Kriterien umgesetzt.]

Modellierung als SOCP?

Quadratische Nebenbedingungen mit SOCP

Sei $Q \in S_+^n$ positiv definit, $q \in \mathbb{R}^n$, $d \in \mathbb{R}$. Die konvex quadratische Nebenbedingung

$$\frac{1}{2}x^T Qx + q^T x + d \leq 0$$

lässt sich mit $Q = Q^{\frac{1}{2}} Q^{\frac{1}{2}}$ über

$$\|Q^{\frac{1}{2}}x + Q^{-\frac{1}{2}}q\| \leq \sqrt{q^T Q^{-1} q - 2d}$$

als SOC-Nebenbedinung schreiben (Beweis: beide Seiten quadrieren).
[Eigenwertzerlegung $Q = P\Lambda P^T \rightarrow Q^{\frac{1}{2}} = P\Lambda^{\frac{1}{2}} P^T$, $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n) \geq 0$]

Quadratische Nebenbedingungen mit SOCP

Sei $Q \in S_+^n$ positiv definit, $q \in \mathbb{R}^n$, $d \in \mathbb{R}$. Die konvex quadratische Nebenbedingung

$$\frac{1}{2}x^T Qx + q^T x + d \leq 0$$

lässt sich mit $Q = Q^{\frac{1}{2}} Q^{\frac{1}{2}}$ über

$$\|Q^{\frac{1}{2}}x + Q^{-\frac{1}{2}}q\| \leq \sqrt{q^T Q^{-1} q - 2d}$$

als SOC-Nebenbedinung schreiben (Beweis: beide Seiten quadrieren).
[Eigenwertzerlegung $Q = P\Lambda P^T \rightarrow Q^{\frac{1}{2}} = P\Lambda^{\frac{1}{2}} P^T$, $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n) \geq 0$]

Für das Markowitz-Modell ist es einfacher, $x_0 \geq \|G^{\frac{1}{2}}x\|$ reicht aus

$$\begin{array}{ll} \min & x_0 \\ \text{s.t.} & \bar{x} = G^{\frac{1}{2}}x \\ & \bar{g}^T x \geq s \\ & \mathbf{1}^T x = 1 \\ & \begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \geq_{\mathcal{Q}} 0, x \geq 0 \end{array}$$

Probabilistische Nebenbedingung, Chance Constraint

Unter der Annahme, dass g mit Varianz G um $\bar{g}$ normalverteilt ist, sollen nun zusätzlich nur Investitionspläne erlaubt werden, für die die Wahrscheinlichkeit, dass der Gewinn über einem Schwellwert $\underline{s} < s$ bleibt, mindestens $\eta \in (0, 1)$ ist,

$$\mathbb{P}(g^T x \geq \underline{s}) \geq \eta$$

Probabilistische Nebenbedingung, Chance Constraint

Unter der Annahme, dass g mit Varianz G um $\bar{g}$ normalverteilt ist, sollen nun zusätzlich nur Investitionspläne erlaubt werden, für die die Wahrscheinlichkeit, dass der Gewinn über einem Schwellwert $\underline{s} < s$ bleibt, mindestens $\eta \in (0,1)$ ist,

$$\mathbb{P}(g^T x \geq \underline{s}) \geq \eta \qquad \rightarrow \qquad \begin{array}{ll} \min & x^T G x \\ \text{s.t.} & \bar{g}^T x \geq s \\ & \mathbb{P}(g^T x \geq \underline{s}) \geq \eta \\ & \mathbf{1}^T x = 1 \\ & x \in \mathbb{R}^n_+ \end{array}$$

Dies modelliert man mit einer Technik der robusten Optimierung: $g^T x \geq \underline{s}$ wird als Ungleichung mit unsicheren Koeffizienten aufgefasst.

Nebenbedingungen mit unsicheren Koeffizienten

Ist in einer Ungleichung $a^T x \leq b$ über die Koeffizienten nur bekannt, dass $a \in \{\bar{a} + Hu : \|u\| = 1\}$ für gegebens $H \in S_+^n$ (pos. semidef), und soll x diese Ungleichung für all diese a erfüllen, so muss

$$\max_{\|u\|=1} \bar{a}^T x + u^T H x = \bar{a}^T x + \|Hx\| \leq b$$

gelten.

Nebenbedingungen mit unsicheren Koeffizienten

Ist in einer Ungleichung $a^T x \leq b$ über die Koeffizienten nur bekannt, dass $a \in \{\bar{a} + Hu : \|u\| = 1\}$ für gegebens $H \in S_+^n$ (pos. semidef), und soll x diese Ungleichung für all diese a erfüllen, so muss

$$\max_{\|u\|=1} \bar{a}^T x + u^T H x = \bar{a}^T x + \|Hx\| \leq b$$

gelten. Letztere Ungleichung ist als SOC darstellbar in der Form

$$\begin{aligned} \xi_0 &= b - \bar{a}^T x \\ \xi &= Hx \\ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} &\in SOC_n = \left\{ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} : \|\xi\| \leq \xi_0, \xi \in \mathbb{R}^n \right\} \end{aligned}$$

Nebenbedingungen mit unsicheren Koeffizienten

Ist in einer Ungleichung $a^T x \leq b$ über die Koeffizienten nur bekannt, dass $a \in \{\bar{a} + Hu : \|u\| = 1\}$ für gegebens $H \in S_+^n$ (pos. semidef), und soll x diese Ungleichung für all diese a erfüllen, so muss

$$\max_{\|u\|=1} \bar{a}^T x + u^T H x = \bar{a}^T x + \|Hx\| \leq b$$

gelten. Letztere Ungleichung ist als SOC darstellbar in der Form

$$\begin{aligned} \xi_0 &= b - \bar{a}^T x \\ \xi &= Hx \\ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} &\in SOC_n = \left\{ \begin{pmatrix} \xi_0 \\ \xi \end{pmatrix} : \|\xi\| \leq \xi_0, \xi \in \mathbb{R}^n \right\} \end{aligned}$$

Für die probabilistische Interpretation sei g normalverteilt um $\bar{g}$ mit Kovarianzmatrix $G = H^2$ und $g^T x \geq \underline{s}$ soll mit Wahrscheinlichkeit $0 < \eta < 1$ erfüllt sein, dann entspricht $\mathbb{P}(g^T x \geq \underline{s}) \geq \eta$ der Nebenbedingung $-\bar{g}^T x + \Phi^{-1}(\eta)\|Hx\| \leq -\underline{s}$. [$\Phi$ . . . Normalverteilung]

Inhaltsübersicht

4.4 Semidefinte Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n\times n} : A = A^T\}$ heißt positiv semidefinit, falls $v^T Av \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S^n_+$ oder $A \succeq 0$.

Sie heißt positiv definit ($A \in S^n_{++}$, $A \succ 0$), falls $v^T Av > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.
[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \ldots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

4.4 Semidefinte Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n\times n} : A = A^T\}$ heißt positiv semidefinit, falls $v^T Av \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S^n_+$ oder $A \succeq 0$.

Sie heißt positiv definit ($A \in S^n_{++}$, $A \succ 0$), falls $v^T Av > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.
[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \ldots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

$\lambda \in \mathbb{R}$ heißt Eigenwert und $v \in \mathbb{R}^n \setminus \{0\}$ Eigenvektor von A, falls $Av = \lambda v$.
Für jedes $A \in S^n$ gibt es eine Eigenwertzerlegung $A = P\Lambda P^T$ mit reellem $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ und $P \in \mathbb{R}^{n\times n}$ orthogonal (d.h., $P^TP = I$).
Für $P = [v_1, \ldots, v_n]$ ist $A = P\Lambda P^T = \sum_{i=1}^n \lambda_i v_i v_i^T$.

4.4 Semidefinte Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n\times n} : A = A^T\}$ heißt positiv semidefinit, falls $v^T Av \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S^n_+$ oder $A \succeq 0$.

Sie heißt positiv definit ($A \in S^n_{++}$, $A \succ 0$), falls $v^T Av > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.
[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \ldots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

$\lambda \in \mathbb{R}$ heißt Eigenwert und $v \in \mathbb{R}^n \setminus \{0\}$ Eigenvektor von A, falls $Av = \lambda v$.
Für jedes $A \in S^n$ gibt es eine Eigenwertzerlegung $A = P\Lambda P^T$ mit reellem $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ und $P \in \mathbb{R}^{n\times n}$ orthogonal (d.h., $P^TP = I$).
Für $P = [v_1, \ldots, v_n]$ ist $A = P\Lambda P^T = \sum_{i=1}^n \lambda_i v_i v_i^T$.

Für $A, B \in S^n$ verwenden wir als inneres Produkt

$$\langle A, B\rangle := \sum_{1\leq i,j\leq n} A_{ij}B_{ij} \qquad [= \mathrm{vec}(A)^T \mathrm{vec}(B)]$$

4.4 Semidefinte Optimierung: Positiv semidefinite Matrizen

Eine symmetrische Matrix $A \in S^n := \{A \in \mathbb{R}^{n\times n} : A = A^T\}$ heißt positiv semidefinit, falls $v^T Av \geq 0 \quad \forall v \in \mathbb{R}^n$; wir schreiben $A \in S^n_+$ oder $A \succeq 0$.

Sie heißt positiv definit ($A \in S^n_{++}$, $A \succ 0$), falls $v^T Av > 0 \quad \forall v \in \mathbb{R}^n \setminus \{0\}$.
[Für $A \succeq 0$ ($\succ 0$) und $J \subseteq \{1, \ldots, n\}$ ist $A_{J,J} \succeq 0$ ($\succ 0$).]

$\lambda \in \mathbb{R}$ heißt Eigenwert und $v \in \mathbb{R}^n \setminus \{0\}$ Eigenvektor von A, falls $Av = \lambda v$.
Für jedes $A \in S^n$ gibt es eine Eigenwertzerlegung $A = P\Lambda P^T$ mit reellem $\Lambda = \mathrm{Diag}(\lambda_1, \ldots, \lambda_n)$ und $P \in \mathbb{R}^{n\times n}$ orthogonal (d.h., $P^TP = I$).
Für $P = [v_1, \ldots, v_n]$ ist $A = P\Lambda P^T = \sum_{i=1}^n \lambda_i v_i v_i^T$.

Für $A, B \in S^n$ verwenden wir als inneres Produkt

$$\langle A, B\rangle := \sum_{1\leq i,j\leq n} A_{ij}B_{ij} \qquad [= \mathrm{vec}(A)^T \mathrm{vec}(B)]$$

Satz

Für $A \in S^n$ sind folgende Bedingungen äquivalent:

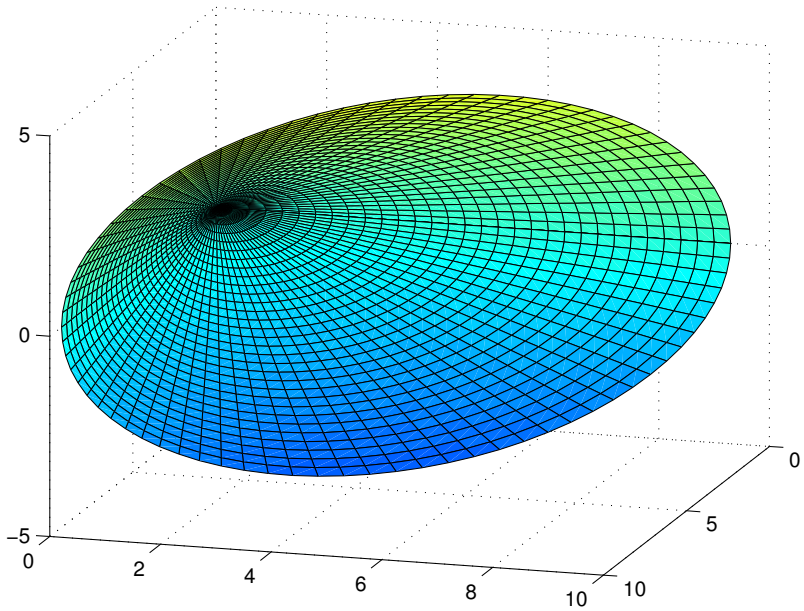
- $A \succeq 0$,
- $\lambda_i(A) \geq 0$, $i = 1, \ldots, n$, $\qquad [\Rightarrow \det(A) \geq 0]$
- $A = C^TC$ *für ein* $C \in \mathbb{R}^{k\times n}$, $\qquad$ [Es gilt: $\mathrm{Rang}(A) = \mathrm{Rang}(C)$]
- $\langle A, B\rangle \geq 0 \quad \forall B \succeq 0$.

Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$ $v^T(\alpha(X+Y))v = \alpha(v^T Xv + v^T Yv) \geq 0$.

Aus $A \in S_+^n \Leftrightarrow \langle A, B \rangle \geq 0 \; \forall B \succeq 0$ folgt: S_+^n ist selbstdual, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0$.

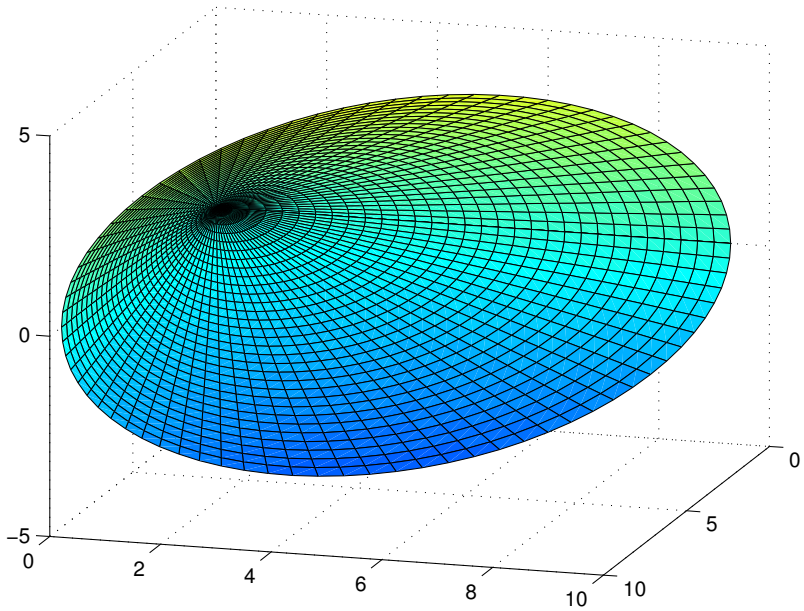


Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$ $v^T(\alpha(X+Y))v = \alpha(v^T Xv + v^T Yv) \geq 0$.

Aus $A \in S_+^n \Leftrightarrow \langle A, B \rangle \geq 0 \; \forall B \succeq 0$ folgt: S_+^n ist selbstdual, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0$.



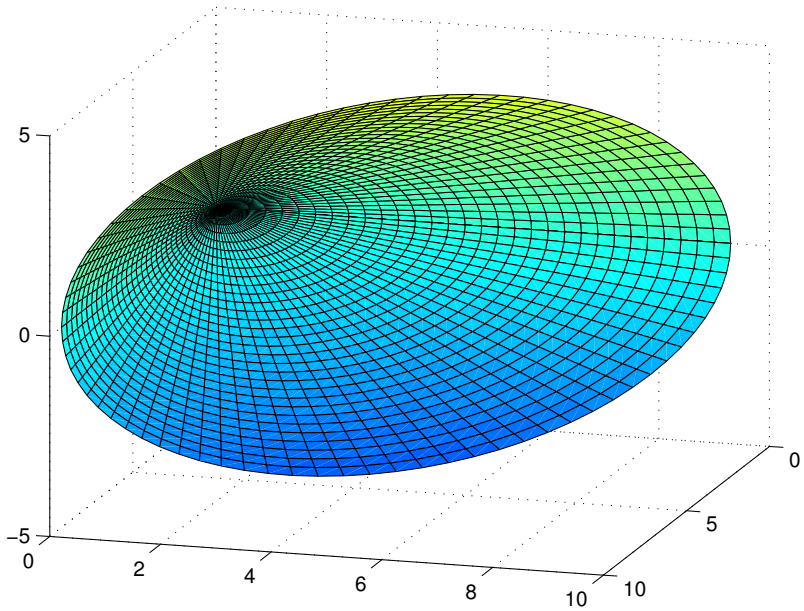
- Ist $B \in \mathbb{R}^{n \times n}$ regulär (=invertierbar), gilt $X \succeq 0 \quad \Leftrightarrow \quad B^T X B \succeq 0$.

Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$ $v^T(\alpha(X+Y))v = \alpha(v^T Xv + v^T Yv) \geq 0$.

Aus $A \in S_+^n \Leftrightarrow \langle A, B\rangle \geq 0 \; \forall B \succeq 0$ folgt: S_+^n ist selbstdual, $(S_+^n)^* = S_+^n$.

Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0$.



- Ist $B \in \mathbb{R}^{n\times n}$ regulär (=invertierbar), gilt $X \succeq 0 \quad \Leftrightarrow \quad B^T XB \succeq 0$.
- Für $A - B \succeq 0$ schreiben wir auch $A \succeq B$.

Der Kegel der positiv semidefiniten Matrizen

Die positiv semidefiniten Matrizen S_+^n bilden einen konvexen Kegel, denn für $X, Y \in S_+^n$, $\alpha \geq 0$ ist $\forall v \in \mathbb{R}^n$
$v^T(\alpha(X+Y))v = \alpha(v^T X v + v^T Y v) \geq 0.$

Aus $A \in S_+^n \Leftrightarrow \langle A, B\rangle \geq 0 \; \forall B \succeq 0$ folgt:
S_+^n ist selbstdual, $(S_+^n)^* = S_+^n$.

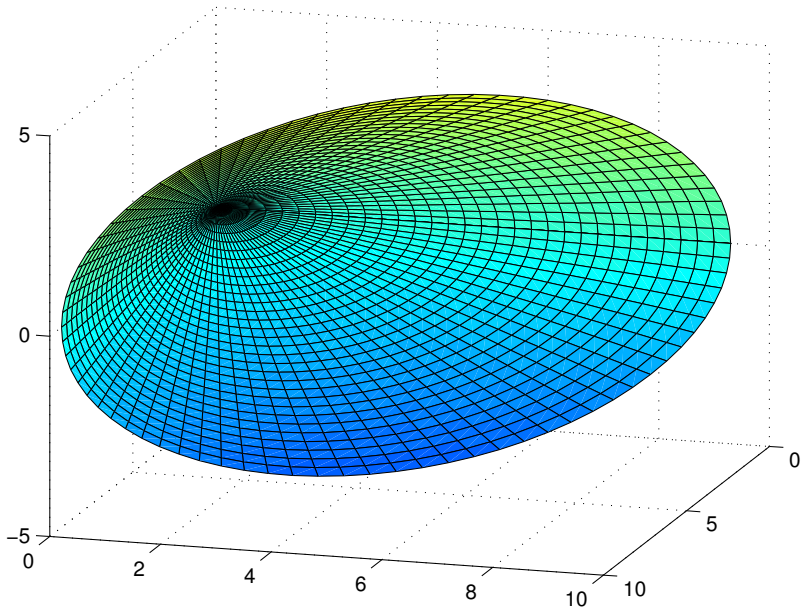


Bild rechts: $S_+^2 = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$

- Ist $B \in \mathbb{R}^{n\times n}$ regulär (=invertierbar), gilt $X \succeq 0 \quad \Leftrightarrow \quad B^T X B \succeq 0.$
- Für $A - B \succeq 0$ schreiben wir auch $A \succeq B$.

In der Modellierung semidefiniter Programme besonders nützlich:

Satz (Schur-Komplement)

Für $A \in S_{++}^m$, $C \in S_+^n$ und $B \in \mathbb{R}^{m\times n}$ gilt

$$\begin{bmatrix} A & B \\ B^T & C \end{bmatrix} \succeq 0 \quad (bzw. \succ 0) \quad \Longleftrightarrow \quad C \succeq B^T A^{-1} B \quad (bzw. \succ 0)$$

LP $\leftrightarrow$ Semidefinites Programm

$$\begin{array}{ll} \min & c^T x \\ \text{s.t.} & Ax = b \\ & x \geq 0 \end{array} \qquad \begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}X = b \\ & X \succeq 0 \end{array}$$

$$x \in \mathbb{R}^n_+ \qquad X \in S^n_+$$

$$c^T x = \sum_i c_i x_i \qquad \langle C, X \rangle = \sum_{i,j} C_{ij} X_{ij}$$

$$Ax = \begin{pmatrix} a_1^T x \\ \vdots \\ a_m^T x \end{pmatrix} \qquad \mathcal{A}X = \begin{pmatrix} \langle A_1, X \rangle \\ \vdots \\ \langle A_m, X \rangle \end{pmatrix}$$

$$A^T y = \sum_i a_i y_i \qquad \mathcal{A}^T y = \sum_i A_i y_i$$

$$\begin{array}{ll} \max & b^T y \\ \text{s.t.} & A^T y + z = c \\ & y \in \mathbb{R}^m, z \geq 0 \end{array} \qquad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^m, Z \succeq 0 \end{array}$$

Semidefinites Programm (SDP) in Normalform

$$
\begin{array}{lll} & \min & \langle C, X\rangle \\ (P) & \text{s.t.} & \mathcal{A}X = b \\ & & X \succeq 0 \end{array}
\qquad
\begin{array}{lll} & \max & b^T y \\ (D) & \text{s.t.} & \mathcal{A}^T y + Z = C \\ & & y \in \mathbb{R}^n, Z \succeq 0 \end{array}
$$

Ist eines der beiden streng zulässig, gilt $v(P) = v(D)$.

Semidefinites Programm (SDP) in Normalform

$$
(P) \quad \begin{array}{ll} \min & \langle C, X \rangle \\ \text{s.t.} & \mathcal{A}X = b \\ & X \succeq 0 \end{array} \qquad (D) \quad \begin{array}{ll} \max & b^T y \\ \text{s.t.} & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^n, Z \succeq 0 \end{array}
$$

Ist eines der beiden streng zulässig, gilt $v(P) = v(D)$.

In der Praxis gibt es oft mehrere $X_i \succeq 0$, für die Theorie reicht eine:

$$
X_1 \succeq 0,\ X_2 \succeq 0,\ \ldots,\ X_k \succeq 0 \quad \Leftrightarrow \quad \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & X_k \end{bmatrix} \succeq 0.
$$

$\Rightarrow$ Semidefinite Optimierung enthält Lineare Optimierung ($X_i \in S^1_+$).

Semidefinites Programm (SDP) in Normalform

$$(P)\quad \begin{array}{ll}\min & \langle C, X\rangle \\ \text{s.t.} & \mathcal{A}X = b \\ & X \succeq 0\end{array} \qquad (D)\quad \begin{array}{ll}\max & b^T y \\ \text{s.t.} & \mathcal{A}^T y + Z = C \\ & y \in \mathbb{R}^n, Z \succeq 0\end{array}$$

Ist eines der beiden streng zulässig, gilt $v(P) = v(D)$.

In der Praxis gibt es oft mehrere $X_i \succeq 0$, für die Theorie reicht eine:

$$X_1 \succeq 0,\ X_2 \succeq 0,\ \ldots,\ X_k \succeq 0 \quad \Leftrightarrow \quad \begin{bmatrix} X_1 & 0 & \cdots & 0 \\ 0 & X_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & X_k \end{bmatrix} \succeq 0.$$

$\Rightarrow$ Semidefinite Optimierung enthält Lineare Optimierung ($X_i \in S^1_+$).

Mit semidefiniter Optimierung sind auch SOC-Bedingungen darstellbar:

$$\begin{bmatrix} x_0 \\ \bar{x} \end{bmatrix} \geq_{\mathcal{Q}} 0 \quad \overset{x_0 \geq 0}{\Leftrightarrow} \quad x_0 \geq \frac{1}{x_0}\bar{x}^T I \bar{x} \quad \overset{\text{Schur}}{\Leftrightarrow} \quad \begin{bmatrix} x_0 & \bar{x}^T \\ \bar{x} & x_0 I \end{bmatrix} \succeq 0.$$

[für $x_0 = 0$ direkt nachprüfen]

Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0,\ y \geq 0,\ xy - z^2 \geq 0$$

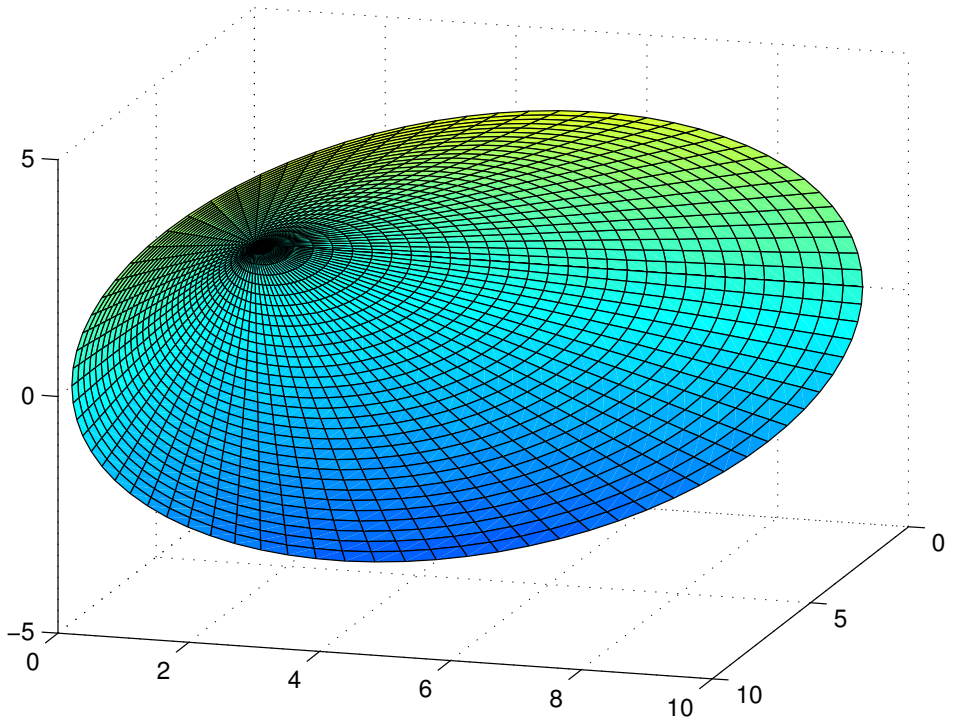


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$$\Rightarrow x \geq 0,\ y \geq 0,\ xy - z^2 \geq 0$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2, \text{ wie LP}$$

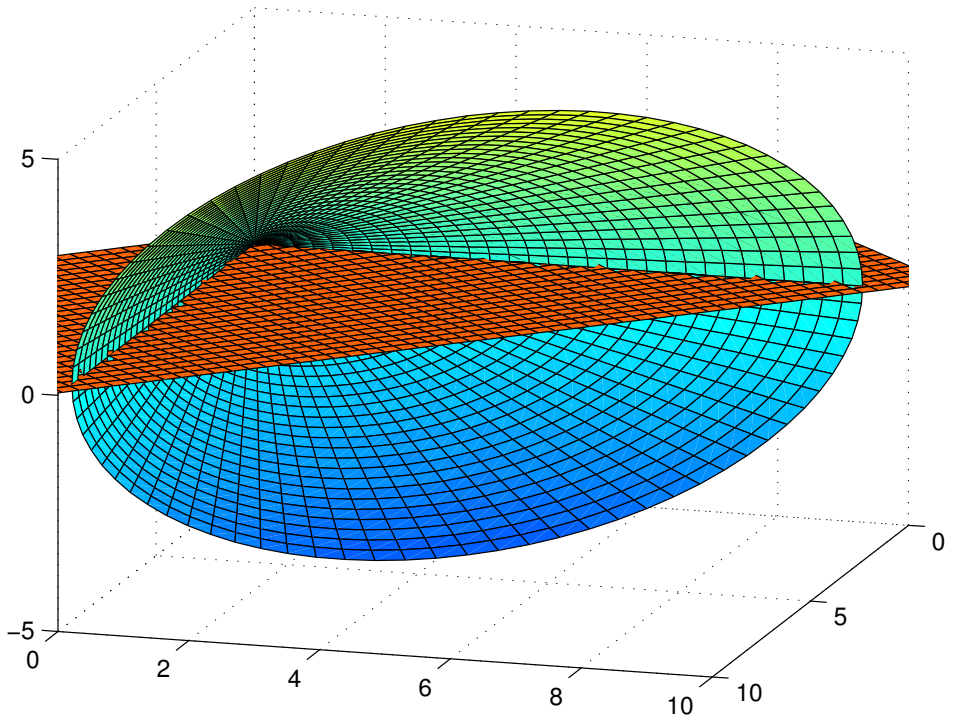


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$\Rightarrow x \geq 0,\ y \geq 0,\ xy - z^2 \geq 0$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2$, wie LP

$A \succ 0, \beta > 0$

$\rightarrow$ „Normalfall Ellipse“

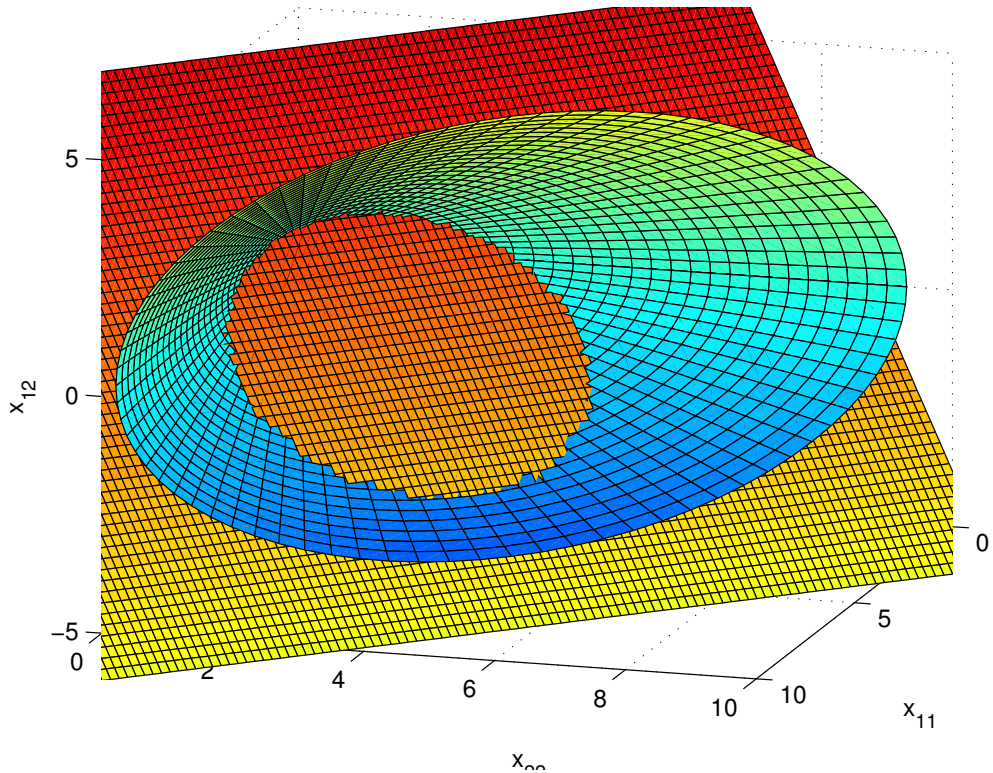


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$\Rightarrow x \geq 0,\ y \geq 0,\ xy - z^2 \geq 0$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}^2_+$, wie LP

$A \succ 0, \beta > 0$

$\rightarrow$ „Normalfall Ellipse“

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta < 0 \rightarrow z = \frac{1}{2}\beta$$

$\Leftrightarrow xy \geq \frac{1}{4}\beta^2$, Hyperbel

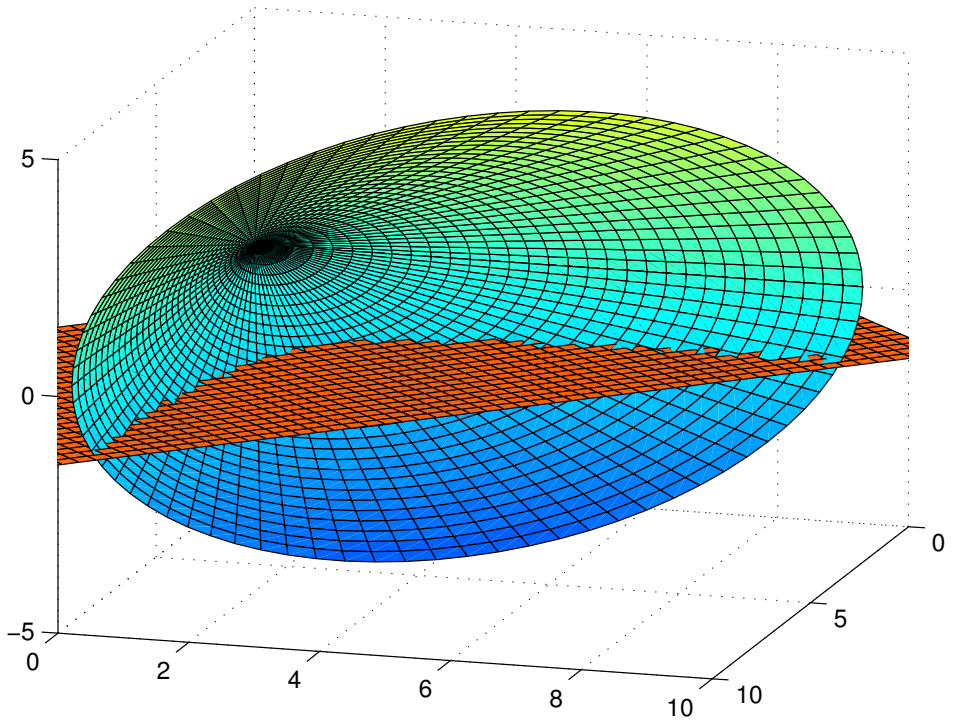


Illustration: $X \in S_+^2$ geschnitten mit $\langle A, X \rangle = \beta$

$$X = \begin{bmatrix} x & z \\ z & y \end{bmatrix} \succeq 0.$$

$\Rightarrow x \geq 0,\ y \geq 0,\ xy - z^2 \geq 0$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta = 0 \rightarrow z = 0$$

$\Leftrightarrow \begin{bmatrix} x \\ y \end{bmatrix} \in \mathbb{R}_+^2$, wie LP

$A \succ 0, \beta > 0$

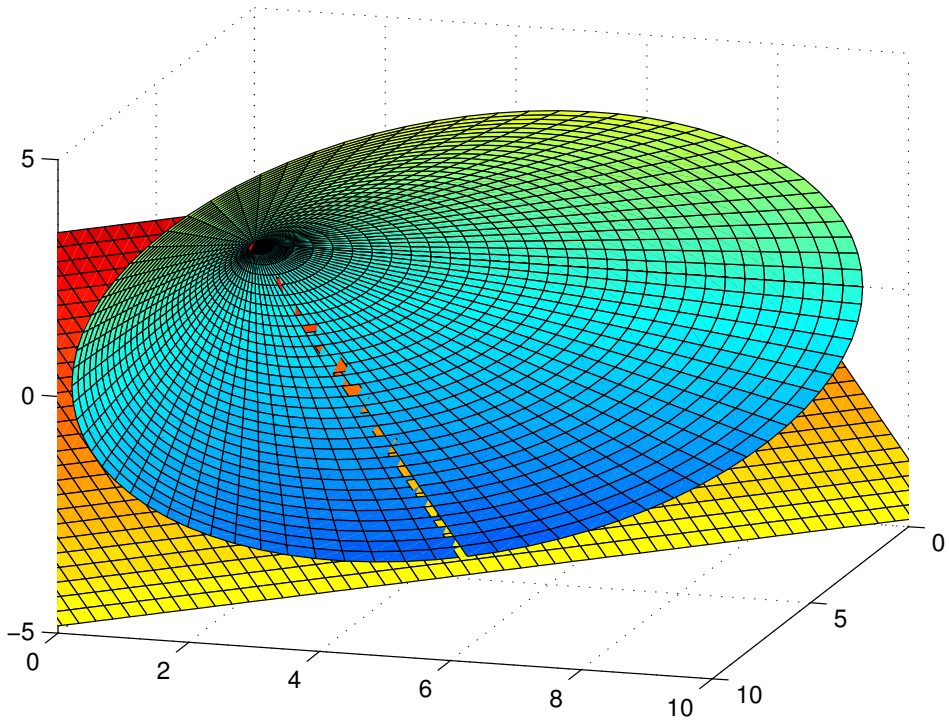
$\rightarrow$ „Normalfall Ellipse“

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \beta < 0 \rightarrow z = \frac{1}{2}\beta$$

$\Leftrightarrow xy \geq \frac{1}{4}\beta^2$, Hyperbel

$A = vv^T, \beta = 0 \rightarrow v$ EV zu $\lambda_1 = 0$

Nur Randpunkte, numerisch schwer!



Beispiel mit Dualitätslücke

$$\min \quad x_{12} \qquad\qquad \max \quad y_1$$

$$\text{s.t.} \begin{bmatrix} 0 & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1+x_{12} \end{bmatrix} \succeq 0 \qquad \text{s.t.} \; Z = \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \succeq 0$$

Entsprechende Matrizen:

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} \max \quad & \langle C, X \rangle \\ & \langle A_1, X \rangle = 1 \\ & \langle A_2, X \rangle = 0 \\ & \langle A_3, X \rangle = 0 \\ & \langle A_4, X \rangle = 0 \\ & X \succeq 0 \end{aligned}$$

$x_{11} = 0 \quad \Rightarrow \quad x_{12} = 0,$ primale Optimallösung ist 0.

$z_{22} = 0 \quad \Rightarrow \quad \frac{1+y_1}{2} = 0,$ duale Optimallösung ist -1.

Problem: Primales Problem instabil

$$\min \quad x_{12} \qquad\qquad \max \quad y_1 + \varepsilon y_2$$

$$\text{s.t.} \begin{bmatrix} \varepsilon & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1+x_{12} \end{bmatrix} \succeq 0 \qquad \text{s.t.} \; Z = \begin{bmatrix} -y_2 & \frac{1+y_1}{2} & -y_3 \\ \frac{1+y_1}{2} & 0 & -y_4 \\ -y_3 & -y_4 & -y_1 \end{bmatrix} \succeq 0$$

Entsprechende Matrizen:

$$C = \begin{bmatrix} 0 & \frac{1}{2} & 0 \\ \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

$$A_1 = \begin{bmatrix} 0 & -\frac{1}{2} & 0 \\ -\frac{1}{2} & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

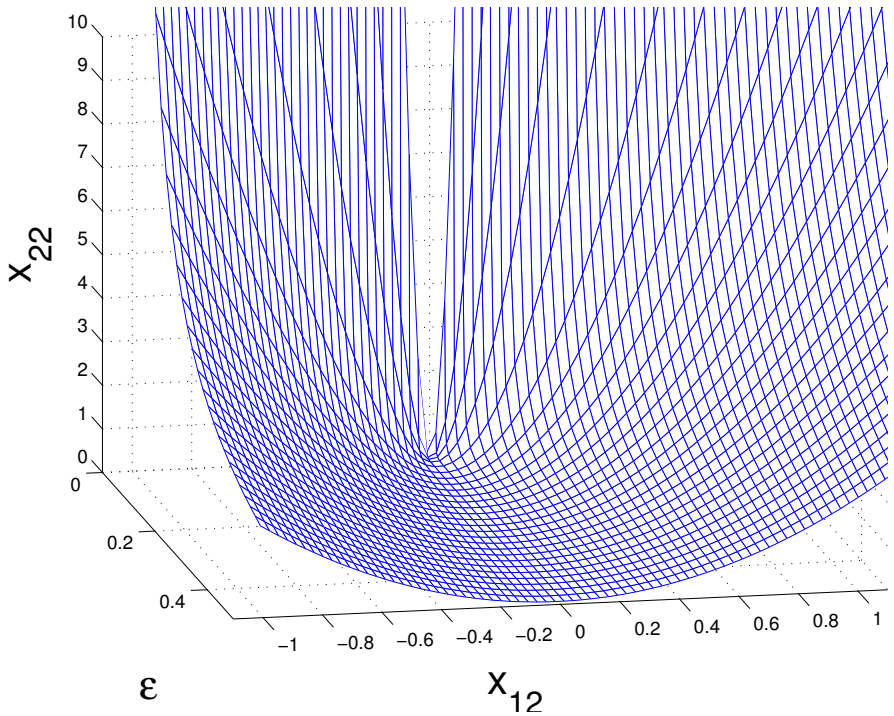
$$A_3 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{aligned} \max \quad & \langle C, X \rangle \\ & \langle A_1, X \rangle = 1 \\ & \langle A_2, X \rangle = \varepsilon \\ & \langle A_3, X \rangle = 0 \\ & \langle A_4, X \rangle = 0 \\ & X \succeq 0 \end{aligned}$$

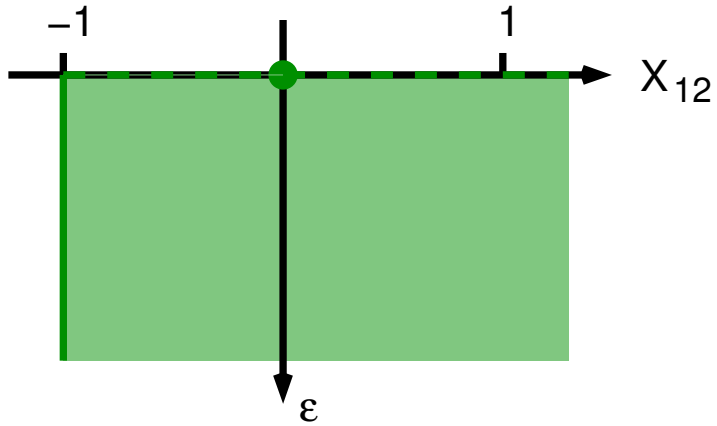
$x_{33} \geq 0 \;\Rightarrow\; x_{12} \geq -1,\; x_{22} \geq \frac{x_{12}^2}{\varepsilon}$, primale Optimallösung ist -1.

$z_{22} = 0 \;\Rightarrow\; \frac{1+y_1}{2} = 0,\; y_2 = 0,$ duale Optimallösung ist -1.

$$\begin{bmatrix} \varepsilon & x_{12} & 0 \\ x_{12} & x_{22} & 0 \\ 0 & 0 & 1+x_{12} \end{bmatrix} \succeq 0 \quad \Leftrightarrow \quad \begin{cases} \varepsilon > 0 & x_{12} \geq -1 & x_{22} \geq \frac{x_{12}^2}{\varepsilon} \\ \varepsilon = 0 & x_{12} = 0 & x_{22} \geq 0 \end{cases}$$



Projektion auf (ε, x_{12})-Ebene:



Für $\varepsilon > 0$ ist $x_{12} \in [-1, -\infty)$, für $\varepsilon = 0$ nur noch $x_{12} \in \{0\}$ zulässig!

Mathem. Grund: die Menge $\left\{ \begin{bmatrix} \langle C, X \rangle \\ \mathcal{A}X \end{bmatrix} : X \succeq 0 \right\}$ ist nicht abgeschlossen.

SDP und Eigenwertoptimierung

Für $A \in S^n$ bezeichne $\lambda_{\min}(A) := \lambda_1(A) \leq \cdots \leq \lambda_n(A) =: \lambda_{\max}(A)$. Es gilt $\lambda_i(A + y_0 I) = \lambda_i(A) + y_0$ für $i = 1, \ldots, n$ und $y_0 \in \mathbb{R}$.

SDP und Eigenwertoptimierung

Für $A \in S^n$ bezeichne $\lambda_{\min}(A) := \lambda_1(A) \leq \cdots \leq \lambda_n(A) =: \lambda_{\max}(A)$. Es gilt $\lambda_i(A + y_0 I) = \lambda_i(A) + y_0$ für $i = 1, \ldots, n$ und $y_0 \in \mathbb{R}$.

In der Optimalsteuerung ist die Stabilität eines Systems gewährleistet, wenn man für die von Steuerparametern $y \in \mathbb{R}^m$ abhängige Systemmatrix $A(y)$ nachweisen kann, dass $\lambda_{\max}(A(y)) < 0$.

Ist $A(y)$ affin, etwa $A(y) := C - \sum_{i=1}^m y_i A_i$ mit $C, A_i \in S^n$, führt das auf

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y)$$

SDP und Eigenwertoptimierung

Für $A \in S^n$ bezeichne $\lambda_{\min}(A) := \lambda_1(A) \leq \cdots \leq \lambda_n(A) =: \lambda_{\max}(A)$. Es gilt $\lambda_i(A + y_0 I) = \lambda_i(A) + y_0$ für $i = 1, \ldots, n$ und $y_0 \in \mathbb{R}$.

In der Optimalsteuerung ist die Stabilität eines Systems gewährleistet, wenn man für die von Steuerparametern $y \in \mathbb{R}^m$ abhängige Systemmatrix $A(y)$ nachweisen kann, dass $\lambda_{\max}(A(y)) < 0$.

Ist $A(y)$ affin, etwa $A(y) := C - \sum_{i=1}^m y_i A_i$ mit $C, A_i \in S^n$, führt das auf

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y)$$

Zur Modellierung als SDP: $\lambda_{\max}(A) = -\lambda_{\min}(-A)$ und

$$y_0 \geq \lambda_{\max}(C - \mathcal{A}^T y) \;\Leftrightarrow\; y_0 + \lambda_{\min}(\mathcal{A}^T y - C) \geq 0 \;\Leftrightarrow\; \lambda_{\min}(y_0 I + \mathcal{A}^T y - C) \geq 0$$

Wegen $Z \succeq 0 \;\Leftrightarrow\; \lambda_{\min}(Z) \geq 0$ gilt daher

$$\min_{y \in \mathbb{R}^m} \lambda_{\max}(C - \mathcal{A}^T y) \quad \Leftrightarrow \quad \begin{array}{ll} \min & y_0 \\ \text{s.t.} & Z = y_0 I + \mathcal{A}^T y - C \\ & y \in \mathbb{R}^m, Z \succeq 0 \end{array}$$

Lineare Matrix Ungleichungen (LMI)

Eine Bedingung der Form

$$y_1 A_1 + y_2 A_2 + \cdots + y_m A_m \preceq C$$

mit $A_i, C \in S^n$ heißt lineare Matrix Ungleichung (Linear Matrix Inequality).

Zulässige $y \in \mathbb{R}^m$ sind SDP-darstellbar, $\{y \in \mathbb{R}^m : \mathcal{A}^T y + Z = C, Z \succeq 0\}$.

Lineare Matrix Ungleichungen (LMI)

Eine Bedingung der Form

$$y_1 A_1 + y_2 A_2 + \cdots + y_m A_m \preceq C$$

mit $A_i, C \in S^n$ heißt lineare Matrix Ungleichung (Linear Matrix Inequality).

Zulässige $y \in \mathbb{R}^m$ sind SDP-darstellbar, $\{y \in \mathbb{R}^m : \mathcal{A}^T y + Z = C, Z \succeq 0\}$.

Bsp: Die Lyapunov Ungleichung fordert für festes $P = [p_1, \ldots, p_n] \in \mathbb{R}^{n \times n}$

$$P^T X + X P \prec 0, \quad X \succ 0.$$

In LMI-Darstellung wäre $y = [x_{11}, x_{12}, \ldots, x_{1n}, x_{22}, x_{23}, \ldots, x_{nn}]^T$, aber es wäre umständlich/sinnlos, die A_i für diese Ungleichungen anzugeben, die Struktur lässt sich im SDP besser direkt nutzen.

Lineare Matrix Ungleichungen (LMI)

Eine Bedingung der Form

$$y_1 A_1 + y_2 A_2 + \cdots + y_m A_m \preceq C$$

mit $A_i, C \in S^n$ heißt lineare Matrix Ungleichung (Linear Matrix Inequality).

Zulässige $y \in \mathbb{R}^m$ sind SDP-darstellbar, $\{y \in \mathbb{R}^m : \mathcal{A}^T y + Z = C, Z \succeq 0\}$.

Bsp: Die Lyapunov Ungleichung fordert für festes $P = [p_1, \ldots, p_n] \in \mathbb{R}^{n \times n}$

$$P^T X + XP \prec 0, \quad X \succ 0.$$

In LMI-Darstellung wäre $y = [x_{11}, x_{12}, \ldots, x_{1n}, x_{22}, x_{23}, \ldots, x_{nn}]^T$, aber es wäre umständlich/sinnlos, die A_i für diese Ungleichungen anzugeben, die Struktur lässt sich im SDP besser direkt nutzen.

Um LMIs zu erkennen, reicht es festzustellen, dass die Matrizen linear von den jeweiligen Variablen abhängen:

Die Matrix-Multiplikation $P^T X$ (bzw . XP) ist linear in X.

Wie erzwingt man positive Definitheit?

Die semidefinite Barrierefunktion $-\log\det X$

Wegen $\det X = \prod_{k=1}^{n} \lambda_i(X)$ ist

$$-\log\det X \;=\; -\log\prod_{k=1}^{n}\lambda_i(X) \;=\; -\sum_{k=1}^{n}\log\lambda_i(X)$$

eine Barrierefunktion für $X \succeq 0$ ($\Leftrightarrow \lambda_i(X) \geq 0$) und nur für $X \succ 0$ ($\Leftrightarrow \lambda_i(X) > 0$) erklärt. [vergl. Barriere $-\sum \log x_i$ für $x \geq 0$ in LP]

Innere-Punkte-Verfahren für SDP nutzen diese Barrierefunktion. [wie LP!]

Die semidefinite Barrierefunktion $-\log\det X$

Wegen $\det X = \prod_{k=1}^{n} \lambda_i(X)$ ist

$$-\log\det X \;=\; -\log\prod_{k=1}^{n}\lambda_i(X) \;=\; -\sum_{k=1}^{n}\log\lambda_i(X)$$

eine Barrierefunktion für $X \succeq 0$ ($\Leftrightarrow \lambda_i(X) \geq 0$) und nur für $X \succ 0$ ($\Leftrightarrow \lambda_i(X) > 0$) erklärt. [vergl. Barriere $-\sum \log x_i$ für $x \geq 0$ in LP]

Innere-Punkte-Verfahren für SDP nutzen diese Barrierefunktion. [wie LP!]

$-\log\det X$ wird auch genutzt, um streng zulässige Lösungen zu finden, z.B.

$$\begin{array}{ll} \min & -\log\det X - \log\det Z \\ \text{s.t.} & Z = P^T X + PX \\ & X \succ 0, Z \succ 0 \end{array}$$

$\rightarrow$ Innere-Punkte-Verfahren für konstanten Barriereparameter $\mu = 1$.

Das Maximieren der Determinante hat zahlreiche Anwendungen, insbesondere im Entwurf von Experimenten (*experiment design*)

Anwendungen der Semidefiniten Optimierung

- Optimalsteuerung und Kontrolltheorie
- Signalverarbeitung
- Kombinatorische Optimierung
- Globale Optimierung über Polynomen
- Robuster Entwurf von Stabkonstruktionen (truss topology design)
- Entwurf von Materialien (free material design)
- Robuste Optimierung
- Momenten-Probleme in der Wahrscheinlichkeitstheorie
- Entwurf von Experimenten in der Statistik
- Eigenwert-Optimierung
- Optimierung (trust-region Bestimmung, quadratische Relaxationen)

4.4.1 Robuste Stabilität dynamischer Systeme

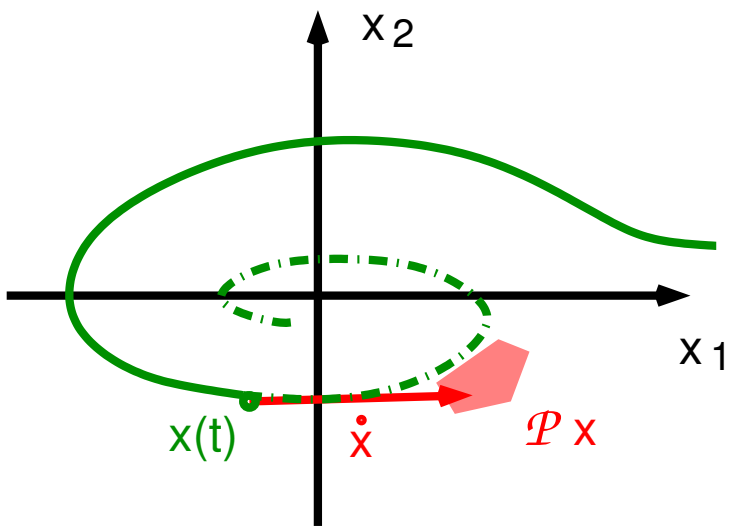
In einem (homogenen linearen) dynamischen System mit unsicheren Daten,

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n\times n},$$

beschreibt
- $x(t)$ . . . Zustand des Systems zur Zeit t.
- $\dot{x} := \frac{d}{dt}x(t)$ . . . (infinitesimale) Veränderung von $x(\cdot)$
- $P(t)$ . . . unsichere Übergangsmatrix zur Zeit t.

(DS) heißt stabil, wenn $x(t) \to 0$ für $t \to \infty$ und beliebige $P(t) \in \mathcal{P}$.

[In der Regelungstechnik würde $\mathcal{P}$ die möglichen Auswirkungen der Regelung/Steuerung umfassen. Man möchte wissen, ob diese auch bei unsauberer Realisierung in der Praxis den Zweck erfüllt.]



4.4.1 Robuste Stabilität dynamischer Systeme

In einem (homogenen linearen) dynamischen System mit unsicheren Daten,

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n\times n},$$

beschreibt
- $x(t)$ ... Zustand des Systems zur Zeit t.
- $\dot{x} := \frac{d}{dt}x(t)$ ... (infinitesimale) Veränderung von $x(\cdot)$
- $P(t)$ ... unsichere Übergangsmatrix zur Zeit t.

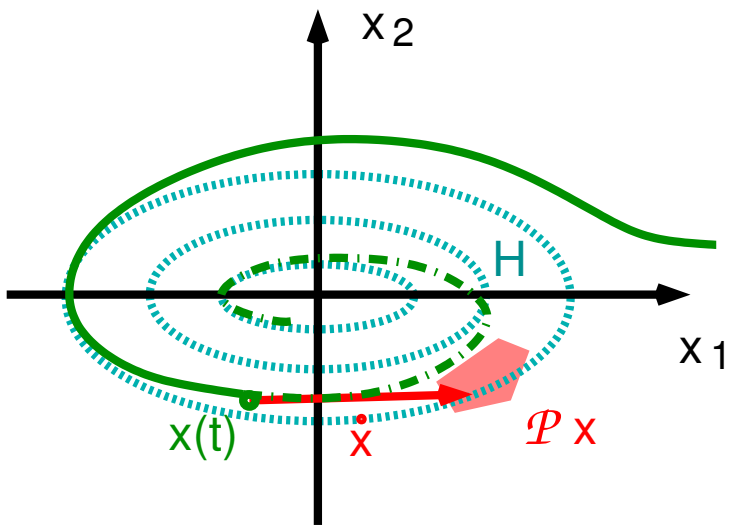
(DS) heißt stabil, wenn $x(t) \to 0$ für $t \to \infty$ und beliebige $P(t) \in \mathcal{P}$.

Hinreichend: Es gibt eine Norm

$$\|x\|_H := \sqrt{x^T H x} \quad \text{mit } H \succ 0$$

mit $\frac{d}{dt}\|x(t)\|_H^2 < 0$ auf allen Trajektorien

(das System heißt dann quadratisch stabil, $x^T Hx$ quadratische Lyapunov Funktion).



Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\dfrac{d}{dt}\|x(t)\|_H^2 < 0$.

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt}\|x(t)\|_H^2 < 0$.

$$\frac{d}{dt}\|x(t)\|_H^2 = \frac{d}{dt}x^T Hx = \dot{x}^T Hx + x^T H\dot{x} = x^T(P(t)^T H + HP(t))x$$

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt}\|x(t)\|_H^2 < 0$.

$$\frac{d}{dt}\|x(t)\|_H^2 = \frac{d}{dt}x^T Hx = \dot{x}^T Hx + x^T H\dot{x} = x^T(P(t)^T H + HP(t))x$$

Falls $A := P^T H + HP \prec 0$ (negativ definit), gilt $v^T Av < 0 \ \forall v \in \mathbb{R}^n \setminus \{0\}$.

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt}\|x(t)\|_H^2 < 0$.

$$\frac{d}{dt}\|x(t)\|_H^2 = \frac{d}{dt}x^T Hx = \dot{x}^T Hx + x^T H\dot{x} = x^T(P(t)^T H + HP(t))x$$

Falls $A := P^T H + HP \prec 0$ (negativ definit), gilt $v^T Av < 0 \; \forall v \in \mathbb{R}^n \setminus \{0\}$.
$\Rightarrow$ Das System ist quadratisch stabil, falls

$$H \succ 0, \qquad P_i^T H + HP_i \prec 0 \quad \text{für } i = 1, \ldots, k$$

zulässig lösbar ist, denn für so ein H erfüllt auch jede Konvexkombination $P \in \mathcal{P}$ die Bedingung $P^T H + HP \prec 0$.

Robuste Lyapunov Stabilität über SDP

$$(DS) \quad \dot{x} = P(t)x(t) \quad \text{mit } P(t) \in \mathcal{P} := \operatorname{conv}\{P_1, \ldots, P_k\} \subset \mathbb{R}^{n \times n}$$

Wir suchen $H \succ 0$ mit $\frac{d}{dt}\|x(t)\|_H^2 < 0$.

$$\frac{d}{dt}\|x(t)\|_H^2 = \frac{d}{dt}x^T Hx = \dot{x}^T Hx + x^T H\dot{x} = x^T(P(t)^T H + HP(t))x$$

Falls $A := P^T H + HP \prec 0$ (negativ definit), gilt $v^T Av < 0 \; \forall v \in \mathbb{R}^n \setminus \{0\}$.
$\Rightarrow$ Das System ist quadratisch stabil, falls

$$H \succ 0, \qquad P_i^T H + HP_i \prec 0 \quad \text{für } i = 1, \ldots, k$$

zulässig lösbar ist, denn für so ein H erfüllt auch jede Konvexkombination $P \in \mathcal{P}$ die Bedingung $P^T H + HP \prec 0$.
Suche H über Determinanten-Maximierung oder Eigenwert-Optimierung:

$$\max \; \lambda \; \text{s.t. } H \succeq \lambda I, \quad P_i^T H + HP_i \preceq -\lambda I \quad \text{für } i = 1, \ldots, k.$$

4.4.2 Entwurf von Experimenten

Um die Werte eines Parametervektors $\xi \in \mathbb{R}^p$ zu schätzen, stehen $\mathcal{R} = \{r_i \in \mathbb{R}^p : i = 1, \ldots, n\}$ mögliche Experimente zur Verfügung. Experiment i liefert pro Durchführung einen Messwert $r_i^T \xi + \rho_i$ mit unabhängig $(\mu = 0, \sigma^2 = 1)$-normalverteiltem Messfehler ρ_i.

4.4.2 Entwurf von Experimenten

Um die Werte eines Parametervektors $\xi \in \mathbb{R}^p$ zu schätzen, stehen $\mathcal{R} = \{r_i \in \mathbb{R}^p : i = 1, \ldots, n\}$ mögliche Experimente zur Verfügung. Experiment i liefert pro Durchführung einen Messwert $r_i^T \xi + \rho_i$ mit unabhängig $(\mu = 0, \sigma^2 = 1)$-normalverteiltem Messfehler ρ_i.

Werden m Experimente $a_j \in \mathcal{R}$ (Wiederholungen sind erlaubt) mit Ergebnissen $\eta_j = a_j^T \xi + \rho_j$ durchgeführt, ergibt der Maximum-Likelihood-Schätzer bei $\mathrm{Rang}[a_1, \ldots, a_m] = n$ ein geschätztes

$$\hat{\xi} = G \sum_{j=1}^{m} \eta_j a_j \qquad \text{mit} \quad G = \left(\sum_{j=1}^{m} a_j a_j^T \right)^{-1},$$

dessen Fehlerverteilung Erwartungswert 0 und Kovarianzmatrix G hat.

4.4.2 Entwurf von Experimenten

Um die Werte eines Parametervektors $\xi \in \mathbb{R}^p$ zu schätzen, stehen $\mathcal{R} = \{r_i \in \mathbb{R}^p : i = 1, \ldots, n\}$ mögliche Experimente zur Verfügung. Experiment i liefert pro Durchführung einen Messwert $r_i^T \xi + \rho_i$ mit unabhängig $(\mu = 0, \sigma^2 = 1)$-normalverteiltem Messfehler ρ_i.

Werden m Experimente $a_j \in \mathcal{R}$ (Wiederholungen sind erlaubt) mit Ergebnissen $\eta_j = a_j^T \xi + \rho_j$ durchgeführt, ergibt der Maximum-Likelihood-Schätzer bei $\mathrm{Rang}[a_1, \ldots, a_m] = n$ ein geschätztes

$$\hat{\xi} = G \sum_{j=1}^{m} \eta_j a_j \qquad \text{mit} \quad G = \left(\sum_{j=1}^{m} a_j a_j^T \right)^{-1},$$

dessen Fehlerverteilung Erwartungswert 0 und Kovarianzmatrix G hat.

Sind G und G' zwei Kovarianzmatrizen dieser Art und gilt $G \preceq G'$, dann ist die zu G gehörende Experimentfolge besser, weil die Varianz des Schätzfehlers kleiner ist.
$\rightarrow$ Finde die bzgl. $\preceq$ minimalen Elemente von

$$\left\{ G = \left(\sum_{i=1}^{n} m_i r_i r_i^T \right)^{-1} : m_i \in \mathbb{N}_0, \sum_i m_i = m \right\}.$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^{n} \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^{n} \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. $\preceq$ minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{\zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta\}.$$

D-**optimales Design**: Minimiere das Volumen des Konfidenzellipsoids.
E-**optimales Design**: Minimiere die längste Halbachse.
A-**optimales Design**: Minimiere die Summe der Halbachsen.

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^{n} \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. $\preceq$ minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{\zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta\}.$$

D-**optimales Design**: Minimiere das Volumen des Konfidenzellipsoids.
E-**optimales Design**: Minimiere die längste Halbachse.
A-**optimales Design**: Minimiere die Summe der Halbachsen.

D-**optimales Design.** Das Volumen ist zu $\det G = \prod \lambda_j(G)$ proportional. Wegen $\det(G^{-1}) = \det(G)^{-1} \Leftrightarrow$ maximiere die Determinante von G^{-1},

$$\begin{array}{ll} \min & -\log\det X \\ \text{s.t.} & X = \sum_{i=1}^{n} \alpha_i r_i r_i^T \\ & \mathbf{1}^T \alpha = 1 \\ & \alpha \geq 0, [X \succ 0] \end{array}$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^{n} \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. $\preceq$ minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{\zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta\}.$$

D-**optimales Design**: Minimiere das Volumen des Konfidenzellipsoids.
E-**optimales Design**: Minimiere die längste Halbachse.
A-**optimales Design**: Minimiere die Summe der Halbachsen.

E-**optimales Design.** Die längste Halbachse ist $\lambda_{\max}(G)$.
Wegen $\lambda_{\min}(G^{-1}) = \lambda_{\max}(G)^{-1} \Leftrightarrow$ maximiere $\lambda_{\min}(G^{-1})$,

$$\begin{array}{ll} \max & -\lambda \\ \text{s.t.} & \sum_{i=1}^{n} \alpha_i r_i r_i^T \succeq \lambda I \\ & \mathbf{1}^T \alpha = 1 \\ & \alpha \geq 0, \lambda \in \mathbb{R} \end{array}$$

Relaxationen

Statt m Experimente ganzzahlig zu wählen, bestimmt man relative Anteile,

$$\left\{ G = \left(\sum_{i=1}^{n} \alpha_i r_i r_i^T \right)^{-1} : \mathbf{1}^T \alpha = 1, \alpha \geq 0 \right\}.$$

Es gibt mehrere Ansätze, ein bzgl. $\preceq$ minimales G zu finden. Interpretiere G dazu als ein „Konfidenzellipsoid“ mit Halbachsenlängen $\lambda_j(G)$,

$$\mathcal{E} = \{\zeta : (\zeta - \hat{\xi})^T G^{-1} (\zeta - \hat{\xi}) \leq \beta\}.$$

D-**optimales Design**: Minimiere das Volumen des Konfidenzellipsoids.
E-**optimales Design**: Minimiere die längste Halbachse.
A-**optimales Design**: Minimiere die Summe der Halbachsen.

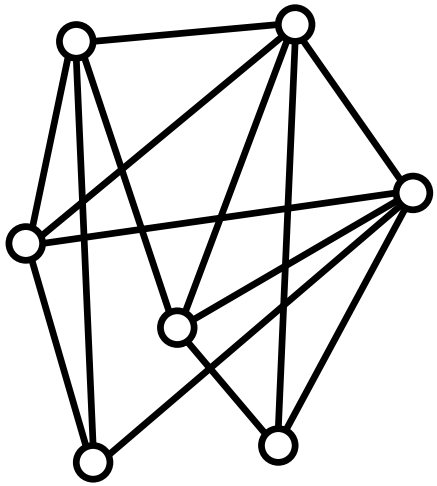
A-**optimales Design.** $\sum_{j=1}^{p} \lambda_j(G) = \sum_{j=1}^{p} G_{jj} = \sum_{j=1}^{p} e_j^T G e_j$.
Für jedes j ist die Unglg. $u_j \succeq e_j^T G e_j$ über Schur-Komplement darstellbar:

$$\begin{array}{ll} \min & \mathbf{1}^T u \\ \text{s.t.} & \begin{bmatrix} \sum_{i=1}^{n} \alpha_i r_i r_i^T & e_j \\ e_j^T & u_j \end{bmatrix} \succeq 0, \quad j = 1, \ldots, p \\ & \mathbf{1}^T \alpha = 1, \alpha \geq 0, u \in \mathbb{R}^p \end{array}$$

(4.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \ldots, n\}$,
$E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$



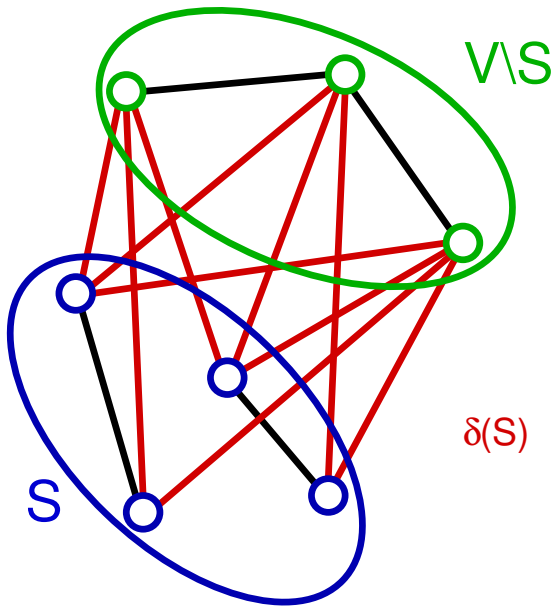
$$\text{(MC)} \qquad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \qquad [NP\text{-vollst.}]$$

(4.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \ldots, n\}$,
$E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$

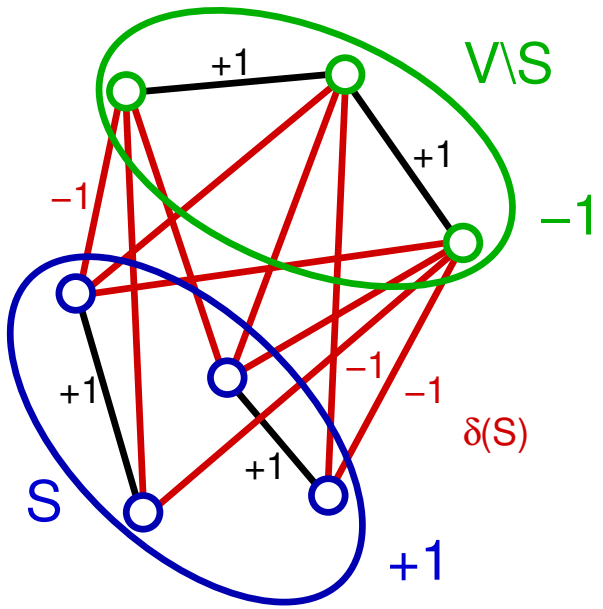
$$\text{(MC)} \qquad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \qquad [NP\text{-vollst.}]$$



(4.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \ldots, n\}$,
$E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$



$$\text{(MC)} \qquad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \qquad [NP\text{-vollst.}]$$

Modellierung: Repräsentiere die Partition durch

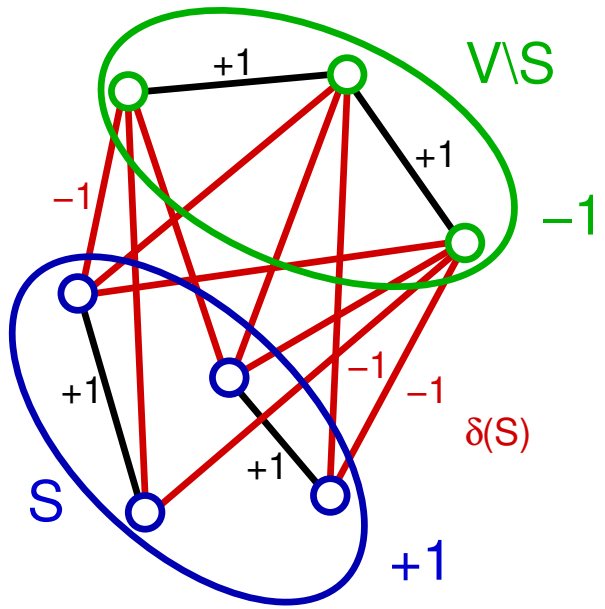
$$x \in \{-1, 1\}^n \quad \text{mit} \quad x_i = \begin{cases} 1 & i \in S \\ -1 & i \in V \setminus S \end{cases}$$

$$\text{Dann ist } x_i x_j = \begin{cases} -1 & ij \in \delta(S) \\ 1 & \text{sonst} \end{cases}, \text{ bzw. } \frac{1 - x_i x_j}{2} = \begin{cases} 1 & ij \in \delta(S) \\ 0 & \text{sonst.} \end{cases}$$

(4.4.3 Graphenpartition: Max-Cut)

Gegeben: Graph $G = (V, E)$, $V = \{1, \ldots, n\}$,
$E \subseteq \{ij : i, j \in V, i < j\}$, Kantengewichte a_{ij}

Gesucht: $S \subset V$ mit gewichtsmaximalem
Schnitt $\delta(S) := \{ij \in E : i \in S, j \in V \setminus S\}$



$$\text{(MC)} \qquad \max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} \qquad [NP\text{-vollst.}]$$

Modellierung: Repräsentiere die Partition durch

$$x \in \{-1, 1\}^n \quad \text{mit} \quad x_i = \begin{cases} 1 & i \in S \\ -1 & i \in V \setminus S \end{cases}$$

$$\text{Dann ist } x_i x_j = \begin{cases} -1 & ij \in \delta(S) \\ 1 & \text{sonst} \end{cases}, \text{ bzw. } \frac{1 - x_i x_j}{2} = \begin{cases} 1 & ij \in \delta(S) \\ 0 & \text{sonst.} \end{cases}$$

$$\max_{S \subseteq V} \sum_{ij \in \delta(S)} a_{ij} = \max_{x \in \{-1,1\}^n} \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2} \quad \rightarrow \quad \max_{x \in \{-1,1\}^n} x^T C x$$

$[C \in S^n$: $C_{ii} = \frac{1}{4} \sum_{j:ij \in E} a_{ij}$ (für $i \in V$), $C_{ij} = -\frac{1}{4} a_{ij}$ (für $ij \in E$), 0 sonst$]$

Äquivalent zu quadratischer 0-1 Optimierung!

Semidefinite Max-Cut Relaxation

Beachte: $x^T Cx = \langle Cx, x\rangle = \langle C, xx^T\rangle$

Eigenschaften von $xx^T = [x_i x_j]$ für $x \in \{-1, 1\}^n$:

- $x_i^2 = 1 \quad \Rightarrow \quad \operatorname{diag}(xx^T) = \mathbf{1}$
- xx^T ist positiv semidefinit, $xx^T \succeq 0$
- $\operatorname{Rang}(xx^T) = 1$

Semidefinite Max-Cut Relaxation

Beachte: $x^T Cx = \langle Cx, x\rangle = \langle C, xx^T\rangle$

Eigenschaften von $xx^T = [x_i x_j]$ für $x \in \{-1,1\}^n$:

- $x_i^2 = 1 \quad \Rightarrow \quad \mathrm{diag}(xx^T) = \mathbf{1}$
- xx^T ist positiv semidefinit, $xx^T \succeq 0$
- $\mathrm{Rang}(xx^T) = 1$

Relaxationsidee: Ersetze xx^T durch eine positiv semidefinite Matrix X.

$$\max_{x\in\{-1,1\}^n} x^T Cx \;\le\; \begin{array}{ll} \max & \langle C, X\rangle \\ \text{s.t.} & \mathrm{diag}(X) = \mathbf{1} \\ & X \succeq 0 \\ & [\mathrm{Rang}(X) = 1] \end{array}$$

[mit Rang 1 $\Leftrightarrow$ (MC), *NP*-vollst.]

Semidefinite Max-Cut Relaxation

Beachte: $x^T Cx = \langle Cx, x\rangle = \langle C, xx^T\rangle$

Eigenschaften von $xx^T = [x_i x_j]$ für $x \in \{-1,1\}^n$:

- $x_i^2 = 1 \quad \Rightarrow \quad \operatorname{diag}(xx^T) = \mathbf{1}$
- xx^T ist positiv semidefinit, $xx^T \succeq 0$
- $\operatorname{Rang}(xx^T) = 1$

Relaxationsidee: Ersetze xx^T durch eine positiv semidefinite Matrix X.

$$\max_{x\in\{-1,1\}^n} x^T Cx \;\le\; \begin{array}{ll} \max & \langle C, X\rangle \\ \text{s.t.} & \operatorname{diag}(X) = \mathbf{1} \\ & X \succeq 0 \\ & [\operatorname{Rang}(X) = 1] \end{array}$$

[mit Rang 1 $\Leftrightarrow$ (MC), *NP*-vollst.]

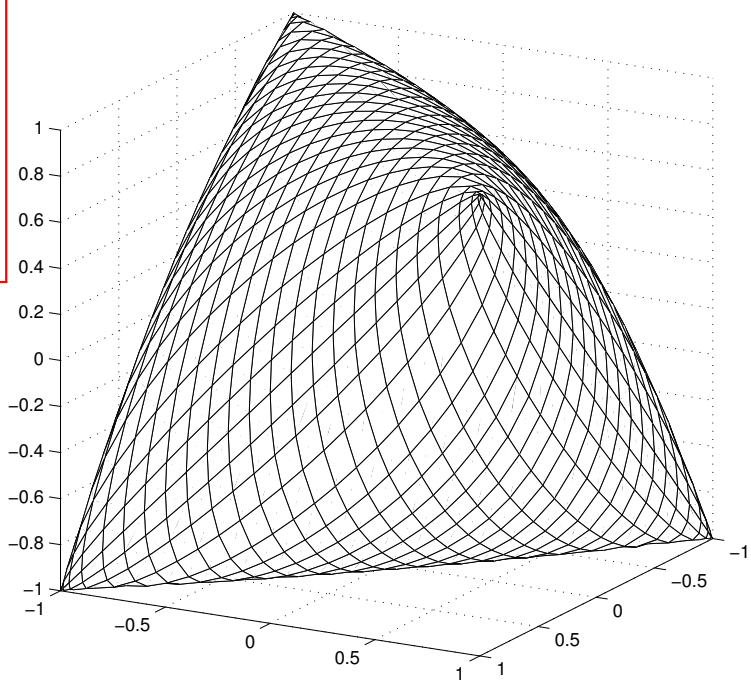


Illustration für $n = 3$:

Rand wird beschrieben von

$$\det \begin{bmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{bmatrix} = 0.$$

Geometrische Interpretation und Rundung

Interpretiere OL X^* als Gram-Matrix $X^* = R^T R$ von $R = [r_1, \ldots, r_n] \in \mathbb{R}^{k \times n}$,

$$X^*_{ij} = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2)$$

Geometrische Interpretation und Rundung

Interpretiere OL X^* als Gram-Matrix $X^* = R^T R$ von $R = [r_1, \ldots, r_n] \in \mathbb{R}^{k \times n}$,

$$X^*_{ij} = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\mathrm{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X^*_{ij}}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Geometrische Interpretation und Rundung

Interpretiere OL X^* als Gram-Matrix $X^* = R^T R$ von $R = [r_1, \ldots, r_n] \in \mathbb{R}^{k \times n}$,

$$X^*_{ij} = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\mathrm{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X^*_{ij}}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig?

Geometrische Interpretation und Rundung

Interpretiere OL X^* als Gram-Matrix $X^* = R^T R$ von $R = [r_1, \ldots, r_n] \in \mathbb{R}^{k \times n}$,

$$X^*_{ij} = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

$\|r_i\| = 1$ wegen $\mathrm{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

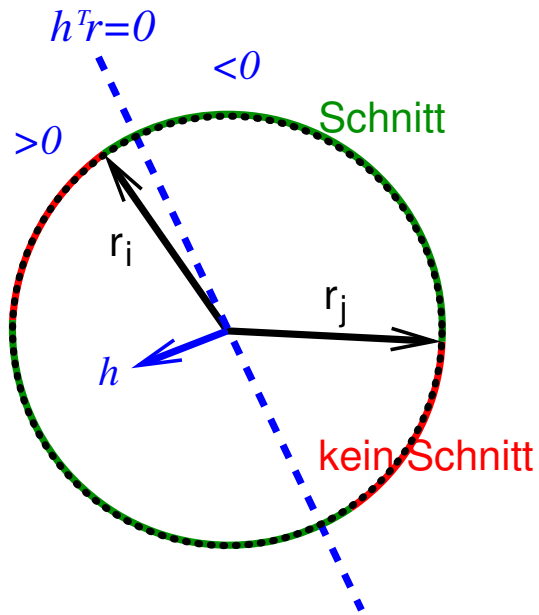
$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X^*_{ij}}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig? **„Zufälliges Runden mit Hyperebenen“**

Runde zu $x \in \{-1, 1\}^n$ über einen normalv. Zufallsvektor $h \in \mathbb{R}^k$ durch

$$x_i = \begin{cases} 1 & h^T r_i \geq 0 \\ -1 & \text{sonst} \end{cases} \quad \rightarrow \quad H = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$



Geometrische Interpretation und Rundung

Interpretiere OL X^* als Gram-Matrix $X^* = R^T R$ von $R = [r_1, \ldots, r_n] \in \mathbb{R}^{k \times n}$,

$$X^*_{ij} = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

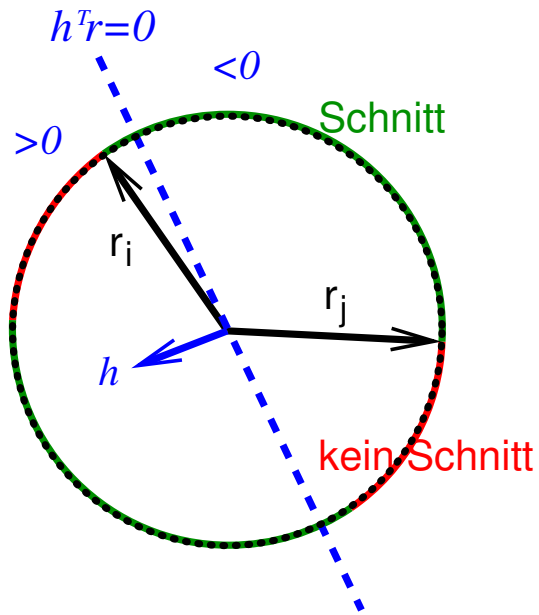
$\|r_i\| = 1$ wegen $\mathrm{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X^*_{ij}}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig? **„Zufälliges Runden mit Hyperebenen“**
Runde zu $x \in \{-1, 1\}^n$ über einen normalv. Zufallsvektor $h \in \mathbb{R}^k$ durch

$$x_i = \begin{cases} 1 & h^T r_i \geq 0 \\ -1 & \text{sonst} \end{cases} \quad \rightarrow \quad H = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$



$\mathbb{E}(H)$ errechnet sich pro $ij \in E$ durch Projektion von h auf die $\{r_i, r_j\}$-Ebene:

$$\mathbb{P}(x_i x_j = -1) = \arccos(r_i^T r_j)/\pi$$

Geometrische Interpretation und Rundung

Interpretiere OL X^* als Gram-Matrix $X^* = R^T R$ von $R = [r_1, \ldots, r_n] \in \mathbb{R}^{k \times n}$,

$$X^*_{ij} = r_i^T r_j = \|r_i\| \|r_j\| \cos \angle(r_1, r_2) \quad [= \cos \angle(r_1, r_2).]$$

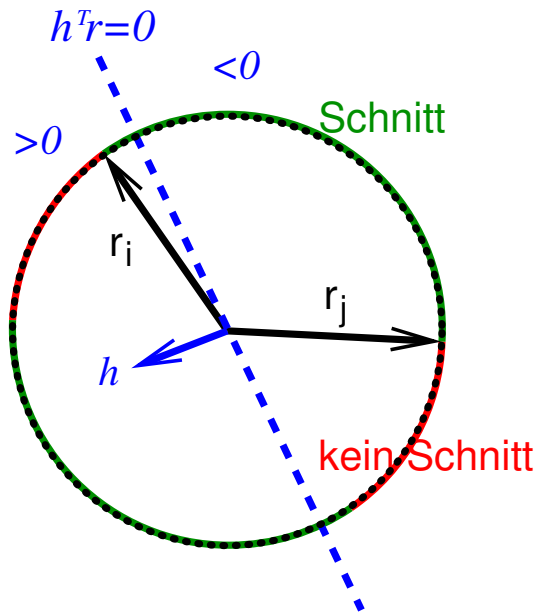
$\|r_i\| = 1$ wegen $\operatorname{diag}(X^*) = \mathbf{1}$. Knoten i entspricht Vektor/Punkt $r_i \in \mathbb{R}^k$,

$$\langle C, X^* \rangle = \sum_{ij \in E} a_{ij} \frac{1 - X^*_{ij}}{2} = \sum_{ij \in E} a_{ij} \frac{1 - r_i^T r_j}{2}$$

Ist $r_i^T r_j$ nahe bei -1 (großer Winkel $\angle(r_1, r_2)$), sollte man i und j trennen.

Geht das für alle gleichzeitig? **„Zufälliges Runden mit Hyperebenen“**
Runde zu $x \in \{-1, 1\}^n$ über einen normalv. Zufallsvektor $h \in \mathbb{R}^k$ durch

$$x_i = \begin{cases} 1 & h^T r_i \geq 0 \\ -1 & \text{sonst} \end{cases} \quad \rightarrow \quad H = \sum_{ij \in E} a_{ij} \frac{1 - x_i x_j}{2}$$



$\mathbb{E}(H)$ errechnet sich pro $ij \in E$ durch Projektion von h auf die $\{r_i, r_j\}$-Ebene:

$$\mathbb{P}(x_i x_j = -1) = \arccos(r_i^T r_j)/\pi$$

Es gilt $\arccos(t)/\pi \geq 0.878 \frac{1 - t}{2}$

und wenn $a_{ij} \geq 0 \Rightarrow \quad \mathbb{E}(H) \geq 0.878 \langle C, X_* \rangle$

„0.878 Approximationsalgorithmus“ von Goemans und Williamson

(4.4.4 SDP: Gram-Matrix und geometrische Einbettung)

Sind n Punkte $r_i \in \mathbb{R}^k$ gegeben und ist $R = [r_1, r_2, .., r_n]$, dann ist die Gram-Matrix $X = R^T R \succeq 0$ positiv semidefinit mit Rang k, und erfüllt

$$X_{ij} = r_i^T r_j,$$

$$\|r_i - r_j\|^2 = r_i^T r_i - 2 r_i^T r_j + r_j^T r_j = X_{ii} - 2X_{ij} + X_{jj} = \langle E_{ij}, X \rangle.$$

$E_{ij} \in \mathbb{R}^{n \times n}$, $E_{ii} = E_{jj} = 1$, $E_{ij} = E_{ji} = -1$, und 0 sonst: $E_{ij} = \begin{bmatrix} & & \\ & 1 & -1 \\ & -1 & 1 \end{bmatrix} \begin{matrix} \\ i \\ j \end{matrix}$

(4.4.4 SDP: Gram-Matrix und geometrische Einbettung)

Sind n Punkte $r_i \in \mathbb{R}^k$ gegeben und ist $R = [r_1, r_2, .., r_n]$, dann ist die Gram-Matrix $X = R^T R \succeq 0$ positiv semidefinit mit Rang k, und erfüllt

$$X_{ij} = r_i^T r_j,$$

$$\|r_i - r_j\|^2 \;=\; r_i^T r_i - 2 r_i^T r_j + r_j^T r_j \;=\; X_{ii} - 2X_{ij} + X_{jj} \;=\; \langle E_{ij}, X\rangle.$$

$E_{ij} \in \mathbb{R}^{n\times n}$, $E_{ii} = E_{jj} = 1$, $E_{ij} = E_{ji} = -1$, und 0 sonst: $E_{ij} = \begin{bmatrix} & i & & j \\ i & 1 & -1 & \\ & & & \\ j & -1 & 1 & \end{bmatrix}$

Ist umgekehrt die Lage der n Punkte nicht bekannt, sondern nur einige Distanzen $d_{ij} \in \mathbb{R}_+$ zwischen Paaren $ij \in E \subseteq \{ij : 1 \le i < j \le n\}$, sucht man eine niedrigdimensionale Einbettung der Punkte mit diesen Distanzen:

$$\begin{array}{ll} \min & \mathrm{Rang}(X) \\ \text{s.t.} & \langle E_{ij}, X\rangle = d_{ij} \quad ij \in E \\ & X \succeq 0 \end{array}$$

(4.4.4 SDP: Gram-Matrix und geometrische Einbettung)

Sind n Punkte $r_i \in \mathbb{R}^k$ gegeben und ist $R = [r_1, r_2, .., r_n]$, dann ist die Gram-Matrix $X = R^T R \succeq 0$ positiv semidefinit mit Rang k, und erfüllt

$$X_{ij} = r_i^T r_j,$$

$$\|r_i - r_j\|^2 \;=\; r_i^T r_i - 2 r_i^T r_j + r_j^T r_j \;=\; X_{ii} - 2X_{ij} + X_{jj} \;=\; \langle E_{ij}, X\rangle.$$

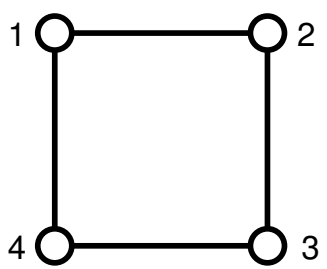
$E_{ij} \in \mathbb{R}^{n\times n}$, $E_{ii} = E_{jj} = 1$, $E_{ij} = E_{ji} = -1$, und 0 sonst: $E_{ij} = \begin{bmatrix} & i & & j \\ i & 1 & -1 & \\ & & & \\ j & -1 & 1 & \end{bmatrix}$

Ist umgekehrt die Lage der n Punkte nicht bekannt, sondern nur einige Distanzen $d_{ij} \in \mathbb{R}_+$ zwischen Paaren $ij \in E \subseteq \{ij : 1 \le i < j \le n\}$, sucht man eine niedrigdimensionale Einbettung der Punkte mit diesen Distanzen:

$$\begin{array}{ll} \min & \mathrm{Rang}(X) \\ \text{s.t.} & \langle E_{ij}, X\rangle = d_{ij} \quad ij \in E \\ & X \succeq 0 \end{array}$$

Zwei Schwierigkeiten:

- Das geht gar nicht für beliebige d_{ij}!
  Bsp: Kanten-Abstand im Kreis der Länge 4:
  $n = 4$, $d_{12} = d_{14} = d_{23} = d_{34} = 1$, $d_{13} = d_{24} = 2$
- Rang-Minimierung ist nicht konvex und *NP*-schwer.



Einbettung mit kleiner Verzerrung

Seien für n Punkte alle paarweisen Distanzen $d_{ij} > 0$ gegeben, und die Dreiecksungleichung $d_{ij} \leq d_{ih} + d_{hj}$ sei für alle i, j, h erfüllt. Eine Einbettung $\rho : \{1, \ldots, n\} \to \mathbb{R}^k$ hat Verzerrung (*distortion*) $D > 0$ falls

$$d_{ij} \leq \|\rho(i) - \rho(j)\| \leq D d_{ij} \qquad \forall 1 \leq i < j \leq n$$

Einbettung mit kleiner Verzerrung

Seien für n Punkte alle paarweisen Distanzen $d_{ij} > 0$ gegeben, und die Dreiecksungleichung $d_{ij} \leq d_{ih} + d_{hj}$ sei für alle i, j, h erfüllt. Eine Einbettung $\rho : \{1, \ldots, n\} \to \mathbb{R}^k$ hat Verzerrung (*distortion*) $D > 0$ falls

$$d_{ij} \leq \|\rho(i) - \rho(j)\| \leq D d_{ij} \qquad \forall 1 \leq i < j \leq n$$

Eine Einbettung geringster Verzerrung in $\mathbb{R}^n$ ist per SDP konstruierbar:

$$\begin{array}{lll} \min & \delta & \\ \text{s.t.} & d_{ij} \leq \langle E_{ij}, X \rangle \leq \delta d_{ij} & 1 \leq i < j \leq n \\ & X \succeq 0, \delta \in \mathbb{R} & [\delta = D^2] \end{array}$$

Faktorisiere X^* zum Beispiel über die Eigenwertzerlegung:

$$X^* = P \Lambda^* P^T = P (\Lambda^*)^{\frac{1}{2}} \underbrace{(\Lambda^*)^{\frac{1}{2}} P^T}_{=:R} = R^T R$$

Der Rang von R ist mit geringem Verlust in D auf $\sim \log n$ reduzierbar.

Inhaltsübersicht

5 Freie Nichtlineare Optimierung

Verfahren zur Minimierung glatter Funktionen ohne Nebenbedingungen,

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

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

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

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

In welcher Form soll f für die Verfahren zugänglich sein?

Inhaltsübersicht

5.1 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 $\longrightarrow$ „Orakel“.

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

5.1 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.

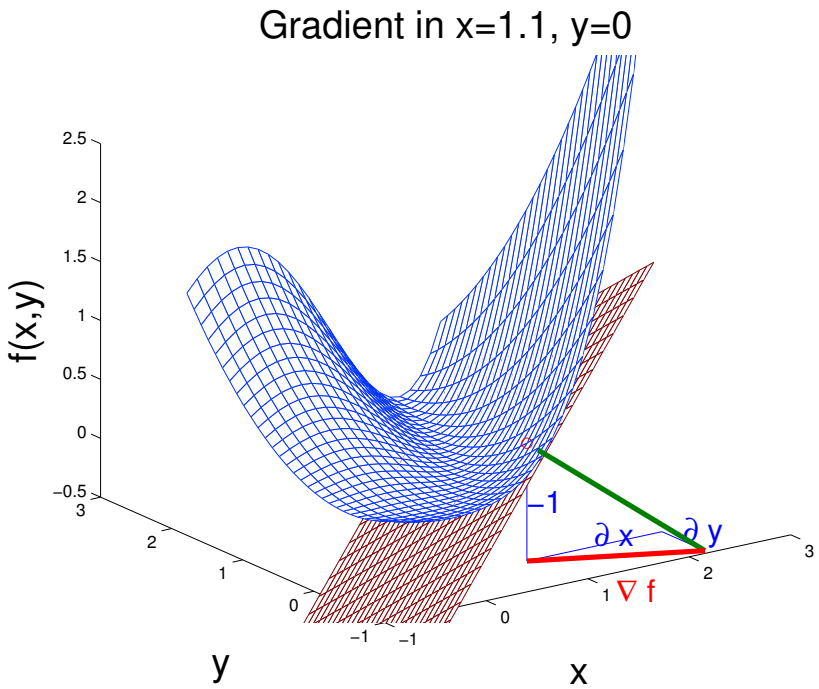
Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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 $\begin{bmatrix} x \\ f(x) \end{bmatrix}$.

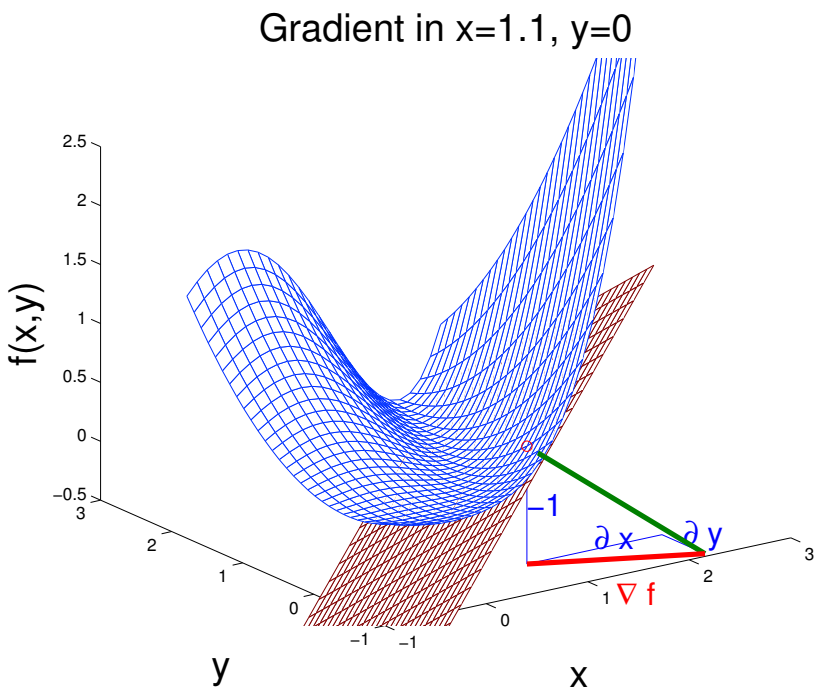


Orakel 1. Ordnung: $f(x)$, $\nabla f(x)$

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

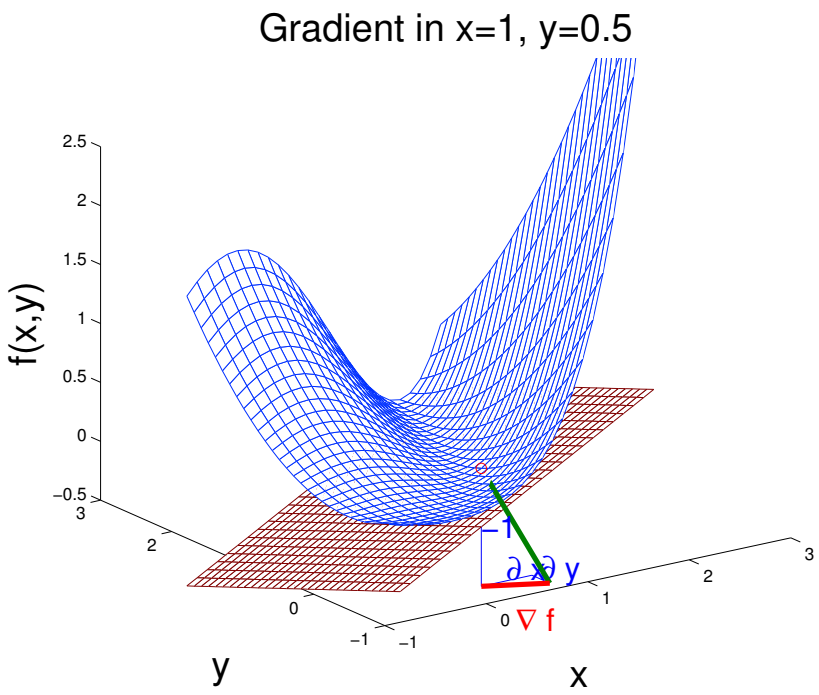
$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x})}{\|x - \bar{x}\|} = \lim_{h \to 0} \frac{f(\bar{x} + h) - f(\bar{x}) - \nabla f(\bar{x})^T h}{\|h\|} = 0.$$

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

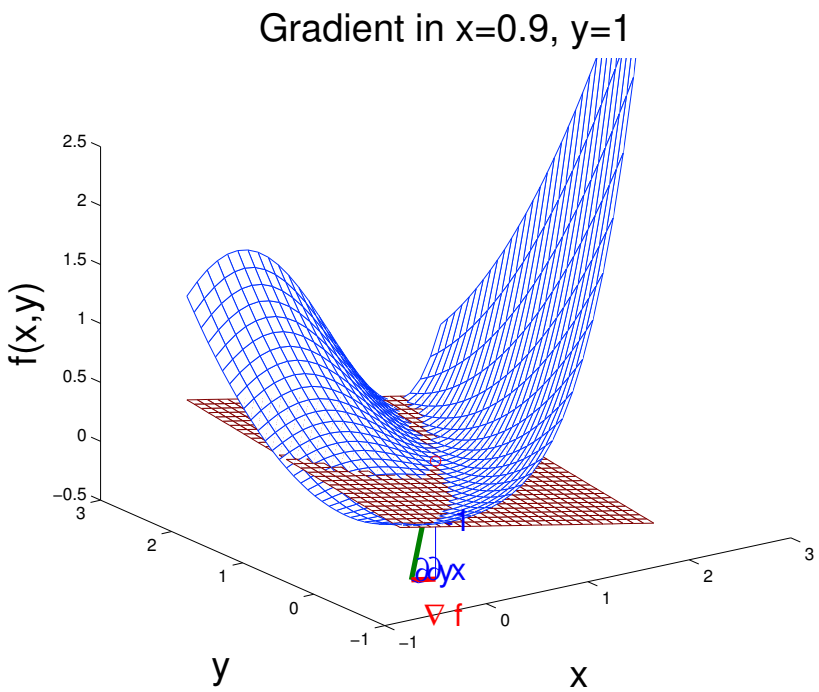
$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x})}{\|x - \bar{x}\|} = \lim_{h \to 0} \frac{f(\bar{x} + h) - f(\bar{x}) - \nabla f(\bar{x})^T h}{\|h\|} = 0.$$

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

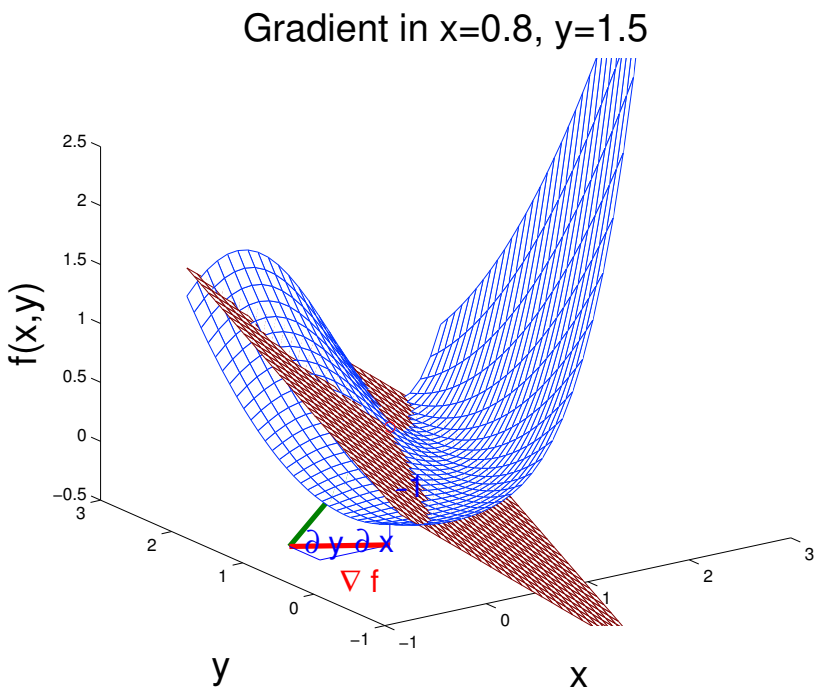
$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x})}{\|x - \bar{x}\|} = \lim_{h \to 0} \frac{f(\bar{x} + h) - f(\bar{x}) - \nabla f(\bar{x})^T h}{\|h\|} = 0.$$

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

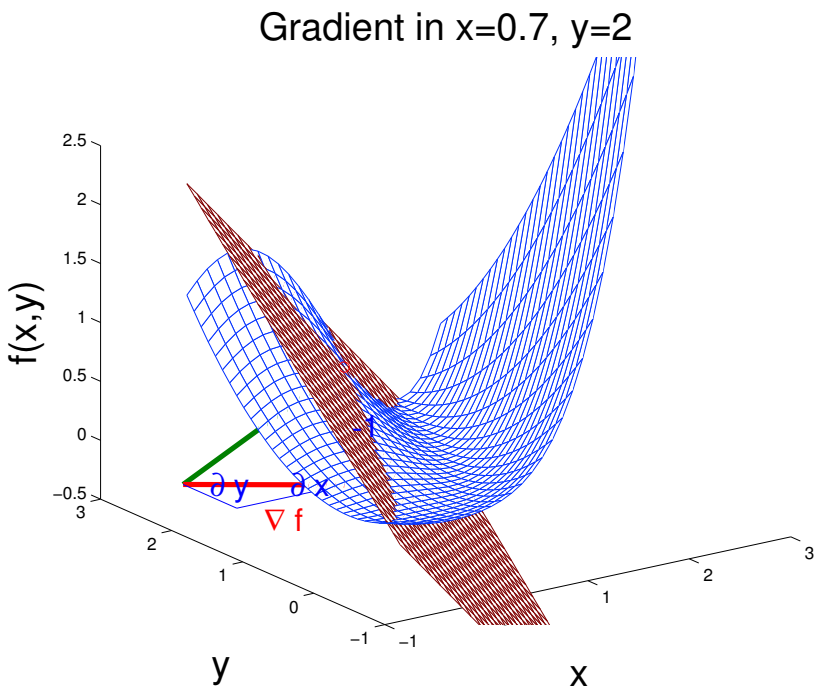
$$\lim_{x \to \bar{x}} \frac{f(x) - f(\bar{x}) - \nabla f(\bar{x})^T (x - \bar{x})}{\|x - \bar{x}\|} = \lim_{h \to 0} \frac{f(\bar{x} + h) - f(\bar{x}) - \nabla f(\bar{x})^T h}{\|h\|} = 0.$$

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

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

Gradient und Richtungsableitung, lineares Modell

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

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

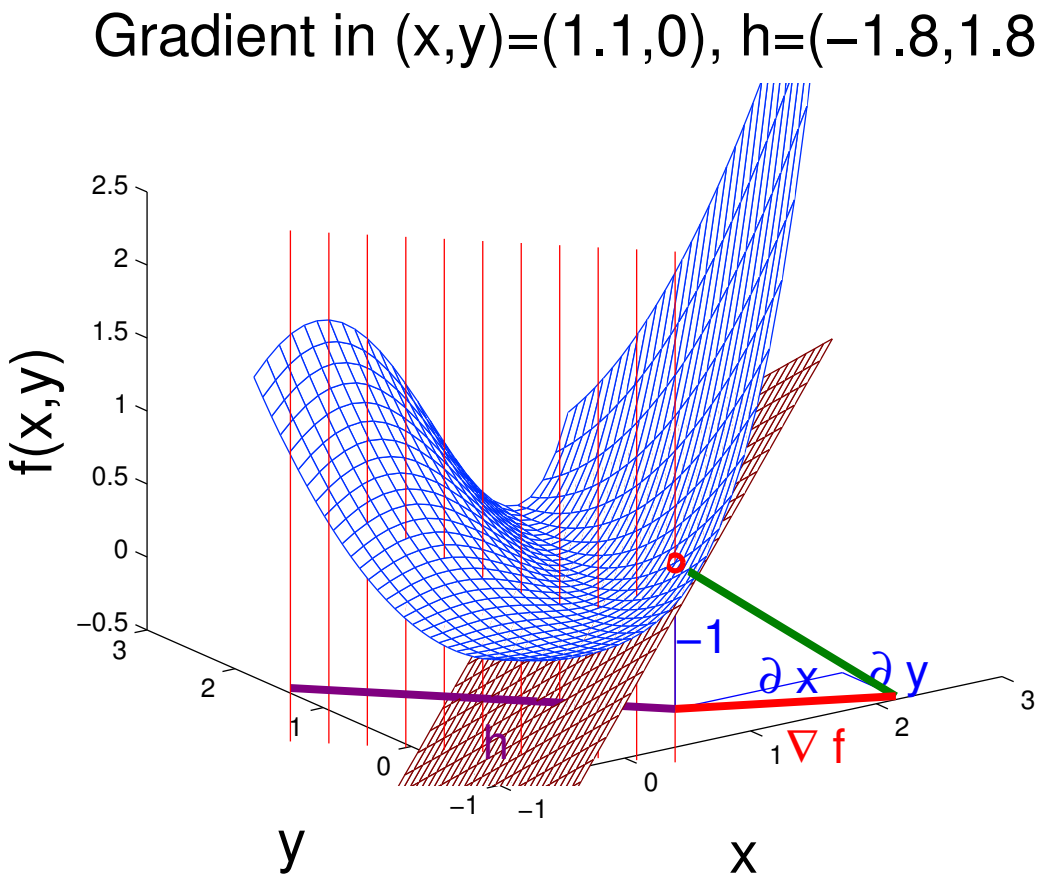
[= Ableitung $\frac{d}{d\alpha}\Phi(0)$ der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]

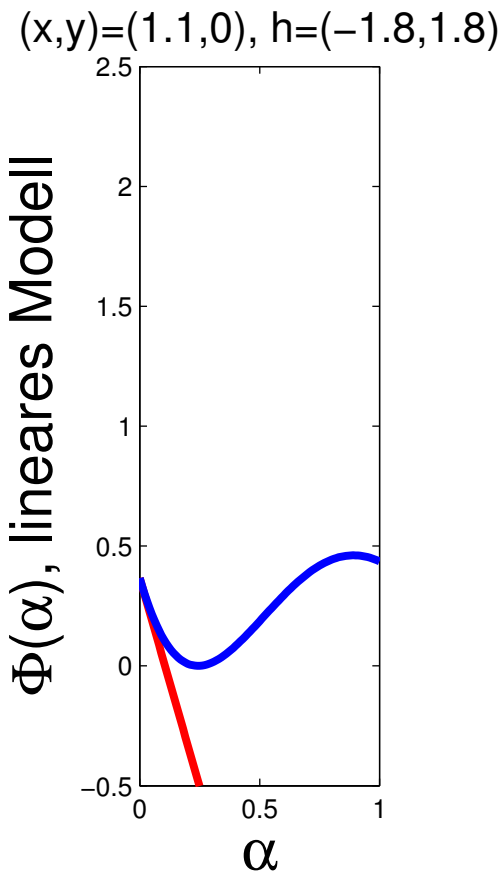
Gradient und Richtungsableitung, lineares Modell

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

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

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



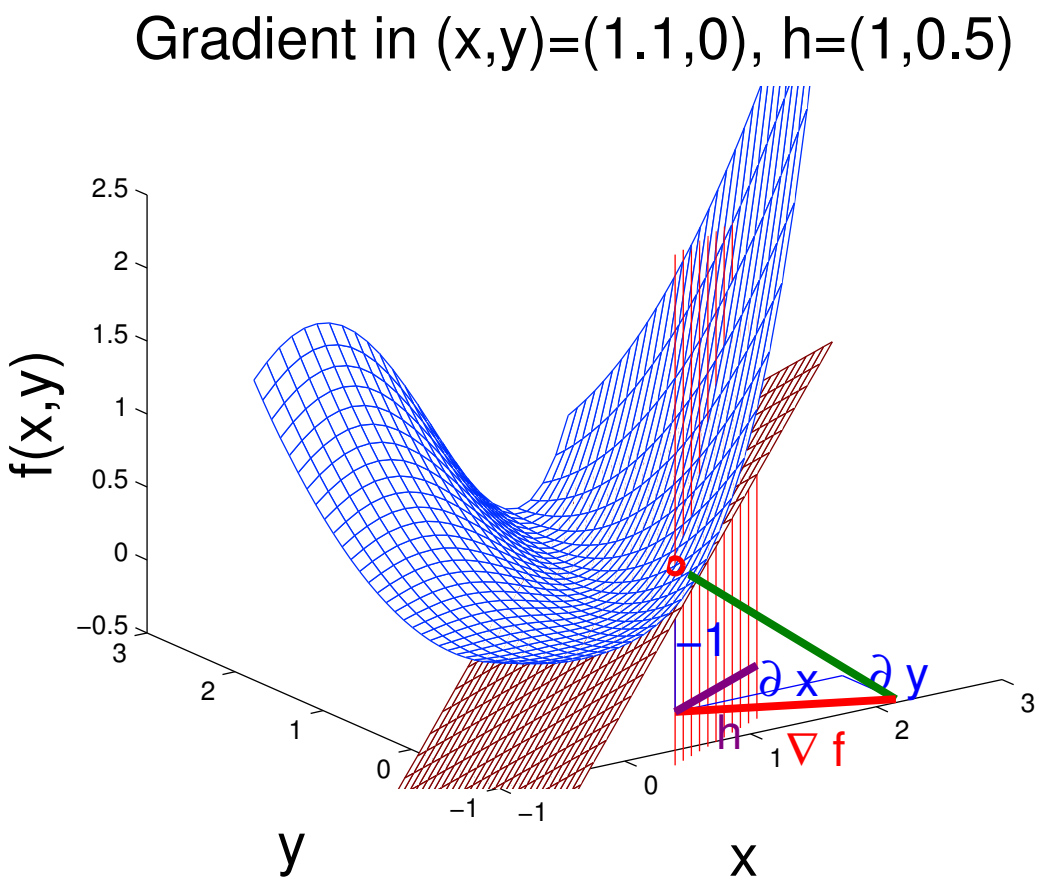


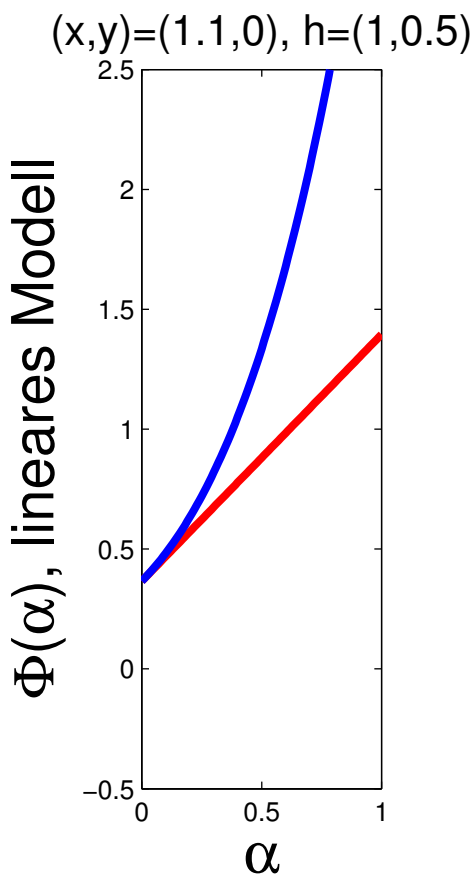
Gradient und Richtungsableitung, lineares Modell

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

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

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



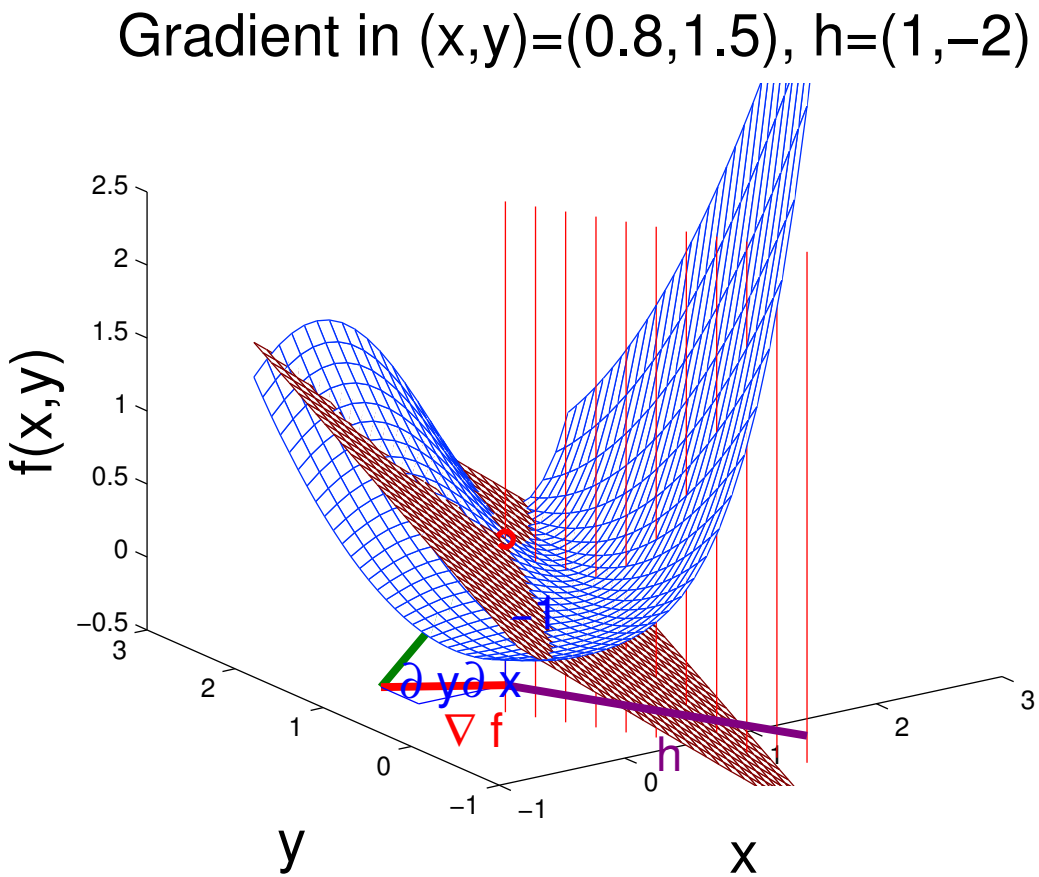


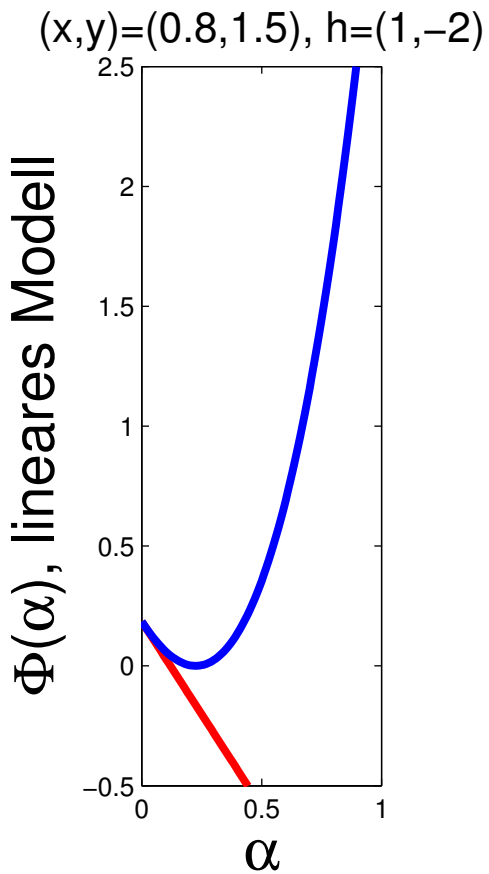
Gradient und Richtungsableitung, lineares Modell

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

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

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



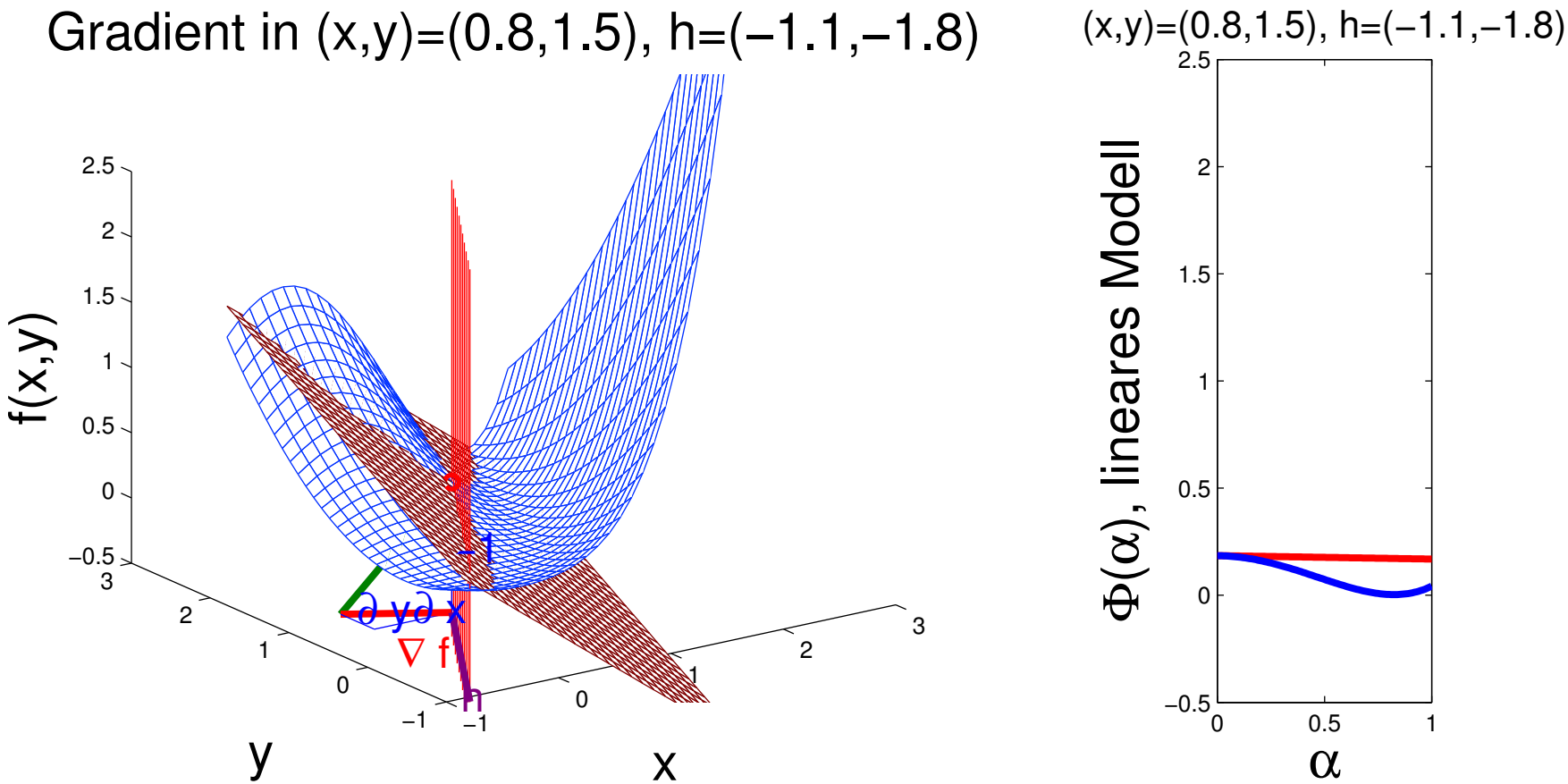


Gradient und Richtungsableitung, lineares Modell

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

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

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

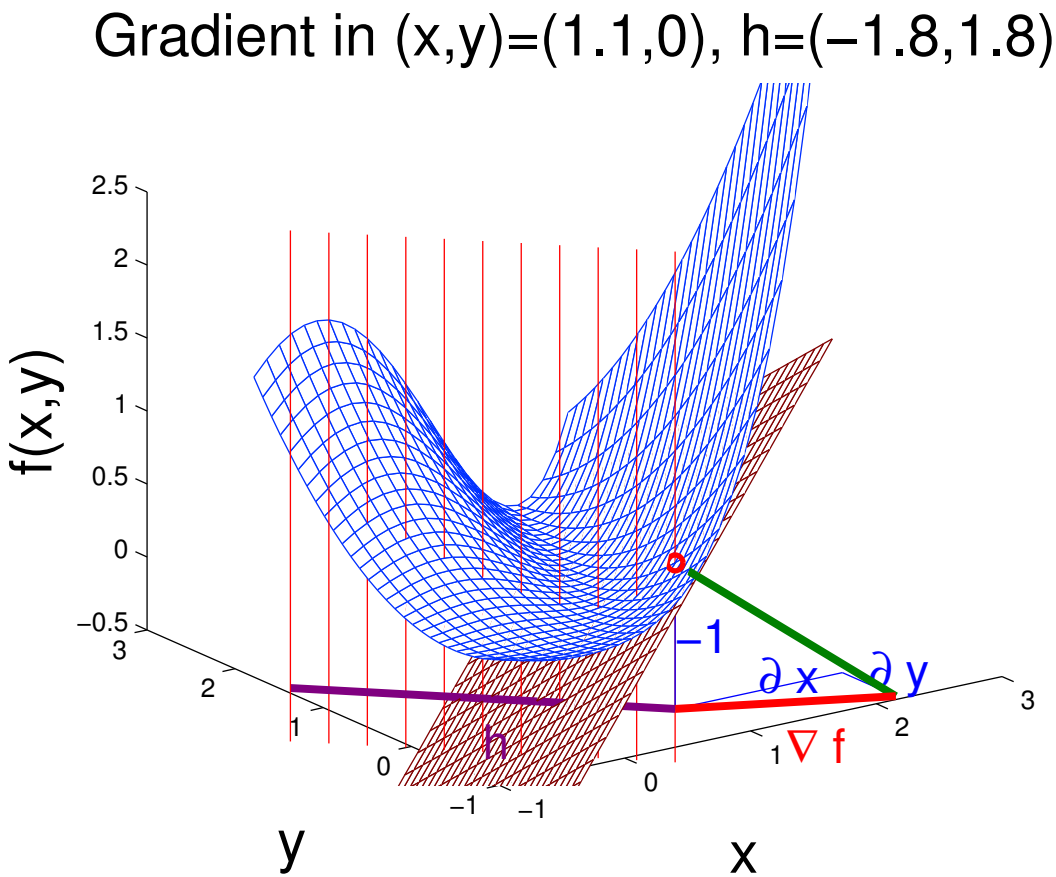


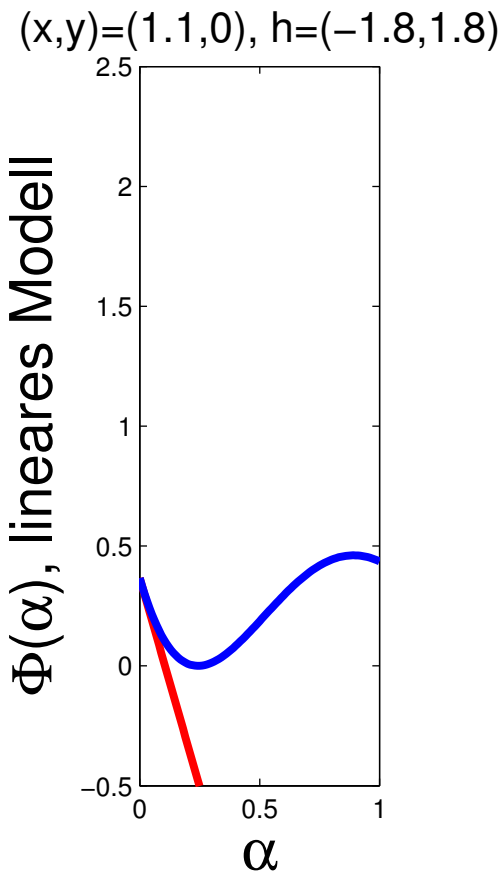
Gradient und Richtungsableitung, lineares Modell

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

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

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



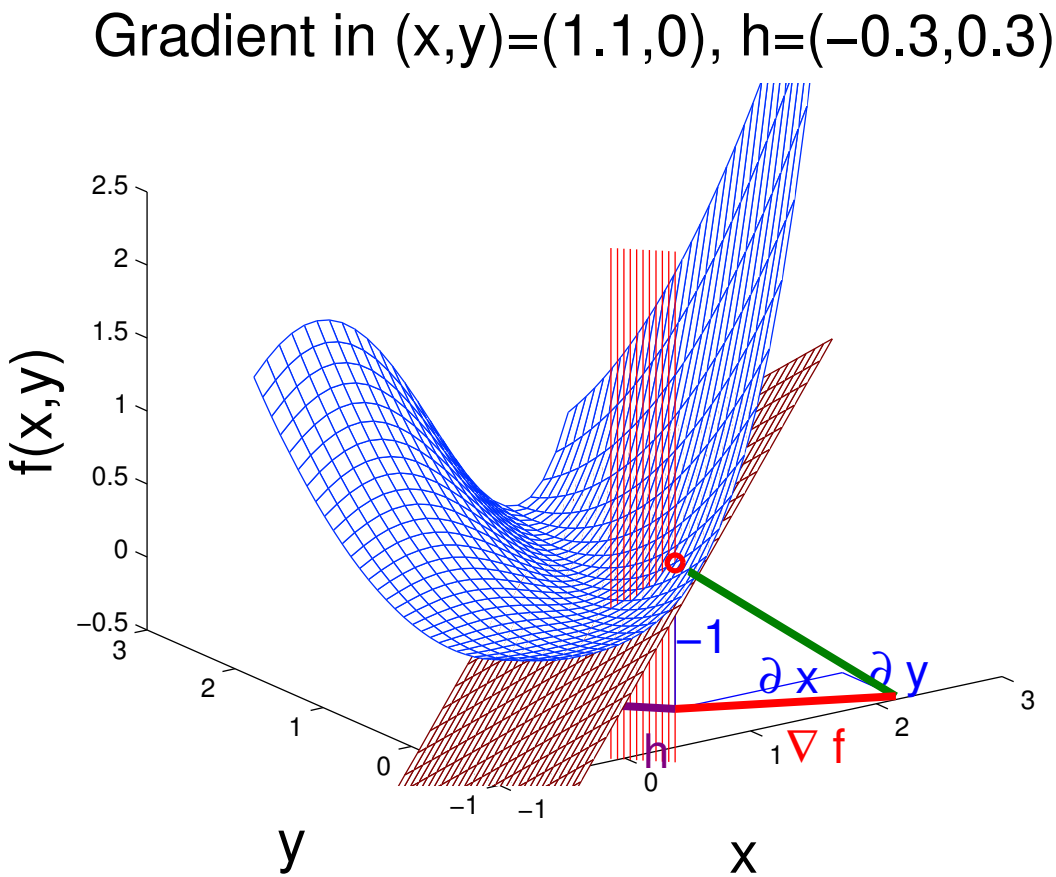


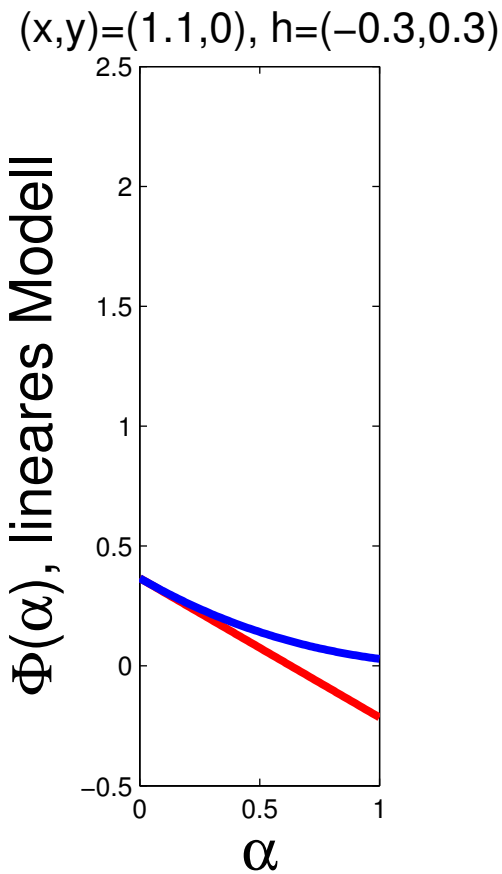
Gradient und Richtungsableitung, lineares Modell

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

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

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





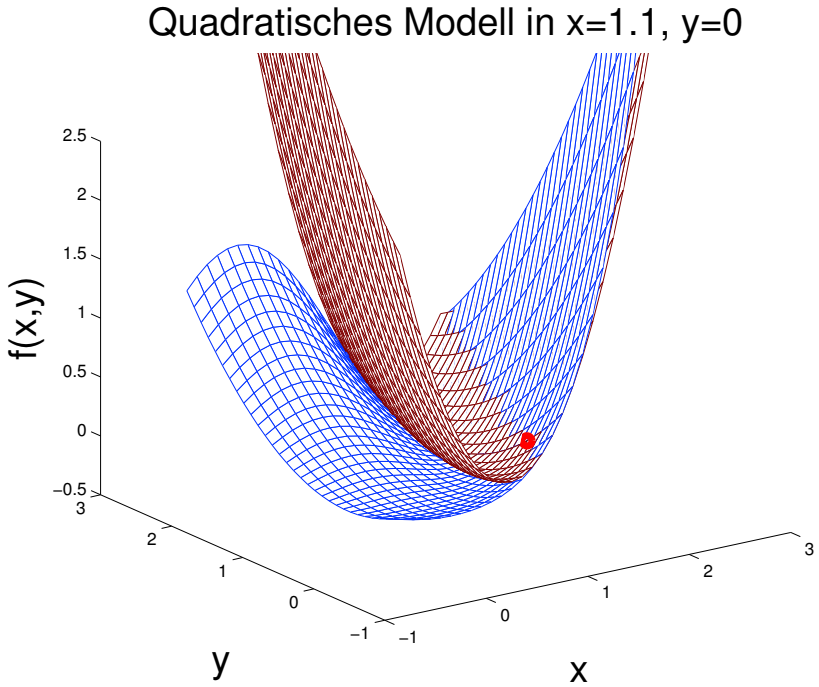
Orakel 2. Ordnung: $f(x)$, $\nabla f(x)$, $\nabla^2 f(x)$

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

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

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

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



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

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

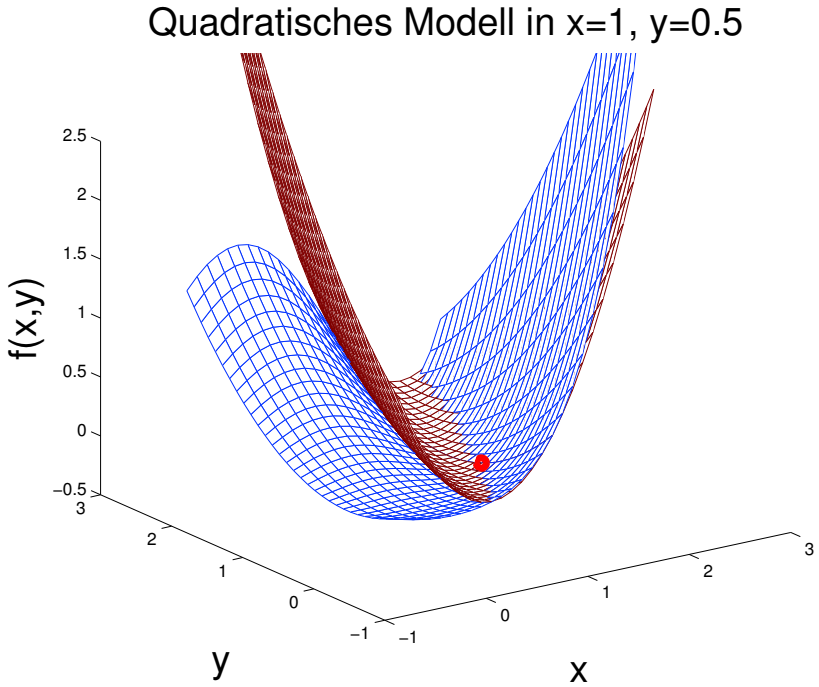
Für x nahe bei $\bar{x}$ ist es eine sehr gute Näherung an f: für glattes f erfüllt $\nabla^2 f$

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

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

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

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



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

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

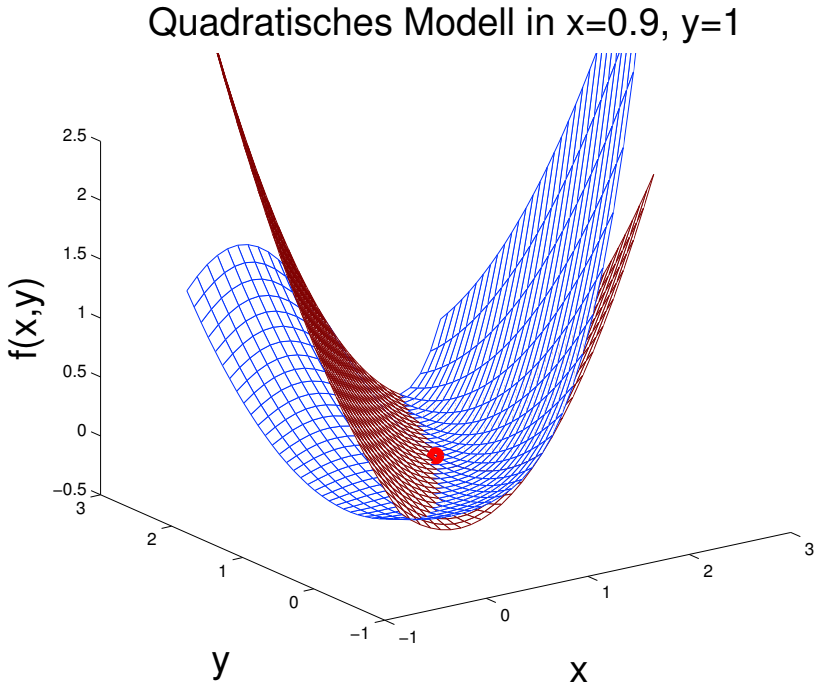
Für x nahe bei $\bar{x}$ ist es eine sehr gute Näherung an f: für glattes f erfüllt $\nabla^2 f$

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

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

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

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



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

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

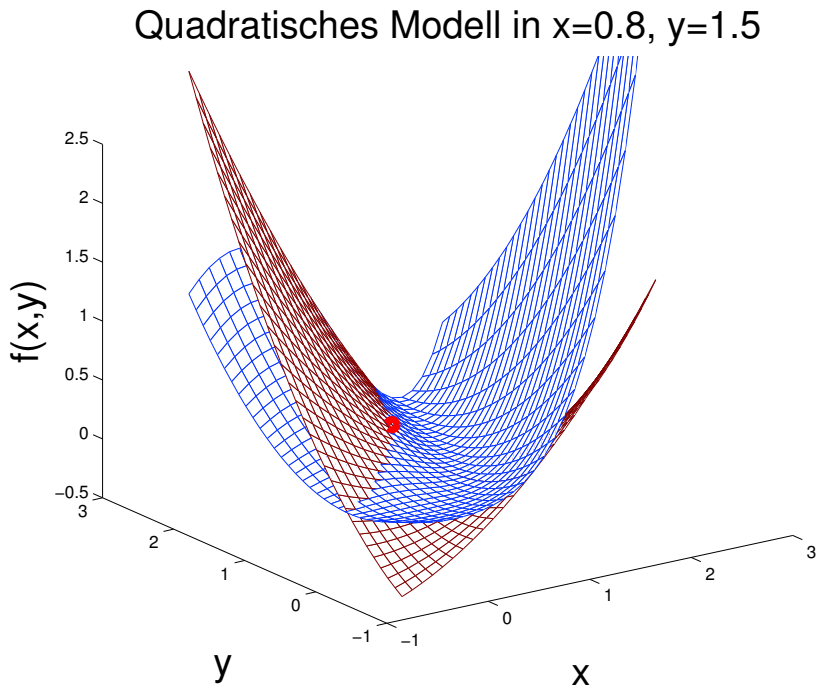
Für x nahe bei $\bar{x}$ ist es eine sehr gute Näherung an f: für glattes f erfüllt $\nabla^2 f$

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

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

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

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



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

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

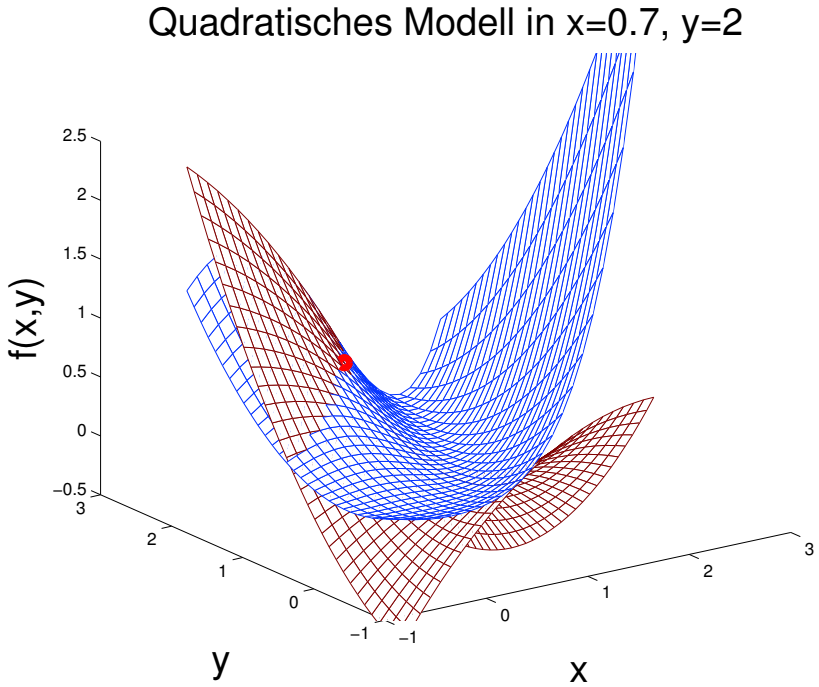
Für x nahe bei $\bar{x}$ ist es eine sehr gute Näherung an f: für glattes f erfüllt $\nabla^2 f$

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

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

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

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



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

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

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

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

Quadratisches Modell in Richtung h

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

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

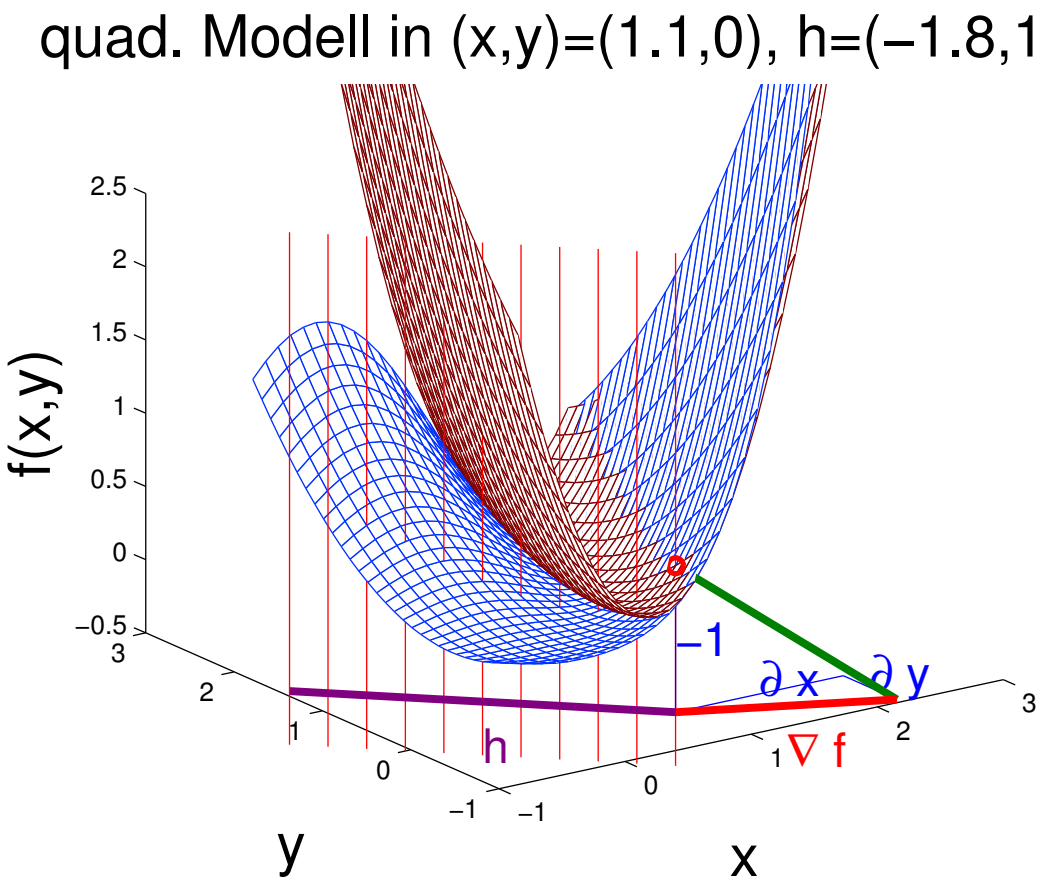
[Taylor-Entw. 2. Ord. der 1-D Funktion $\Phi : \mathbb{R} \to \mathbb{R}$, $\Phi(\alpha) := f(x + \alpha h)$]

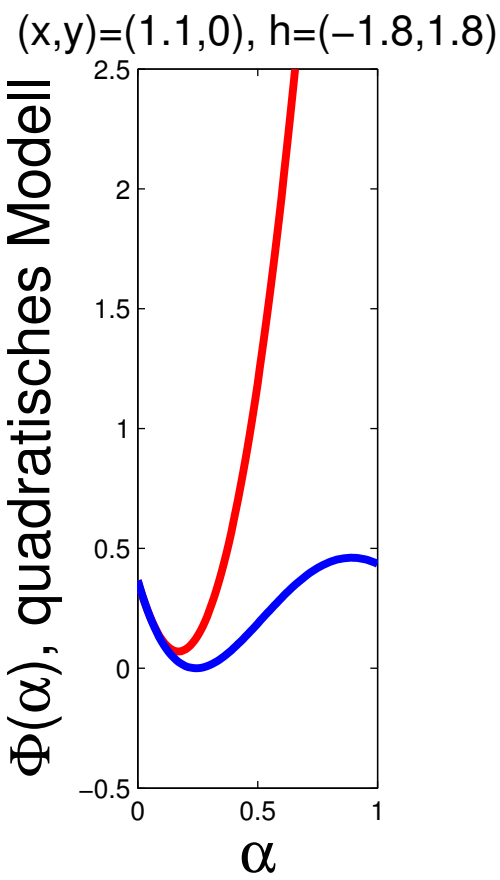
Quadratisches Modell in Richtung h

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

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

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



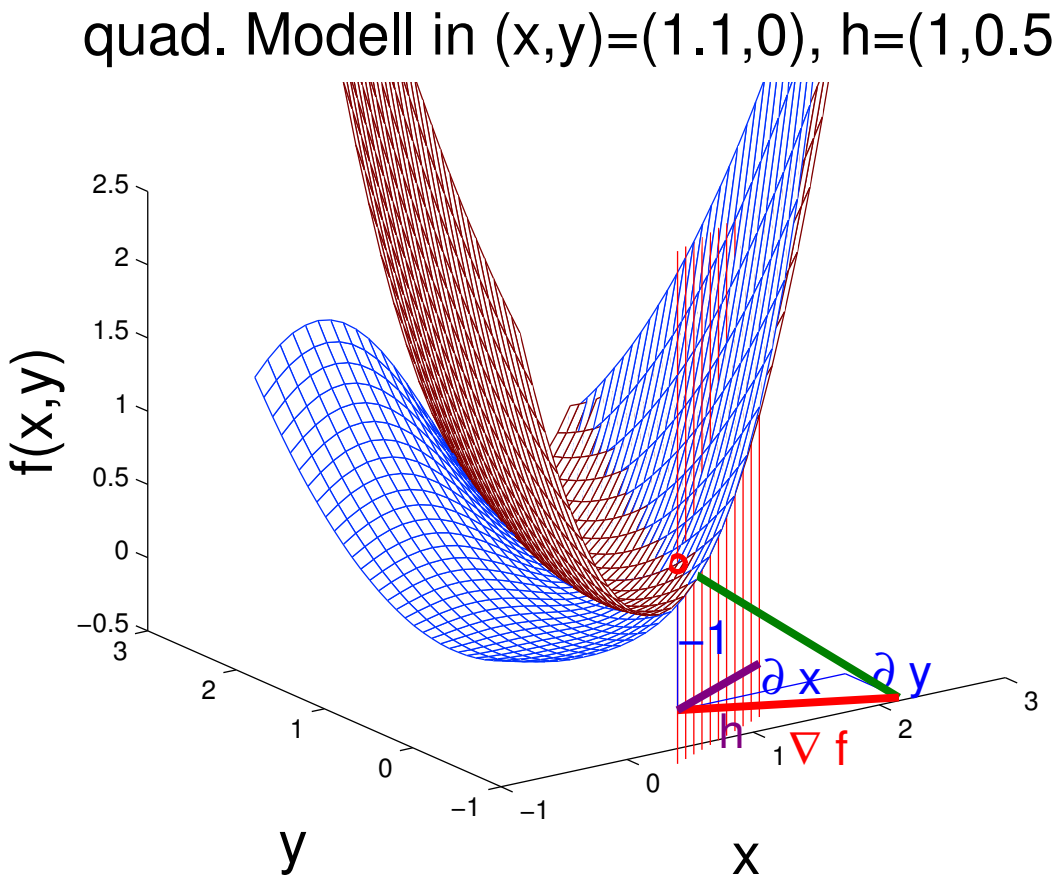


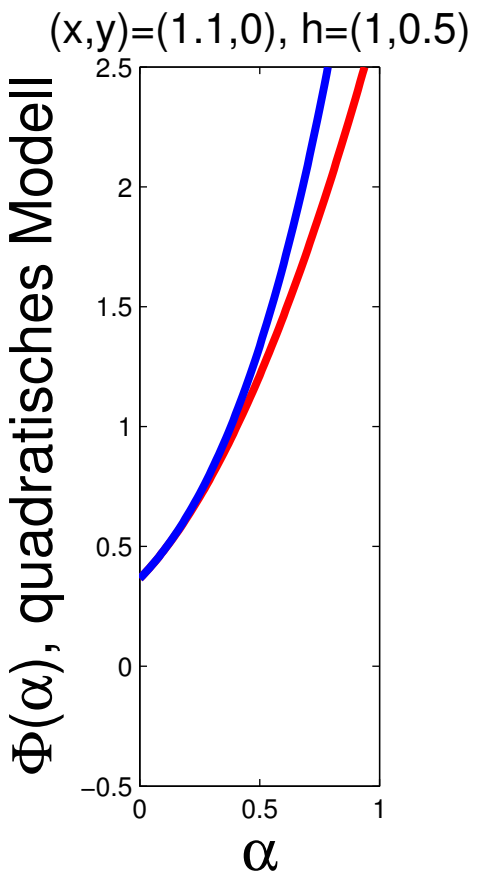
Quadratisches Modell in Richtung h

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

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

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



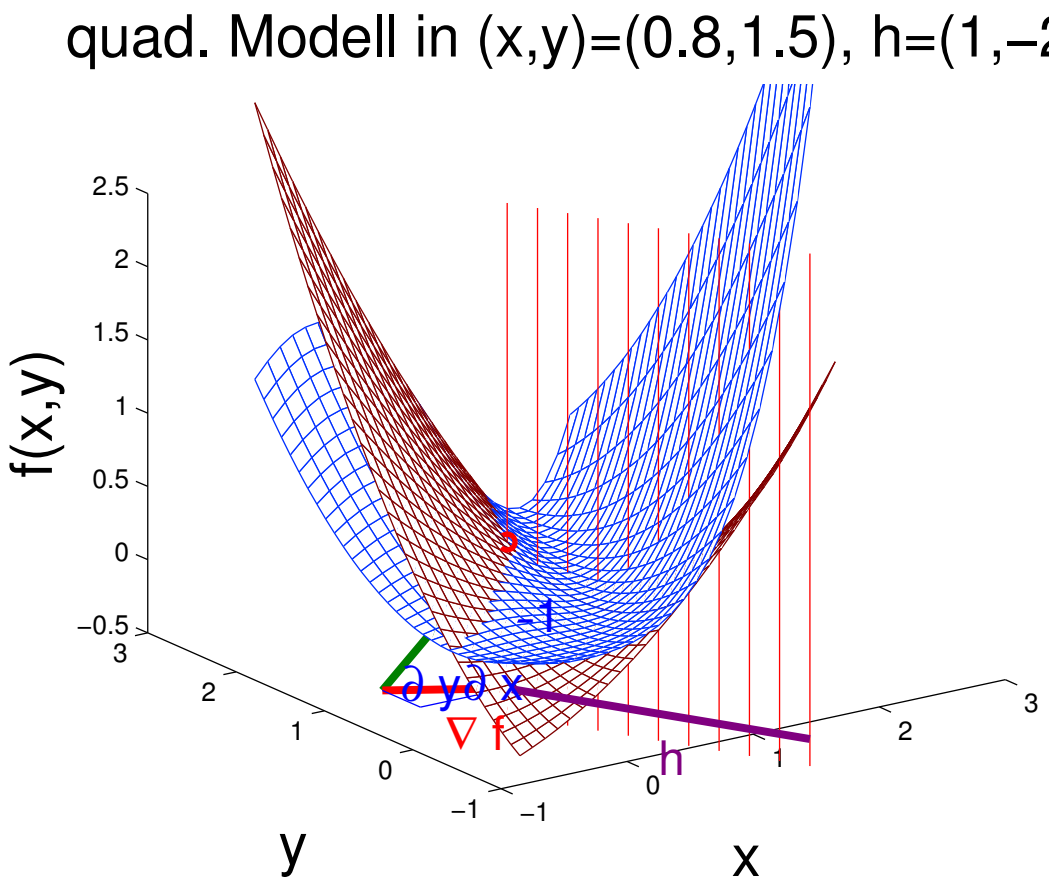


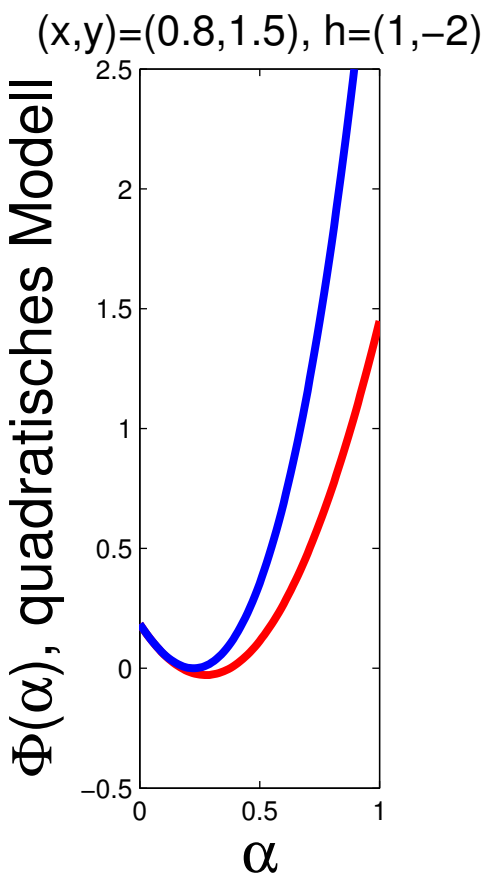
Quadratisches Modell in Richtung h

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

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

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



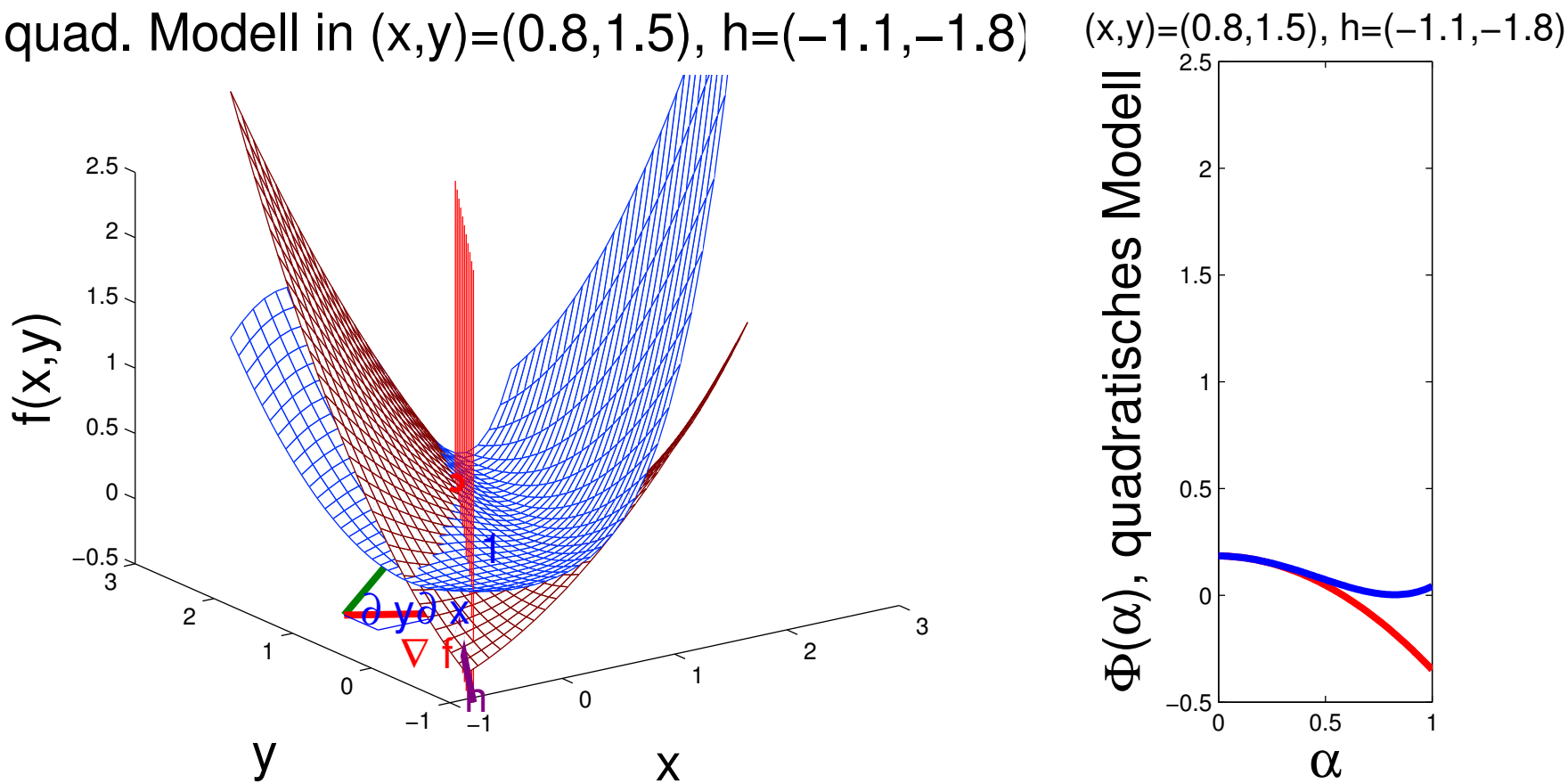


Quadratisches Modell in Richtung h

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

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

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



Der Satz von Taylor/Mittelwertsatz

Satz (Taylor/Mittelwertsatz)

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

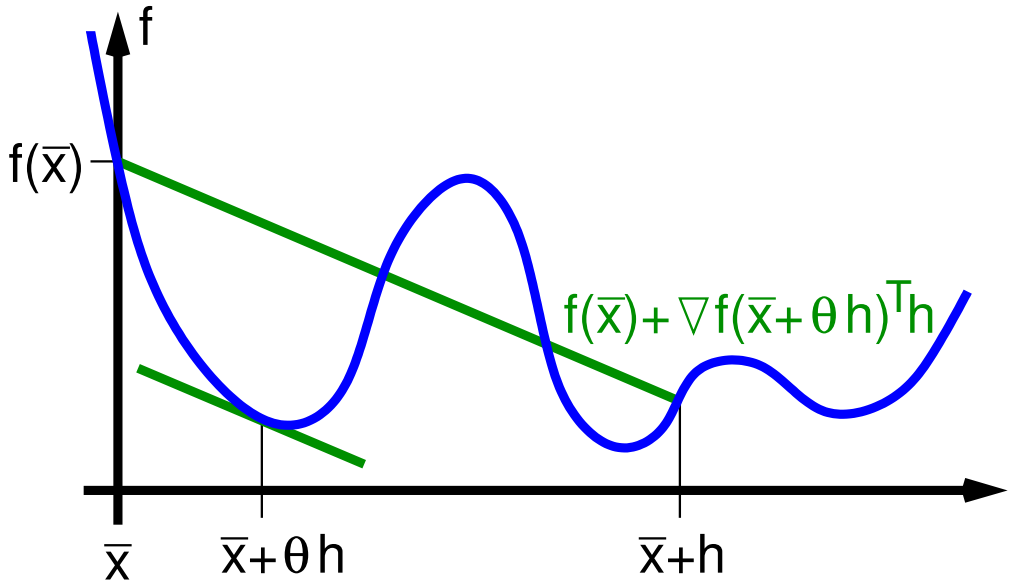
$$\exists \theta_1 \in (0,1): \quad f(\bar{x}+h) = f(\bar{x}) + \nabla f(\bar{x}+\theta_1 h)^T h,$$

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

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

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

Illustration des ersten Falls des Mittelwertsatzes:



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

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

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

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

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

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

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

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

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

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

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

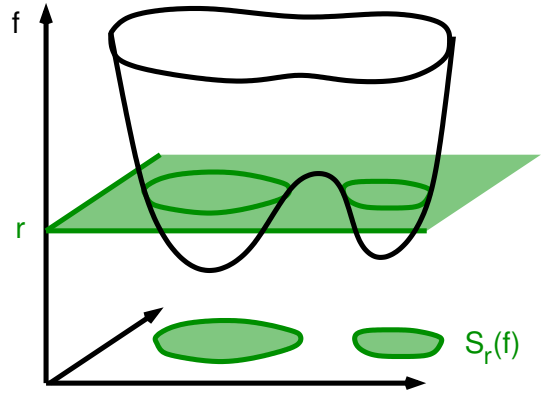
Aus dem Satz von Taylor und der Lipschitz-Stetigkeit von $\nabla^k f$ folgt
$f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \mathbf{o}(\|h\|)$,
$f(\bar{x} + h) = f(\bar{x}) + \nabla f(\bar{x})^T h + \frac{1}{2} h^T \nabla^2 f(\bar{x}) h + \mathbf{o}(\|h\|^2)$

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

Niveaumengen und Niveaulinien

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

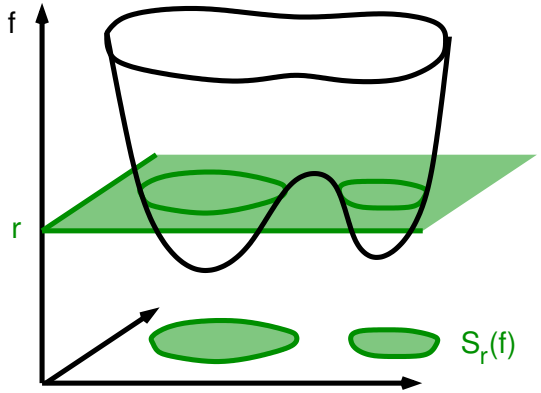
$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$



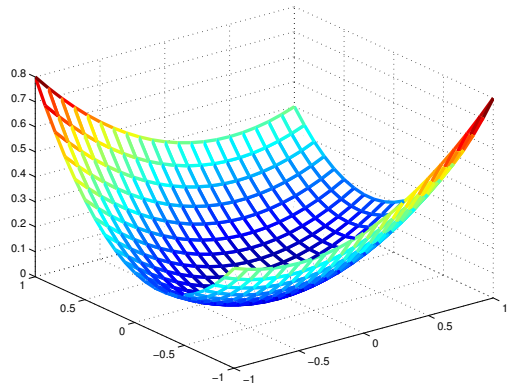
Niveaumengen und Niveaulinien

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

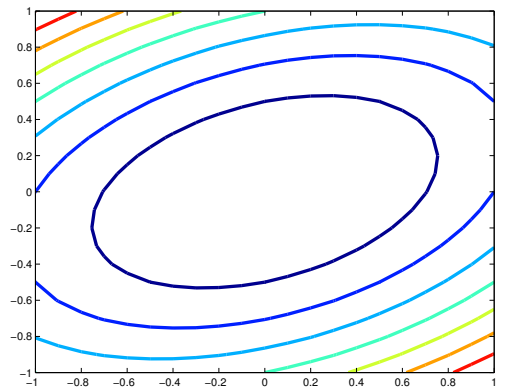
$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$



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



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

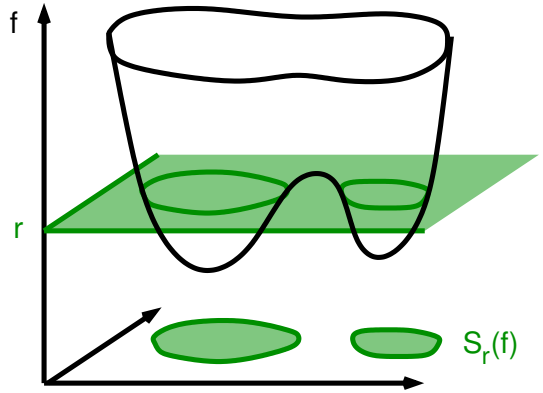


Bsp: quadratische Funktion

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

mit Q positiv definit.

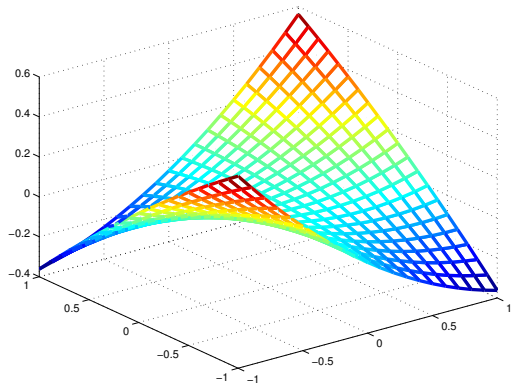
Niveaumengen und Niveaulinien



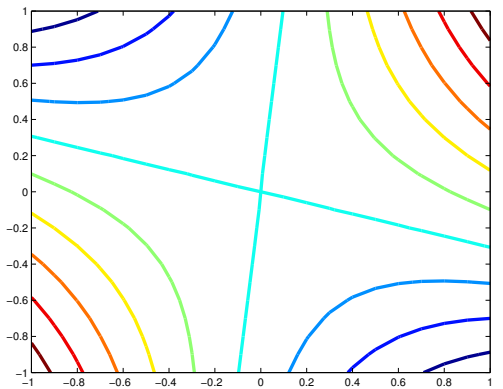
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

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

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



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



Bsp: quadratische Funktion

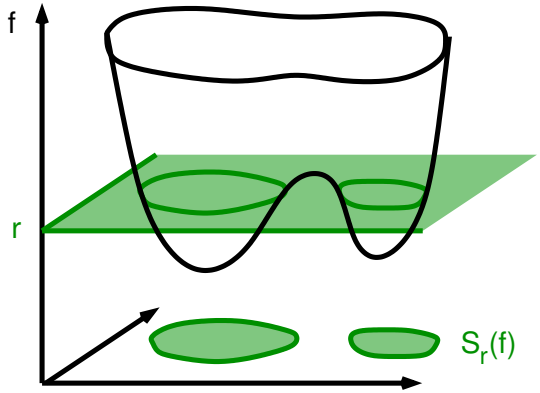
$$\tfrac{1}{2} x^T Q x + q^T x + d$$

mit Q indefinit.

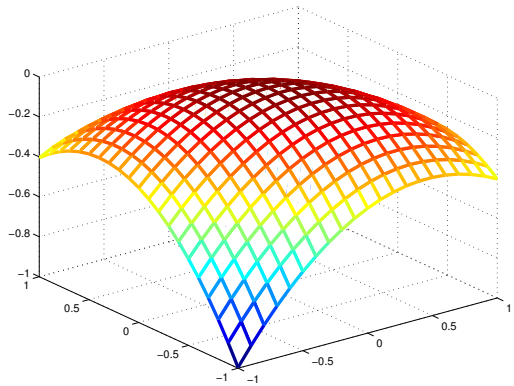
Niveaumengen und Niveaulinien

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

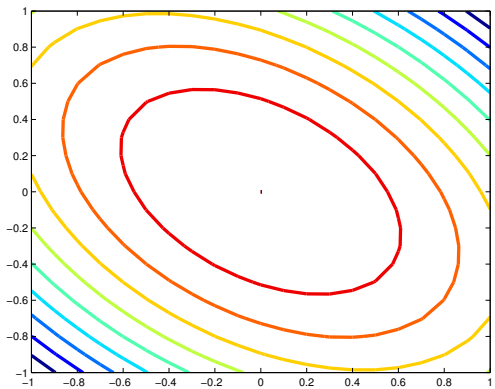
$$S_r(f) := \{x \in \mathbb{R}^n : f(x) \leq r\}.$$



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



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

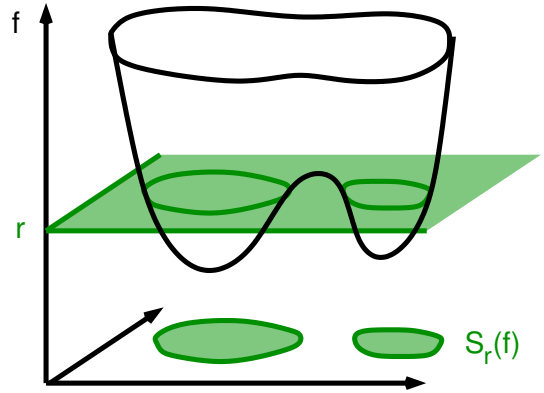


Bsp: quadratische Funktion

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

mit Q negativ definit.

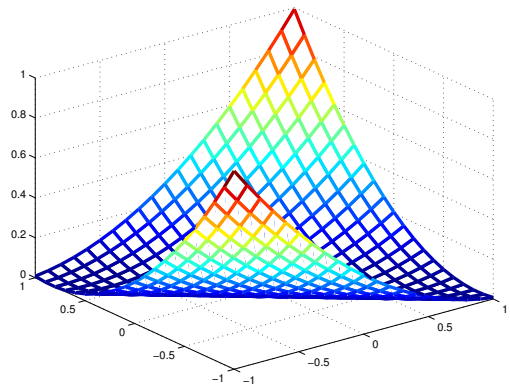
Niveaumengen und Niveaulinien



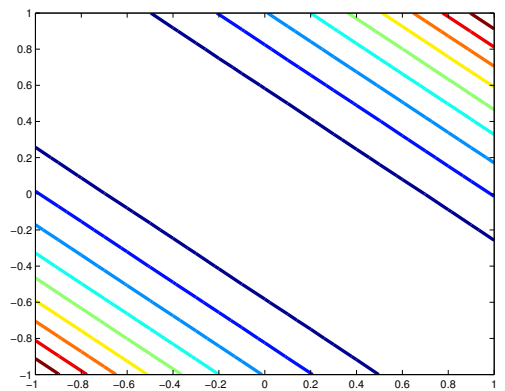
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

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

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



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

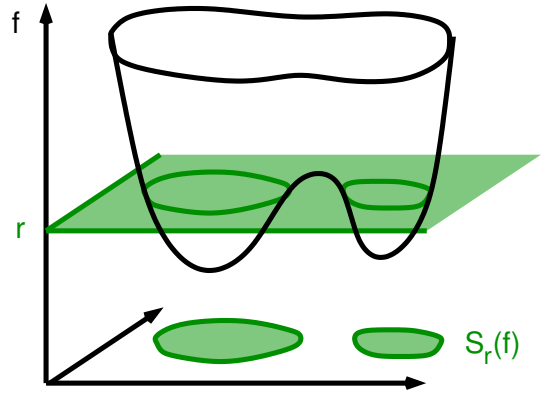


Bsp: quadratische Funktion

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

mit Q positiv semidefinit.

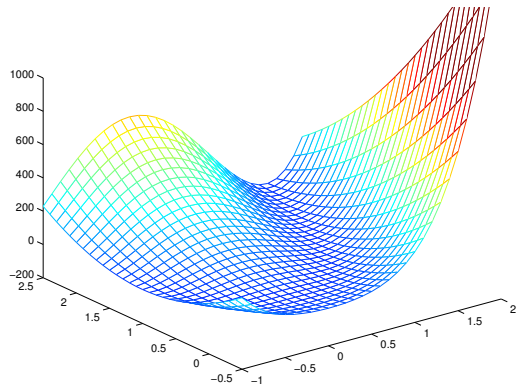
Niveaumengen und Niveaulinien



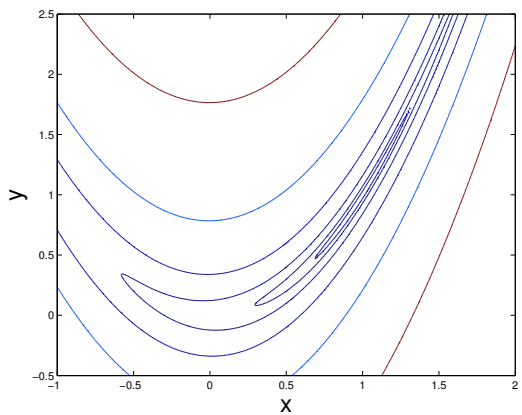
Verfahren der freien nichtlinearen Optimierung suchen nur Punkte mit besserem Zielfunktionswert als dem derzeitigen, also nur Punkte aus der Niveaumenge von f zu einem Wert $r \in \mathbb{R}$,

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

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



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



Bsp: Rosenbrock-Funktion (*banana shape*)

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

Minimum wird in (1,1) angenommen.

Inhaltsübersicht

5.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

5.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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

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

5.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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

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

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

5.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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

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

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

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

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

5.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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

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

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

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

5.2 Optimalitätsbedingungen

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

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

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

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

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

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

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

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

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

Notwendige Optimalitätsbedingung 2. Ordnung

Satz (Notwendige Optimalitätsbedingung 2. Ordnung)

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

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

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

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

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

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

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

Hinreichende Optimalitätsbedingung 2. Ordnung

Satz (Hinreichende Optimalitätsbedingung 2. Ordnung)

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

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

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

Hinreichende Optimalitätsbedingung 2. Ordnung

Satz (Hinreichende Optimalitätsbedingung 2. Ordnung)

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

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

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

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

Hinreichende Optimalitätsbedingung 2. Ordnung

Satz (Hinreichende Optimalitätsbedingung 2. Ordnung)

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

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

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

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

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

Bemerkungen

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

Bemerkungen

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

Bemerkungen

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

Inhaltsübersicht

5.3 Das Newton-Verfahren

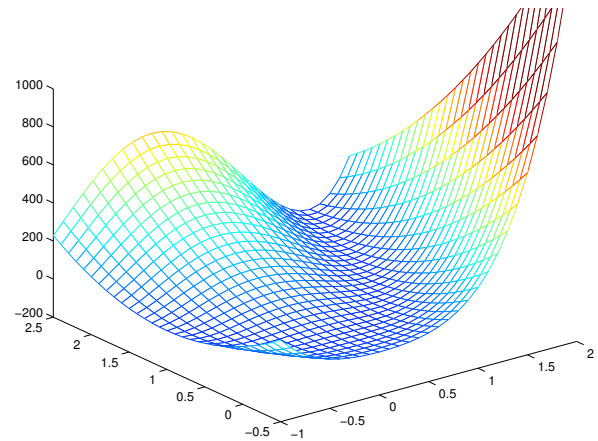
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

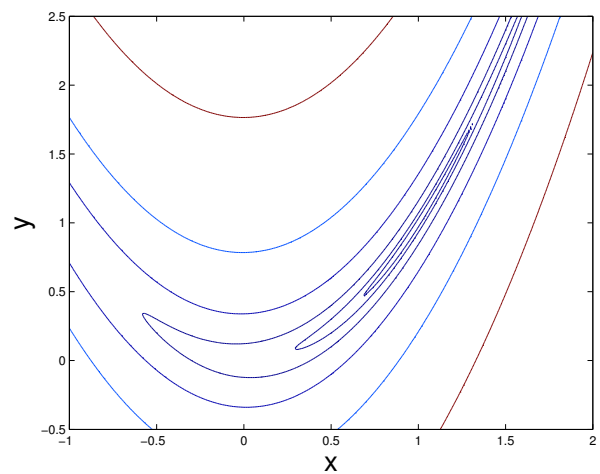
Ist x^* ein lokales Minimum mit $\nabla^2 f(x^*) \succ 0$ und ändert sich $\nabla^2 f$ nicht zu schnell ($\nabla^2 f$ Lipschitz-stetig), gilt $\nabla^2 f(x) \succ 0$ für alle x nahe bei x^*.

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

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

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





5.3 Das Newton-Verfahren

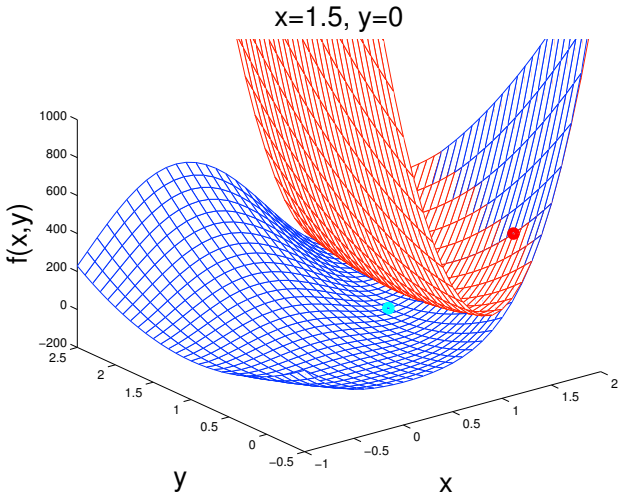
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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

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

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

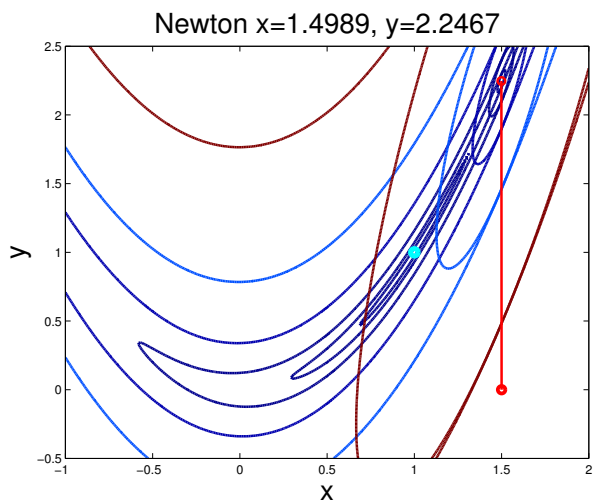
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



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

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

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



5.3 Das Newton-Verfahren

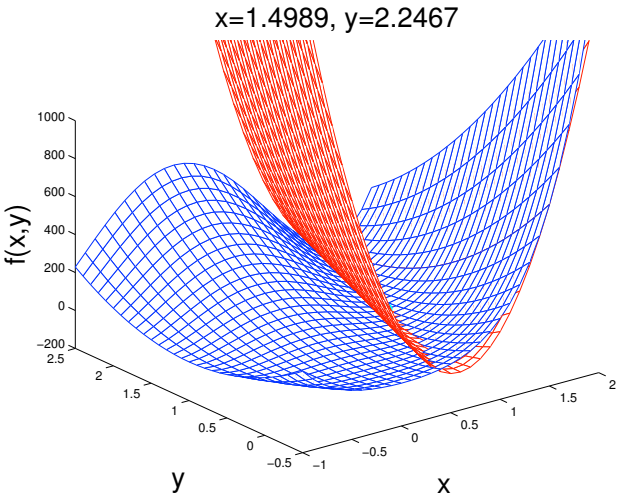
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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

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

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

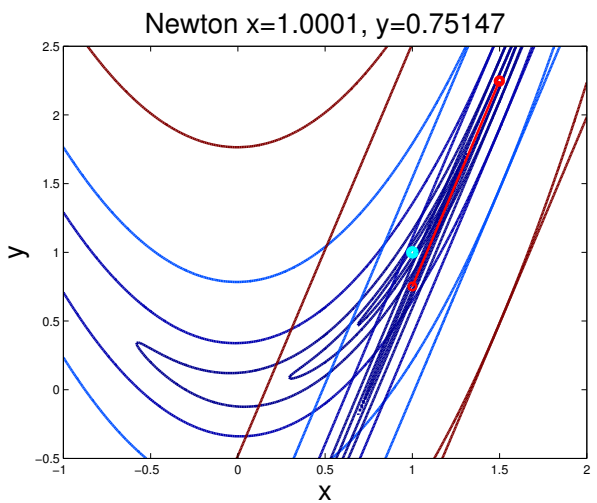
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



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

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

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



5.3 Das Newton-Verfahren

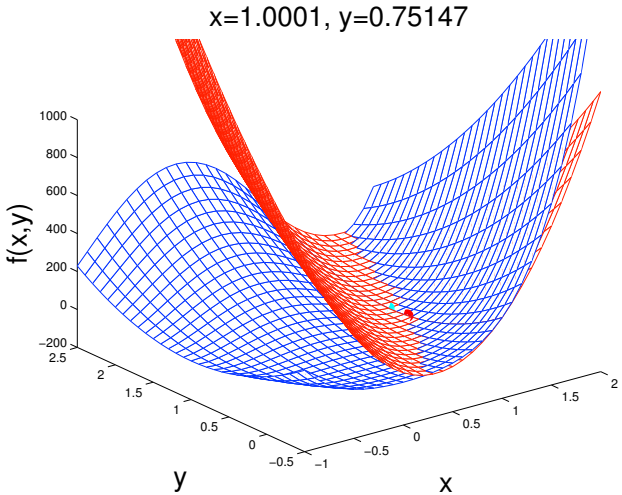
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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

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

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

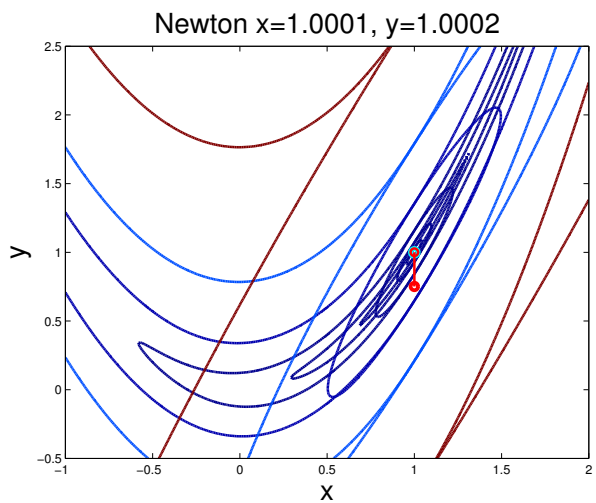
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



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

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

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



5.3 Das Newton-Verfahren

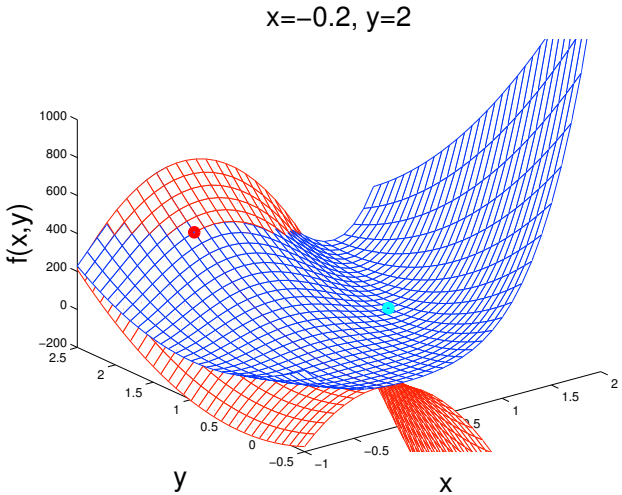
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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

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

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

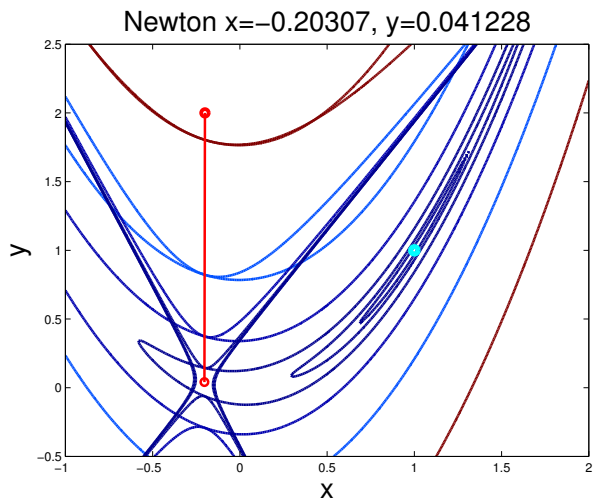
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



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

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

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



5.3 Das Newton-Verfahren

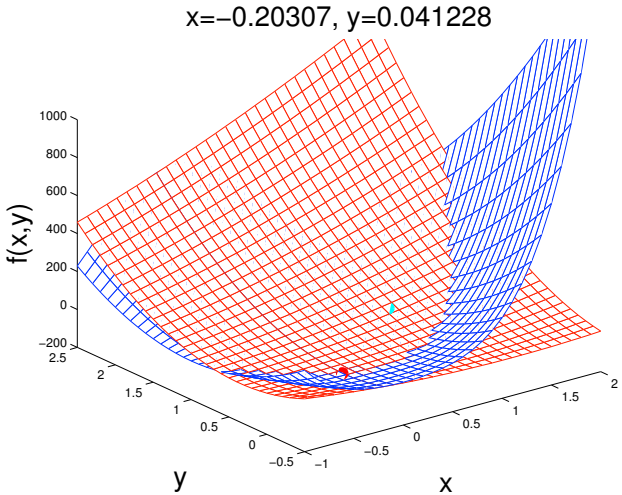
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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

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

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

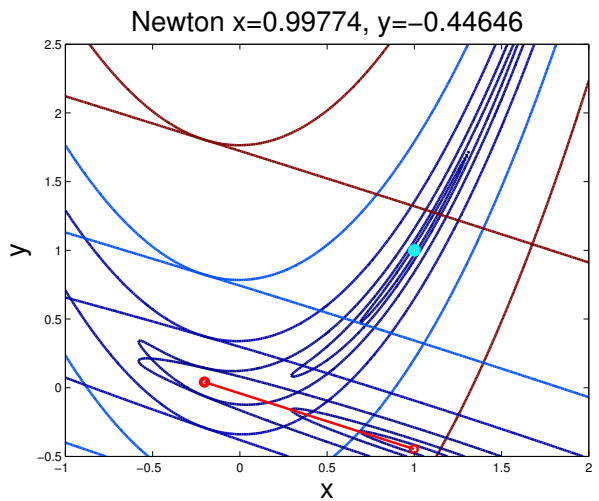
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



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

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

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



5.3 Das Newton-Verfahren

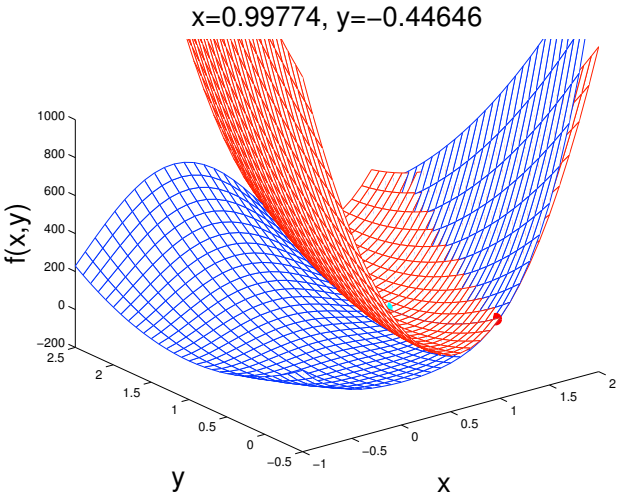
[Eigentlich sucht es Nullstellen von Funktionen $F : \mathbb{R}^n \to \mathbb{R}^m$ (s. später), hier suchen wir ein x mit $\nabla f(x) = 0$.]

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

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

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

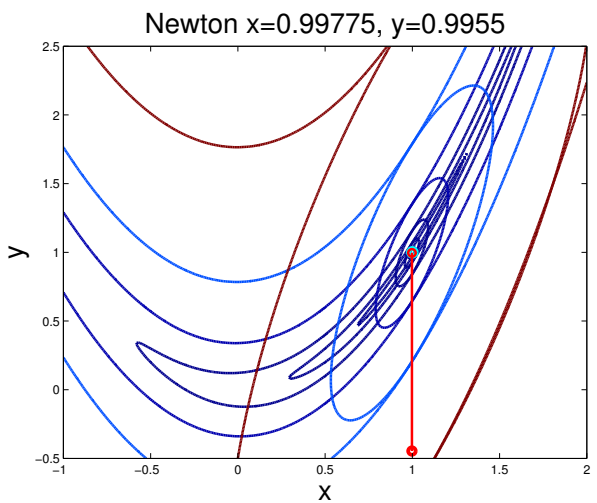
$$[= c + q^T h + \tfrac{1}{2} h^T Q h]$$



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

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

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



Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

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

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

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

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

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

Satz (lokal-quadratische Konvergenz des Newton-Verfahrens)

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

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

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

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

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

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

Bemerkungen

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

Bemerkungen

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

Bemerkungen

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

Bemerkungen

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

Bemerkungen

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

Inhaltsübersicht

5.4 Line-Search-Verfahren

Schematischer Ablauf von Line-Search-Verfahren:

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

Zwei Hauptaufgaben:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

---

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

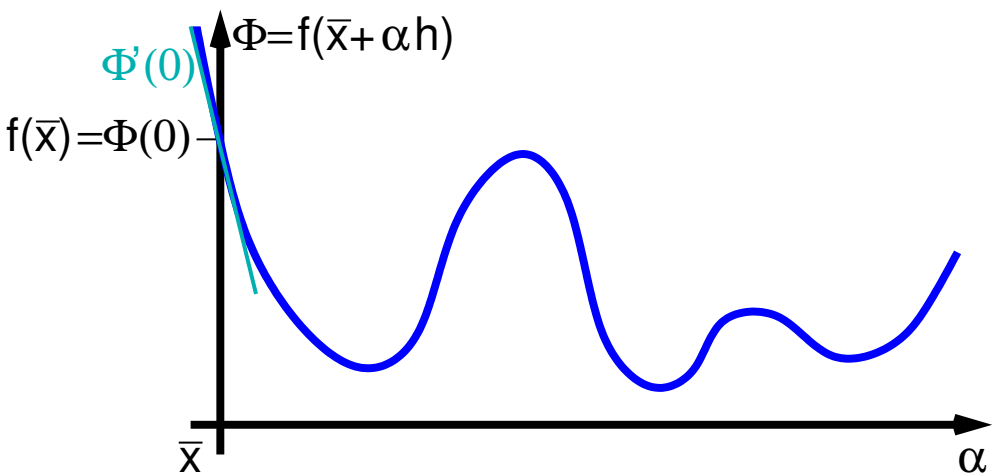
$$\exists \delta > 0 : \frac{-\nabla f_k}{\|\nabla f_k\|}^T \frac{h^{(k)}}{\|h^{(k)}\|} = \cos \angle(-\nabla f_k, h^{(k)}) \geq \delta > 0 \text{ für } k > 0.$$

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

Line-Search für Abstiegsrichtung h

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

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

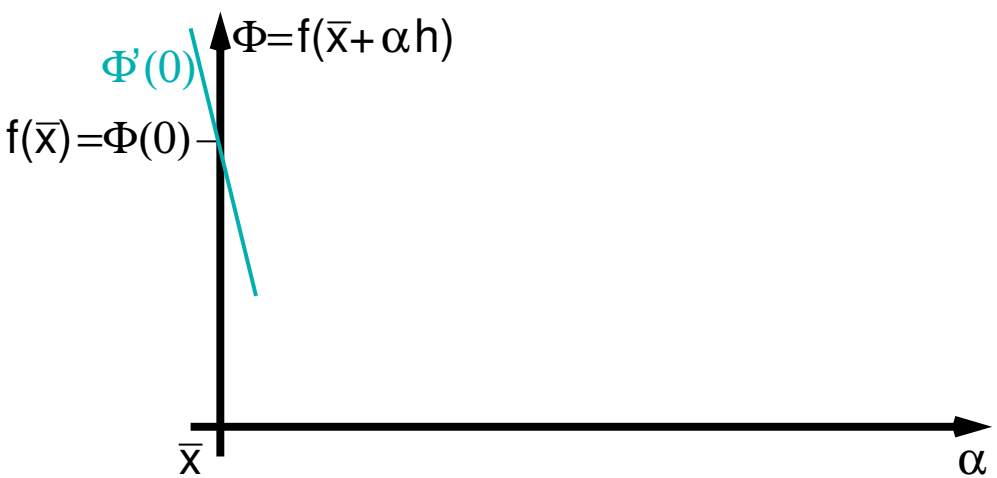


Line-Search für Abstiegsrichtung h

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

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

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



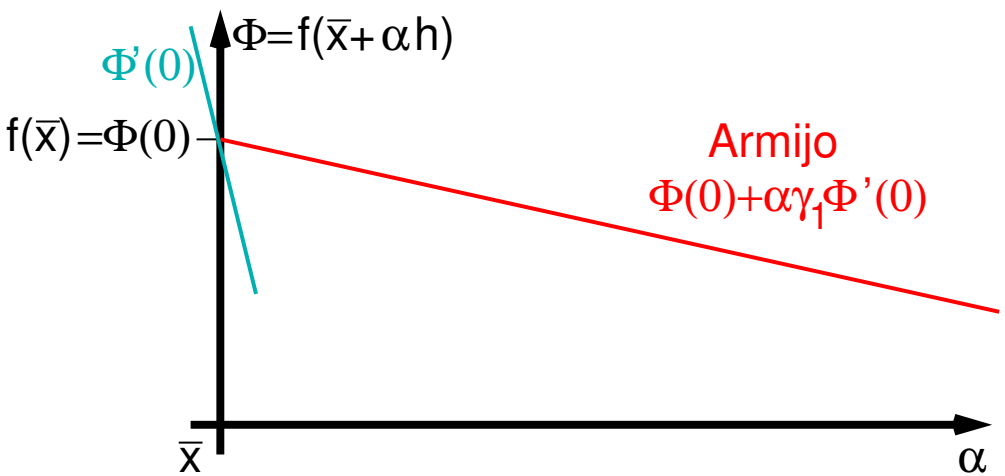
Line-Search für Abstiegsrichtung h

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

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

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

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



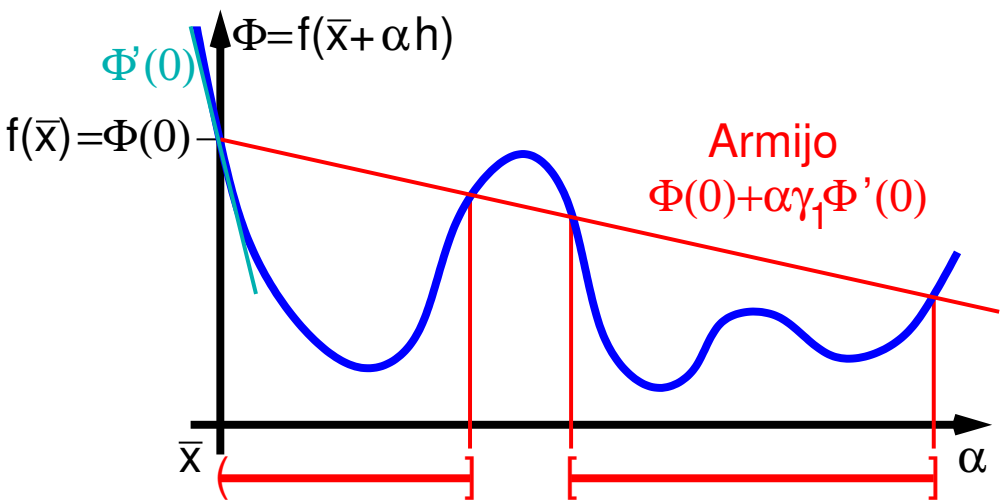
Line-Search für Abstiegsrichtung h

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

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

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

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



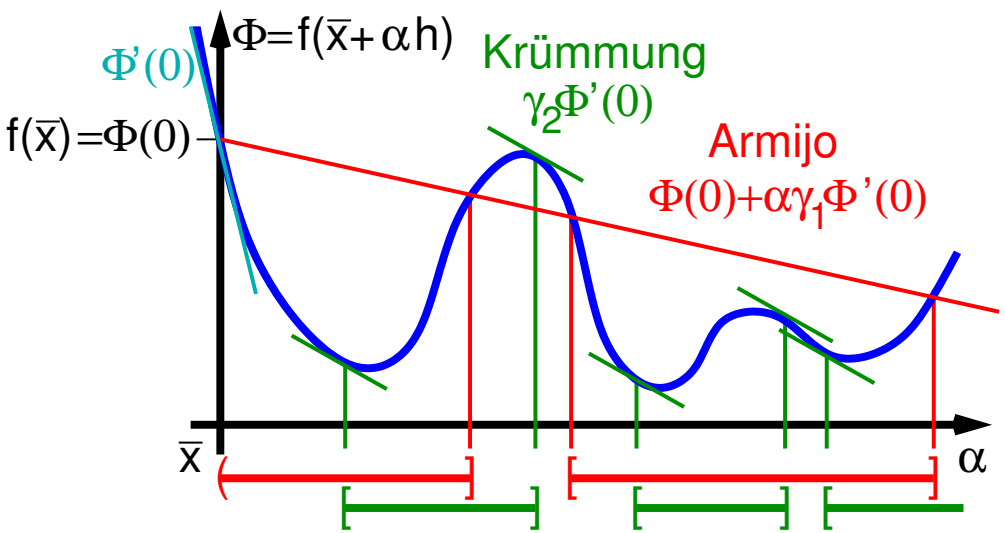
Line-Search für Abstiegsrichtung h

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

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

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

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



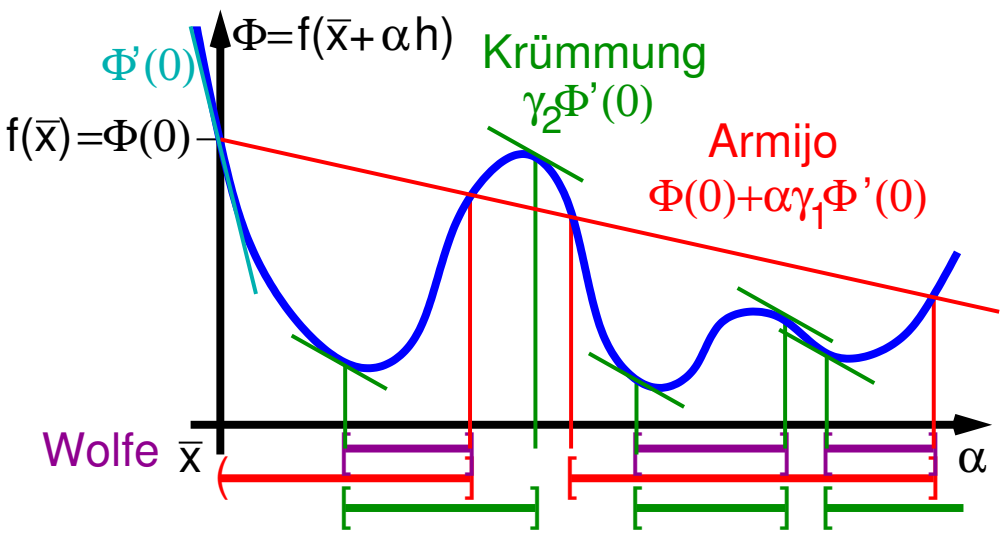
Line-Search für Abstiegsrichtung h

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

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

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

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



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

Wolfe-Bedingungen und globale Konvergenz

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

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

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

Wolfe-Bedingungen und globale Konvergenz

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

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

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

Satz (Globale Konvergenz von Line-Search-Verfahren)

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

Wolfe-Bedingungen und globale Konvergenz

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

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

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

Satz (Globale Konvergenz von Line-Search-Verfahren)

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

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

Bestimmung der Schrittweite in der Praxis

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

5.5 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 $\bar{x}$ ($B \succ 0$):

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

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

5.5 Skalierung

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

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

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

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

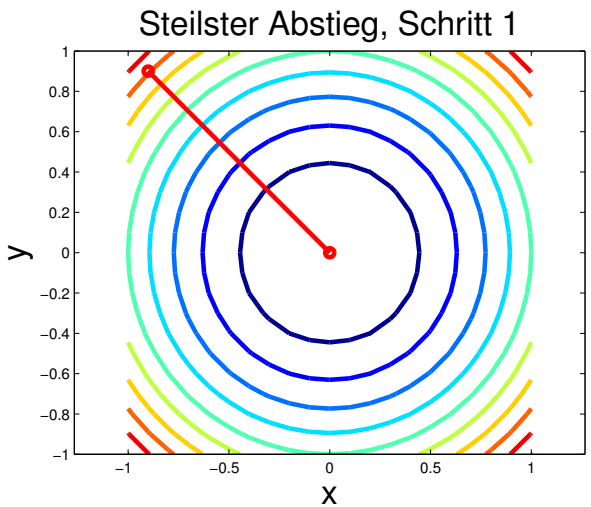
Ideal skaliert ($Q = I$):

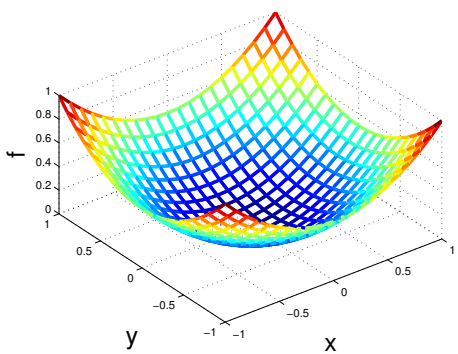
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0, c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

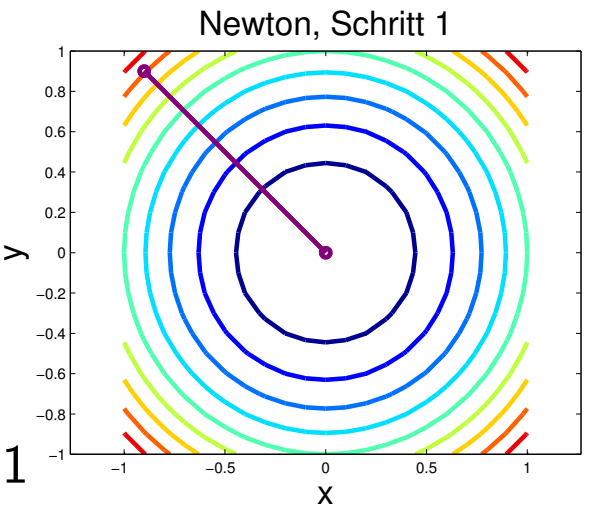
Ideal skaliert ($Q = I$):

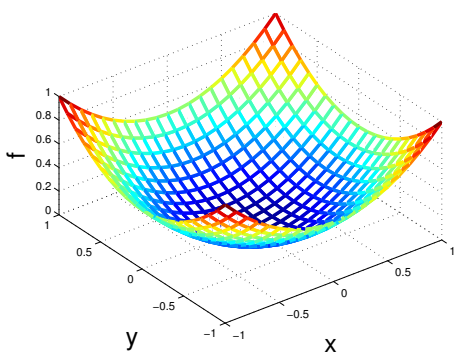
$$Q = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0, c = 0$$

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

Newton ($B = Q$):

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





5.5 Skalierung

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

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

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

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

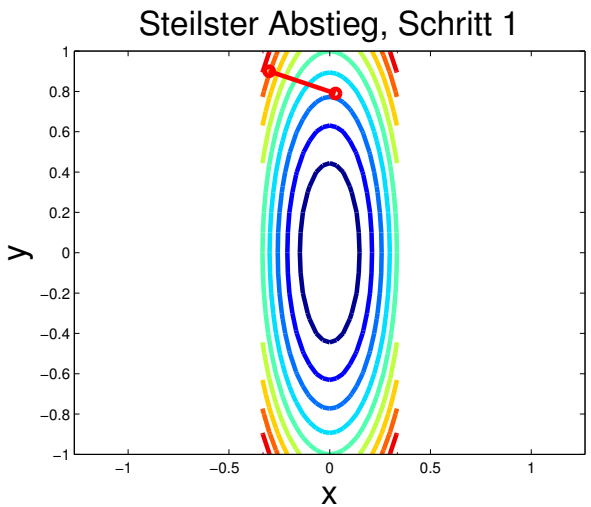
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

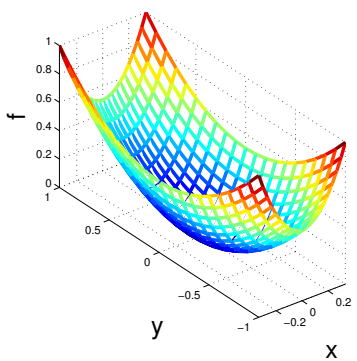
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

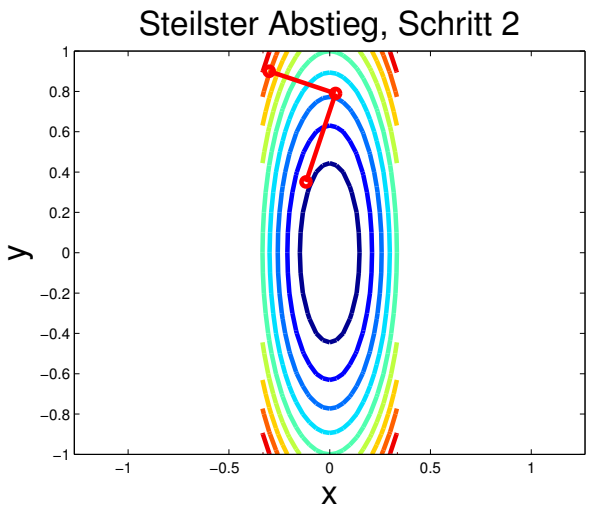
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

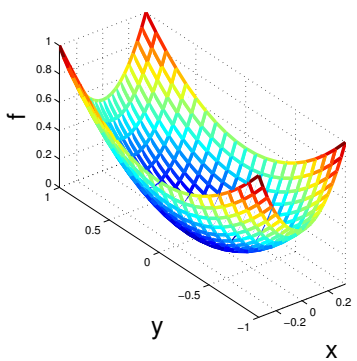
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

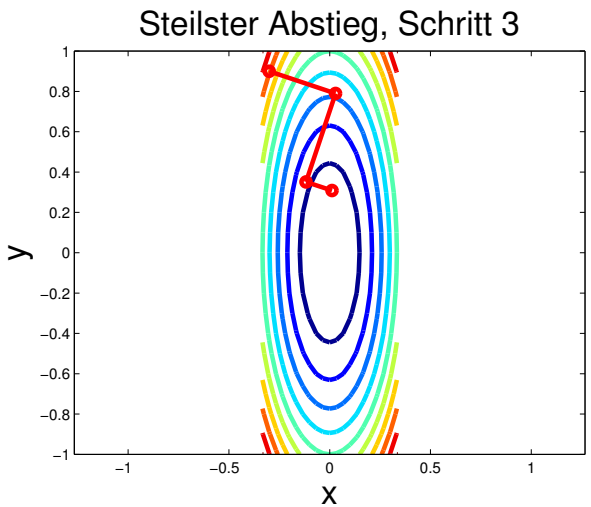
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

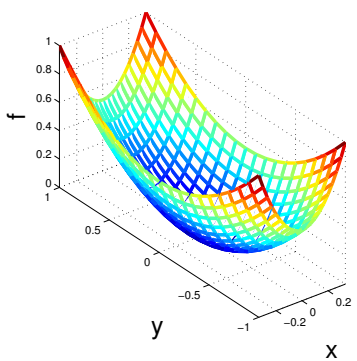
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

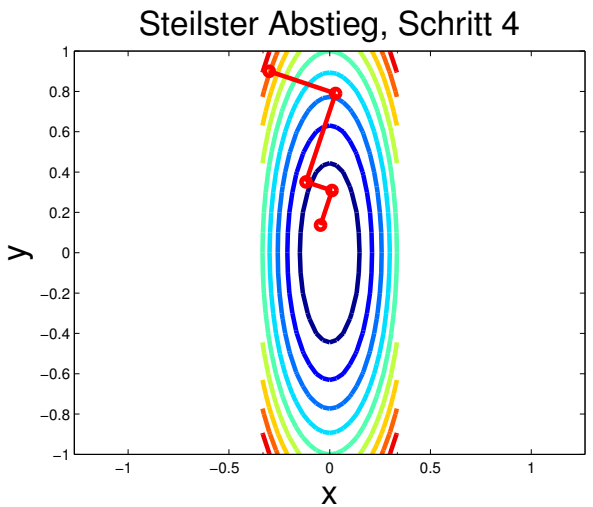
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

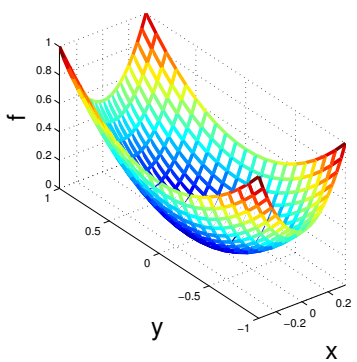
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

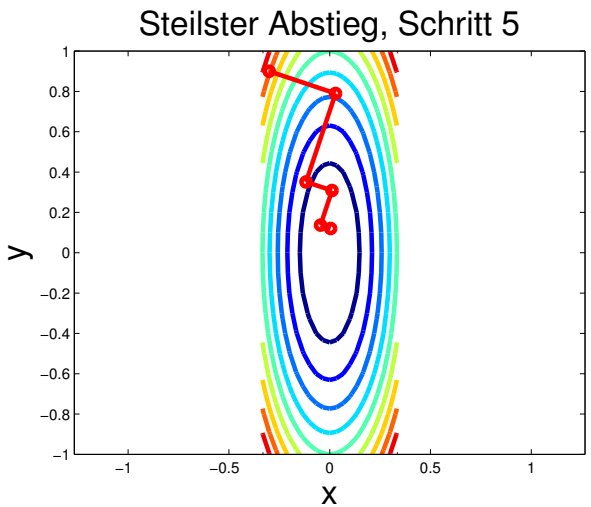
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

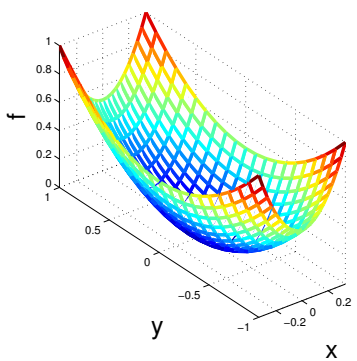
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

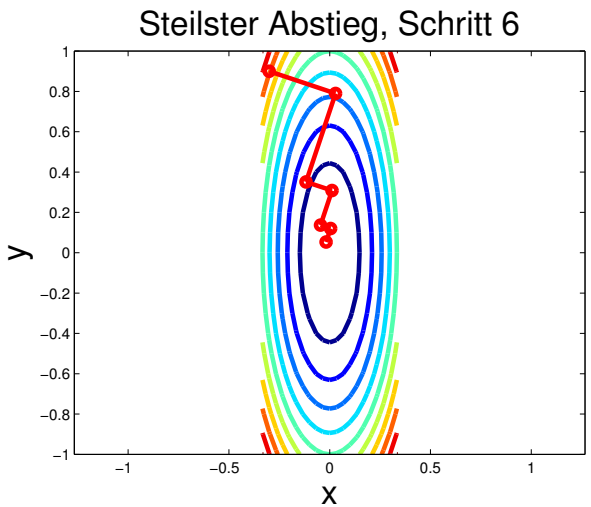
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

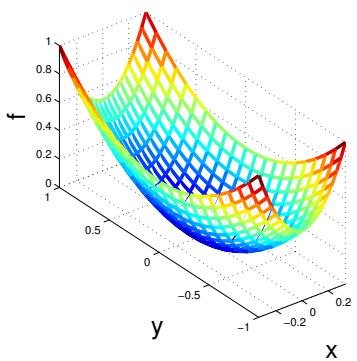
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

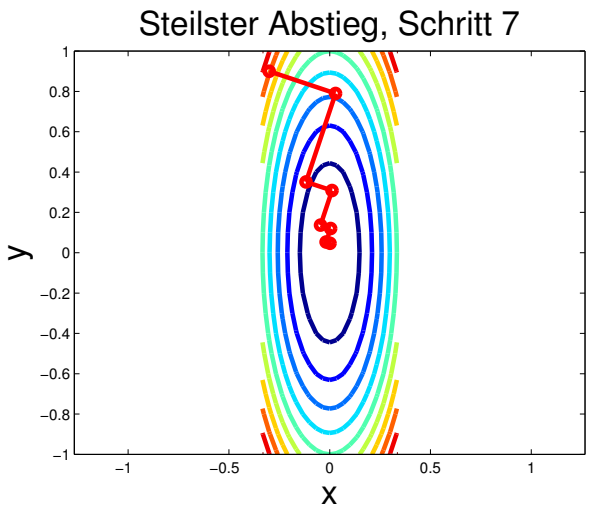
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

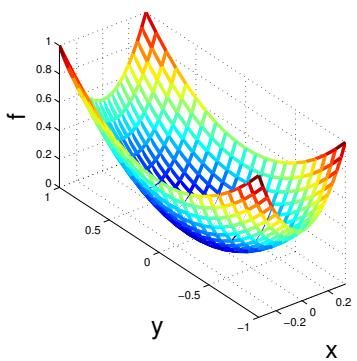
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

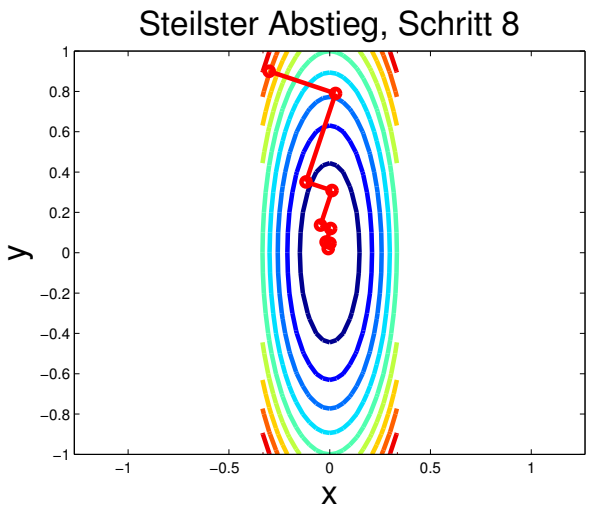
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

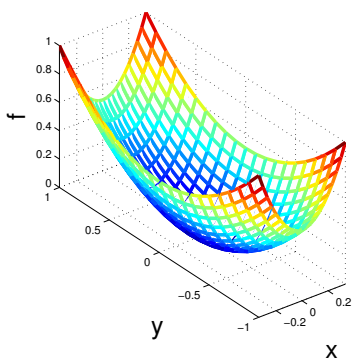
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

Steilster Abstieg ($B = I$):

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





5.5 Skalierung

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

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

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

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

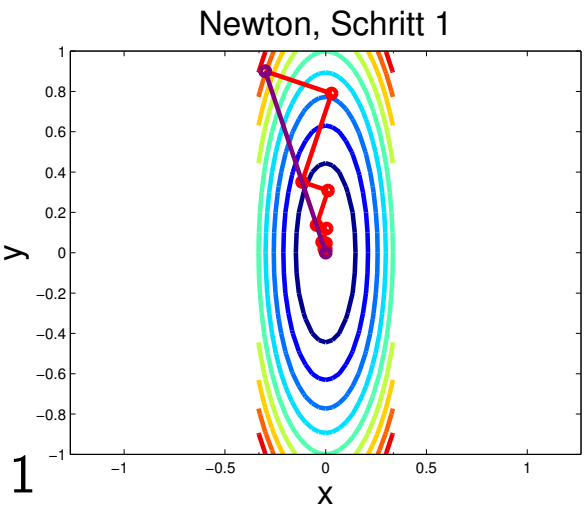
Noch gut skaliert! ($x_1 = 3\tilde{x}_1$):

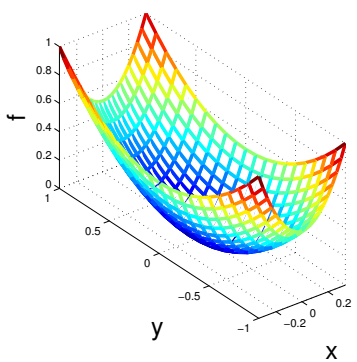
$$\tilde{Q} = \begin{bmatrix} 3^2 & 0 \\ 0 & 1 \end{bmatrix},\ q = 0,\ c = 0$$

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

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

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





Steilster Abstieg konvergiert SEHR langsam

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

Satz

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

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

Steilster Abstieg konvergiert SEHR langsam

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

Satz

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

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

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

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

$\rightarrow$ Newton konvergiert am Ende fast immer hervorragend

Steilster Abstieg konvergiert SEHR langsam

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

Satz

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

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

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

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

$\rightarrow$ Newton konvergiert am Ende fast immer hervorragend

---

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

Inhaltsübersicht

5.6 Quasi-Newton-Verfahren

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

Satz

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

5.6 Quasi-Newton-Verfahren

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

Satz

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

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

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

5.6 Quasi-Newton-Verfahren

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

Satz

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

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

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

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

5.6 Quasi-Newton-Verfahren

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

Satz

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

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

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

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

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

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

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

5.6 Quasi-Newton-Verfahren

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

Satz

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

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

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

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

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

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

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

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

Quasi-Newton mit BFGS-Update

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

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

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

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

Quasi-Newton mit BFGS-Update

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

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

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

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

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

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

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

Quasi-Newton mit BFGS-Update

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

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

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

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

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

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

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

Quasi-Newton mit BFGS-Update

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

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

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

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

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

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

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

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

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

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

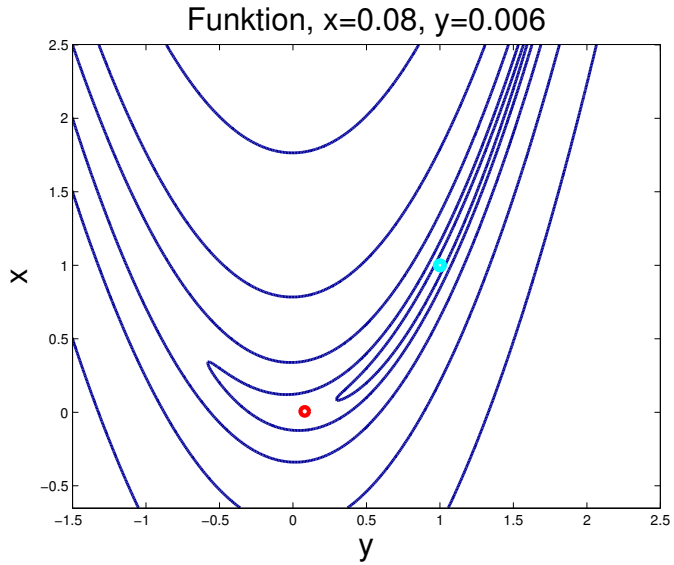
In der Nähe eines x^*, das die hinreichenden Optimalitätsbedingungen erfüllt, ist ein hinreichend glattes f annähernd streng konvex und quadratisch.

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

Inhaltsübersicht

5.7 Trust-Region-Verfahren

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

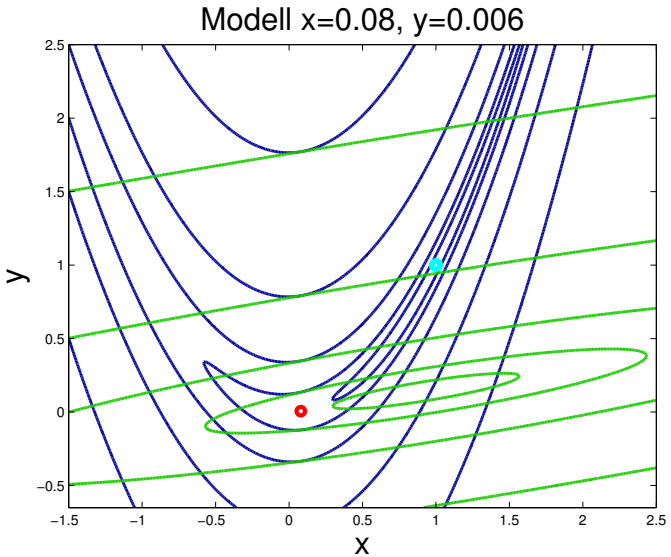


5.7 Trust-Region-Verfahren

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]

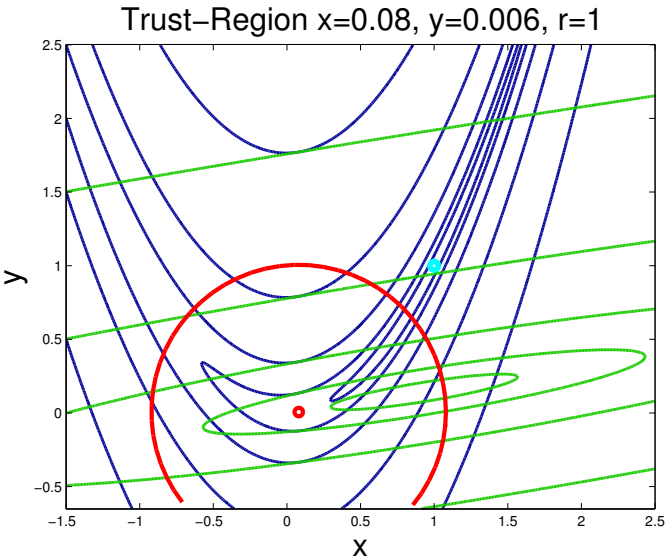


5.7 Trust-Region-Verfahren

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

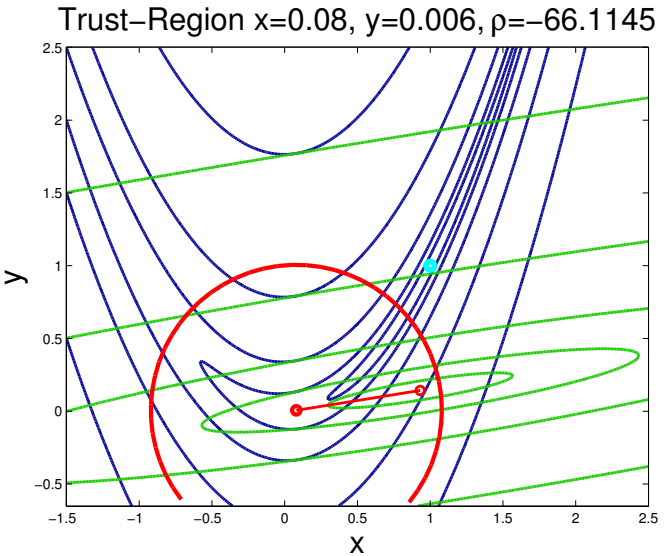


5.7 Trust-Region-Verfahren

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
$$m_0(h) \approx f(x^{(0)} + h) \; \forall \|h\| \leq \Delta_0$$
1. Löse das Trust-Region-Unterproblem:
$$h^{(k)} \approx \underset{\|h\| \leq \Delta_k}{\operatorname{Argmin}} \, m_k(h) \qquad \text{[nur annähern]}$$

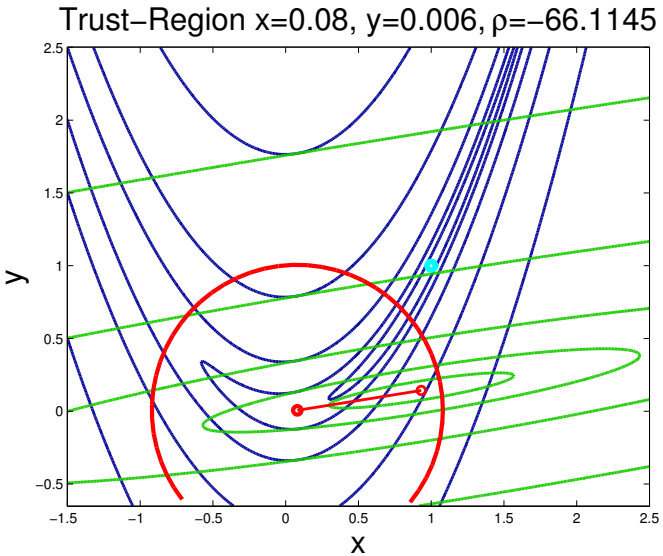


5.7 Trust-Region-Verfahren

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

---

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



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

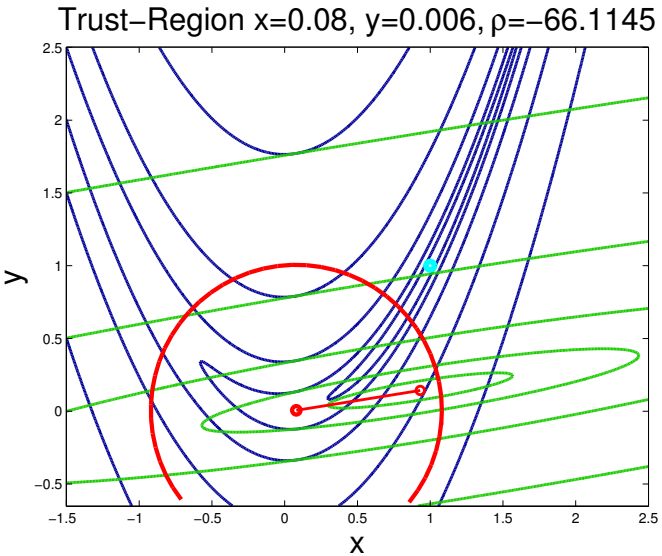
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert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

5.7 Trust-Region-Verfahren

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

---

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



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

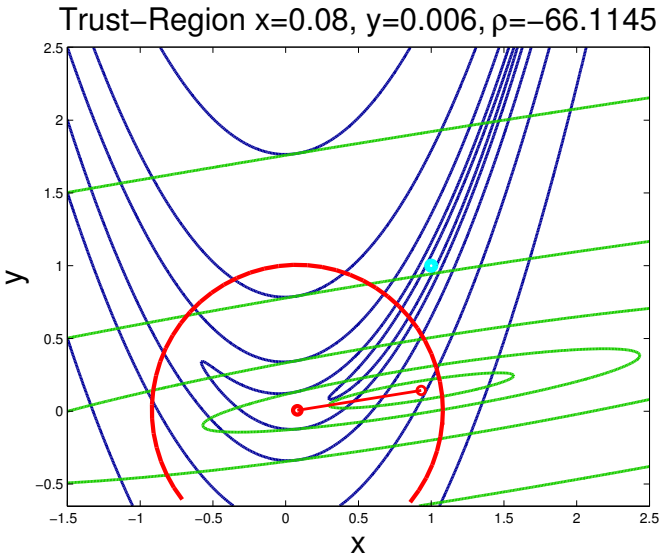
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert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

5.7 Trust-Region-Verfahren

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

---

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



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

2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$

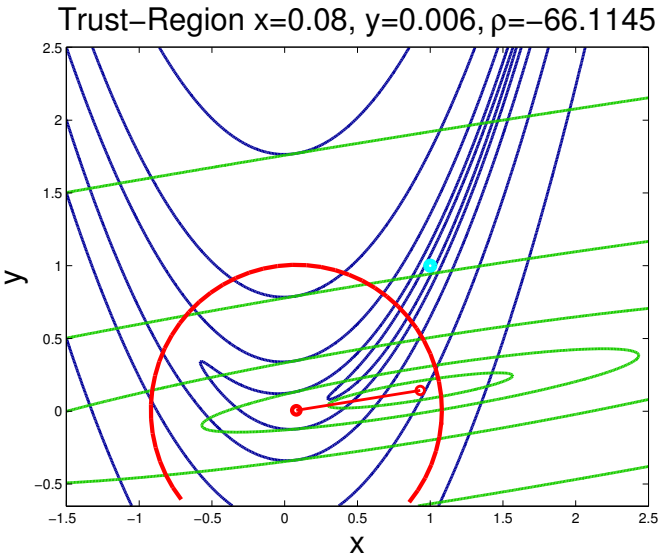
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

5.7 Trust-Region-Verfahren

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

---

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



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

2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$

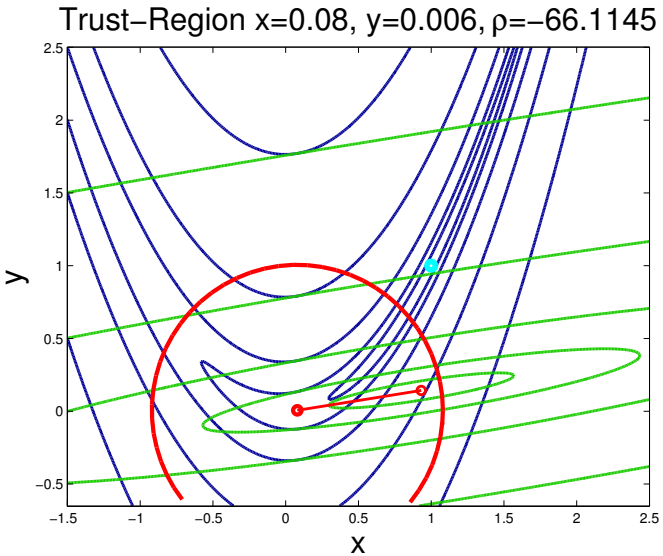
3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
setze $x^{(k+1)} := x^{(k)} + h^{(k)}$, $m_{k+1}(h) = f_{k+1} + \nabla f_{k+1}^T h + \frac{1}{2} h^T B_{k+1} h$,
sonst ändere nichts, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

5.7 Trust-Region-Verfahren

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

---

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



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

2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$

3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
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, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

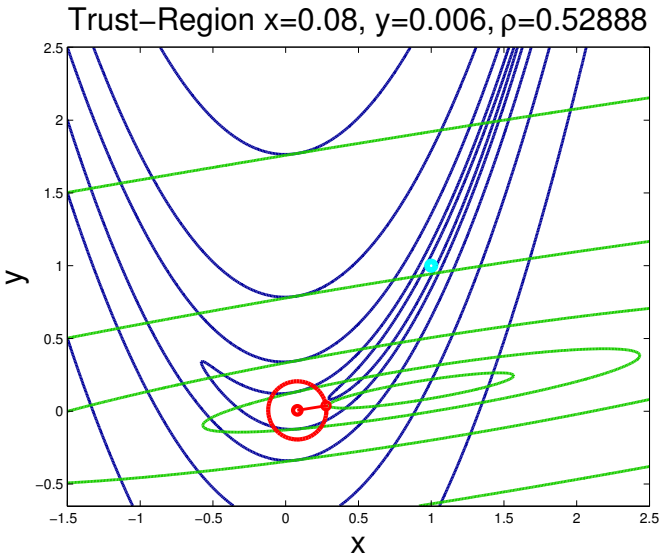
4. $k \leftarrow k + 1$. Falls $\|\nabla f_k\| > \varepsilon$, GOTO 1, sonst STOP.

5.7 Trust-Region-Verfahren

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

---

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



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

2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$

3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
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, $x^{(k+1)} = x^{(k)}$, $m_{k+1} := m_k$.

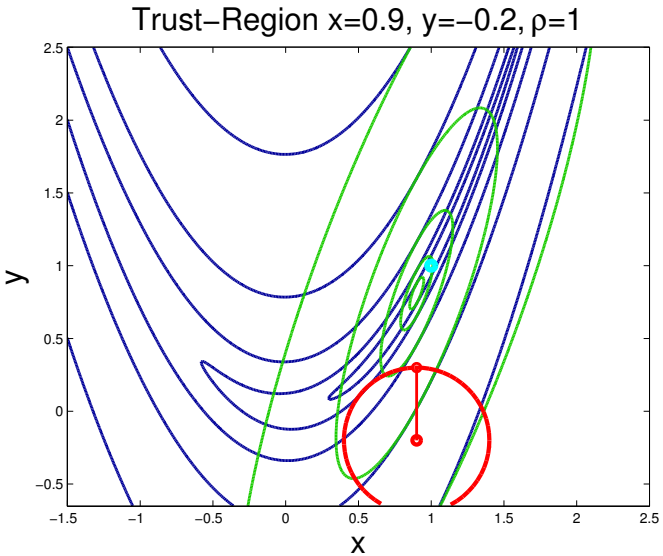
4. $k \leftarrow k + 1$. Falls $\|\nabla f_k\| > \varepsilon$, GOTO 1, sonst STOP.

5.7 Trust-Region-Verfahren

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

---

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



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

2. Berechne $f(x^{(k)} + h^{(k)})$ und den Fortschrittsfaktor $\rho_k := \frac{f(x^{(k)}) - f(x^{(k)} + h^{(k)})}{m_k(0) - m_k(h^{(k)})}$,
setze $\Delta_{k+1} := \begin{cases} \frac{1}{4}\|h^{(k)}\| & \text{für } \rho_k < \frac{1}{4} \quad [m_k(h^{(k)}) \text{ zu schlecht}] \\ \min\{2\Delta_k, \bar{\Delta}\} & \rho_k > \frac{3}{4} \text{ und } \|h^{(k)}\| = \Delta_k \quad [\Delta \text{ zu klein}] \\ \Delta_k & \text{sonst.} \quad [\text{Wert ok}] \end{cases}$

3. Ist $\rho_k > \eta$ für ein festes Akzeptanzniveau $\eta \in [0, 1)$,
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

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

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

Konvergenz von Trust-Region-Verfahren

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

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

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

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

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

Konvergenz von Trust-Region-Verfahren

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

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

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

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

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

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

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

Konvergenz von Trust-Region-Verfahren

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

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

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

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

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

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

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

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

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

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

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

Falls f nur als Unterprogramm gegeben ist oder symbolisches Differenzieren fehlschlägt:

- Numerisches Differenzieren
- Automatisches Differenzieren (falls Source-Code verfügbar)

(Für beides gibt es kommerzielle und frei verfügbare Software)

Inhaltsübersicht

5.8 Numerisches Differenzieren

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

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

5.8 Numerisches Differenzieren

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

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

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

5.8 Numerisches Differenzieren

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

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

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

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

5.8 Numerisches Differenzieren

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

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

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

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

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

Inhaltsübersicht

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

$x_4 := x_1 x_2$

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6/x_3$

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

$x_7 := x_6/x_3$

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

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

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

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

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

5.9 Automatisches Differenzieren

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

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

wird konsequent rekursiv auf alle Teilausdrücke angewandt.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$x_4 := x_1 x_2$

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

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

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

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

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

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

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

$x_5 := e^{x_4}$

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

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

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

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

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

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

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

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

$x_6 := x_4 + x_5$

$x_7 := x_6 / x_3$

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

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

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

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

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

$x_7 := x_6 / x_3$

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$\begin{aligned}
x_4 := x_1 x_2 \quad &\to D_h x_4 = x_2 \nabla_x x_1^T h + x_1 \nabla_x x_2^T h = x_2 D_h x_1 + x_1 D_h x_2 \quad = x_2 h_1 + x_1 h_2 \\
x_5 := e^{x_4} \quad &\to D_h x_5 = e^{x_4} \nabla_x x_4^T h \quad = e^{x_4} D_h x_4 \\
x_6 := x_4 + x_5 \quad &\to D_h x_6 = \quad \dots \quad = D_h x_4 + D_h x_5 \\
x_7 := x_6 / x_3 \quad &\to D_h x_7 = \quad \dots \quad = x_3^{-1} D_h x_6 - x_6 x_3^{-2} D_h x_3
\end{aligned}$$

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

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

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

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

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

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

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

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

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

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

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

Vorwärts-Berechnung:

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

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

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

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

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

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

Vorwärts-Berechnung: x_1

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

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

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

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

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

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

Vorwärts-Berechnung: x_2

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

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

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

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

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

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

Vorwärts-Berechnung: x_3

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

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

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

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

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

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

Vorwärts-Berechnung: x_4

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} =$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} =$	
$x_6 := x_4 + x_5$	$=$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$	
$x_7 := x_6/x_3$	$=$	$\emptyset$		

Vorwärts-Berechnung: x_5

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} =$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} =$	
$x_7 := x_6/x_3$	$=$	$\emptyset$		

Vorwärts-Berechnung: x_6

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5,6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	

Vorwärts-Berechnung: x_7

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_7}$

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_6}$

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_5}$

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

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

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

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

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	$\frac{\partial f}{\partial x_2} = \frac{2}{3} \cdot 2 = \frac{4}{3}$
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_2}$

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	$\frac{\partial f}{\partial x_1} = \frac{2}{3} \cdot 0 = 0$
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	$\frac{\partial f}{\partial x_2} = \frac{2}{3} \cdot 2 = \frac{4}{3}$
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\frac{\partial f}{\partial x_1}$

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

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

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

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

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

x_i		N_i	$\frac{\partial x_j}{\partial x_i}$	$\frac{\partial f}{\partial x_i}$
x_1	$=2$	$\{4\}$	$\frac{\partial x_4}{\partial x_1} = x_2 = 0$	$\frac{\partial f}{\partial x_1} = \frac{2}{3} \cdot 0 = 0$
x_2	$=0$	$\{4\}$	$\frac{\partial x_4}{\partial x_2} = x_1 = 2$	$\frac{\partial f}{\partial x_2} = \frac{2}{3} \cdot 2 = \frac{4}{3}$
x_3	$=3$	$\{7\}$	$\frac{\partial x_7}{\partial x_3} = \frac{x_6}{x_3^2} = \frac{1}{9}$	$\frac{\partial f}{\partial x_3} = 1 \cdot \frac{1}{9} = \frac{1}{9}$
$x_4 := x_1 x_2$	$=0$	$\{5, 6\}$	$\frac{\partial x_5}{\partial x_4} = e^{x_4} = 1$, $\frac{\partial x_6}{\partial x_4} = 1$	$\frac{\partial f}{\partial x_4} = \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 = \frac{2}{3}$
$x_5 := e^{x_4}$	$=1$	$\{6\}$	$\frac{\partial x_6}{\partial x_5} = 1$	$\frac{\partial f}{\partial x_5} = \frac{1}{3} \cdot 1 = \frac{1}{3}$
$x_6 := x_4 + x_5$	$=1$	$\{7\}$	$\frac{\partial x_7}{\partial x_6} = \frac{1}{x_3} = \frac{1}{3}$	$\frac{\partial f}{\partial x_6} = 1 \cdot \frac{1}{3} = \frac{1}{3}$
$x_7 := x_6/x_3$	$=\frac{1}{3}$	$\emptyset$	— $(f = x_7$!!!)	$\frac{\partial f}{\partial x_7} = \frac{\partial x_7}{\partial x_7} = 1$

Rückwärts-Berechnung: $\nabla f = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3})^T = (0, \frac{4}{3}, \frac{1}{9})^T$

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).
- $\nabla f^T h$ ($\nabla^2 fh$) in Rechengenauigkeit! (im Gegensatz zu num. Diff.)

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).
- $\nabla f^T h$ ($\nabla^2 fh$) in Rechengenauigkeit! (im Gegensatz zu num. Diff.)
- Grenzen: AD-Software kommt z.B. nicht immer mit Verzweigungen/bedingten Berechnungen im Source-code zurecht.

Bemerkungen zum Automatischen Differenzieren

- Die Rückwärts-Ber. benötigt den gesamten Berechnungsablauf im Speicher, dafür ist die Laufzeit viel kürzer, als wenn $[\nabla f]_i$ über $\nabla f^T e_i$ für jedes i vorwärts berechnet wird.
- Der Gradient der Funktion $\nabla f(x)^T h$ von x ist $\nabla^2 f(x)h$ und benötigt eine Vorwärts- und Rückwärts-Berechnung. Ohne $\nabla^2 f$ zu bilden, ist so Hessematrix mal Vektor in etwa 12 mal #Operationen zur Berechnung von f berechenbar, aber der Speicherbedarf ist groß.
- Ist f als Source-code (z.B. in C) gegeben, gibt es Software, die daraus automatisch alle oberen Varianten in Orakelform erzeugt (wurde z.B. schon eingesetzt, wenn f über einen Löser für partielle Differentialgleichungen berechnet wurde).
- $\nabla f^T h$ ($\nabla^2 fh$) in Rechengenauigkeit! (im Gegensatz zu num. Diff.)
- Grenzen: AD-Software kommt z.B. nicht immer mit Verzweigungen/bedingten Berechnungen im Source-code zurecht.

Verfahren, die $\nabla^2 f(x)$ nicht explizit benötigen, sondern nur eine Routine, die $\nabla^2 f(x)$ mal Vektor berechnet, heißen „Hessian-free methods“.

Inhaltsübersicht

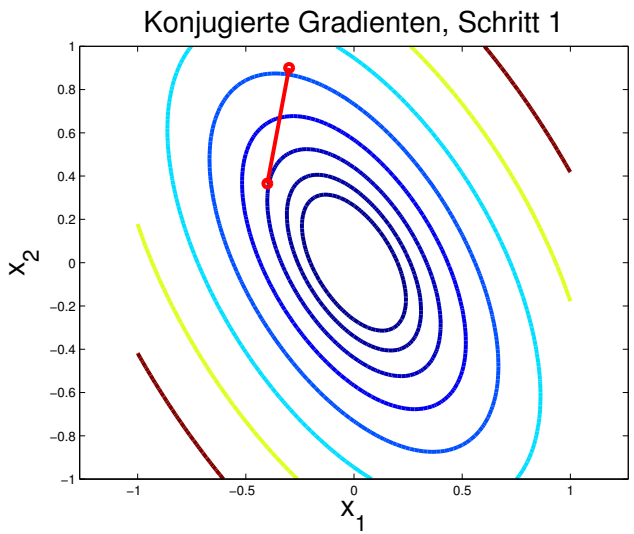
5.10 Das lineare Konjugierte-Gradienten-Verfahren

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.

5.10 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.

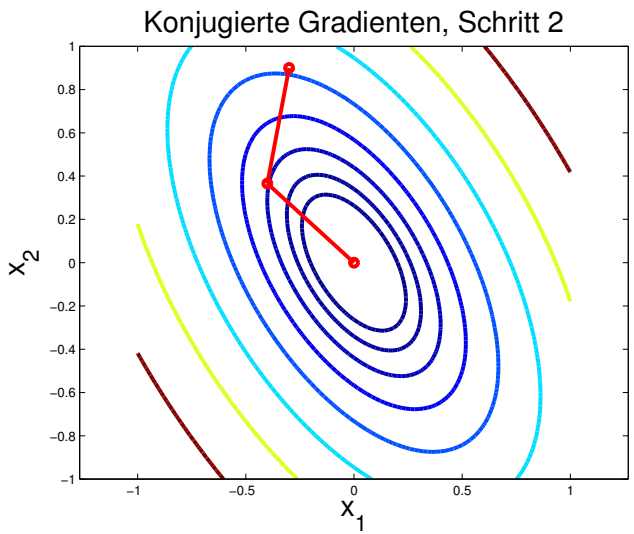
Beginnend mit steilstem Abstieg wählt es im jeweils nächsten, mit exaktem Line-Search bestimmten Punkt die nächste Richtung so, dass alle alten Richtungen keine Verbesserung mehr bringen $\rightarrow$ konjugierte Richtungen.



5.10 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.



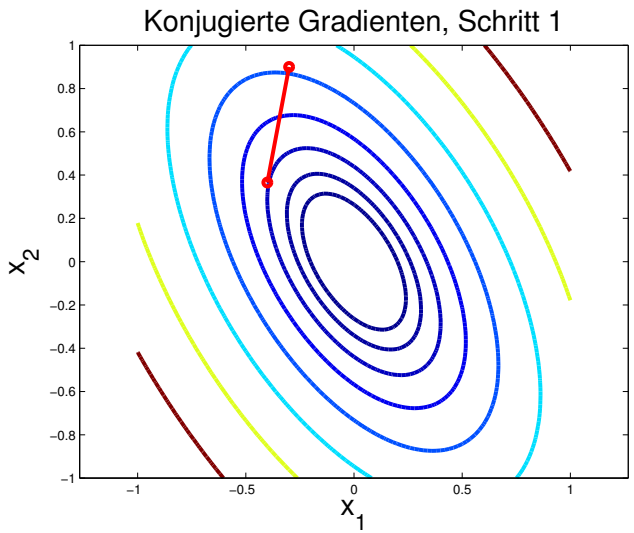
5.10 Das lineare Konjugierte-Gradienten-Verfahren

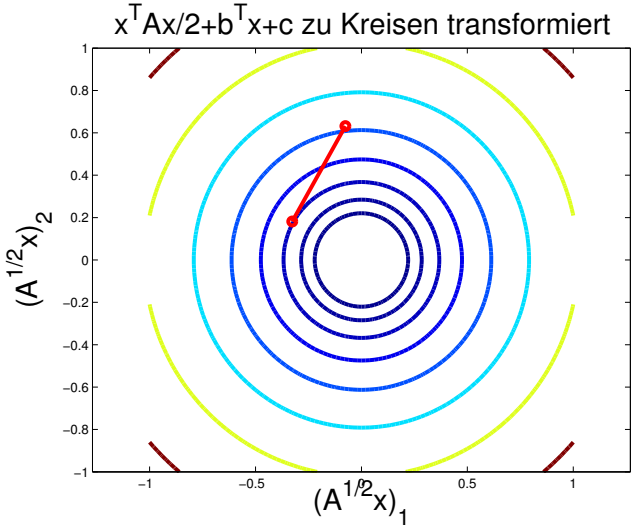
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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 $\longrightarrow$ 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.





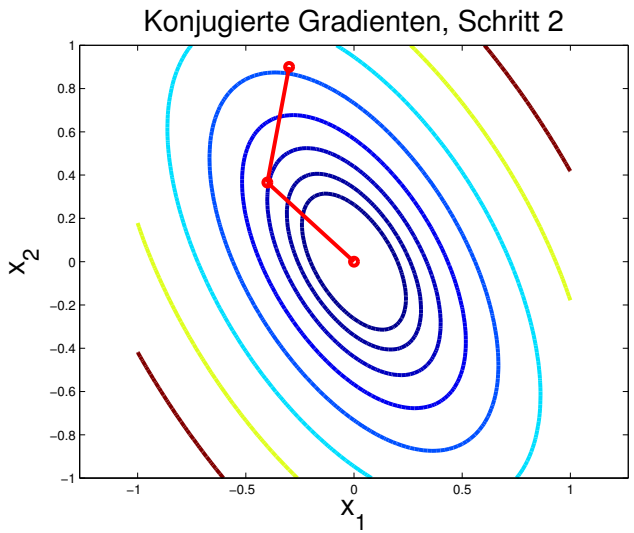
5.10 Das lineare Konjugierte-Gradienten-Verfahren

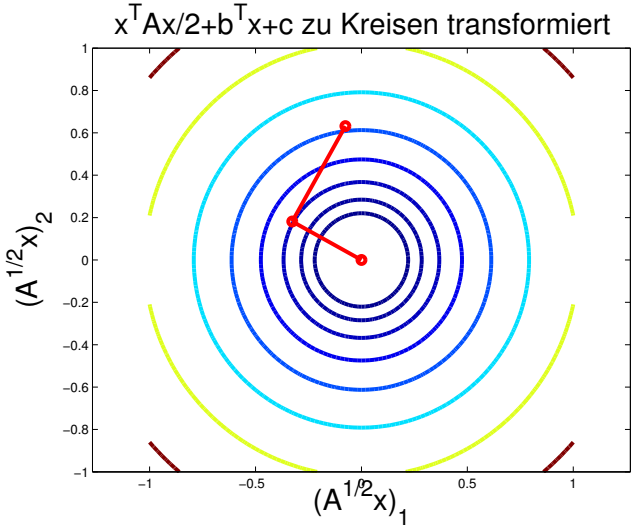
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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!]

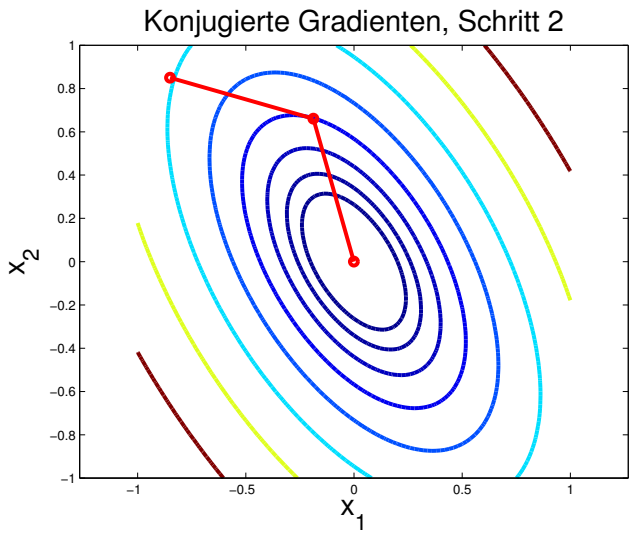
5.10 Das lineare Konjugierte-Gradienten-Verfahren

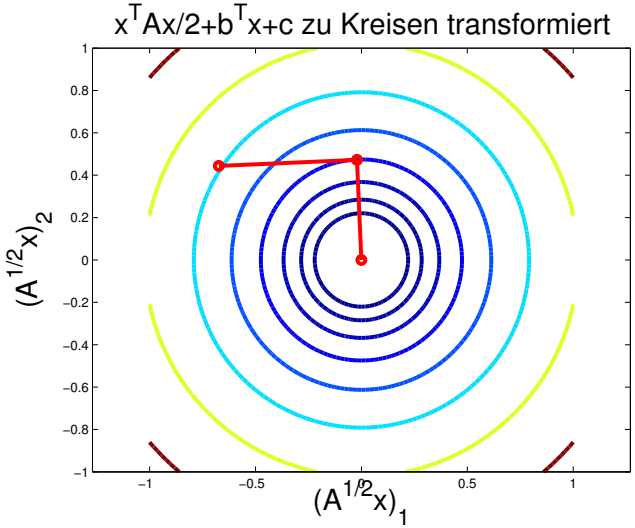
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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!]

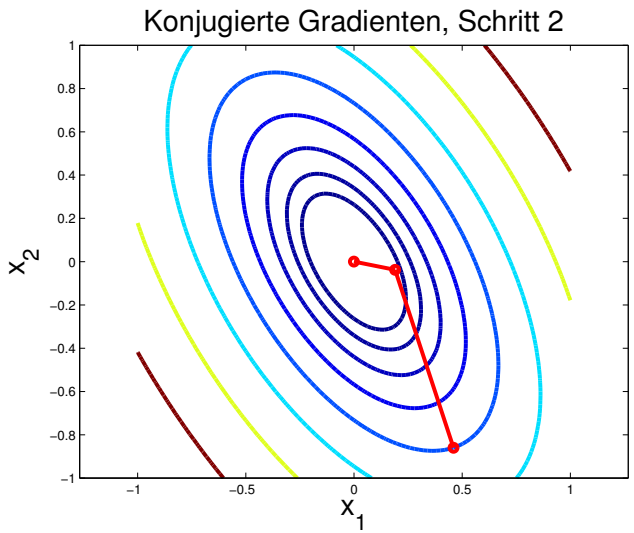
5.10 Das lineare Konjugierte-Gradienten-Verfahren

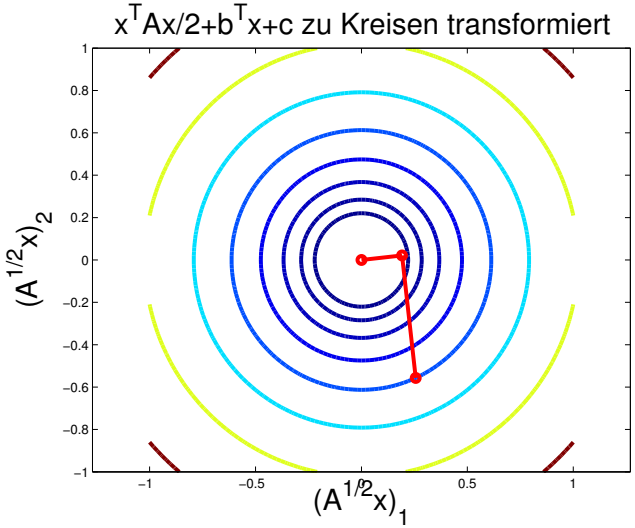
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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!]

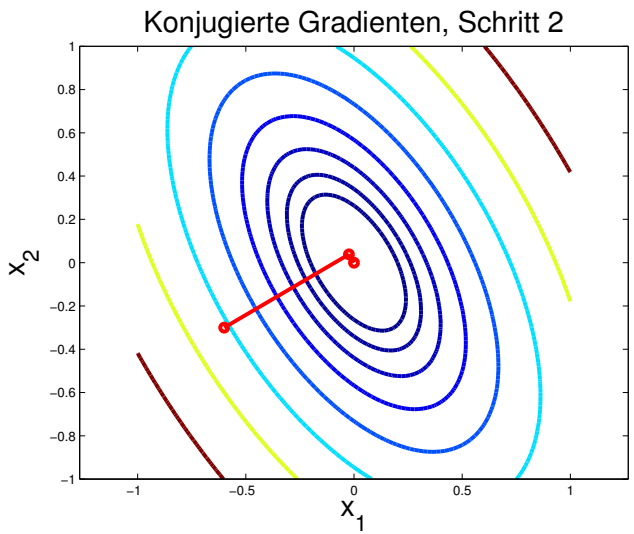
5.10 Das lineare Konjugierte-Gradienten-Verfahren

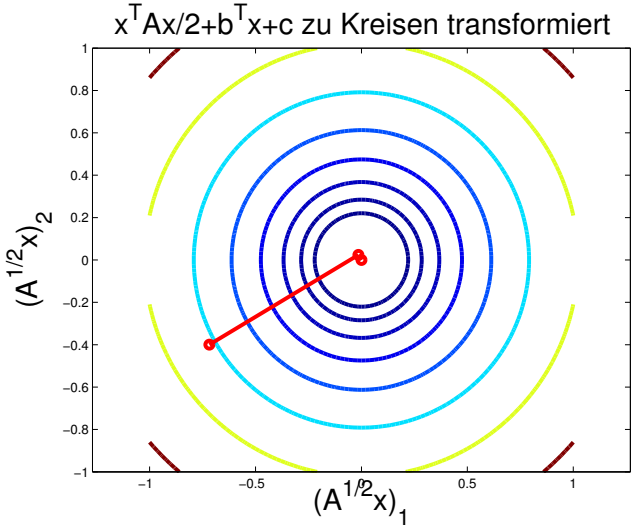
Wir betrachten zuerst nur $\min f(x) = \frac{1}{2}x^T Ax - b^T x + c$ mit $A \succ 0$.
In diesem Spezialfall hinreichende Opt.-Bed.: $\nabla f(x) = 0 \Leftrightarrow Ax = b$
Das **lineare CG-Verfahren** (conjugate gradients) löst große pos. def. Gleichungssysteme iterativ nur mit Matrix-Vektor-Multiplikationen.

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. $r^{(k+1)} = Ax^{(k+1)} - b = r^{(k)} + \alpha_k A h^{(k)}$ $\quad [\in \operatorname{span}\{r^{(0)}, \ldots, A^{k+1} r^{(0)}\}]$

   Für jedes $i \leq k$ gilt $(h^{(i)})^T r^{(k+1)} = 0$.

Aus exaktem Line-Search folgt $(r^{(k+1)})^T h^{(k)} = 0$

und für jedes $0 \leq i < k$, wegen $x^{(k+1)} = x^{(i+1)} + \sum_{j=i+1}^{k} \alpha_j h^{(j)}$,

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

Die Schritte des linearen CG-Verfahrens

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

Wähle β_{k+1} so, dass $h^{(k+1)}$ konjugiert zu $h^{(k)}$: $0 = (h^{(k+1)})^T A h^{(k)}$, also
$0 = -(r^{(k+1)})^T A h^{(k)} + \beta_{k+1} (h^{(k)})^T A h^{(k)} \Rightarrow \beta_k = \frac{(r^{(k+1)})^T A h^{(k)}}{(h^{(k)})^T A h^{(k)}}$.

Für jedes $0 \leq i \leq k$ gilt (mit $\beta_0 = 0$) wegen $r^{(i)} = \beta_i h^{(i-1)} - h^{(i)}$
$(r^{(i)})^T r^{(k+1)} = \beta_i (h^{(i-1)})^T r^{(k+1)} - (h^{(i)})^T r^{(k+1)} = 0$
und jedes $0 \leq i < k$ wegen $A h^{(i)} = (r^{(i+1)} - r^{(i)})/\alpha_i$
$(h^{(k+1)})^T A h^{(i)} = -(r^{(k+1)})^T A h^{(i)} + \beta_{k+1} \underbrace{(h^{(k)})^T A h^{(i)}}_{=0} = -\frac{(r^{(k+1)})^T (r^{(i+1)} - r^{(i)})}{\alpha_i} = 0$

Die Schritte des linearen CG-Verfahrens

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

Die Schritte des linearen CG-Verfahrens

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

Satz (Lineares CG-Verfahren)

Ist in Iteration k der Gradient $r_k \neq 0$, so gilt
$\operatorname{span}\{r^{(0)}, \ldots, A^k r^{(0)}\} = \operatorname{span}\{r^{(0)}, \ldots, r^{(k)}\} = \operatorname{span}\{h^{(0)}, \ldots, h^{(k)}\}$
und $(r^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T r^{(k)} = 0$, $(h^{(i)})^T A h^{(k)} = 0$ *für* $0 \leq i < k$.
Insbesondere ist $x^{(k)} = \operatorname{argmin}\{f(x) : x \in x^{(0)} + \operatorname{span}\{r^{(0)}, \ldots, A^{k-1} r^{(0)}\}\}$
und nach höchstens n Iterationen endet das Verfahren mit $r^{(k)} = 0$.

Der lineare CG-Algorithmus in Standardform

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

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

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

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k+1$, GOTO 1.

Der lineare CG-Algorithmus in Standardform

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

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

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

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k+1$, GOTO 1.

- Pro Iteration wird nur eine Matrix-Vektor-Multiplikation benötigt.
  [Besonders effizient für dünn besetzte Matrix A oder Hessian-free]

Der lineare CG-Algorithmus in Standardform

Nutzt man die Orthogonalität, lässt sich der Alg. weiter vereinfachen:

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

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

Input: $A \succ 0$, $b \in \mathbb{R}^m$, $x^{(0)} \in \mathbb{R}^n$, $\varepsilon > 0$

0. $r^{(0)} = Ax^{(0)} - b$, $h^{(0)} = -r^{(0)}$, $k = 0$
1. Falls $\|r_k\| < \varepsilon$ STOP.
2. $\alpha_k = \frac{\|r^{(k)}\|^2}{(h^{(k)})^T A h^{(k)}}$
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. $r^{(k+1)} = r^{(k)} + \alpha_k A h^{(k)}$
5. $\beta_{k+1} = \frac{\|r^{(k+1)}\|^2}{\|r^{(k)}\|^2}$
6. $h^{(k+1)} = -r^{(k+1)} + \beta_{k+1} h^{(k)}$
7. $k \leftarrow k+1$, GOTO 1.

- Pro Iteration wird nur eine Matrix-Vektor-Multiplikation benötigt.
  [Besonders effizient für dünn besetzte Matrix A oder Hessian-free]
- Stellenauslöschung verursacht numerische Probleme, $k \leq n$ reicht nicht.

$\rightarrow$ In der Praxis ist nur mit Präkonditionierung gute Genauigkeit erzielbar.

[Präkond. $\approx$ Koordinatentrafo Richtung Kreis, stark problemabhängig!]

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz
- Gilbert-Nocedal: $\beta_{k+1}^{GN} = \max\{\beta_{k+1}^{PR}, 0\}$ praktisch gut, mit Konvergenz

Das nichtlineare Konjugierte-Gradienten-Verfahren

Ersetze im linearen CG-Verf. $r^{(k)}$ durch ∇f_k und α_k-Formel durch Line-Search.

Input: $x^{(0)} \in \mathbb{R}^n$, Orakel 1. Ordnung, $\varepsilon > 0$

0. $\nabla f_0 = \nabla f(x^{(0)})$, $h^{(0)} = -\nabla f_0$, $k = 0$
1. Falls $\|\nabla f_k\| < \varepsilon$ STOP.
2. Bestimme α_k durch Line-Search.
3. $x^{(k+1)} = x^{(k)} + \alpha_k h^{(k)}$
4. Berechne $\nabla f_{k+1} = \nabla f(x^{(k+1)})$.
5. $\beta_{k+1}^{FR} = \frac{\|\nabla f_{k+1}\|^2}{\|\nabla f_k\|^2}$
6. $h^{(k+1)} = -\nabla f_{k+1} + \beta_{k+1}^{FR} h^k$
7. $k \leftarrow k+1$, GOTO 1.

Schwierigkeiten: wegen des inexakten LS ist i.A. $\nabla f_{k+1}^T h^{(k)} \neq 0$ und manchmal bleibt $\nabla f_{k+1}^T \nabla f_k$ groß $\rightarrow$ Korrekturversuche über Wahl von β:

- Fletcher-Reeves: β_{k+1}^{FR} mit verschärften Wolfe-Bed. $\rightarrow$ Abstiegsrichtung
- Polak-Ribière: $\beta_{k+1}^{PR} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}$ praktisch gut, ohne Konvergenz
- Gilbert-Nocedal: $\beta_{k+1}^{GN} = \max\{\beta_{k+1}^{PR}, 0\}$ praktisch gut, mit Konvergenz
- Regelm. Neustart oder $\beta_k = 0$ ist jeweils ein steilster Abstiegsschritt

$\rightarrow$ global konvergent

Inhaltsübersicht

5.11 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.

5.11 Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Iterative Verfahren brechen ab, wenn das Residuum

$$r^{(k)} = \nabla^2 f_k h^{(k)} + \nabla f_k$$

klein genug ist. Inexakte Newton-Verfahren fordern

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

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

5.11 Inexakte Newton-Verfahren

Für Glgssysteme $Ax = b$ mit $A \succ 0$ bestimmt das lineare CG-Verfahren schnell eine Näherungslösung durch Matrix-Vektor Multiplikationen.

Idee: Bestimme Newtonrichtung $h_N^{(k)}$ näherungsweise durch Lösen von

$$\nabla^2 f_k h_N^{(k)} = -\nabla f_k,$$

und nutze z.B. automatisches Differenzieren für Hessematrix mal Vektor.

Iterative Verfahren brechen ab, wenn das Residuum

$$r^{(k)} = \nabla^2 f_k h^{(k)} + \nabla f_k$$

klein genug ist. Inexakte Newton-Verfahren fordern

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

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

Satz (Inexakte Newton-Verfahren)

Sei f hinr. glatt und x^* erfülle die hinr. Opt.-Bed. Ist $x^{(0)}$ nahe genug bei x^*, so gilt für jede Punktfolge $x^{(k+1)} = x^{(k)} + h^{(k)}$ mit $h^{(k)}$ erfüllt (INC):
Falls $\eta_k \leq \eta < 1$, konvergiert $x^{(k)}$ linear gegen x^*.
Falls $\eta_k \to 0$, konvergiert $x^{(k)}$ superlinear gegen x^*.
Falls $\eta_k \leq \gamma \|\nabla f_k\|$ mit festem $\gamma > 0$, konvergieren die $x^{(k)}$ quadratisch.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$,[„Newton“-Schritt] sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
   Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$,[„Newton“-Schritt]
   sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.
3. $k \leftarrow k + 1$, GOTO 1.

- Das CG-Verfahren garantiert: $h^{(k)}$ ist Abstiegsrichtung.
  $p^{(i)}$ minimiert $m(p) = \frac{1}{2} p^T \nabla^2 f_k p + \nabla f_k^T p$ über $\operatorname{span}\{d^{(0)}, \ldots, d^{(i-1)}\}$.
  Weil $p^{(i)} = \sum_{j=0}^{i-1} \xi_j d^{(j)}$, $(d^{(j)})^T \nabla^2 f_k d^{(j)} > 0$ und $(d^{(j)})^T \nabla^2 f_k d^{(l)} = 0$ $(j \neq l)$
  folgt $0 = m(0) \geq m(p^{(i)}) = \nabla f_k^T p^{(i)} + \frac{1}{2} \sum_{j=0}^{i-1} \xi_j^2 (d^{(j)})^T \nabla^2 f_k d^{(j)}$

Das Line-Search-Newton-CG-Verfahren

Im Folgenden bezeichnen $x^{(k)}$, $h^{(k)}$ die üblichen Punkt/Richtungsfolgen und $p^{(i)}$, $d^{(i)}$ die des CG-Verfahrens. Die η_k hier ergeben superlineare K.

---

0. Wähle Startpunkt $x^{(0)}$, setze $k = 0$.
1. Berechne $h^{(k)}$ mit dem linearen CG-Verfahren angewandt auf $\nabla^2 f_k p = -\nabla f_k$ mit Startpunkt $p^{(0)} = 0$, $i = 0$. Beende CG, sobald
   a) $(d^{(i)})^T \nabla^2 f_k d^{(i)} \leq 0$ [CG-Richtung negativer Krümmung, $\nabla^2 f_k \not\succ 0$]
      Ist $i = 0$ (erster CG-Schritt), setze $h^{(k)} := -\nabla f_k$, sonst $h^{(k)} := p^{(i)}$.
   b) $\|\nabla f_k + \nabla^2 f_k p^{(i)}\| \leq \min\{\frac{1}{2}, \sqrt{\|\nabla f_k\|}\}\|\nabla f_k\|$, setze $h^{(k)} := p^{(i)}$.
2. Erfüllt $x^{(k)} + h^{(k)}$ Armijo, setze $x^{(k+1)} := x^{(k)} + h^{(k)}$,[„Newton“-Schritt] sonst bestimme α_k mit $x^{(k+1)} := x^{(k)} + \alpha_k h^{(k)}$ erfüllt Wolfe.
3. $k \leftarrow k + 1$, GOTO 1.

- Das CG-Verfahren garantiert: $h^{(k)}$ ist Abstiegsrichtung.
- CG benötigt nur Hessematrix mal Vektor $\rightarrow$ *Hessian-free*
- Immer zuerst Schrittlänge 1 für superlineare Konvergenz versuchen!
- CG braucht problemspezifische Präkonditionierer, gefährdet gute Konv.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h $[g = \nabla f$, Newton: $B = \nabla^2 f]$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h $[g = \nabla f$, Newton: $B = \nabla^2 f]$

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h $[g = \nabla f$, Newton: $B = \nabla^2 f]$

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.

1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.

2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$

Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$
8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$
8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$
9. $i \leftarrow i + 1$, GOTO 1.

Das Steihaug CG-Verf. für das Trust-Region-Problem

Input: Model $m(h) = f + g^T h + \frac{1}{2} h^T B h$, Radius $\Delta > 0$, $\varepsilon_{opt} > 0$, $\varepsilon_{term} > 0$
Output: Schritt h [$g = \nabla f$, Newton: $B = \nabla^2 f$]

0. $p^{(0)} := 0$, $r^{(0)} := g$, $d^{(0)} := -r^{(0)}$, $i := 0$. Ist $\|r_0\| < \varepsilon_{opt}$, RETURN $h := p^{(0)}$.
1. Ist $(d^{(i)})^T B d^{(i)} \leq 0$ (negative Krümmung), finde OL $\bar{\alpha}$ zu $\min_\alpha \{m(p) : p = p^{(i)} + \alpha d^{(i)}, \|p\| = \Delta\}$, RETURN $h := p^{(i)} + \bar{\alpha} d^{(i)}$.
2. $\alpha_i := \frac{\|r^{(i)}\|^2}{(d^{(i)})^T B d^{(i)}}$ [exakter LS für $m(\cdot)$ in Richtung $d^{(i)}$]
3. $p^{(i+1)} := p^{(i)} + \alpha_i d^{(i)}$
4. Ist $\|p^{(i+1)}\| \geq \Delta$, [verlässt Trust Region]
   finde $\bar{\alpha} \geq 0$ mit $\|p^{(i)} + \bar{\alpha}_i d^{(i)}\| = \Delta$, RETURN $h := p^{(i)} + \bar{\alpha}_i d^{(i)}$
5. $r^{(i+1)} := r^{(i)} + \alpha_i B d^{(i)}$
6. Ist $\|r^{(i+1)}\| < \varepsilon_{term} \|r^{(0)}\|$, RETURN $h := p^{(i+1)}$. [Inexakter Newton]
7. $\beta_{i+1} := \frac{\|r^{(i+1)}\|^2}{\|r^{(i)}\|^2}$
8. $d^{(i+1)} := -r^{(i+1)} + \beta_{i+1} d^{(i)}$
9. $i \leftarrow i + 1$, GOTO 1.

- $p^{(1)}$ ist Cauchy-Punkt, CG-Verfahren verbessert monoton $\Rightarrow$ global konv.
- Benötigt auch nur Hessematrix mal Vektor $\rightarrow$ Hessian-free
- Es gilt $\|p^{(0)}\| < \cdots < \|p^{(i)}\| < \cdots \leq \Delta \Rightarrow$ lokal inexakter Newton

Inhaltsübersicht

5.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2} \sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

5.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.

5.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.
Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte $\tilde{y}$ mit Wahrscheinlichkeit

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

auftreten.

5.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.
Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte $\tilde{y}$ mit Wahrscheinlichkeit

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

auftreten. Betrachtet man $p(\tilde{y}; x, t)$ als Likelihood-Funktion dafür, dass x die korrekten Parameter sind, ist die beste Parameterwahl für $\tilde{y}$ und t in diesem Sinn ein Maximum-Likelihood-Schätzer $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

5.12 Nichtlineare kleinste Quadrate

Typisches Problem für optimale Parameterwahl: Minimiere ein f der speziellen Gestalt $f(x) = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x)$ mit $r_j : \mathbb{R}^n \to \mathbb{R}$ glatt.

Bsp: An m Messpunkten t_j werden Werte $\tilde{y}_j \in \mathbb{R}$ mit unabhängigen, $(0, \sigma^2)$-normalverteilten Messfehlern gemessen. Für die korrekten Werte y_j wird vermutet, dass sie sich für geeignet gewählte Parameter x als Funktion $y_j = \Phi(t_j; x)$ der Messpunkte darstellen lassen. Bestimme x.
Hat ein Messfehler von $\varepsilon \in \mathbb{R}$ die „Wahrscheinlichkeit“

$$\varphi(\varepsilon) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{\varepsilon^2}{2\sigma^2}},$$

würden für korrekte Parameter x die Messwerte $\tilde{y}$ mit Wahrscheinlichkeit

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

auftreten. Betrachtet man $p(\tilde{y}; x, t)$ als Likelihood-Funktion dafür, dass x die korrekten Parameter sind, ist die beste Parameterwahl für $\tilde{y}$ und t in diesem Sinn ein Maximum-Likelihood-Schätzer $\bar{x} \in \operatorname{Argmax}_x p(\tilde{y}; x, t)$.

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

Spezialfall: Lineare kleinste Quadrate

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

Spezialfall: Lineare kleinste Quadrate

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

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

x^* ist Optimallösung $\Leftrightarrow$ x^* erfüllt die Normalgleichungen $A^TAx = A^Tb$.

Spezialfall: Lineare kleinste Quadrate

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

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

x^* ist Optimallösung $\Leftrightarrow$ x^* erfüllt die Normalgleichungen $A^TAx = A^Tb$.

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

- Cholesky-Faktorisierung von A^TA (mit pivotisieren),
- QR-Faktorisierung von A (recht stabil, wird aber dicht besetzt),
- SVD-Dekomposition (am stabilsten und teuersten)
- Präkonditionierte CG-Verfahren (PCG)
  (für Näherungslösung im *large scale*-Bereich, falls Av und A^Tw billig)

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).
Betrachte Gradienten und Hessematrix von f (Kettenregel nutzen):

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

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

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

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

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

Nichtlineare kleinste Quadrate mit kleinen Residuen

$$\min_x f(x) := \frac{1}{2}\|r(x)\|^2 = \frac{1}{2}\sum_{j=1}^{m} r_j^2(x) \quad \text{mit} \quad r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_m(x) \end{bmatrix}$$

und in einer Umgebung von x^* sei $\|r(x)\|$ klein (=kleine Residuen).
Betrachte Gradienten und Hessematrix von f (Kettenregel nutzen):

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

$\longrightarrow$ die Jacobimatrix liefert eine gute Näherung für die Hessematrix.

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

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

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

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

5.12.1 Das Gauss-Newton-Verfahren

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

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

5.12.1 Das Gauss-Newton-Verfahren

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

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

Vorteile:

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

5.12.1 Das Gauss-Newton-Verfahren

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

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

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\mathrm{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) 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 $J_k h_k^{GN} = 0 \Leftrightarrow J_k^T r_k = \nabla f_k = 0$.

5.12.1 Das Gauss-Newton-Verfahren

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

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

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\mathrm{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) 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.

5.12.1 Das Gauss-Newton-Verfahren

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

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

Vorteile:

- Es muss keine zweite Ableitung berechnet werden.
- Ist r_k klein (kleine Residuen), ist $J_k^T J_k$ gute Näherung für $\nabla^2 f$
- Ist $\mathrm{Rang}(J_k) = n$ ($J_k^T J_k \succ 0$) 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 $\mathrm{Rang}(J_k) < n$, 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

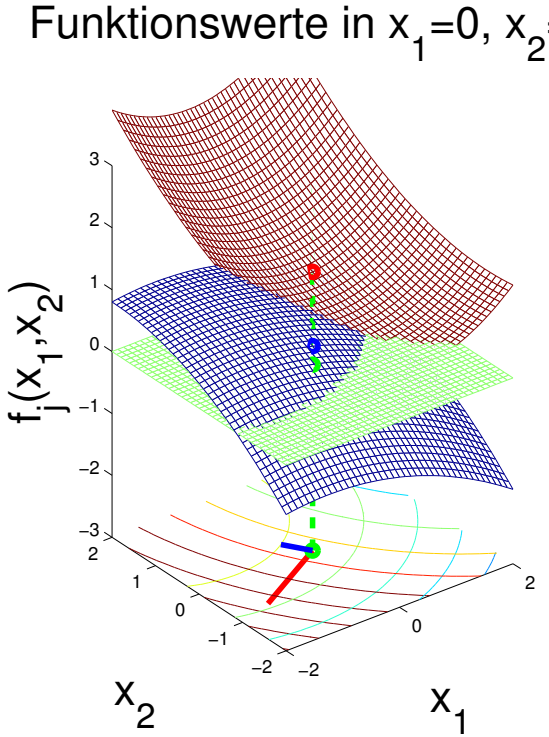
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

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

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



5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

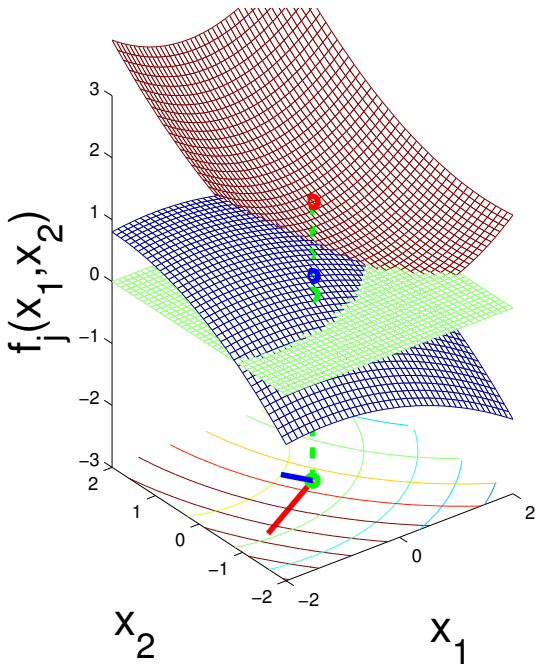
Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

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

Bsp1: Primal-Duales System für LP

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

Funktionswerte in x_1=0, x_2=0



5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

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

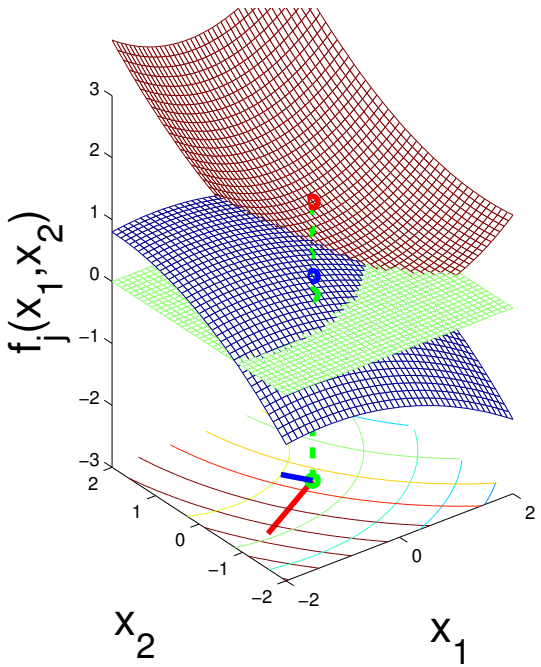
Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Bsp1: Primal-Duales System für LP

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

Bsp2: Opt.-Bed. für $\min_x f(x)$

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



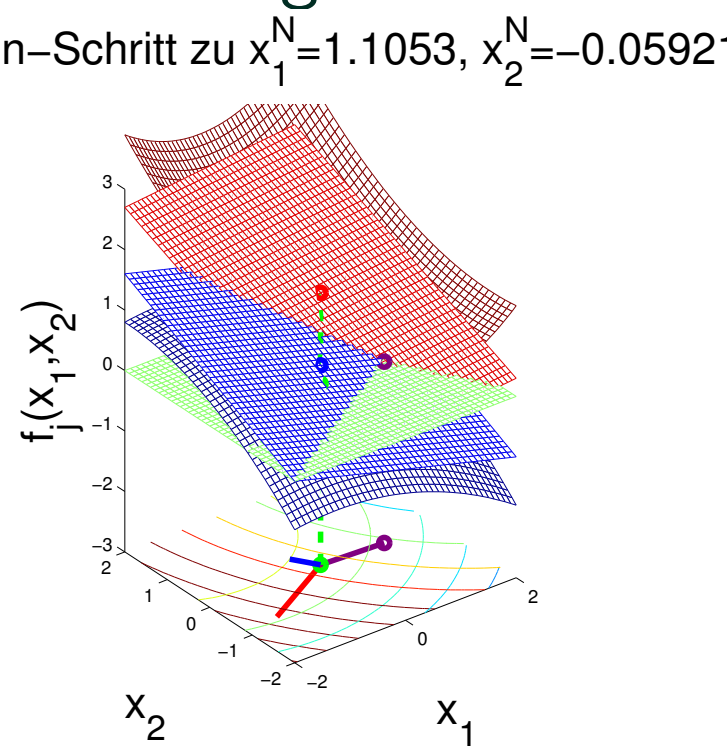
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

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

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



Bsp1: Primal-Duales System für LP

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

Bsp2: Opt.-Bed. für $\min_x f(x)$

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

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

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

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

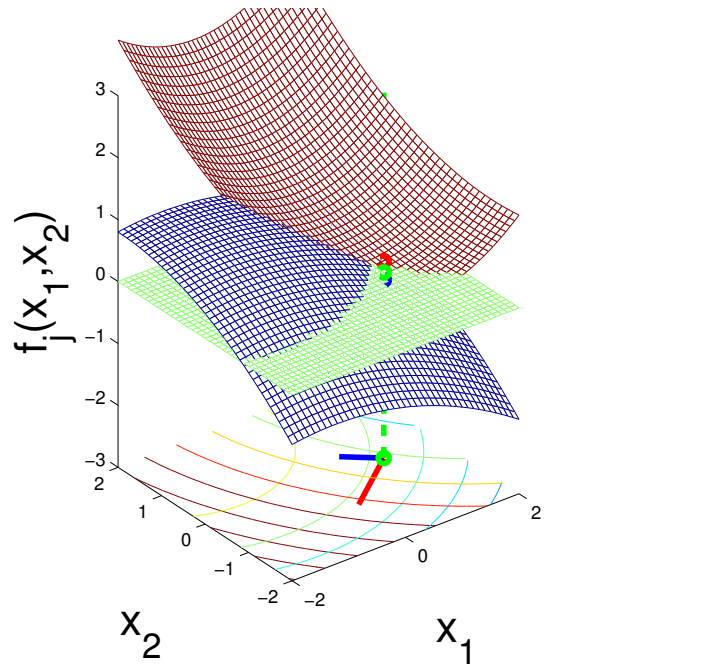
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

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

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



Bsp1: Primal-Duales System für LP

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

Bsp2: Opt.-Bed. für $\min_x f(x)$

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

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

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

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

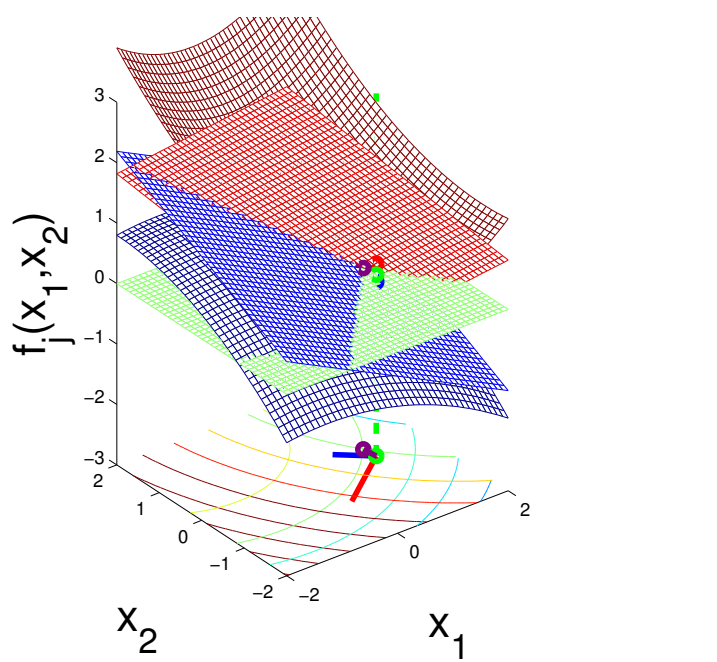
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

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

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



Bsp1: Primal-Duales System für LP

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

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

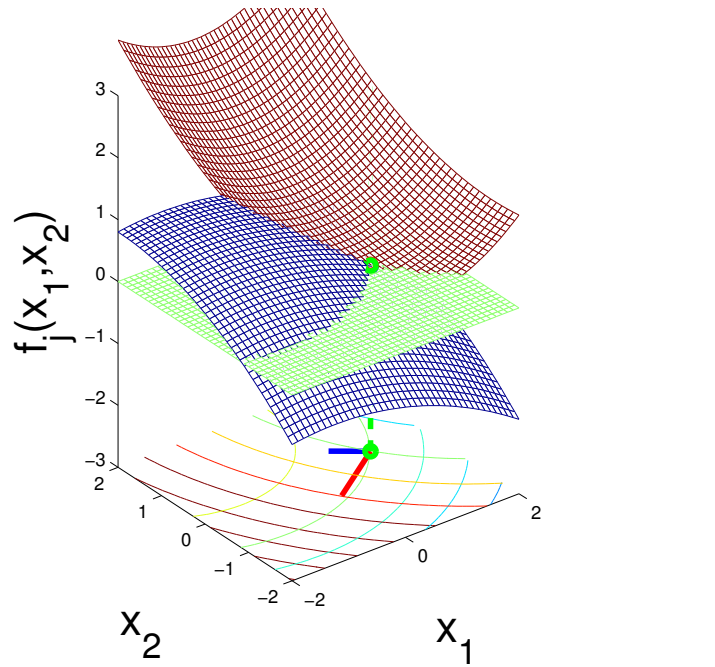
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

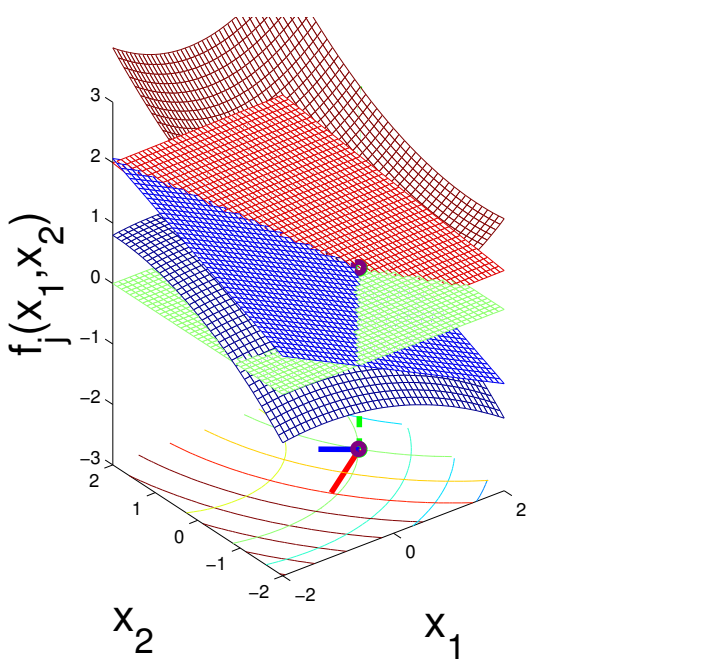
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

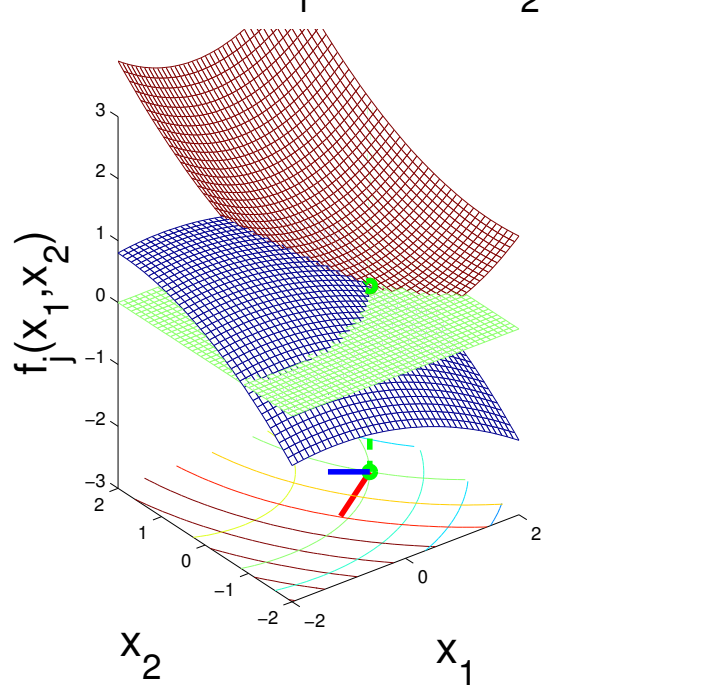
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

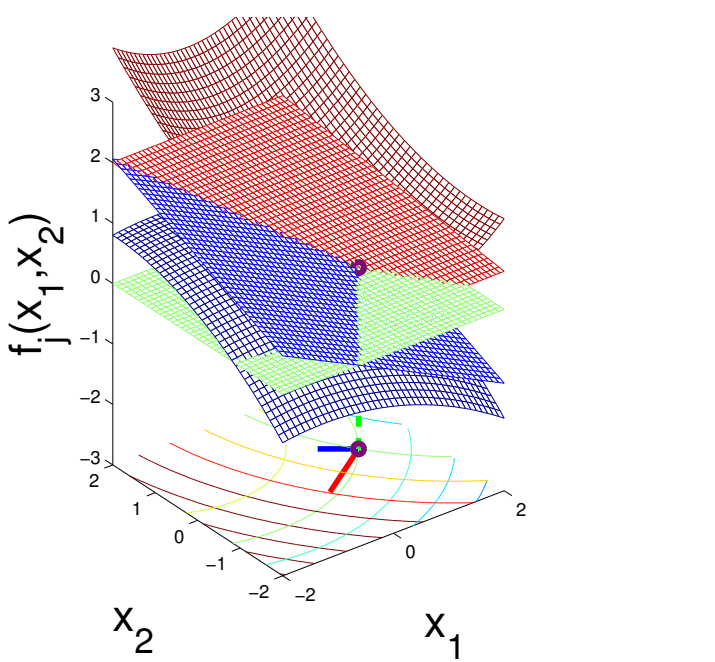
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

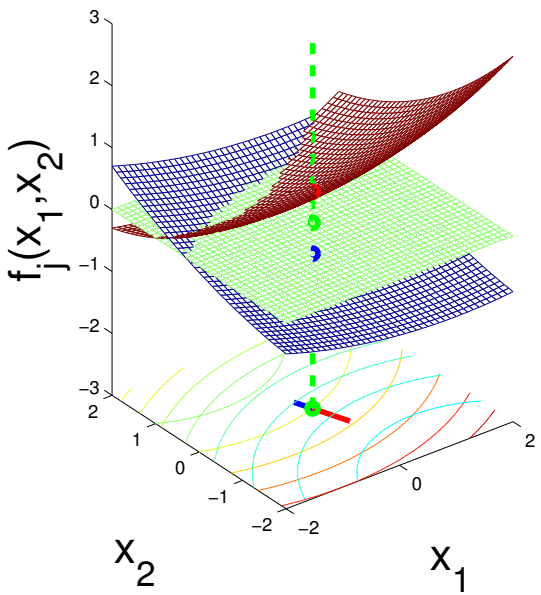
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

Ist $J_F(x^{(k)})$ singulär ($\Leftrightarrow$ die ∇f_j lin. abh.), versagt es.

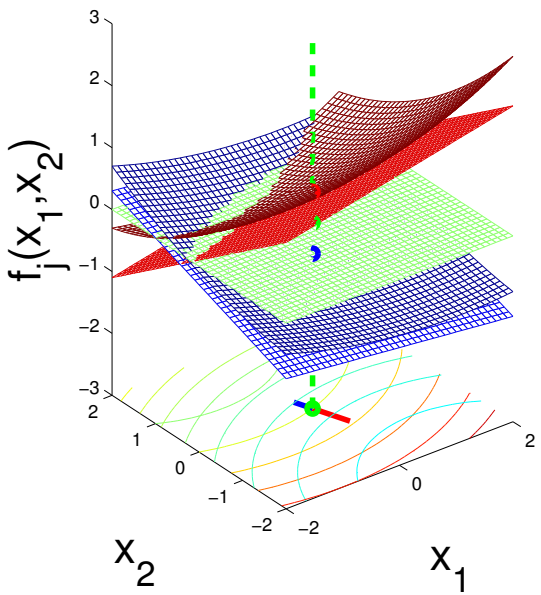
5.13 Newton für nichtlineare Gleichungen

Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

Ist $J_F(x^{(k)})$ singulär ($\Leftrightarrow$ die ∇f_j lin. abh.), versagt es.

5.13 Newton für nichtlineare Gleichungen

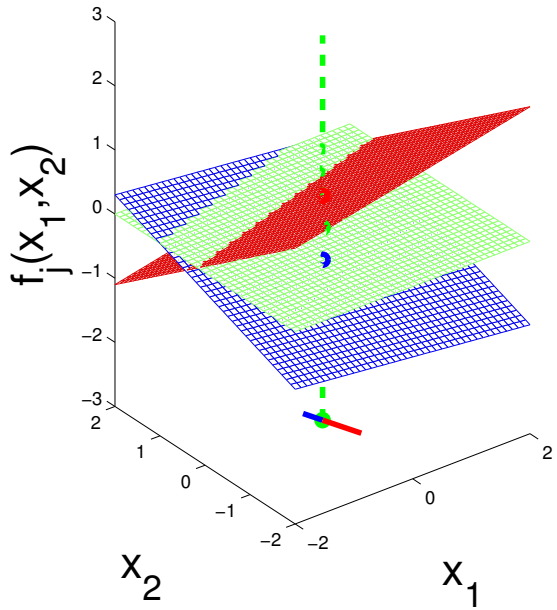
Lösung nichtlinearer Gleichungen

(n Unbekannte, n Gleichungen)

Gegeben: $F : \mathbb{R}^n \to \mathbb{R}^n$

Gesucht: $x \in \mathbb{R}^n$ mit $F(x) = 0$

Rang(Jacobimatrix)=1



Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) := \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$

$F(x) := \nabla f(x) = 0$

Newton-Verfahren: Bestimme $x^{(k+1)}$ als Lösung der Linearisierung in $x^{(k)}$,

$$F(x^{(k)} + h) \approx F(x^{(k)}) + J_F(x^{(k)})h = \begin{bmatrix} f_1(x^{(k)}) \\ \vdots \\ f_n(x^{(k)}) \end{bmatrix} + \begin{bmatrix} \nabla f_1(x^{(k)})^T \\ \vdots \\ \nabla f_n(x^{(k)})^T \end{bmatrix} h = 0$$

Für $J_F(x^{(k)})$ regulär (invertierbar) ist der Newton-Schritt die eindeutige Lösung $h_{NF}^{(K)} := -J_F(x^{(k)})^{-1} F(x^{(k)})$ und $x^{(k+1)} := x^{(k)} + h_{NF}^{(K)}$.

Ist $J_F(x^{(k)})$ singulär ($\Leftrightarrow$ die ∇f_j lin. abh.), versagt es.

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$, $F(x) := \nabla f(x) = 0$, $J_F(x) = \nabla^2 f(x)$

$$0 = F(x) + J_F(x)h = \nabla f(x) + \nabla^2 f(x) h \Rightarrow h_N = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$, $F(x) := \nabla f(x) = 0$, $J_F(x) = \nabla^2 f(x)$

$$0 = F(x) + J_F(x)h = \nabla f(x) + \nabla^2 f(x) h \Rightarrow h_N = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Zur Globalisierung wird der Fortschritt über eine *merit-function*, meist $f(x) := \frac{1}{2}\|F(x)\|^2$ (s. nichtlin. kleinste Quadrate), bewertet. h_{NF} ist (falls definiert) Abstiegsrichtung für f: $\nabla f_k^T h_{NF}^{(k)} = -F_k^T J_k J_k^{-1} F_k = -\|F_k\|^2 < 0$.

Ein Punkt $\bar{x} \in \mathbb{R}^n$ heißt reguläre Lösung von $F(x) = 0$, falls $F(\bar{x}) = 0$ und $J_F(\bar{x})$ regulär. Für diese konvergiert Newton lokal quadratisch.

Satz (Newton-Verfahren für Nichtlineare Gleichungen)

Sei x^* eine reguläre Lösung von $F(x) = 0$, J_F Lipschitz-stetig in einer Umgebung von x^* und $x^{(0)}$ hinreichend nahe an x^*, dann konvergiert die Folge $x^{(k+1)} = x^{(k)} - J_F(x^{(k)})^{-1} F(x^{(k)})$ quadratisch gegen x^*.

Bsp1: Primal-Duales System für LP

$$F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) + J_F\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} A^T y + z - c \\ Ax - b \\ x \circ z - \mu \mathbf{1} \end{bmatrix} + \begin{bmatrix} 0 & A^T & I \\ A & 0 & 0 \\ \mathrm{Diag}(z) & 0 & \mathrm{Diag}(x) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = 0$$

Bsp2: Opt.-Bed. für $\min_x f(x)$, $F(x) := \nabla f(x) = 0$, $J_F(x) = \nabla^2 f(x)$

$$0 = F(x) + J_F(x)h = \nabla f(x) + \nabla^2 f(x) h \Rightarrow h_N = -\nabla^2 f(x)^{-1} \nabla f(x)$$

Zur Globalisierung wird der Fortschritt über eine *merit-function*, meist $f(x) := \frac{1}{2}\|F(x)\|^2$ (s. nichtlin. kleinste Quadrate), bewertet. h_{NF} ist (falls definiert) Abstiegsrichtung für f: $\nabla f_k^T h_{NF}^{(k)} = -F_k^T J_k J_k^{-1} F_k = -\|F_k\|^2 < 0$.
Vorsicht: Lokale Minima der Merit-Funktion sind i.A. keine Lösungen!

Inhaltsübersicht

Inhaltsübersicht

6.1 Restringierte Nichtlineare Optimierung

Aufgabenstellung:

$$(P) \qquad \begin{array}{lllll} \min & f(x) & & & \text{Zielfunktion} \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} & & \text{Gleichungsnebenbedingungen} \\ & g_i(x) \leq 0 & i \in \mathcal{I} & & \text{Ungleichungsnebenbedingungen} \\ & x \in \mathbb{R}^n & & & \text{Grundmenge (meist } \mathbb{R}^n\text{, manchmal } [0,1]^n) \end{array}$$

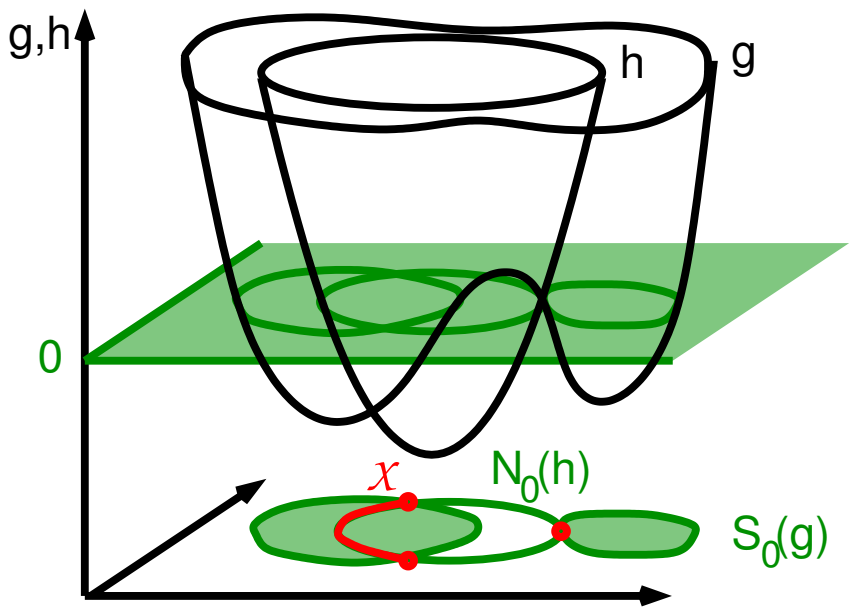
mit f, g_i, $h_i : \mathbb{R}^n \to \mathbb{R}$ hinreichend glatt und $\mathcal{E}$, $\mathcal{I}$ endliche Indexmengen.

6.1 Restringierte Nichtlineare Optimierung

Aufgabenstellung:

$$(P) \qquad \begin{array}{lll} \min & f(x) & \text{Zielfunktion} \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbedingungen} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbedingungen} \\ & x \in \mathbb{R}^n & \text{Grundmenge (meist } \mathbb{R}^n\text{, manchmal } [0,1]^n) \end{array}$$

mit f, g_i, $h_i : \mathbb{R}^n \to \mathbb{R}$ 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, $\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i)$.



6.1 Restringierte Nichtlineare Optimierung

Aufgabenstellung:

$$(P) \qquad \begin{array}{lll} \min & f(x) & \text{Zielfunktion} \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} & \text{Gleichungsnebenbedingungen} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} & \text{Ungleichungsnebenbedingungen} \\ & x \in \mathbb{R}^n & \text{Grundmenge (meist } \mathbb{R}^n\text{, manchmal } [0,1]^n) \end{array}$$

mit f, g_i, $h_i : \mathbb{R}^n \to \mathbb{R}$ 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

$$f(x^*) \leq f(x) \quad \forall x \in \mathcal{X} \cap B_\varepsilon(x^*) := \{x : \|x - x^*\| \leq \varepsilon\} \text{ für ein } \varepsilon > 0.$$

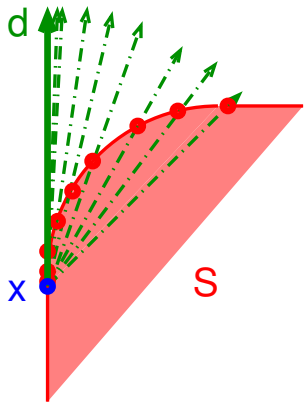
Für Algorithmen ist diese Beschreibung ungeeignet. Wir benötigen eine algebraische Beschreibung der lokalen Optimalität, die für gegebenes $\bar{x} \in \mathbb{R}^n$ nur auf den Funktionswerten und Gradienten (und eventuell ∇^2) von f, g_i, h_i in $\bar{x}$ beruht (Orakel 1./2. Ordnung für f, g_i, h_i).
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

6.2 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 $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

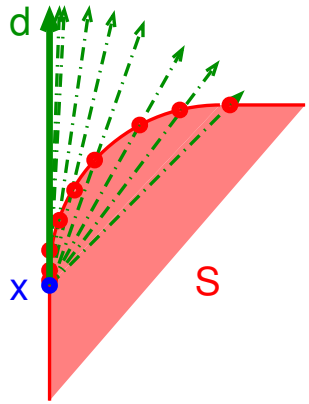
[Die α_k strecken die Vektoren auf die Länge von d]



6.2 Zulässige Richtungen und Tangentialkegel

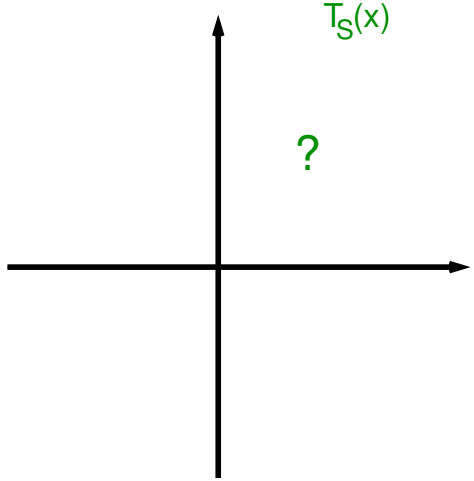
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

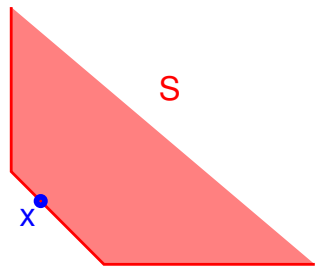
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

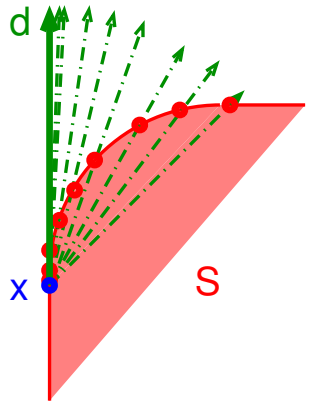




6.2 Zulässige Richtungen und Tangentialkegel

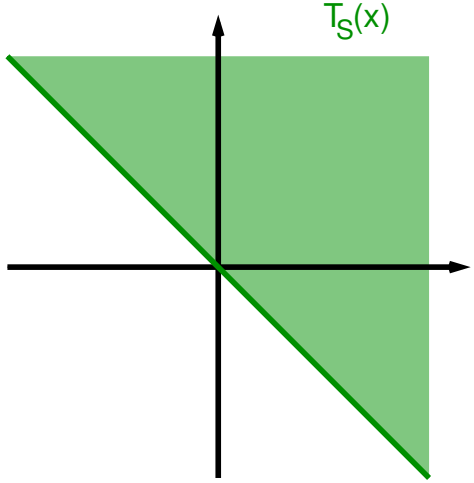
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

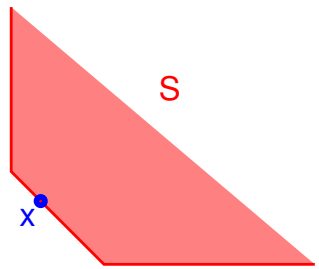
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

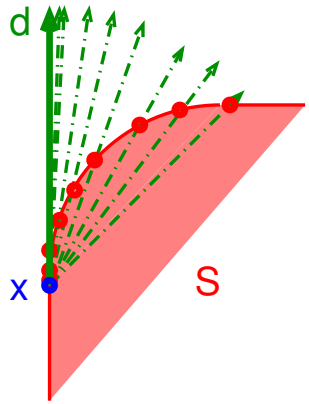




6.2 Zulässige Richtungen und Tangentialkegel

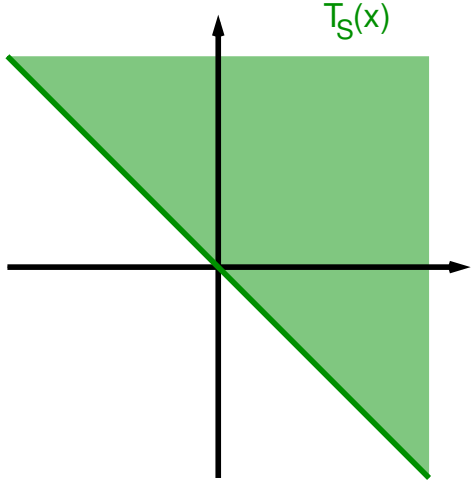
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

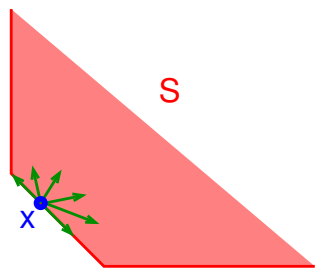
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

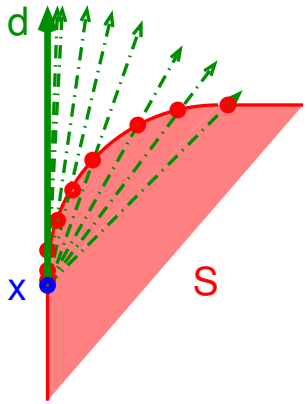




6.2 Zulässige Richtungen und Tangentialkegel

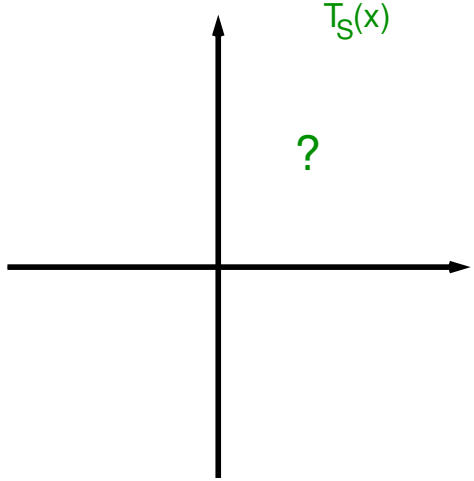
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

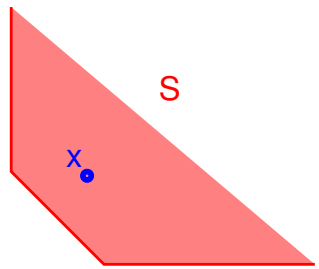
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

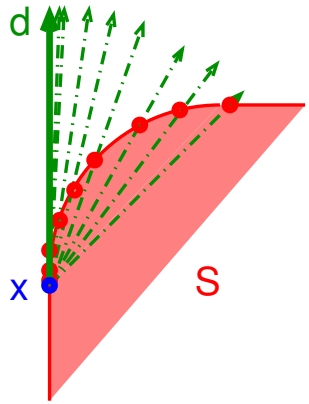




6.2 Zulässige Richtungen und Tangentialkegel

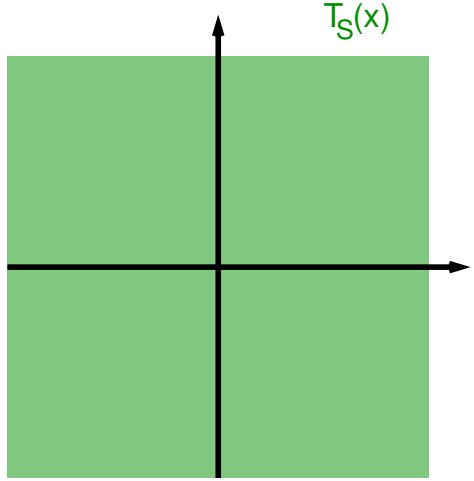
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

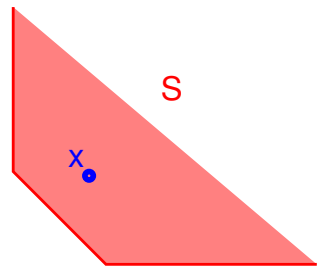
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

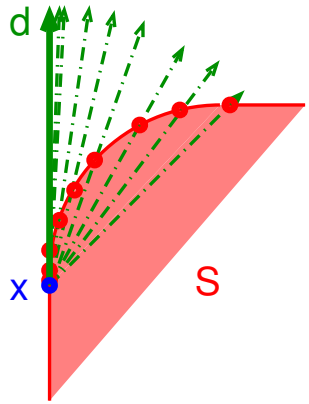




6.2 Zulässige Richtungen und Tangentialkegel

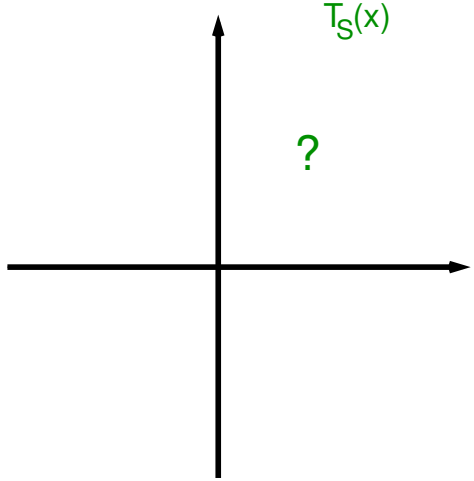
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

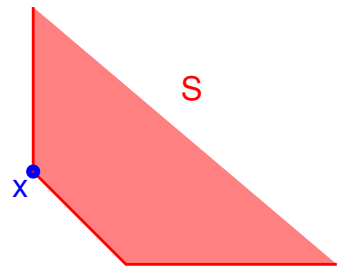
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

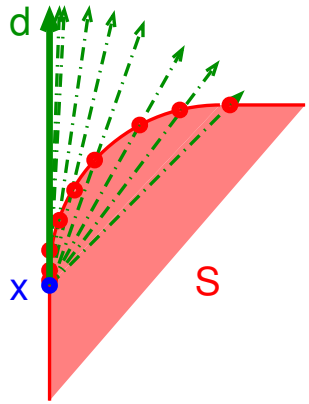




6.2 Zulässige Richtungen und Tangentialkegel

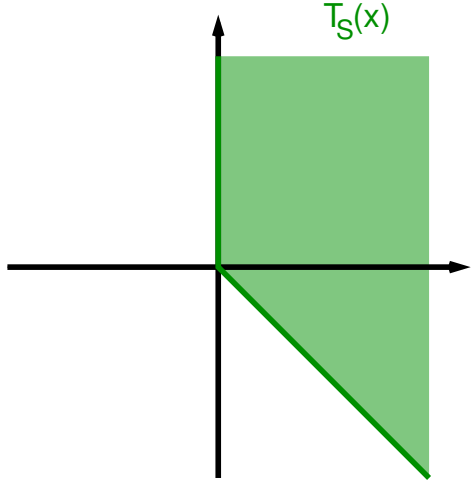
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

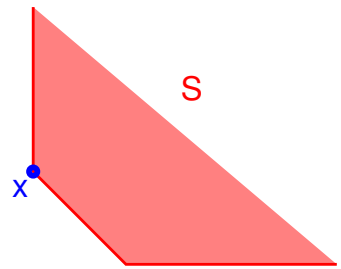
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

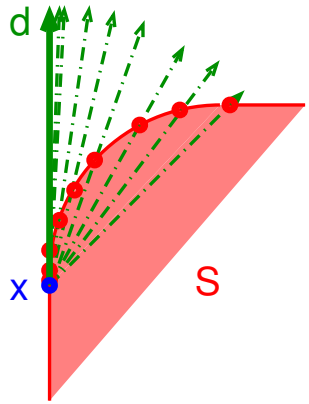




6.2 Zulässige Richtungen und Tangentialkegel

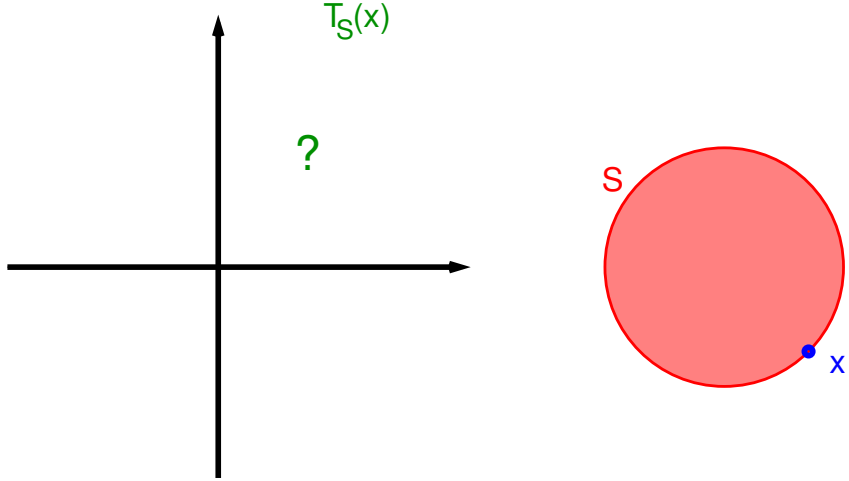
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

[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.

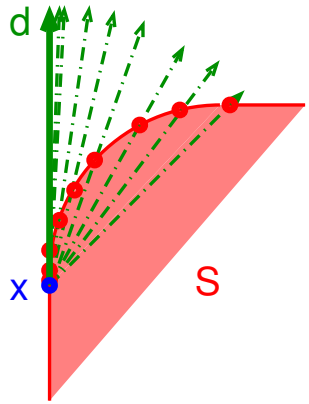
[$T_S(x)$ ist Kegel, denn für $\alpha \geq 0$ ist mit $d \in T_S(x)$ auch $\alpha d \in T_S(x)$: Für $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]



6.2 Zulässige Richtungen und Tangentialkegel

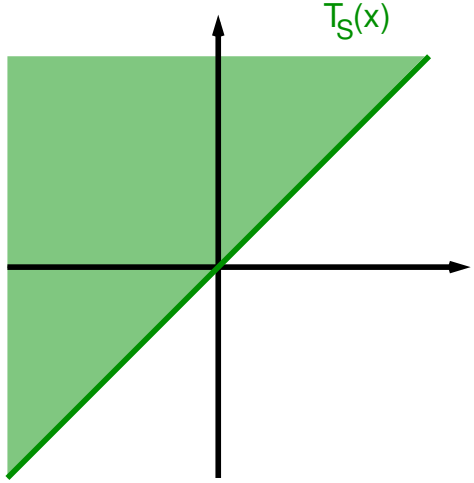
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

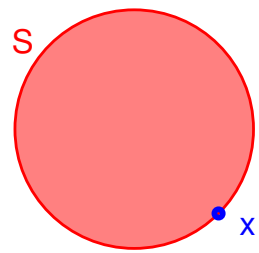
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

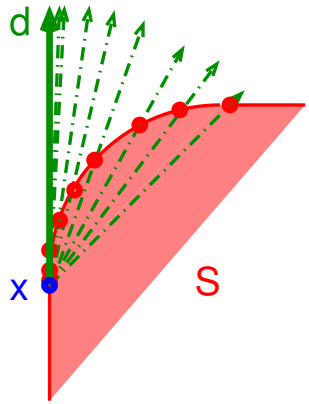




6.2 Zulässige Richtungen und Tangentialkegel

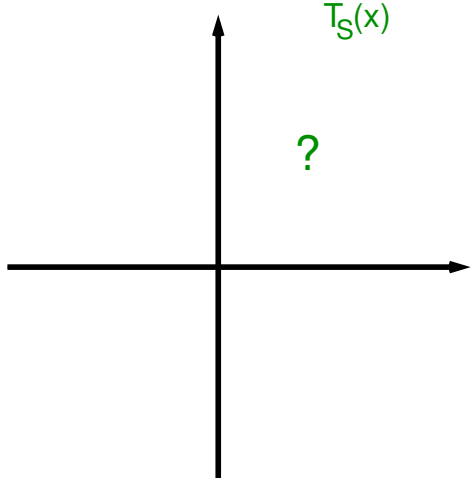
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

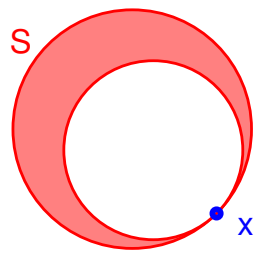
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

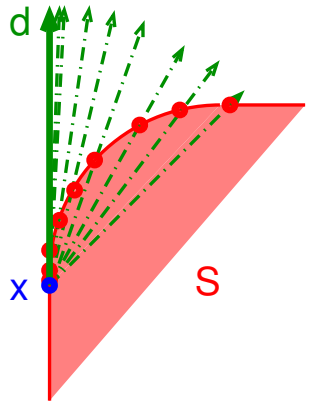




6.2 Zulässige Richtungen und Tangentialkegel

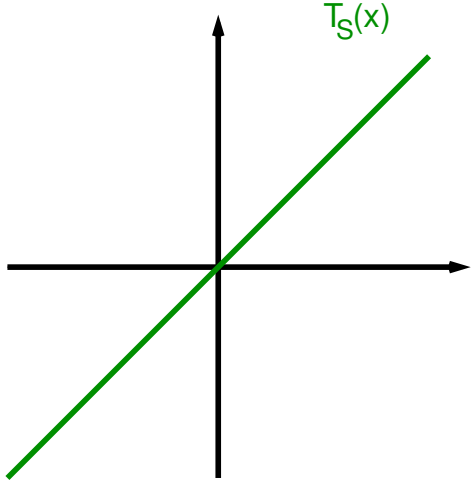
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

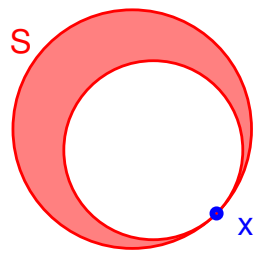
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

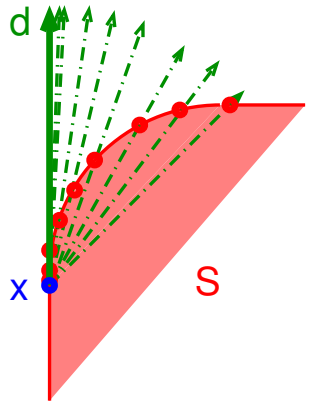




6.2 Zulässige Richtungen und Tangentialkegel

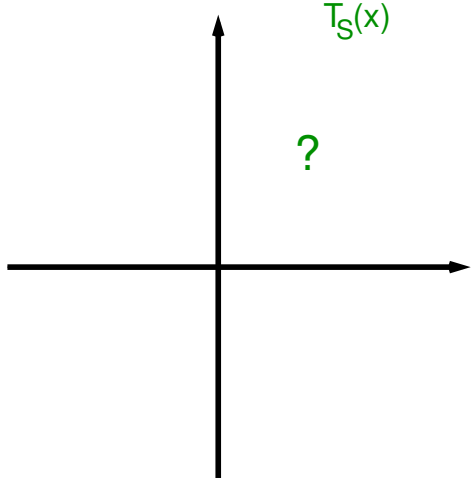
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

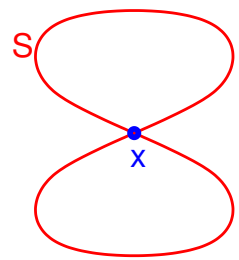
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]

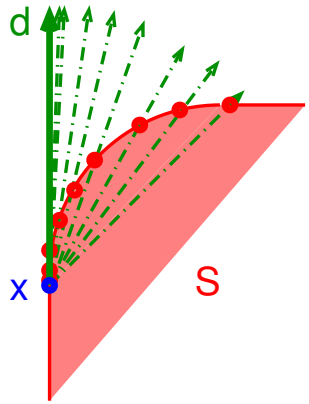




6.2 Zulässige Richtungen und Tangentialkegel

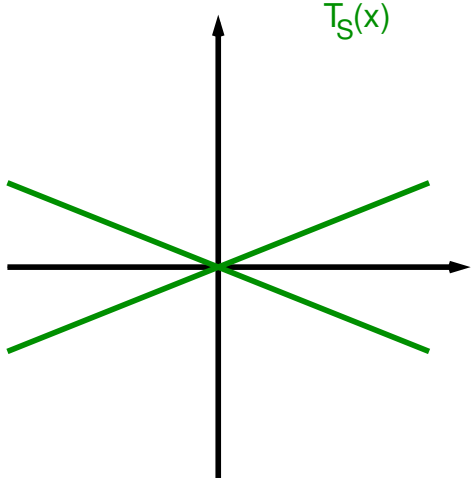
Für eine Menge $S \subseteq \mathbb{R}^n$ und ein $x \in S$ heißt $d \in \mathbb{R}^n$ zulässige Richtung (Tangentialrichtung), wenn es eine Folge $x^{(k)} \in S$ $(k \in \mathbb{N})$ mit $\lim_{k\to\infty} x^{(k)} = x$ und $\alpha_k \geq 0$ gibt mit $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$.

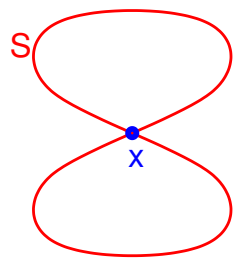
[Die α_k strecken die Vektoren auf die Länge von d]



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 $d = \lim_{k\to\infty} \alpha_k(x^{(k)} - x)$ ist $\alpha d = \lim_{k\to\infty} \bar{\alpha}_k(x^{(k)} - x)$ mit $\bar{\alpha}_k := \alpha\alpha_k$.]





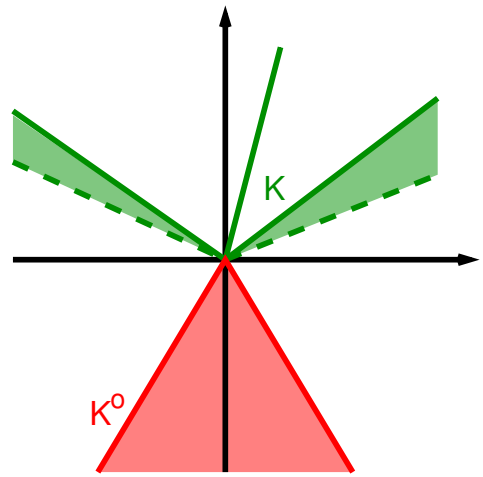
Der Polarkegel

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

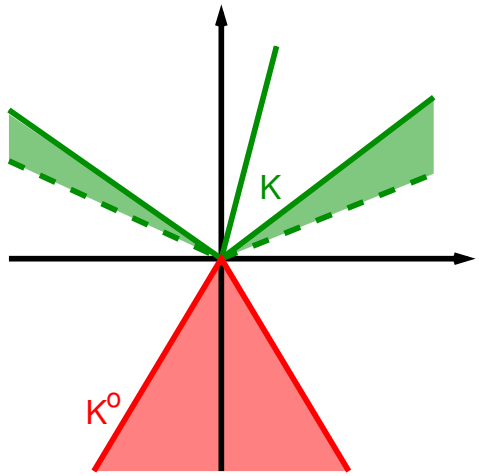
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

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

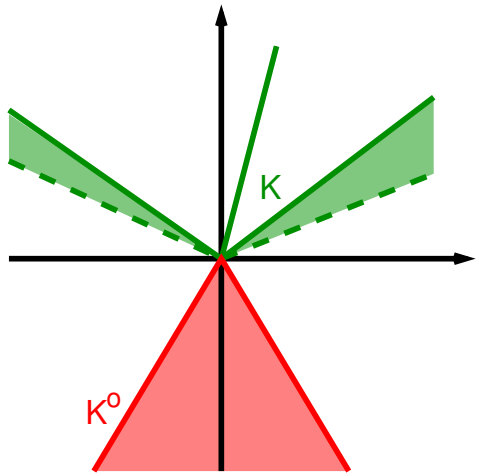
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

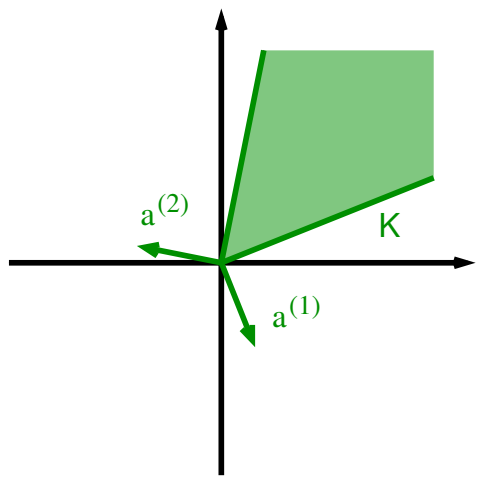
den Polarkegel zu K.

[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 $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \ldots, m\}$, was ist K°?



Der Polarkegel

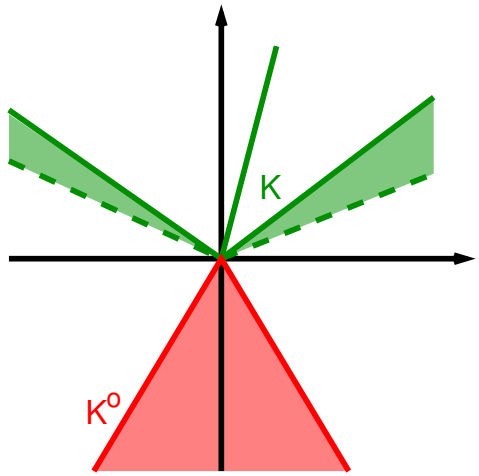
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \quad \forall d \in K\}$$

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 $\bar{K}^\circ \subseteq K^\circ$.

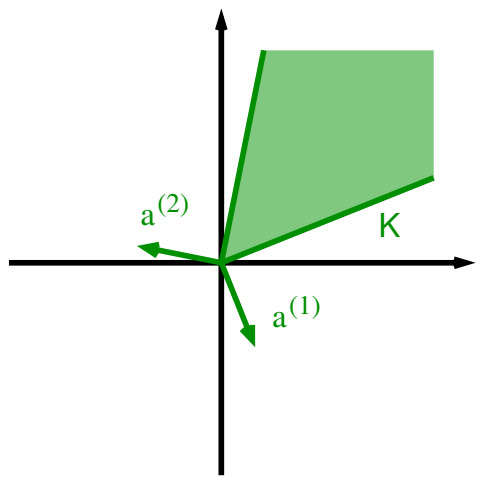


Gegeben $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \ldots, m\}$, was ist K°?

Setze dazu $A := [a^{(1)}, \ldots, a^{(m)}]$.

$$p \in K^\circ \Leftrightarrow$$

$$\begin{aligned} 0 = \max \quad & p^T d \\ \text{s.t.} \quad & A^T d \leq 0 \\ & d \text{ frei} \end{aligned}$$



Der Polarkegel

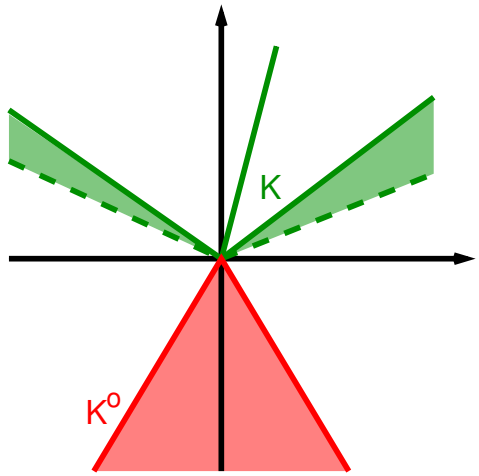
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \ \ \forall d \in K\}$$

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 $\bar{K}^\circ \subseteq K^\circ$.

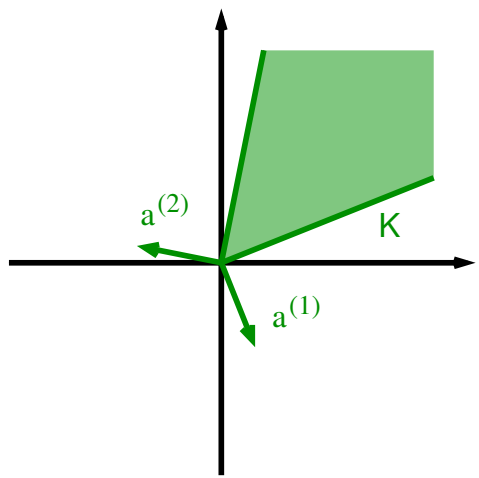


Gegeben $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \ldots, m\}$, was ist K°?

Setze dazu $A := [a^{(1)}, \ldots, a^{(m)}]$.

$$p \in K^\circ \Leftrightarrow$$

$$\begin{array}{rll} 0 = \max & p^T d & \\ \text{s.t.} & A^T d \leq 0 & \Leftrightarrow \\ & d \text{ frei} & \end{array} \qquad \begin{array}{rll} 0 = \min & 0^T \lambda & \\ \text{s.t.} & A\lambda = p & \\ & \lambda \geq 0 & \end{array}$$



Der Polarkegel

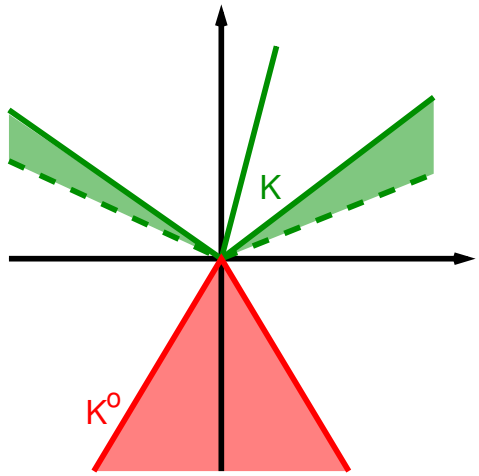
Ist $K \subseteq \mathbb{R}^n$ ein Kegel, nennt man

$$K^\circ := \{p \in \mathbb{R}^n : p^T d \leq 0 \ \ \forall d \in K\}$$

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 $K = \{d \in \mathbb{R}^n : (a^{(i)})^T d \leq 0, i = 1, \ldots, m\}$, was ist K°?

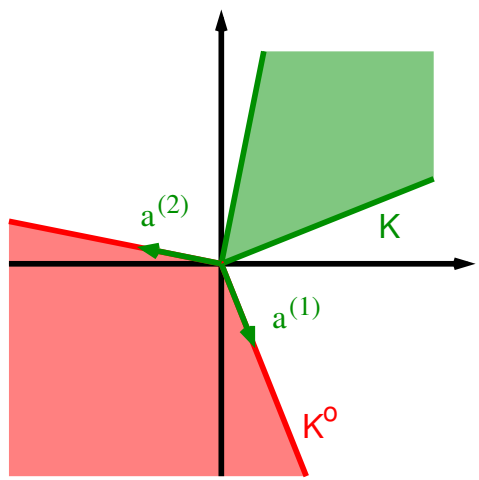
Setze dazu $A := [a^{(1)}, \ldots, a^{(m)}]$.

$$p \in K^\circ \Leftrightarrow$$

$$\begin{array}{rll} 0 = \max & p^T d & \\ \text{s.t.} & A^T d \leq 0 & \Leftrightarrow \\ & d \text{ frei} & \end{array} \qquad \begin{array}{rll} 0 = \min & 0^T \lambda & \\ \text{s.t.} & A\lambda = p & \\ & \lambda \geq 0 & \end{array}$$

$$\Leftrightarrow \exists \lambda \geq 0 : A\lambda = p$$

Also ist $K^\circ = \{p \in \mathbb{R}^n : p = \sum_{i=1}^m \lambda_i a^{(i)}, \lambda_i \geq 0\}$.



Inhaltsübersicht

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x \in \mathcal{X}} f(x)$

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x \in \mathcal{X}} f(x)$

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 $\mathcal{X} \ni x^{(k)} \to x^*$, für große k

$$f(x^*) \leq \; f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|)$$

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x \in \mathcal{X}} f(x)$

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 $\mathcal{X} \ni x^{(k)} \longrightarrow x^*$, für große k

$$f(x^*) \leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|)$$

$$\Rightarrow \quad \nabla f(x^*)^T (x^{(k)} - x^*) \geq -\mathbf{o}(\|x^{(k)} - x^*\|) \qquad | \cdot \alpha_k \geq 0$$

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x \in \mathcal{X}} f(x)$

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 $\mathcal{X} \ni x^{(k)} \to x^*$, für große k

$$
\begin{aligned}
& f(x^*) \leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T (x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\
\Rightarrow \quad & \nabla f(x^*)^T (x^{(k)} - x^*) \geq -\mathbf{o}(\|x^{(k)} - x^*\|) \qquad | \cdot \alpha_k \geq 0 \\
\Rightarrow \quad & \underbrace{\nabla f(x^*)^T [\alpha_k (x^{(k)} - x^*)]}_{\lim \to \nabla f(x^*)^T d} \geq - \underbrace{\mathbf{o}(\alpha_k \|x^{(k)} - x^*\|)}_{\lim \to \mathbf{o}(\|d\|) = 0}
\end{aligned}
$$

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x\in\mathcal{X}} f(x)$

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 $\mathcal{X} \ni x^{(k)} \to x^*$, für große k

$$
\begin{aligned}
f(x^*) \leq\ & f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T(x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\
\Rightarrow \quad & \nabla f(x^*)^T(x^{(k)} - x^*) \geq -\mathbf{o}(\|x^{(k)} - x^*\|) \qquad | \cdot \alpha_k \geq 0 \\
\Rightarrow \quad & \underbrace{\nabla f(x^*)^T[\alpha_k(x^{(k)} - x^*)]}_{\lim\to \nabla f(x^*)^T d} \geq -\underbrace{\mathbf{o}(\alpha_k\|x^{(k)} - x^*\|)}_{\lim\to \mathbf{o}(\|d\|)=0}
\end{aligned}
$$

Satz (Notwendige Optimalitätsbedingung 1. Ordnung)

Ist x^* ein lokales Optimum von (P), so ist jede Richtungsableitung in eine zulässige Richtung nichtnegativ,

$$\nabla f(x^*)^T d \geq 0 \quad \forall d \in T_{\mathcal{X}}(x^*).$$

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x\in\mathcal{X}} f(x)$
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 $\mathcal{X} \ni x^{(k)} \to x^*$, für große k

$$
\begin{aligned}
f(x^*) \leq\ & f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T(x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\
\Rightarrow \quad & \nabla f(x^*)^T(x^{(k)} - x^*) \geq -\mathbf{o}(\|x^{(k)} - x^*\|) \qquad | \cdot \alpha_k \geq 0 \\
\Rightarrow \quad & \underbrace{\nabla f(x^*)^T[\alpha_k(x^{(k)} - x^*)]}_{\lim\to \nabla f(x^*)^T d} \geq -\underbrace{\mathbf{o}(\alpha_k\|x^{(k)} - x^*\|)}_{\lim\to \mathbf{o}(\|d\|)=0}
\end{aligned}
$$

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!]

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x\in\mathcal{X}} f(x)$
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 $\mathcal{X} \ni x^{(k)} \to x^*$, für große k

$$
\begin{aligned}
f(x^*) \leq\ & f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T(x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\
\Rightarrow \quad & \nabla f(x^*)^T(x^{(k)} - x^*) \geq -\mathbf{o}(\|x^{(k)} - x^*\|) \qquad | \cdot \alpha_k \geq 0 \\
\Rightarrow \quad & \underbrace{\nabla f(x^*)^T[\alpha_k(x^{(k)} - x^*)]}_{\lim\to \nabla f(x^*)^T d} \geq -\underbrace{\mathbf{o}(\alpha_k\|x^{(k)} - x^*\|)}_{\lim\to \mathbf{o}(\|d\|)=0}
\end{aligned}
$$

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.

6.3 Notwendige Optimalitätsbedingung 1. Ordnung

Sei $\mathcal{X} \subseteq \mathbb{R}^n$ und betrachte: $(P) \quad \min_{x\in\mathcal{X}} f(x)$
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 $\mathcal{X} \ni x^{(k)} \to x^*$, für große k

$$
\begin{aligned}
& f(x^*) \leq f(x^{(k)}) = f(x^*) + \nabla f(x^*)^T(x^{(k)} - x^*) + \mathbf{o}(\|x^{(k)} - x^*\|) \\
\Rightarrow \quad & \nabla f(x^*)^T(x^{(k)} - x^*) \geq -\mathbf{o}(\|x^{(k)} - x^*\|) \qquad | \cdot \alpha_k \geq 0 \\
\Rightarrow \quad & \underbrace{\nabla f(x^*)^T[\alpha_k(x^{(k)} - x^*)]}_{\lim\to \nabla f(x^*)^T d} \geq -\underbrace{\mathbf{o}(\alpha_k\|x^{(k)} - x^*\|)}_{\lim\to \mathbf{o}(\|d\|)=0}
\end{aligned}
$$

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

6.4 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

$$\mathcal{X} = \bigcap_{i \in \mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i \in \mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

[Alle $x^{(k)} \in \mathcal{X}$ für d sind auch in den anderen Mengen.]

6.4 Der linearisierte Tangentialkegel

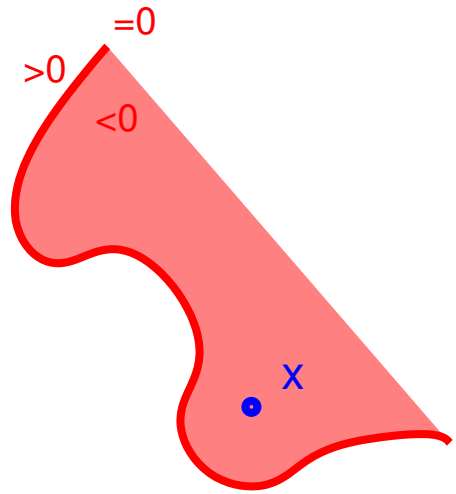
Schreibt man $\mathcal{X} = \{x \in \mathbb{R}^n : h_i(x) = 0 \ (i \in \mathcal{E}), g_i(x) \leq 0 \ (i \in \mathcal{I})\}$ als

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

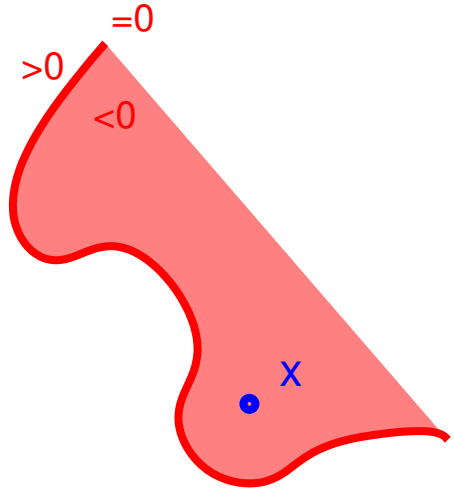
$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

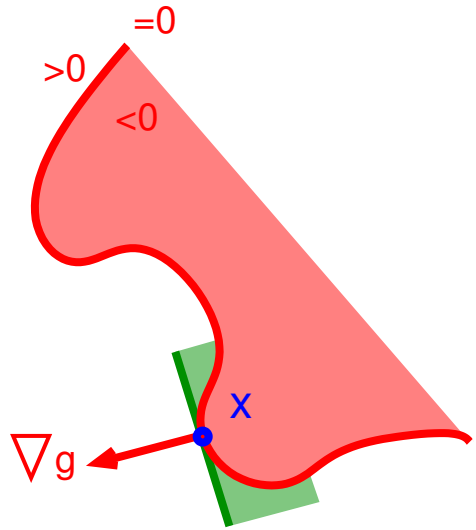
Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

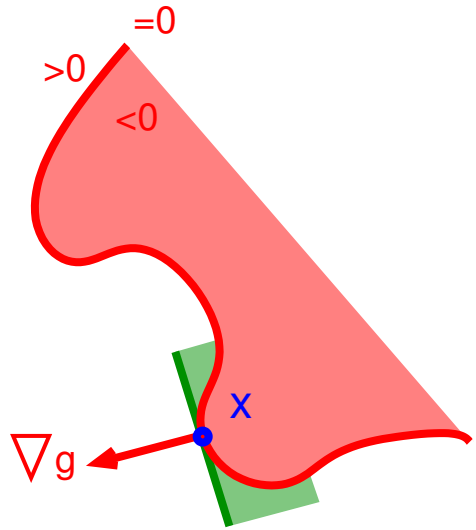
1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

Für jedes einzelne h_i und g_i ist der Tangentialkegel meist gut darstellbar:

1. $g_i(x) < 0$: $T_{S_0(g_i)}(x) = \mathbb{R}^n$

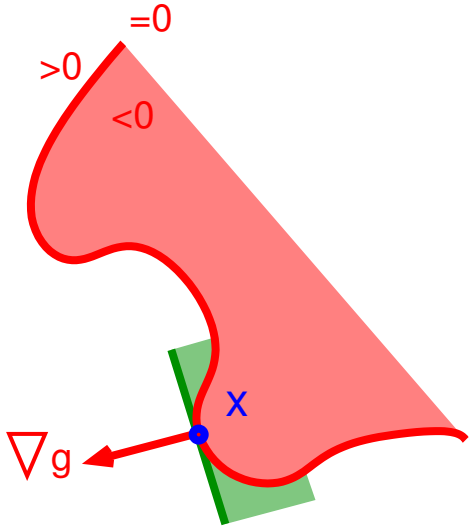
Keine Einschränkung der Richtungen!

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

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!

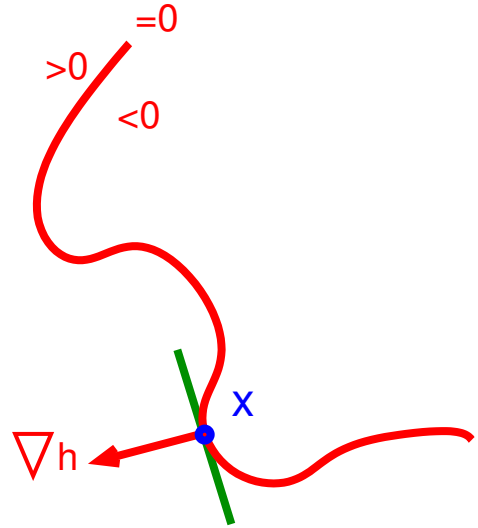
Eine Ungleichung mit $g_i(x) < 0$ heißt inaktiv (in x).

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

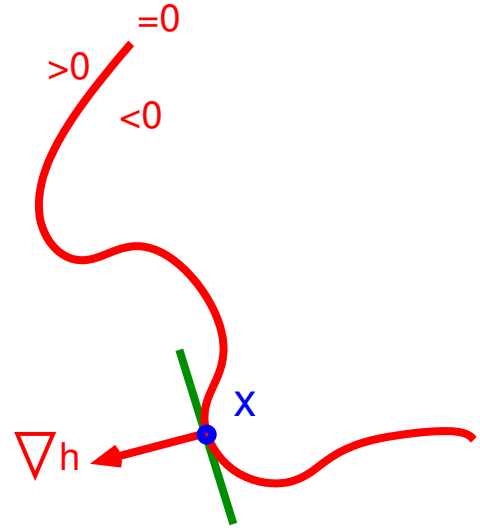
$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \ \cap \ \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

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).

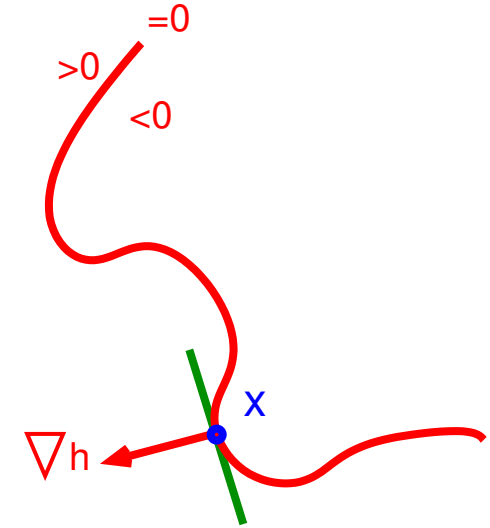
2. $g_i(x) = 0$: $T_{S_0(g_i)}(x) = \{d \in \mathbb{R}^n : \nabla g_i(x)^T d \leq 0\}$,

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,

$$T_P(x) := \{d \in \mathbb{R}^n : \nabla h_i(x)^T d = 0 \ (i \in \mathcal{E}), \nabla g_i(x)^T d \leq 0 \ (i \in \mathcal{A}(x))\}.$$

6.4 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

$$\mathcal{X} = \bigcap_{i\in\mathcal{E}} N_0(h_i) \cap \bigcap_{i\in\mathcal{I}} S_0(g_i), \quad \text{sieht man leicht}$$

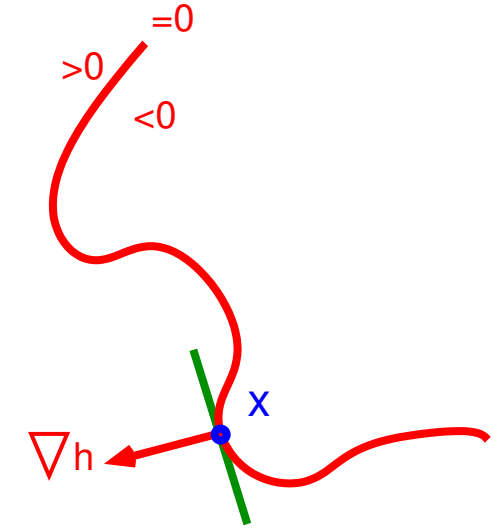
$$d \in T_{\mathcal{X}}(x) \quad \Rightarrow \quad d \in T_{N_0(h_i)}(x) \ (i \in \mathcal{E}) \ \text{ und } \ d \in T_{S_0(g_i)}(x) \ (i \in \mathcal{I}).$$

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!

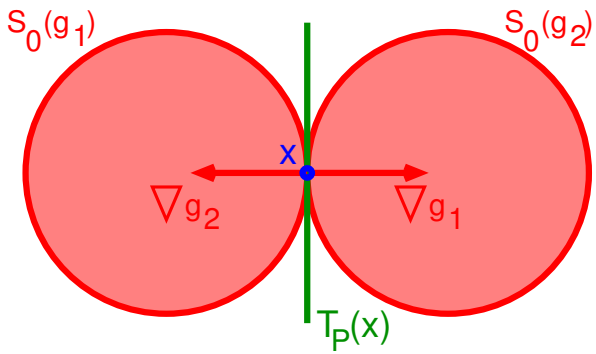
Bsp: $\bar{x} = 0$ und 2 Nebenbedingungen:

$g_1(x) := \frac{1}{2}\|x + e_1\|^2 - 1 \leq 0$, $\nabla g_1(\bar{x}) = e_1$,

$g_2(x) := \frac{1}{2}\|x - e_1\|^2 - 1 \leq 0$, $\nabla g_2(\bar{x}) = -e_1$,

$$\Rightarrow T_P(\bar{x}) = \{d \in \mathbb{R}^n : e_1^T d \leq 0, -e_1^T d \leq 0\} = \{d \in \mathbb{R}^n : d_1 = 0\},$$

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!

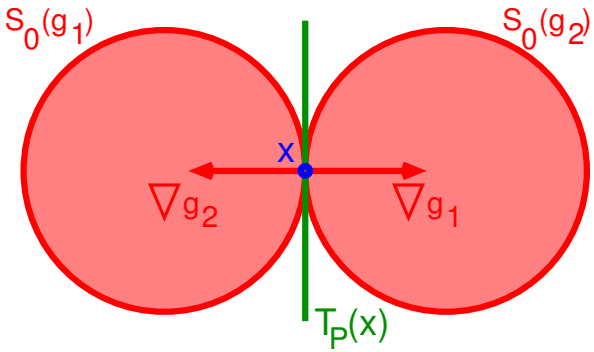
Bsp: $\bar{x} = 0$ und 2 Nebenbedingungen:

$g_1(x) := \frac{1}{2}\|x + e_1\|^2 - 1 \leq 0$, $\nabla g_1(\bar{x}) = e_1$,

$g_2(x) := \frac{1}{2}\|x - e_1\|^2 - 1 \leq 0$, $\nabla g_2(\bar{x}) = -e_1$,

$$\Rightarrow T_P(\bar{x}) = \{d \in \mathbb{R}^n : e_1^T d \leq 0, -e_1^T d \leq 0\} = \{d \in \mathbb{R}^n : d_1 = 0\},$$

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!

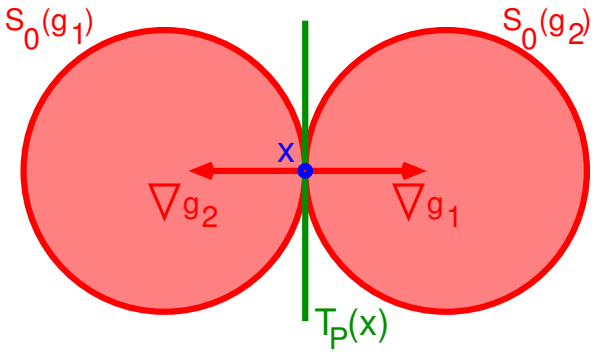
Bsp: $\bar{x} = 0$ und 2 Nebenbedingungen:

$g_1(x) := \frac{1}{2}\|x + e_1\|^2 - 1 \leq 0$, $\nabla g_1(\bar{x}) = e_1$,

$g_2(x) := \frac{1}{2}\|x - e_1\|^2 - 1 \leq 0$, $\nabla g_2(\bar{x}) = -e_1$,

$$\Rightarrow T_P(\bar{x}) = \{d \in \mathbb{R}^n : e_1^T d \leq 0, -e_1^T d \leq 0\} = \{d \in \mathbb{R}^n : d_1 = 0\},$$

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.

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

6.5 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

$$\begin{aligned} \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i \in \mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad \textit{[Kompl.]} \end{aligned}$$

6.5 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

$$\begin{aligned} \nabla f(x^*) + \sum_{i\in\mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i\in\mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad \textit{[Kompl.]} \end{aligned}$$

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 $\lambda_i^* := 0$. □

6.5 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

$$\begin{aligned} \nabla f(x^*) + \sum_{i\in\mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i\in\mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad \textit{[Kompl.]} \end{aligned}$$

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.“

6.5 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

$$\begin{aligned} \nabla f(x^*) + \sum_{i\in\mathcal{E}} \mu_i^* \nabla h_i(x^*) + \sum_{i\in\mathcal{I}} \lambda_i^* \nabla g_i(x^*) &= 0 \\ \lambda_i^* g_i(x^*) &= 0 \quad i \in \mathcal{I} \quad \textit{[Kompl.]} \end{aligned}$$

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

$$\mathcal{L}(x, \bar{\mu}, \bar{\lambda}) \;\geq\; \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \;\geq\; \mathcal{L}(\bar{x}, \mu, \lambda)$$

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 $\max_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}^{\mathcal{I}}_+} \mathcal{L}(\bar{x}, \mu, \lambda)$:

$$\nabla_\mu \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \;\Leftrightarrow\; h_i(\bar{x}) = 0$$

$$\nabla_\lambda \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \leq 0 \text{ und } \nabla_\lambda \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda})^T \bar{\lambda} = 0 \;\Leftrightarrow\; g_i(\bar{x}) \leq 0 \text{ und } g_i(\bar{x})\bar{\lambda}_i = 0$$

[Komplementarität: $(i \notin \mathcal{A}(\bar{x}) \Rightarrow \bar{\lambda}_i = 0)$ und $(\bar{\lambda}_i > 0 \Rightarrow g_i(\bar{x}) = 0)$]

$\bar{x}$ 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 $\max_{\mu \in \mathbb{R}^{\mathcal{E}}, \lambda \in \mathbb{R}^{\mathcal{I}}_+} \mathcal{L}(\bar{x}, \mu, \lambda)$:

$$\nabla_\mu \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \;\Leftrightarrow\; h_i(\bar{x}) = 0$$

$$\nabla_\lambda \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) \leq 0 \text{ und } \nabla_\lambda \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda})^T \bar{\lambda} = 0 \;\Leftrightarrow\; g_i(\bar{x}) \leq 0 \text{ und } g_i(\bar{x})\bar{\lambda}_i = 0$$

[Komplementarität: $(i \notin \mathcal{A}(\bar{x}) \Rightarrow \bar{\lambda}_i = 0)$ und $(\bar{\lambda}_i > 0 \Rightarrow g_i(\bar{x}) = 0)$]

$\bar{x}$ ist lokales Minimum des unrestringierten Problems $\min_{x \in \mathbb{R}^n} \mathcal{L}(x, \bar{\mu}, \bar{\lambda})$:

$$\nabla_x \mathcal{L}(\bar{x}, \bar{\mu}, \bar{\lambda}) = 0 \;\Leftrightarrow\; \nabla f(\bar{x}) + \textstyle\sum_{i \in \mathcal{E}} \bar{\mu}_i \nabla h_i(\bar{x}) + \sum_{i \in \mathcal{I}} \bar{\lambda}_i \nabla g_i(\bar{x}) = 0,$$

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

$$
\begin{array}{lll}
 & \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) = 0, \\
 & \nabla_\mu \mathcal{L} = & h_i(x) = 0, \quad i \in \mathcal{E}, \\
(KKT) & \nabla_\lambda \mathcal{L} = & g_i(x) \leq 0, \quad i \in \mathcal{I}, \\
 & & \lambda_i \geq 0, \quad i \in \mathcal{I}, \\
 & & g_i(x)\lambda_i = 0, \quad i \in \mathcal{I},
\end{array}
$$

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

$$
\begin{array}{lrcll}
 & \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) & = & 0, \\
 & \nabla_\mu \mathcal{L} = & h_i(x) & = & 0, \quad i \in \mathcal{E}, \\
(KKT) & \nabla_\lambda \mathcal{L} = & g_i(x) & \leq & 0, \quad i \in \mathcal{I}, \\
 & & \lambda_i & \geq & 0, \quad i \in \mathcal{I}, \\
 & & g_i(x)\lambda_i & = & 0, \quad i \in \mathcal{I},
\end{array}
$$

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

$$
\begin{array}{lrcll}
 & \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) & = & 0, \\
 & \nabla_\mu \mathcal{L} = & h_i(x) & = & 0, \quad i \in \mathcal{E}, \\
(KKT) & \nabla_\lambda \mathcal{L} = & g_i(x) & \leq & 0, \quad i \in \mathcal{I}, \\
 & & \lambda_i & \geq & 0, \quad i \in \mathcal{I}, \\
 & & g_i(x)\lambda_i & = & 0, \quad i \in \mathcal{I},
\end{array}
$$

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:

Die KKT-Bedingungen

Optimierungsverfahren suchen nach Lösungen der KKT-Bedingungen

$$
\begin{array}{lrcll}
 & \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) & = & 0, \\
 & \nabla_\mu \mathcal{L} = & h_i(x) & = & 0, \quad i \in \mathcal{E}, \\
(KKT) & \nabla_\lambda \mathcal{L} = & g_i(x) & \leq & 0, \quad i \in \mathcal{I}, \\
 & & \lambda_i & \geq & 0, \quad i \in \mathcal{I}, \\
 & & g_i(x)\lambda_i & = & 0, \quad i \in \mathcal{I},
\end{array}
$$

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

$$
\begin{array}{lrcll}
 & \nabla_x \mathcal{L} = & \nabla f(x) + \sum \mu_i \nabla h_i(x) + \sum \lambda_i \nabla g_i(x) & = & 0, \\
 & \nabla_\mu \mathcal{L} = & h_i(x) & = & 0, \quad i \in \mathcal{E}, \\
(KKT) & \nabla_\lambda \mathcal{L} = & g_i(x) & \leq & 0, \quad i \in \mathcal{I}, \\
 & & \lambda_i & \geq & 0, \quad i \in \mathcal{I}, \\
 & & g_i(x)\lambda_i & = & 0, \quad i \in \mathcal{I},
\end{array}
$$

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

$$
(CP) \qquad \begin{array}{lll} \min & f(x) & \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \\ & g_i(x) \leq 0 & i \in \mathcal{I} \\ & x \in \Omega & \end{array} \qquad \begin{array}{l} f \text{ konvex, glatt} \\ \text{jedes } h_i \text{ affin} \\ \text{jedes } g_i \text{ konvex, glatt} \\ \mathbb{R}^n \text{ oder „einfaches“ Polyeder} \end{array}
$$

Spezialfall: Opt.-Bed. für (glatte) konvexe Optimierung

Ein (glattes) konvexes Optimierungsproblem hat die Gestalt

$$
(CP) \qquad \begin{array}{lll} \min & f(x) & \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \\ & g_i(x) \leq 0 & i \in \mathcal{I} \\ & x \in \Omega & \end{array} \qquad \begin{array}{l} f \text{ konvex, glatt} \\ \text{jedes } h_i \text{ affin} \\ \text{jedes } g_i \text{ konvex, glatt} \\ \mathbb{R}^n \text{ oder „einfaches“ Polyeder} \end{array}
$$

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

$$
(CP) \qquad \begin{array}{lll} \min & f(x) & \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \\ & g_i(x) \leq 0 & i \in \mathcal{I} \\ & x \in \Omega & \end{array} \qquad \begin{array}{l} f \text{ konvex, glatt} \\ \text{jedes } h_i \text{ affin} \\ \text{jedes } g_i \text{ konvex, glatt} \\ \mathbb{R}^n \text{ oder „einfaches“ Polyeder} \end{array}
$$

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

$$
(CP) \qquad \begin{array}{lll} \min & f(x) & \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \\ & g_i(x) \leq 0 & i \in \mathcal{I} \\ & x \in \Omega & \end{array} \qquad \begin{array}{l} f \text{ konvex, glatt} \\ \text{jedes } h_i \text{ affin} \\ \text{jedes } g_i \text{ konvex, glatt} \\ \mathbb{R}^n \text{ oder „einfaches“ Polyeder} \end{array}
$$

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

$$(CP) \qquad \begin{array}{lll} \min & f(x) & \\ \text{s.t.} & h_i(x) = 0 & i \in \mathcal{E} \\ & g_i(x) \leq 0 & i \in \mathcal{I} \\ & x \in \Omega & \end{array} \qquad \begin{array}{l} f \text{ konvex, glatt} \\ \text{jedes } h_i \text{ affin} \\ \text{jedes } g_i \text{ konvex, glatt} \\ \mathbb{R}^n \text{ oder „einfaches“ Polyeder} \end{array}$$

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

6.6 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

$$0 = d^T \nabla f(x^*) \overset{(KKT)}{=} \sum \mu_i^* \underbrace{d^T \nabla h_i(x^*)}_{=0} + \sum \lambda_i^* \underbrace{d^T \nabla g_i(x^*)}_{\leq 0},$$

also Richtungen aus dem Teilkegel

$$T_P'(x^*, \lambda^*) := \{d \in T_P(x^*) : d^T \nabla g_i(x^*) = 0 \text{ mit } \lambda_i^* > 0 \ (i \in \mathcal{A}(x^*))\}.$$

6.6 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

$$0 = d^T \nabla f(x^*) \overset{(KKT)}{=} \sum \mu_i^* \underbrace{d^T \nabla h_i(x^*)}_{=0} + \sum \lambda_i^* \underbrace{d^T \nabla g_i(x^*)}_{\leq 0},$$

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 $h^{(k)} := x^{(k)} - x^* \to 0$, nutze $\mathcal{L}_* := \mathcal{L}(x^*, \mu^*, \lambda^*) = f(x^*) =: f_*$,

$$\mathcal{L}(x^* + h^{(k)}, \mu^*, \lambda^*) = \underbrace{\mathcal{L}_*}_{=f_*} + \underbrace{\nabla_x \mathcal{L}_*^T}_{=0 \ (KKT)} h^{(k)} + \frac{1}{2} \underbrace{(h^{(k)})^T \nabla_{xx} \mathcal{L}_* h^{(k)}}_{\Rightarrow \lim \to d^T \nabla_{xx} \mathcal{L}_* d \geq 0} + \mathbf{o}(\|h\|^2) \geq f_*$$

6.6 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

$$0 = d^T \nabla f(x^*) \overset{(KKT)}{=} \sum \mu_i^* \underbrace{d^T \nabla h_i(x^*)}_{=0} + \sum \lambda_i^* \underbrace{d^T \nabla g_i(x^*)}_{\leq 0},$$

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 $h^{(k)} := x^{(k)} - x^* \to 0$, nutze $\mathcal{L}_* := \mathcal{L}(x^*, \mu^*, \lambda^*) = f(x^*) =: f_*$,

$$\mathcal{L}(x^* + h^{(k)}, \mu^*, \lambda^*) = \underbrace{\mathcal{L}_*}_{=f_*} + \underbrace{\nabla_x \mathcal{L}_*^T}_{=0 \ (KKT)} h^{(k)} + \tfrac{1}{2} \underbrace{(h^{(k)})^T \nabla_{xx} \mathcal{L}_* h^{(k)}}_{\Rightarrow \lim \to d^T \nabla_{xx} \mathcal{L}_* d \geq 0} + \mathbf{o}(\|h\|^2) \geq f_*$$

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

$$d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*, \lambda^*) d \geq 0 \quad \textit{für alle } d \in T_P'(x^*, \lambda^*).$$

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

$$d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*, \lambda^*) d > 0 \quad \text{für alle } d \in T_P'(x^*, \lambda^*) \setminus \{0\},$$

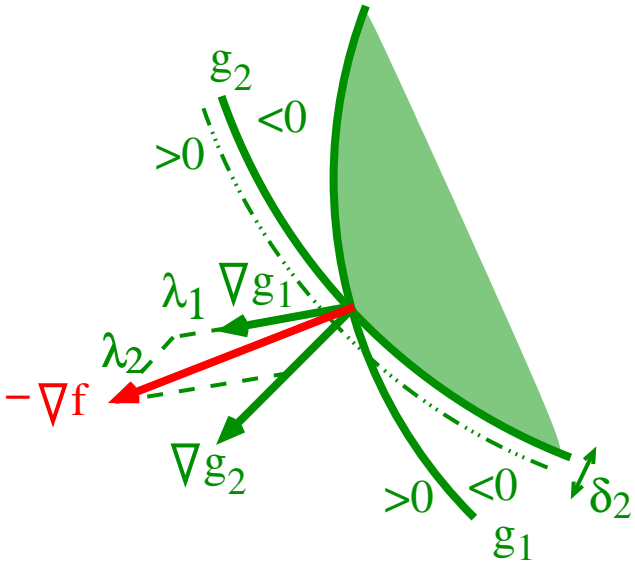
so ist x^* ein lokales Minimum von (P).

Inhaltsübersicht

6.7 Sensitivität

Wie ändert sich die Lösung, wenn man an den rechten Seiten der Nebenbedingungen wackelt?

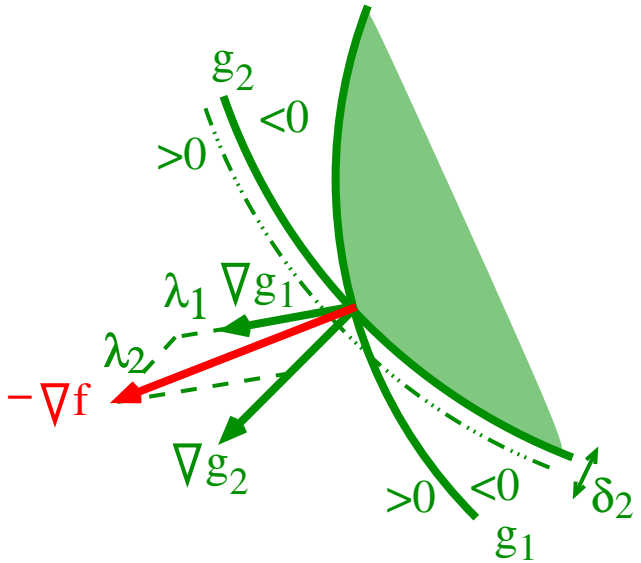
Sind die Multiplikatoren in einer Umgebung der Lösung eindeutig (dies benötigt (LICQ) und strenge Komplementarität, $\lambda_i^* > 0 \Leftrightarrow g_i(x^*) = 0$), geben sie dazu viel Information.



6.7 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)

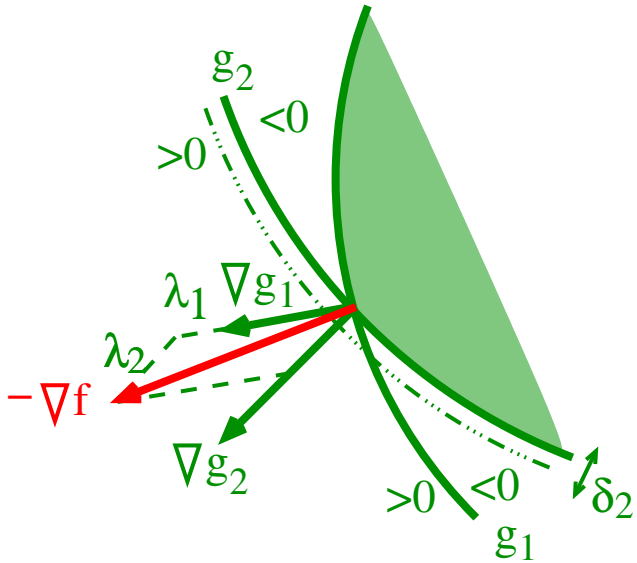
$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = \delta_i, \quad i \in \mathcal{E}, \\ & g_i(x) \le \delta_i, \quad i \in \mathcal{I}, \\ & x \in \mathbb{R}^n. \end{aligned}$$

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 $\nabla_\delta[f(x^*(\cdot))](0) = -\begin{bmatrix} \mu^* \\ \lambda^* \end{bmatrix}$.

6.7 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)

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & h_i(x) = \delta_i, \quad i \in \mathcal{E}, \\ & g_i(x) \le \delta_i, \quad i \in \mathcal{I}, \\ & x \in \mathbb{R}^n. \end{aligned}$$

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

Inhaltsübersicht

7.1 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.

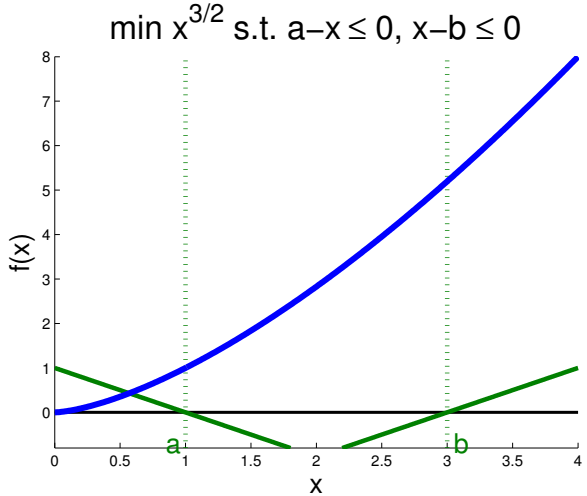
7.1 Strafverfahren (penalty methods)

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: $\min \quad f(x) := x^{\frac{3}{2}}$

s.t. $\quad g_1(x) := a - x \leq 0$

$\quad\quad g_2(x) := x - b \leq 0$



7.1 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: $\min \quad f(x) := x^{\frac{3}{2}}$

s.t. $\quad g_1(x) := a - x \leq 0$

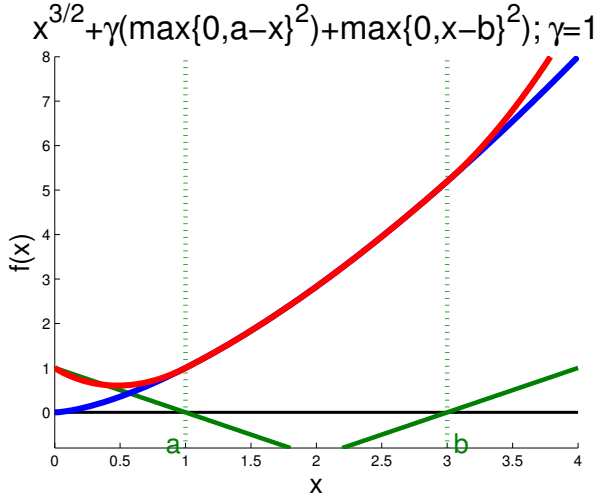
$\quad\quad g_2(x) := x - b \leq 0$

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.



7.1 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: $\min \quad f(x) := x^{\frac{3}{2}}$

s.t. $\quad g_1(x) := a - x \leq 0$

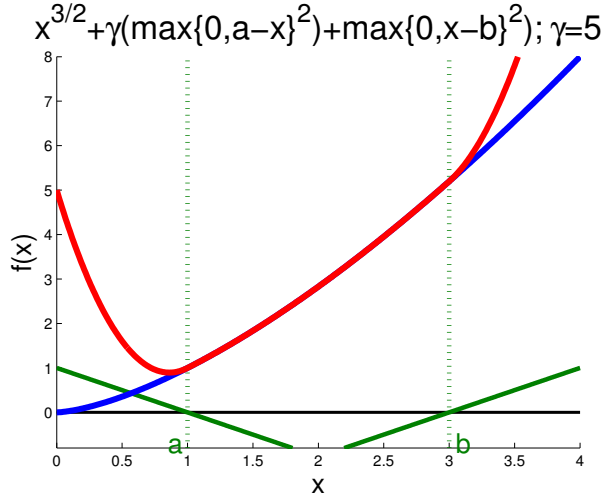
$\quad\quad g_2(x) := x - b \leq 0$

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.



7.1 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: $\min \quad f(x) := x^{\frac{3}{2}}$

s.t. $\quad g_1(x) := a - x \leq 0$

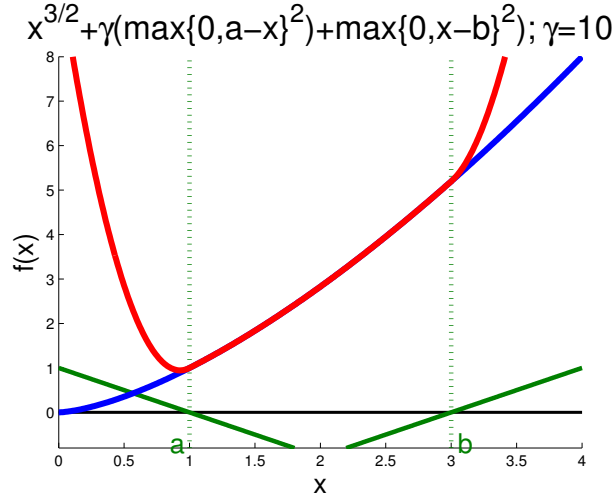
$\quad\quad g_2(x) := x - b \leq 0$

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.



7.1 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: $\min \quad f(x) := x^{\frac{3}{2}}$

s.t. $\quad g_1(x) := a - x \leq 0$

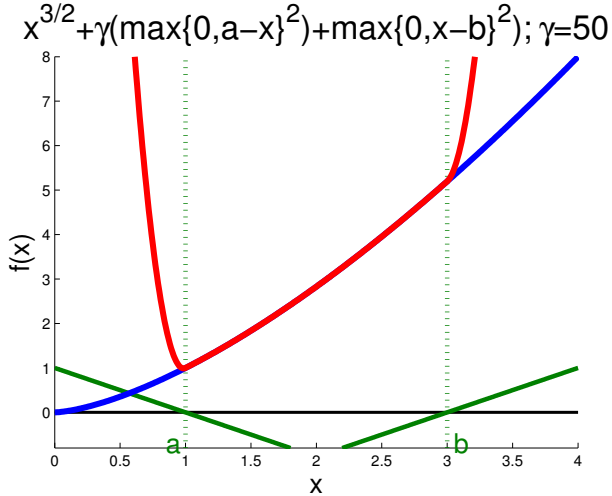
$\quad\quad g_2(x) := x - b \leq 0$

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.



7.1 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.

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } h_i(x) = 0 \quad i \in \mathcal{E} \\ &\qquad g_i(x) \le 0 \quad i \in \mathcal{I} \\ &\qquad x \in \mathbb{R}^n \end{aligned} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} f_\gamma(x) := f(x) + \gamma \Big[\underbrace{\sum_{i \in \mathcal{E}} \Psi(h_i(x)) + \sum_{i \in \mathcal{I}} \Phi(g_i(x))}_{=: \sigma(x) \ \ldots \ \text{Straffunktion}} \Big]$$

$\gamma \ge 0 \ \ldots$ Strafparameter

mit $\begin{cases} \Psi(y) > 0 & \text{für } y \neq 0, \\ \Psi(y) = 0 & \text{für } y = 0, \end{cases}$ und $\begin{cases} \Phi(y) > 0 & \text{für } y > 0, \\ \Phi(y) = 0 & \text{für } y \le 0. \end{cases}$ [Relax.!]

Strafproblem: $\max_{\gamma \ge 0} \Theta(\gamma)$ mit $\Theta(\gamma) := \inf_{x \in \mathbb{R}^n} f_\gamma(x)$

7.1 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.

$$\begin{array}{l}\min f(x)\\ \text{s.t. } h_i(x)=0 \quad i\in\mathcal{E}\\ \qquad g_i(x)\le 0 \quad i\in\mathcal{I}\\ \qquad x\in\mathbb{R}^n\end{array} \quad\rightarrow\quad \min_{x\in\mathbb{R}^n} f_\gamma(x) := f(x)+\gamma\Big[\underbrace{\sum_{i\in\mathcal{E}}\Psi(h_i(x))+\sum_{i\in\mathcal{I}}\Phi(g_i(x))}_{=:\,\sigma(x)\ \ldots\ \text{Straffunktion}}\Big]$$

$\gamma\ge 0$ … Strafparameter

mit $\begin{cases}\Psi(y)>0 & \text{für } y\neq 0,\\ \Psi(y)=0 & \text{für } y=0,\end{cases}$ und $\begin{cases}\Phi(y)>0 & \text{für } y>0,\\ \Phi(y)=0 & \text{für } y\le 0.\end{cases}$ [Relax.!]

Strafproblem: $\max_{\gamma\ge 0}\Theta(\gamma)$ 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 $\gamma\ge 0$ ein globales Optimum x_γ, für das $\Theta(\gamma)$ angenommen wird, und ist $\{x_\gamma:\gamma\ge 0\}$ 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 $f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$.

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 $f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$.

Bsp: $\min \quad f(x) := x^{\frac{3}{2}}$

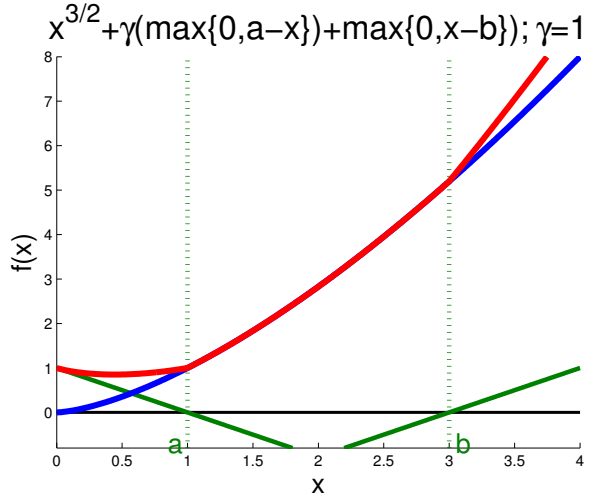
s.t. $\quad g_1(x) := a - x \leq 0$

$\quad\quad g_2(x) := x - b \leq 0$

$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$

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 $f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$.

Bsp: $\min \quad f(x) := x^{\frac{3}{2}}$

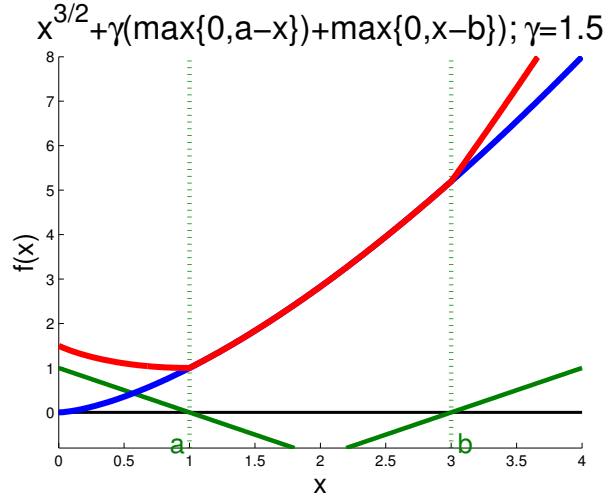
s.t. $\quad g_1(x) := a - x \leq 0$

$\quad\quad g_2(x) := x - b \leq 0$

$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$

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 $f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$.

Bsp: $\min \quad f(x) := x^{\frac{3}{2}}$

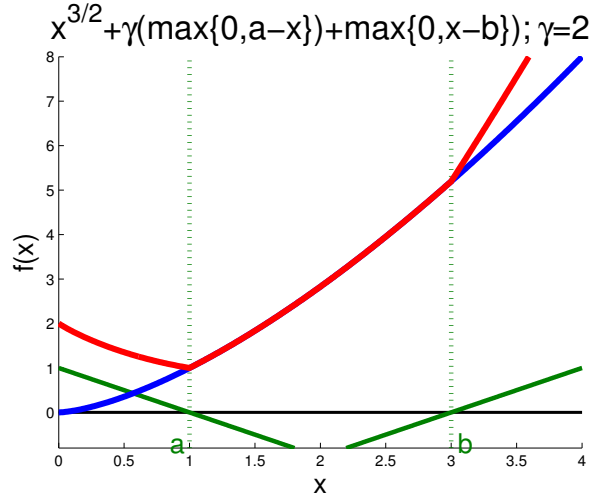
s.t. $\quad g_1(x) := a - x \leq 0$

$\quad\quad g_2(x) := x - b \leq 0$

$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$

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 $f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$.

Bsp: $\min \quad f(x) := x^{\frac{3}{2}}$

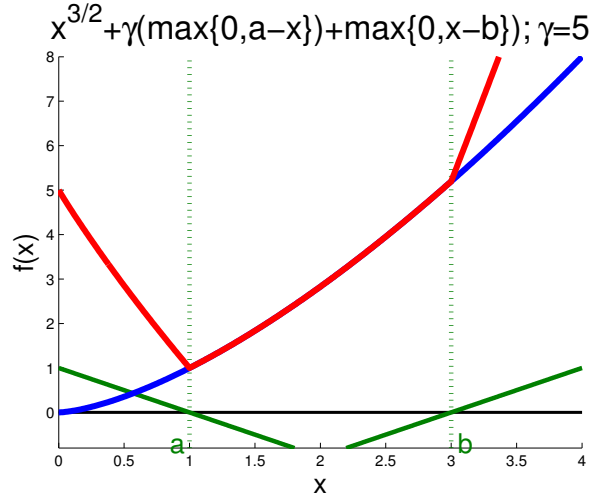
s.t. $\quad g_1(x) := a - x \leq 0$

$\quad\quad g_2(x) := x - b \leq 0$

$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$

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 $f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$.

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-Straffuntkion: 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 $\longrightarrow$ ungünstige Konvergenzeigenschaften.
- Für große γ wird die zweite Ableitung in unzulässige Richtungen sehr groß $\longrightarrow$ numerische Probleme.
- Die quadratische Straffunktion ist i.A. nicht exakt, selbst für sehr großes γ bleibt x_γ^* unzulässig (s. Anfangsbeispiel).

Inhaltsübersicht

7.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator μ^* exakt.

7.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator μ^* exakt.

Erfüllt x^* die hinreichenden Opt.-bed. mit Multiplikator μ^*, dann gilt

$$\nabla_x \mathcal{L}(x^*,\mu^*) = 0 \text{ und } d^T \nabla_{xx}\mathcal{L}(x^*,\mu^*)d > 0 \quad \forall d \in T_P(x^*) [\Leftrightarrow J_h(x^*)d = 0].$$

7.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2,$$

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 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 $\mathcal{L}_\gamma(\cdot,\mu^*)$.

7.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2,$$

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 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.

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2$$

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.

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2$$

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 $k \rightarrow k+1$)

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2$$

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 $0 = \nabla_x \mathcal{L}(x^*,\mu^*) = \nabla f(x^*) + \sum_{i\in\mathcal{E}} \mu_i^* \nabla h_i(x^*)$.

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2$$

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

$$0 \approx \nabla_x \mathcal{L}_{\gamma_k}(x^{(k)},\mu^{(k)}) = \nabla f(x^{(k)}) + \sum_{i\in\mathcal{E}} \underbrace{[\mu_i^{(k)} + \gamma_k h_i(x^{(k)})]}_{\approx\mu^*} \nabla h_i(x^{(k)})$$

$$\mathcal{L}_\gamma(x,\mu) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2}\sum_{i\in\mathcal{E}} h_i(x)^2 \quad = \mathcal{L}(x,\mu) + \frac{\gamma}{2}\|h(x)\|^2$$

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

$$0 \approx \nabla_x \mathcal{L}_{\gamma_k}(x^{(k)},\mu^{(k)}) = \nabla f(x^{(k)}) + \sum_{i\in\mathcal{E}} \underbrace{[\mu_i^{(k)} + \gamma_k h_i(x^{(k)})]}_{\approx\mu^*} \nabla h_i(x^{(k)})$$

$$\rightarrow \quad \mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \qquad (i\in\mathcal{E})$$

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}_($$x^{(k)}, \mu^{(k)})\|$ klein genug, STOP.
   (u.U. Neustart, falls kein Fortschritt Richtung Zulässigkeit, etc.)
3. Aktualisiere die Multiplikatoren: $\mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \qquad (i \in \mathcal{E})$
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 $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\| \le \varepsilon_k$.
2. Ist $\|h(x^{(k)})\|$ und $\|\nabla_x \mathcal{L}_(x^{(k)}, \mu^{(k)})\|$ klein genug, STOP.
   (u.U. Neustart, falls kein Fortschritt Richtung Zulässigkeit, etc.)
3. Aktualisiere die Multiplikatoren: $\mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \qquad (i \in \mathcal{E})$
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 Glgen umgewandelt:

$$g_i(x) + s_i = 0,\ s_i \ge 0 \quad (i \in \mathcal{I})$$

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 Unglgen, s. später).

Inhaltsübersicht

7.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

$\longrightarrow$ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

7.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

$\longrightarrow$ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp:

7.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

$\longrightarrow$ vor allem für Ungleichungen gut geeignet! Im Folgenden $\mathcal{E} = \emptyset$.

Bsp: Innere-Punkte-Verfahren für Lineare Optimierung über Kegeln

7.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

$\rightarrow$ 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 ∞.

7.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

$\rightarrow$ 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 $y < 0$: $-\frac{1}{y}$, $-\log(-y)$

7.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

$\rightarrow$ 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 $\bar{x}^{(0)} \in \mathring{\mathcal{X}}$, $\beta_0 > 0$, 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 $\bar{x}^{(0)} \in \mathring{\mathcal{X}}$, $\beta_0 > 0$, 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

$$0 = \nabla f_\beta(x(\beta)) \quad = \quad \nabla f(x(\beta)) + \sum_{i \in \mathcal{I}} \underbrace{\frac{\beta}{-g_i(x(\beta))}}_{=:\lambda_i(\beta)\ \geq 0} \nabla g_i(x(\beta))$$

Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt $\bar{x}^{(0)} \in \mathring{\mathcal{X}}$, $\beta_0 > 0$, 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 $\|\nabla_x f_{\beta_k}(x^{(k)})\| \le \varepsilon_k$.
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

$$0 = \nabla f_\beta(x(\beta)) \quad = \quad \nabla f(x(\beta)) + \sum_{i\in\mathcal{I}} \underbrace{\frac{\beta}{-g_i(x(\beta))}}_{=:\lambda_i(\beta)\ \ge 0} \nabla g_i(x(\beta))$$

also ist

$$\begin{aligned} \nabla f(x(\beta)) + \sum_{i\in\mathcal{I}} \lambda_i(\beta)\nabla g_i(x(\beta)) &= 0 \\ g_i(x(\beta)) &\le 0 \quad (i\in\mathcal{I}) \\ \lambda_i(\beta) &\ge 0 \quad (i\in\mathcal{I}) \\ -\lambda_i(\beta) g_i(x(\beta)) &= \beta \quad (i\in\mathcal{I}) \text{ [pert. Kompl.]} \end{aligned}$$

Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt $\bar{x}^{(0)} \in \mathring{\mathcal{X}}$, $\beta_0 > 0$, 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 $\|\nabla_x f_{\beta_k}(x^{(k)})\| \le \varepsilon_k$.
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

$$0 = \nabla f_\beta(x(\beta)) \quad = \quad \nabla f(x(\beta)) + \sum_{i\in\mathcal{I}} \underbrace{\frac{\beta}{-g_i(x(\beta))}}_{=:\lambda_i(\beta)\ \ge 0} \nabla g_i(x(\beta))$$

also ist

$$\begin{aligned} \nabla f(x(\beta)) + \sum_{i\in\mathcal{I}} \lambda_i(\beta)\nabla g_i(x(\beta)) &= 0 \\ g_i(x(\beta)) &\le 0 \quad (i\in\mathcal{I}) \\ \lambda_i(\beta) &\ge 0 \quad (i\in\mathcal{I}) \\ -\lambda_i(\beta) g_i(x(\beta)) &= \beta \quad (i\in\mathcal{I}) \text{ [pert. Kompl.]} \end{aligned}$$

Für $\beta \to 0$ wird bei Konvergenz ein KKT-Punkt gefunden!

Satz (Fiacco und Mc Cormick 1968)

Sei $\mathring{\mathcal{X}} \neq \emptyset$ 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 $x(\beta):(0,\bar{\beta}) \to \mathbb{R}^n$ mit $x(\beta) \overset{\beta \to 0}{\to} x^*$ und $x(\beta)$ ist lokales Minimum von f_β. [zent. Pfad]

(ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \overset{\beta \to 0}{\to} \lambda_i^*$ 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 Mc Cormick 1968)

Sei $\mathring{\mathcal{X}} \neq \emptyset$ 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 $x(\beta):(0,\bar{\beta}) \to \mathbb{R}^n$ mit $x(\beta) \overset{\beta \to 0}{\to} x^*$ und $x(\beta)$ ist lokales Minimum von f_β. [zent. Pfad]

(ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \overset{\beta \to 0}{\to} \lambda_i^*$ 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 Mc Cormick 1968)

Sei $\mathring{\mathcal{X}} \neq \emptyset$ 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 $x(\beta):(0,\bar{\beta}) \to \mathbb{R}^n$ mit $x(\beta) \overset{\beta\to 0}{\to} x^*$ und $x(\beta)$ ist lokales Minimum von f_β. [zent. Pfad]

(ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \overset{\beta\to 0}{\to} \lambda_i^*$ 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 Mc Cormick 1968)

Sei $\mathring{\mathcal{X}} \neq \emptyset$ 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 $x(\beta):(0,\bar{\beta}) \to \mathbb{R}^n$ mit $x(\beta) \overset{\beta\to 0}{\to} x^*$ und $x(\beta)$ ist lokales Minimum von f_β. [zent. Pfad]

(ii) Für $x(\beta)$ aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \overset{\beta\to 0}{\to} \lambda_i^*$ 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

7.4 Quadratische Optimierung (Quadratic Programming)

QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.
$\Rightarrow$ Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

7.4 Quadratische Optimierung (Quadratic Programming)

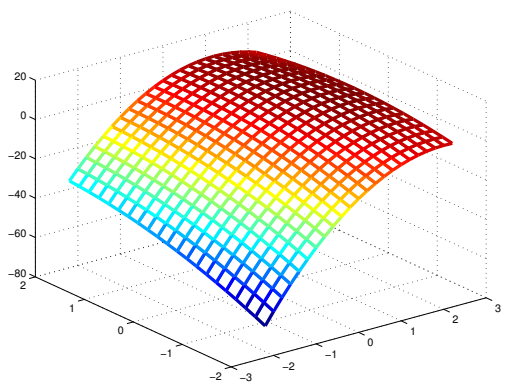
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

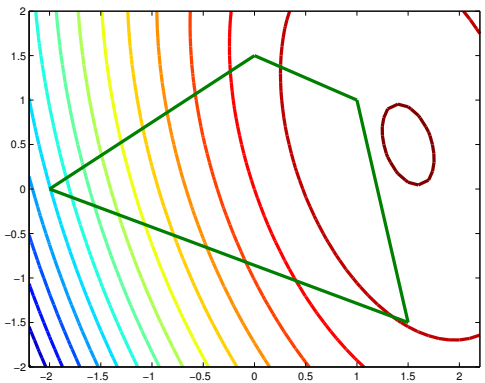
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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.





7.4 Quadratische Optimierung (Quadratic Programming)

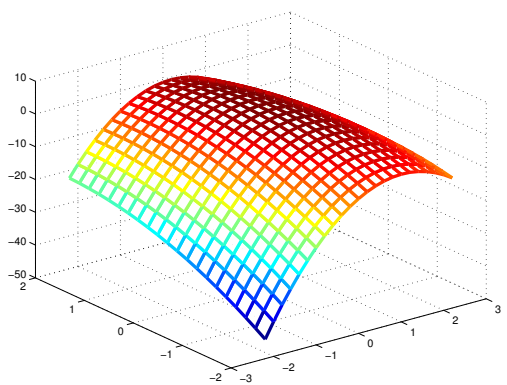
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

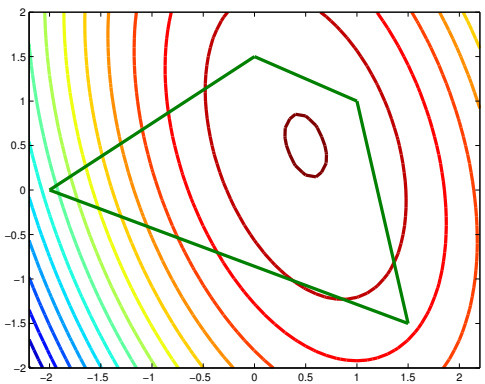
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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.





7.4 Quadratische Optimierung (Quadratic Programming)

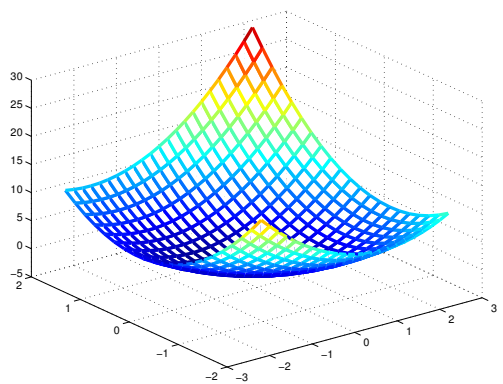
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

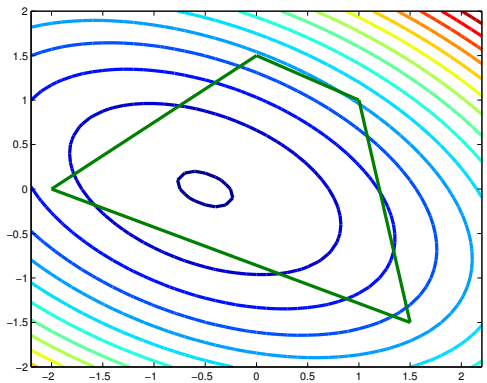
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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





7.4 Quadratische Optimierung (Quadratic Programming)

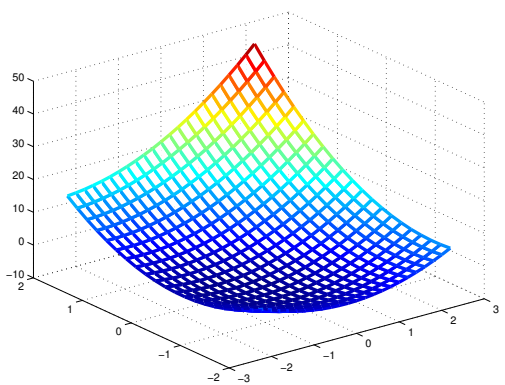
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

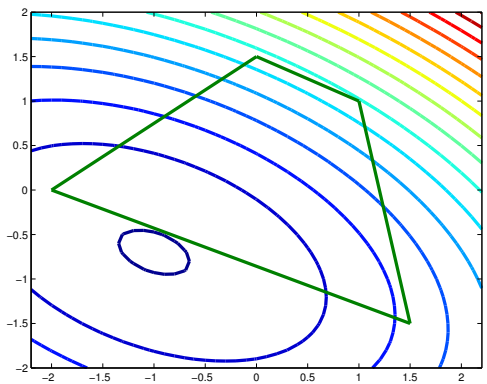
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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





7.4 Quadratische Optimierung (Quadratic Programming)

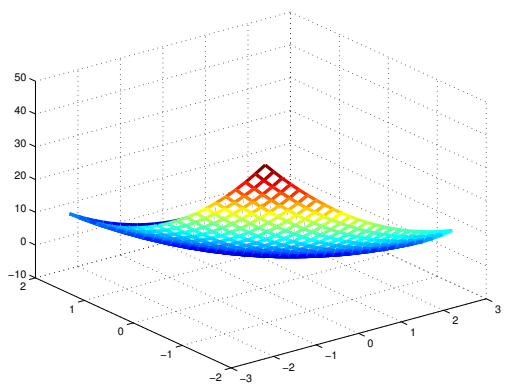
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

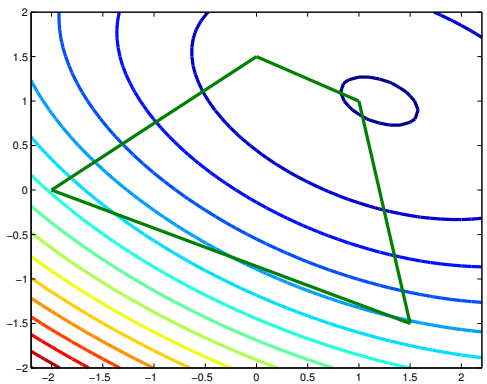
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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





7.4 Quadratische Optimierung (Quadratic Programming)

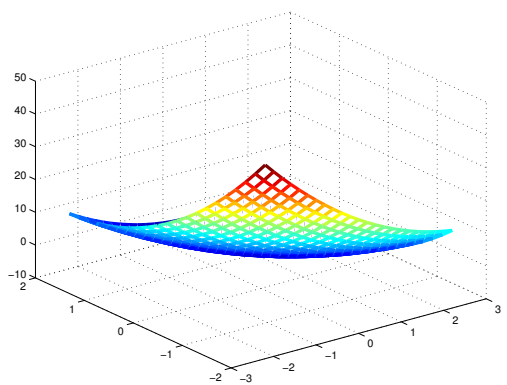
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

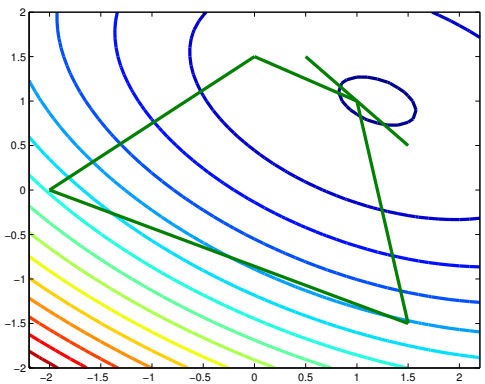
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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





7.4 Quadratische Optimierung (Quadratic Programming)

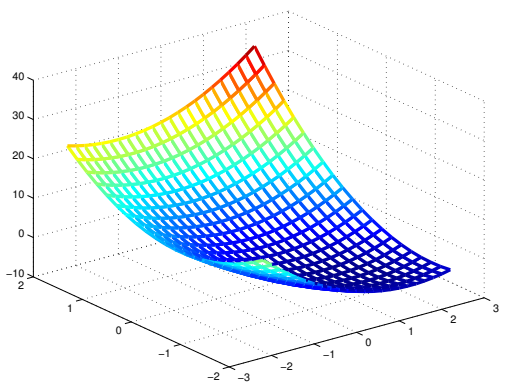
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

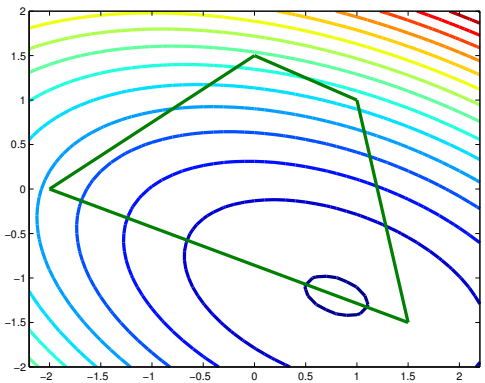
$$\begin{array}{lll} \min & f(x) := \frac{1}{2} x^T Q x + q^T x & \\ \text{s.t.} & b_i - h_i^T x = 0 & i \in \mathcal{E} \quad [h_i \in \mathbb{R}^n] \\ & b_i - g_i^T x \leq 0 & i \in \mathcal{I} \quad [g_i \in \mathbb{R}^n] \\ & x \in \mathbb{R}^n & \end{array} \Bigg\} \rightarrow Ax \geq b$$

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





7.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$\text{(PQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax \geq b$$

7.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$\text{(PQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax \geq b$$

Lagrange-Funktion: $\mathcal{L}(x, \lambda) = \frac{1}{2} x^T Q x + q^T x + \lambda^T (b - Ax)$, $(\lambda \geq 0)$

primales Problem: $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$ [Für x^*: $\exists\ \lambda^*$ mit $(b - Ax^*)^T \lambda^* = 0$]

7.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$\text{(PQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax \geq b$$

Lagrange-Funktion: $\mathcal{L}(x, \lambda) = \frac{1}{2} x^T Q x + q^T x + \lambda^T (b - Ax)$, $(\lambda \geq 0)$

primales Problem: $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$ [Für x^*: $\exists\ \lambda^*$ mit $(b - Ax^*)^T \lambda^* = 0$]

duales Problem: $\sup\limits_{\lambda \geq 0} \inf\limits_x \mathcal{L}(x, \lambda) = \sup\limits_{\lambda \geq 0} \big(b^T \lambda + \underbrace{\inf\limits_x [\frac{1}{2} x^T Q x + (q - A^T \lambda)^T x]}_{\nabla_x = 0 \ \Rightarrow \ Qx + q - A^T \lambda = 0, \text{ sonst } -\infty} \big)$

7.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$\text{(PQP)} \quad \min \ \frac{1}{2}x^T Qx + q^T x \ \text{ s.t. } \ Ax \geq b$$

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: $\sup\limits_{\lambda \geq 0} \inf\limits_x \mathcal{L}(x,\lambda) = \sup\limits_{\lambda \geq 0} \big(b^T\lambda + \underbrace{\inf\limits_x [\frac{1}{2}x^T Qx + (q - A^T\lambda)^T x]}_{\nabla_x = 0 \ \Rightarrow \ Qx + q - A^T\lambda = 0,\ \text{sonst } -\infty}\big)$

I.A. enthält das duale QP auch noch x-Variable(!):

$$\text{(DQP)} \quad \begin{array}{ll} \max & \frac{1}{2}x^T Qx + q^T x + (b - Ax)^T\lambda \\ \text{s.t.} & Qx + q - A^T\lambda = 0 \\ & x \in \mathbb{R}^n, \lambda \geq 0 \end{array}$$

[erfüllt starke Dualität!]

7.4.1 Konvexe quadratische Optimierung

Für $Q \succeq 0$ ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

Herleitung für

$$\text{(PQP)} \quad \min \ \frac{1}{2}x^T Qx + q^T x \ \text{ s.t. } \ Ax \geq b$$

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: $\sup\limits_{\lambda \geq 0} \inf\limits_x \mathcal{L}(x,\lambda) = \sup\limits_{\lambda \geq 0} \big(b^T\lambda + \underbrace{\inf\limits_x [\frac{1}{2}x^T Qx + (q - A^T\lambda)^T x]}_{\nabla_x = 0 \ \Rightarrow \ Qx + q - A^T\lambda = 0,\ \text{sonst } -\infty}\big)$

I.A. enthält das duale QP auch noch x-Variable(!):

$$\text{(DQP)} \quad \begin{array}{ll} \max & \frac{1}{2}x^T Qx + q^T x + (b - Ax)^T\lambda \\ \text{s.t.} & Qx + q - A^T\lambda = 0 \\ & x \in \mathbb{R}^n, \lambda \geq 0 \end{array} \qquad \text{[erfüllt starke Dualität!]}$$

Für $Q \succ 0$ ist x eindeutig bestimmt: $x(\lambda) = Q^{-1}(A^T\lambda - q)$,

$$\longrightarrow \quad \begin{array}{ll} \max & -\frac{1}{2}\lambda^T(AQ^{-1}A^T)\lambda + (b + AQ^{-1}q)^T\lambda - \frac{1}{2}q^T Q^{-1}q \\ \text{s.t.} & \lambda \geq 0 \end{array}$$

Nur Vorzeichenbedingungen! Oft besonders effizient lösbar $\longrightarrow \lambda^*$, $x^* = x(\lambda^*)$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverf., $A = [a_1, \ldots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2} x^T Q x + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \; \frac{1}{2} x^T Q x + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverf., $A = [a_1, \ldots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2} x^T Q x + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \ \frac{1}{2} x^T Q x + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

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 Pfadverfolgungsverf., $A = [a_1, \ldots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2} x^T Q x + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \; \frac{1}{2} x^T Q x + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

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:

$$\begin{array}{rcll} Qx + q - A^T \lambda & = & 0 & \text{„duale Zulässigkeit“} \\ Ax - s & = & b & \text{„primale Zulässigkeit“} \\ s \circ \lambda & = & \gamma \mathbf{1} & \text{„perturb. Kompl.“} \end{array}$$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverf., $A = [a_1, \ldots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2} x^T Q x + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \ \frac{1}{2} x^T Q x + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

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:

$$\begin{array}{rcll} Qx + q - A^T \lambda & = & 0 & \text{„duale Zulässigkeit“} \\ Ax - s & = & b & \text{„primale Zulässigkeit“} \\ s \circ \lambda & = & \gamma \mathbf{1} & \text{„perturb. Kompl.“} \end{array}$$

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 Pfadverfolgungsverf., $A = [a_1, \ldots, a_m]^T$, $b \in \mathbb{R}^m$.

$$\begin{array}{ll} \min & \frac{1}{2} x^T Q x + q^T x \\ \text{s.t.} & Ax \geq b \end{array} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \ \frac{1}{2} x^T Q x + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

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:

$$\begin{array}{rcll} Qx + q - A^T \lambda & = & 0 & \text{„duale Zulässigkeit“} \\ Ax - s & = & b & \text{„primale Zulässigkeit“} \\ s \circ \lambda & = & \gamma \mathbf{1} & \text{„perturb. Kompl.“} \end{array}$$

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).

Innere-Punkte-Verf. sind auch auf nicht-konvexe QPs recht gut anwendbar.

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$x = \bar{x} + d$ mit $d \in \{d : Ad = 0\} =: \ker A$ [Kern/Nullraum von A]

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$x = \bar{x} + d$ mit $d \in \{d : Ad = 0\} =: \ker A$ [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 $\ker A = \{Zu : u \in \mathbb{R}^k\}$,

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$x = \bar{x} + d$ mit $d \in \{d : Ad = 0\} =: \ker A$ [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 $\ker A = \{Zu : u \in \mathbb{R}^k\}$, $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$$x = \bar{x} + d \text{ mit } d \in \{d : Ad = 0\} =: \ker A \quad \text{[Kern/Nullraum von } A\text{]}$$

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

$$\text{(EQP)} \quad \Leftrightarrow \quad \min_{u \in \mathbb{R}^k} \tfrac{1}{2} u^T Z^T Q Z u + (Q\bar{x} + q)^T Z u + \tfrac{1}{2} \bar{x}^T Q \bar{x} + q^T \bar{x}.$$

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$$x = \bar{x} + d \text{ mit } d \in \{d : Ad = 0\} =: \ker A \quad \text{[Kern/Nullraum von } A\text{]}$$

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

$$\text{(EQP)} \quad \Leftrightarrow \quad \min_{u \in \mathbb{R}^k} \tfrac{1}{2} u^T Z^T Q Z u + (Q\bar{x} + q)^T Z u + \tfrac{1}{2} \bar{x}^T Q \bar{x} + q^T \bar{x}.$$

$\Rightarrow$ endliche OL nur, falls $Z^T Q Z \succeq 0$ und $Z^T Q Z u + Z^T (Q\bar{x} + q) = 0$ lösbar,

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2} x^T Q x + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$x = \bar{x} + d$ mit $d \in \{d : Ad = 0\} =: \ker A$ [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

$$\text{(EQP)} \quad \Leftrightarrow \quad \min_{u \in \mathbb{R}^k} \tfrac{1}{2} u^T Z^T Q Z u + (Q\bar{x} + q)^T Z u + \tfrac{1}{2} \bar{x}^T Q \bar{x} + q^T \bar{x}.$$

$\Rightarrow$ endliche OL nur, falls $Z^T Q Z \succeq 0$ und $Z^T Q Z u + Z^T (Q\bar{x} + q) = 0$ lösbar, eindeutig für $Z^T Q Z \succ 0$: $u^* = -(Z^T Q Z)^{-1} Z^T (Q\bar{x} + q)$ und $x^* = \bar{x} + Zu$.

7.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$\text{(EQP)} \quad \min \ \frac{1}{2}x^T Qx + q^T x \ \text{ s.t. } \ Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$ 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

$x = \bar{x} + d$ mit $d \in \{d : Ad = 0\} =: \ker A$ [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

$$\text{(EQP)} \quad \Leftrightarrow \quad \min_{u\in\mathbb{R}^k} \tfrac{1}{2}u^T Z^T QZu + (Q\bar{x} + q)^T Zu + \tfrac{1}{2}\bar{x}^T Q\bar{x} + q^T\bar{x}.$$

$\Rightarrow$ endliche OL nur, falls $Z^TQZ \succeq 0$ und $Z^TQZu + Z^T(Q\bar{x} + q) = 0$ lösbar, eindeutig für $Z^TQZ \succ 0$: $u^* = -(Z^TQZ)^{-1}Z^T(Q\bar{x} + q)$ und $x^* = \bar{x} + Zu$.

Satz (QP mit Gleichungsbedingungen)

Seien die Spalten von Z eine Basis von $\ker A$, $Z^TQZ \succ 0$, *und* $\bar{x} \in \mathcal{X}$. *Ist* $x^* = \bar{x} + d^*$ *die Lösung des KKT Systems zu (EQP),*

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} x^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \quad \Leftrightarrow \quad \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}\begin{bmatrix} -d^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} q + Q\bar{x} \\ A\bar{x} - b \end{bmatrix},$$

dann ist x^* die eindeutige Optimallösung von (EQP).

7.4.3 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) $\longrightarrow$ 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: $\min \frac{1}{2} x^T Q x + q^T x$ s.t. $Ax \geq b$ mit $A = [a_1, a_2, a_3, a_4]^T$

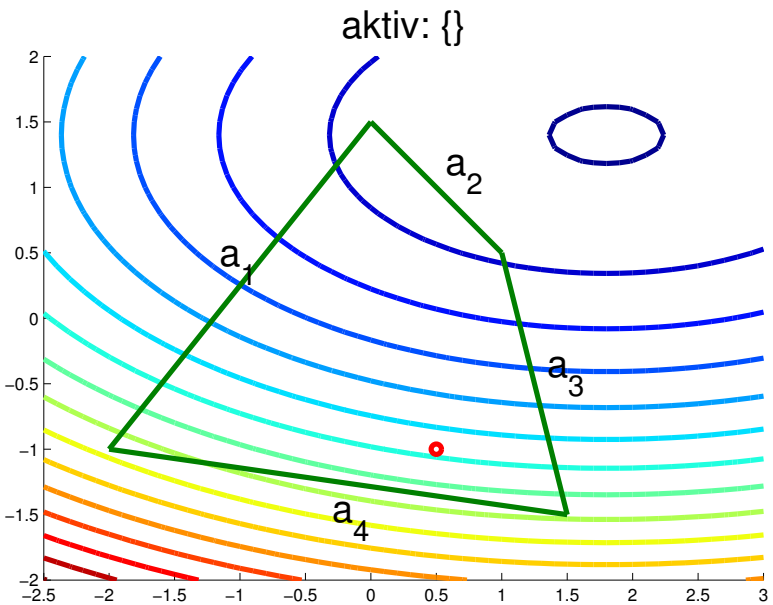
7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ 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: $\min \frac{1}{2} x^T Q x + q^T x$ 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: ?



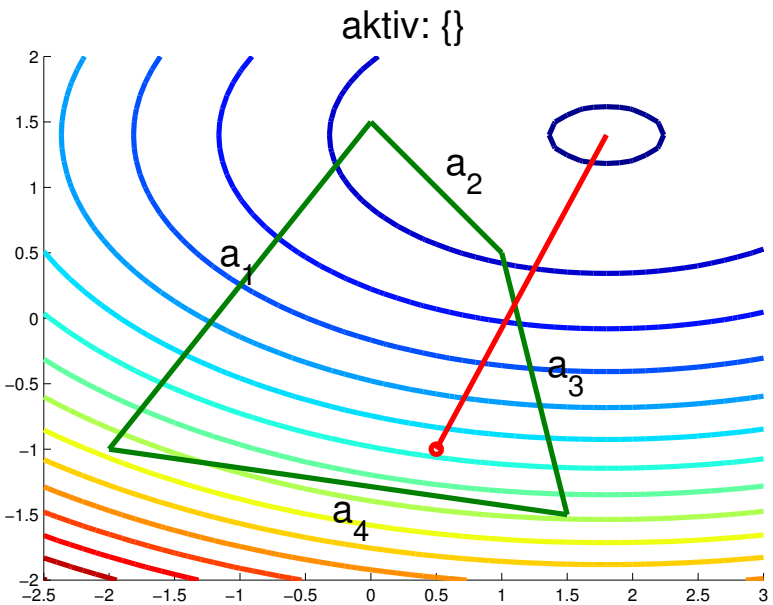
7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ 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: $\min \frac{1}{2} x^T Q x + q^T x$ 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 ?



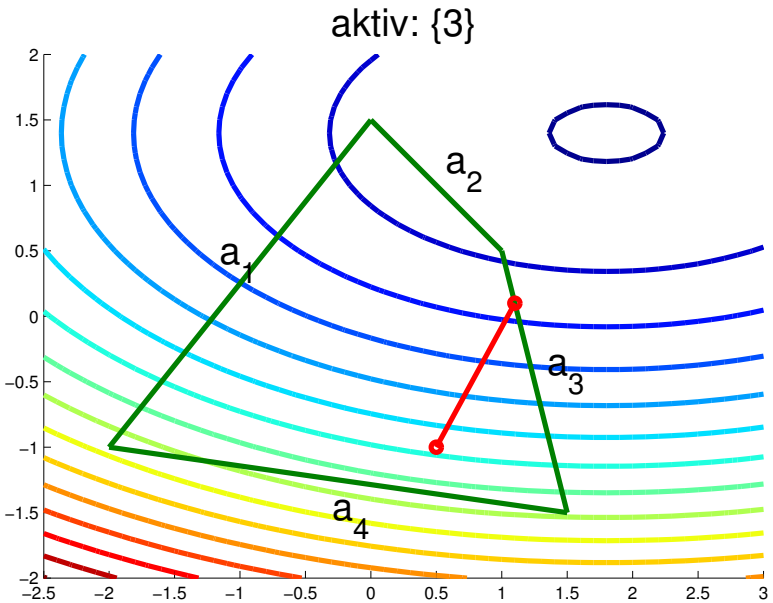
7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ 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: $\min \frac{1}{2} x^T Q x + q^T x$ 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 ?



7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

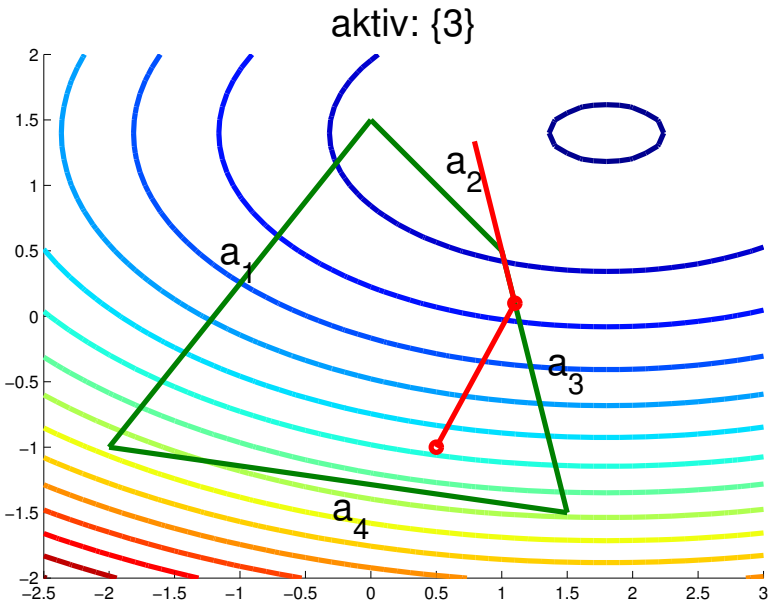
- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ active set,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2} x^T Q x + q^T x$ 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 ?



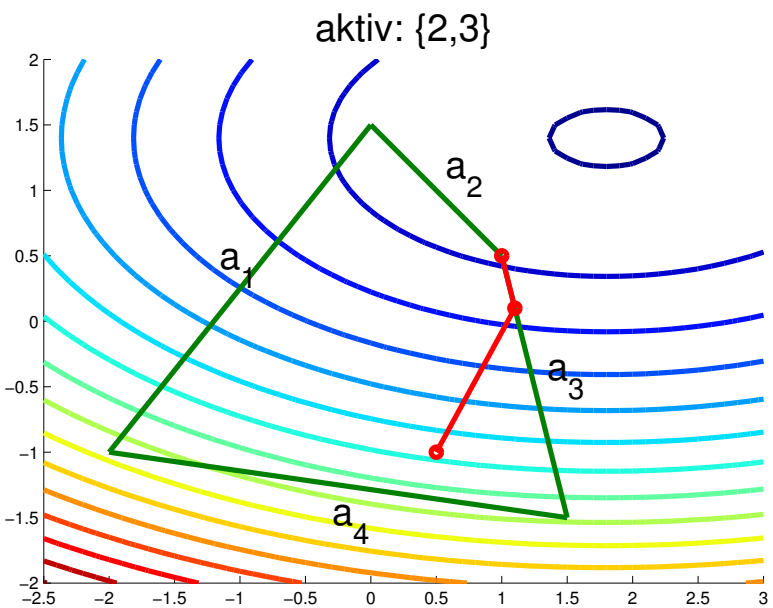
7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ 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: $\min \frac{1}{2} x^T Q x + q^T x$ 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 ?



7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

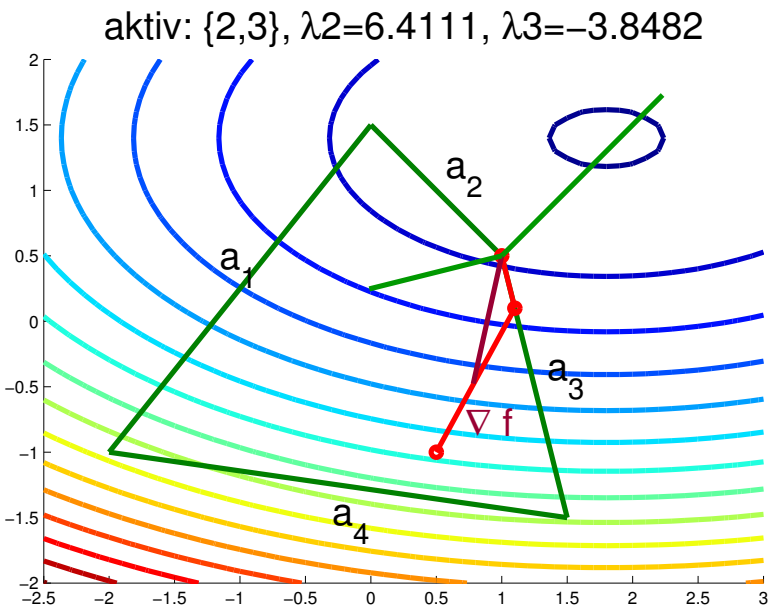
- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ active set,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2} x^T Q x + q^T x$ 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:

$$\begin{bmatrix} Q & a_2 & a_3 \\ a_2^T & 0 & 0 \\ a_3^T & 0 & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} q + Qx^{(2)} \\ a_2^T x^{(2)} - b_2 \\ a_3^T x^{(2)} - b_3 \end{bmatrix}$$

$\longrightarrow d = 0$, $\lambda_3 < 0$ (!),
$x^{(3)} := x^{(2)}$,
aktiv wird ?



7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

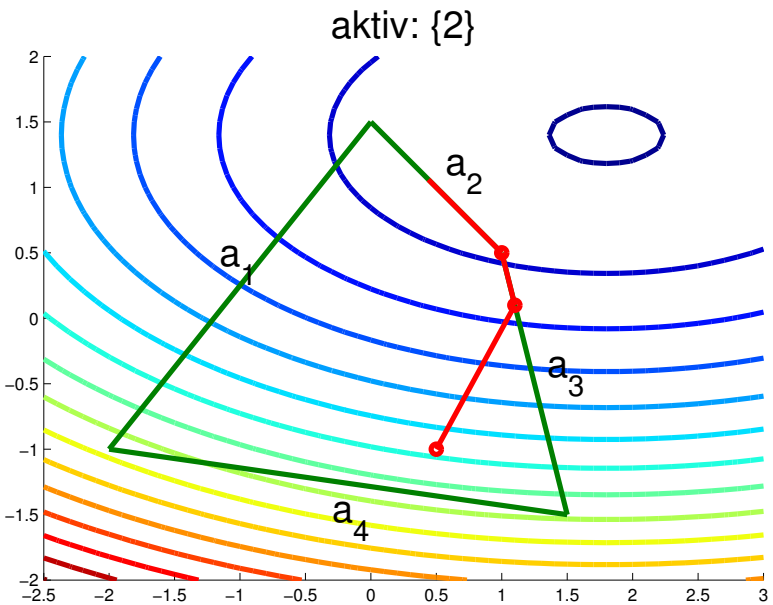
- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ active set,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2} x^T Q x + q^T x$ 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 ?



7.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

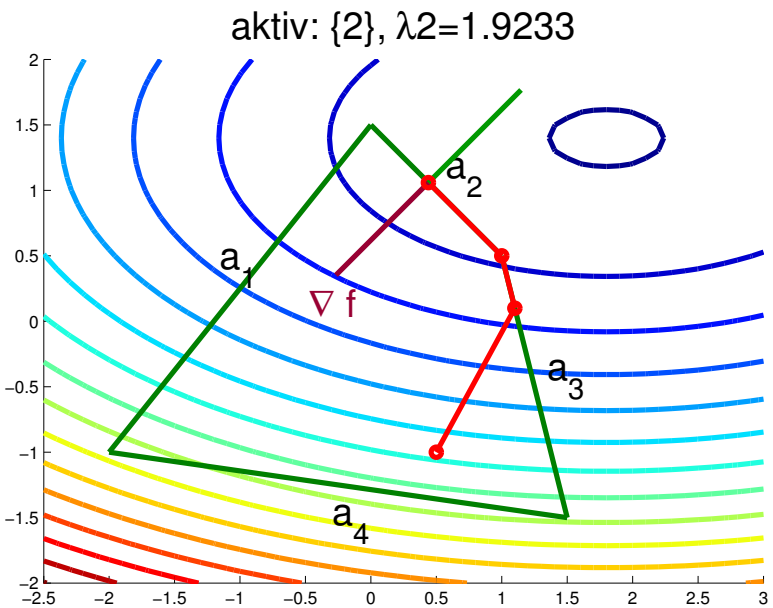
- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) $\longrightarrow$ active set,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp: $\min \frac{1}{2} x^T Q x + q^T x$ 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 $x^T Qx + q^T x$ $s.t.$ $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $x^T Qx + q^T x$ $s.t.$ $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $x^T Qx + q^T x$ $s.t.$ $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $x^T Qx + q^T x$ s.t. $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: min $x^T Qx + q^T x$ s.t. $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.

   Sonst $(d \neq 0)$
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ für $\emptyset$]

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: min $x^T Qx + q^T x$ s.t. $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.

   Sonst $(d \neq 0)$
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ 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 $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{\imath}\}$ mit $\hat{\imath}$ erzeugt α_k.

Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$ und o.B.d.A. nur Unglgen: min $x^T Qx + q^T x$ s.t. $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.
   Sonst $(d \neq 0)$
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ 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 $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{\imath}\}$ mit $\hat{\imath}$ 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 $x^T Qx + q^T x$ s.t. $a_i^T x \geq b_i$ $(i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.
   Sonst $(d \neq 0)$
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ 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 $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{\imath}\}$ mit $\hat{\imath}$ 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: $\min\ x^T Qx + q^T x \ \ s.t. \ \ a_i^T x \geq b_i \ (i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.
   Sonst ($d \neq 0$)
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ 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 $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{\imath}\}$ mit $\hat{\imath}$ 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: $\min\ x^T Qx + q^T x \ \ s.t. \ \ a_i^T x \geq b_i \ (i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.
   Sonst ($d \neq 0$)
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ 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 $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{\imath}\}$ mit $\hat{\imath}$ 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: $\min\ x^T Qx + q^T x \ \ s.t. \ \ a_i^T x \geq b_i \ (i \in \mathcal{I})$

---

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 $\hat{\imath} \in \mathcal{A}_k$ mit $\lambda_{\hat{\imath}} < 0$, setze $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{\imath}\}$, $x^{(k+1)} := x^{(k)}$.
   Sonst ($d \neq 0$)
   a) bestimme $\alpha_k := \min\{\frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0\}$. [$+\infty$ 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 $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{\imath}\}$ mit $\hat{\imath}$ 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

7.5 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.

7.5 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
$\longrightarrow$ Newton führt zu lokal quadratischer Konvergenz von SQP-Verf.

7.5 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
$\longrightarrow$ 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.

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

$$J_F(x, \mu) = \begin{bmatrix} \nabla_{xx} \mathcal{L} & \nabla_{x\mu} \mathcal{L} \\ \nabla_{\mu x} \mathcal{L} & \nabla_{\mu\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx} \mathcal{L}(x, \mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Iteration k: 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 $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \ldots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $$F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Iteration k: 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 $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \ldots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\begin{aligned} \min \quad & \tfrac{1}{2} d_x Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} \quad & A_k d_x = -h^{(k)} \end{aligned} \qquad [\text{Zeile } i\text{: } h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0]$$

7.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen: $\min f(x)$ s.t. $h(x) = 0$. $[h : \mathbb{R}^n \to \mathbb{R}^{\mathcal{E}}]$

Löse die KKT-Bed. für $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$ mit Newton:

KKT: $F(x,\mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_\mu \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$

Newton: $\begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix}$ mit $F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_\mu^N \end{bmatrix} = 0$

$$J_F(x,\mu) = \begin{bmatrix} \nabla_{xx}\mathcal{L} & \nabla_{x\mu}\mathcal{L} \\ \nabla_{\mu x}\mathcal{L} & \nabla_{\mu\mu}\mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx}\mathcal{L}(x,\mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

Iteration k: 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 $Q_k := \nabla_{xx}\mathcal{L}(x^{(k)}, \mu^{(k)})$, $A_k := J_h(x^{(k)})$, $h^{(k)} := [h_1(x^{(k)}), \ldots, h_{|\mathcal{E}|}(x^{(k)})]^T$.

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\begin{aligned} \min \quad & \tfrac{1}{2} d_x Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} \quad & A_k d_x = -h^{(k)} \end{aligned} \qquad [\text{Zeile } i\text{: } h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0]$$

Dafür bestimmt sich eine Optimallösung mit Multiplikatoren y durch

$\begin{bmatrix} Q_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x \\ y \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -h^{(k)} \end{bmatrix}$, setze $y = -(\mu^{(k)} + d_\mu) \Leftrightarrow$ Newton-System

Das QP des SQP-Verfahrens

$$(P) \quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{array}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x),$$

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

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

$$(QP_k)\quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 \quad i \in \mathcal{I} \end{array}$$

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

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

$$(QP_k)\quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 \quad i \in \mathcal{I} \end{array} \quad \rightarrow \quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \le -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}_-] \end{array}$$

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

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

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

$$(QP_k)\quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 \quad i \in \mathcal{I} \end{array} \quad \rightarrow \quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \le -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}_-] \end{array}$$

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

$$(P)\quad \begin{array}{ll}\min & f(x)\\ \text{s.t.} & h_i(x)=0 \quad i\in\mathcal{E}\\ & g_i(x)\le 0 \quad i\in\mathcal{I}\end{array}\quad,\quad \mathcal{L}(x,\mu,\lambda):=f(x)+\sum_{i\in\mathcal{E}}\mu_i h_i(x)+\sum_{i\in\mathcal{I}}\lambda_i g_i(x),$$

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

$$(QP_k)\quad \begin{array}{ll}\min & \frac{1}{2}d_x^T Q_k d_x+\nabla f_k^T d_x\\ \text{s.t.} & h_i(x^{(k)})+\nabla h_i(x^{(k)})^T d_x=0 \quad i\in\mathcal{E}\\ & g_i(x^{(k)})+\nabla g_i(x^{(k)})^T d_x\le 0 \quad i\in\mathcal{I}\end{array}\quad\rightarrow\quad \begin{array}{ll}\min & \frac{1}{2}d_x^T Q_k d_x+\nabla f_k^T d_x\\ \text{s.t.} & A_k^h d_x=-h^{(k)} \quad [y_h\in\mathbb{R}^{\mathcal{E}}]\\ & A_k^g d_x\le -g^{(k)} \quad [y_g\in\mathbb{R}^{\mathcal{I}}_-]\end{array}$$

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,
  $Q_k=\nabla^2 f(x^{(k)})+\sum_{i\in\mathcal{E}}\mu_i^{(k)}\nabla^2 h_i(x^{(k)})+\sum_{i\in\mathcal{I}}\lambda_i^{(k)}\nabla^2 g_i(x^{(k)})$,
  $\rightarrow$ kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.

Das QP des SQP-Verfahrens

$$(P)\quad \begin{array}{ll} \min & f(x) \\ \text{s.t.} & h_i(x) = 0 \quad i \in \mathcal{E} \\ & g_i(x) \le 0 \quad i \in \mathcal{I} \end{array}, \qquad \mathcal{L}(x,\mu,\lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x),$$

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

$$(QP_k)\quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \le 0 \quad i \in \mathcal{I} \end{array} \quad \rightarrow \quad \begin{array}{ll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ \text{s.t.} & A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & A_k^g d_x \le -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}_-] \end{array}$$

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,
  $Q_k = \nabla^2 f(x^{(k)}) + \sum_{i\in\mathcal{E}} \mu_i^{(k)} \nabla^2 h_i(x^{(k)}) + \sum_{i\in\mathcal{I}} \lambda_i^{(k)} \nabla^2 g_i(x^{(k)})$,
  $\rightarrow$ kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.
- Die Hessematrizen in Q_k sind durch BFGS, etc. approximierbar.

7.5.2 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.

7.5.2 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

$$f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$$

- f_γ ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
- $d_x^{(k)}$ ist Abstiegsrichtung für f_γ für $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$.

7.5.2 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

$$f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$$

- 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. $\gamma > 0$,

$$\mathcal{L}_\gamma(x, \mu, \lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x) + \gamma\big[\sum_{i\in\mathcal{E}} h_i(x)^2 + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}^2\big]$$

- differenzierbar und für γ groß genug ist $d_x^{(k)}$ Abstiegsrichtung
- Die Wahl von γ hat starken Einfluss auf das Konvergenzverhalten ...

7.5.2 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

$$f_\gamma(x) := f(x) + \gamma\big[\sum_{i\in\mathcal{E}} |h_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}\big]$$

- 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. $\gamma > 0$,

$$\mathcal{L}_\gamma(x, \mu, \lambda) := f(x) + \sum_{i\in\mathcal{E}} \mu_i h_i(x) + \sum_{i\in\mathcal{I}} \lambda_i g_i(x) + \gamma\big[\sum_{i\in\mathcal{E}} h_i(x)^2 + \sum_{i\in\mathcal{I}} \max\{0, g_i(x)\}^2\big]$$

- 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 → schlechte lokale Konvergenz.

Es gibt heuristische Gegenstrategien ...

7.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \ldots, n\} \leq \Delta$.
- 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.

7.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \ldots, n\} \leq \Delta$.
- 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 $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.

7.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \ldots, n\} \leq \Delta$.
- 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 $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.
$\longrightarrow$ Löse als Trust-Region-Unterproblem für Strafparameter $\gamma > 0$

$$
\begin{array}{lll}
\min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s & \\
\text{s.t.} & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s & i \in \mathcal{E} \\
& g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s & i \in \mathcal{I} \\
& -\Delta \leq [d_x]_i \leq \Delta \ (i = 1, \ldots, n), s \geq 0 &
\end{array}
$$

7.5.3 Globalisierung mit Trust-Region-Ansätzen

Klassischer Ansatz:

- Fordere zusätzlich $\|d_x\| \leq \Delta$ oder, damit über lineare Nebenbed. darstellbar, $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \ldots, n\} \leq \Delta$.
- 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 $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$.
$\rightarrow$ Löse als Trust-Region-Unterproblem für Strafparameter $\gamma > 0$

$$
\begin{array}{lll}
\min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s & \\
\text{s.t.} & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s & i \in \mathcal{E} \\
& g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s & i \in \mathcal{I} \\
& -\Delta \leq [d_x]_i \leq \Delta \ (i = 1, \ldots, n), s \geq 0 &
\end{array}
$$

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.

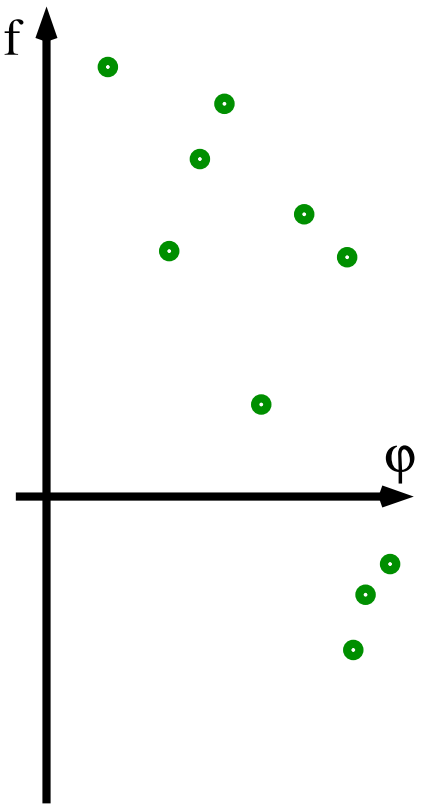
7.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.



7.5.4 Globalisierung mit einem Filter-Ansatz

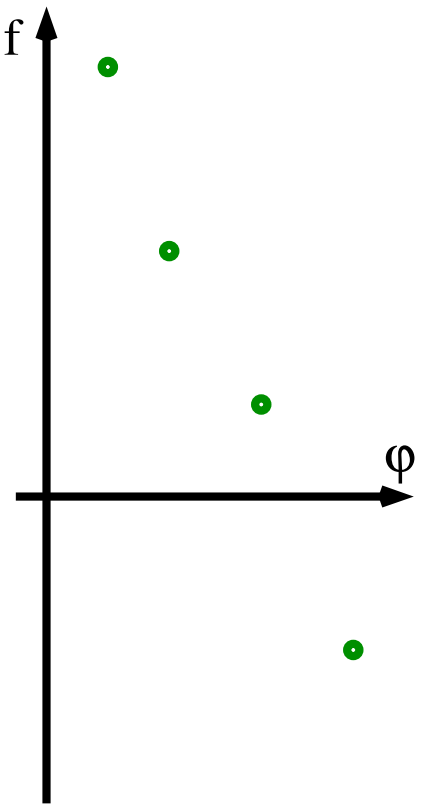
Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$ . . .Verletzungsmaß, z.B. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$



7.5.4 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. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

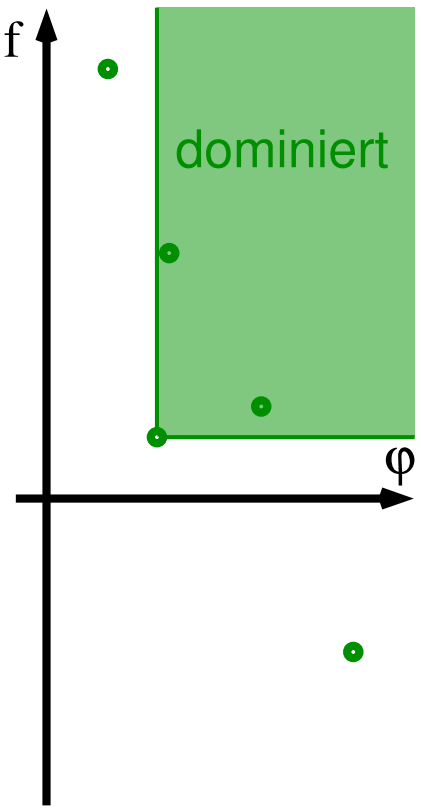
Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$



7.5.4 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. $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$.

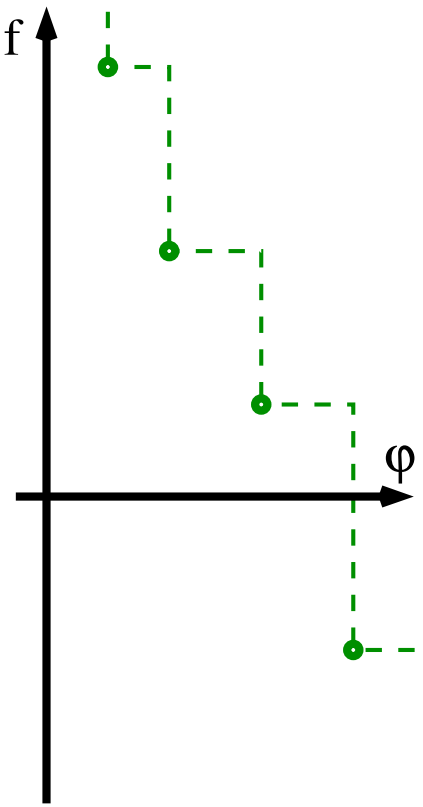
Betrachte $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

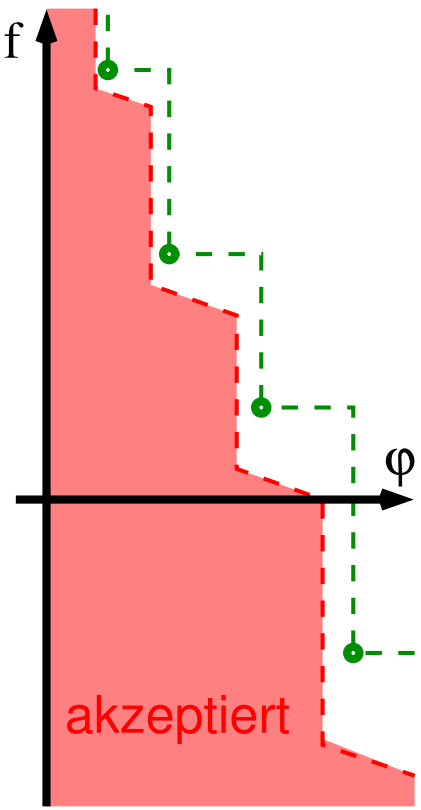
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

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]



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

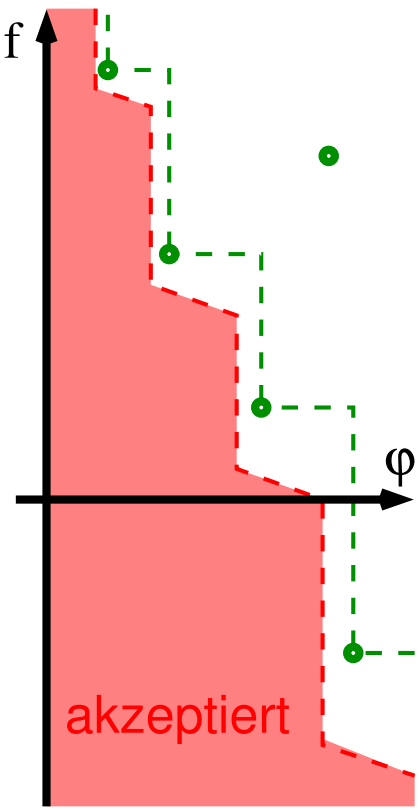
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

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]



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

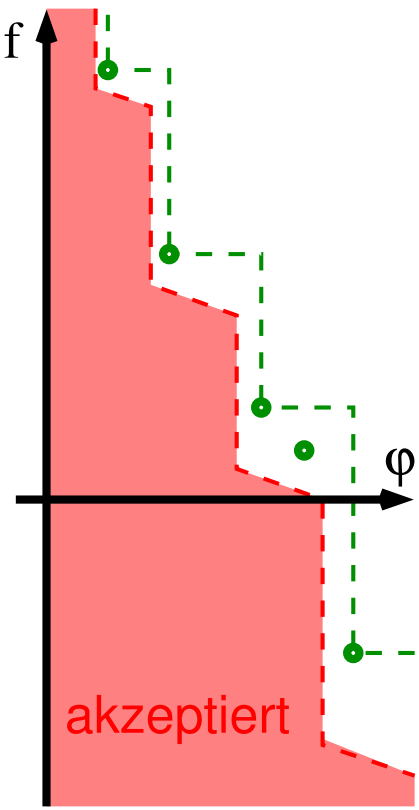
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

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]



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

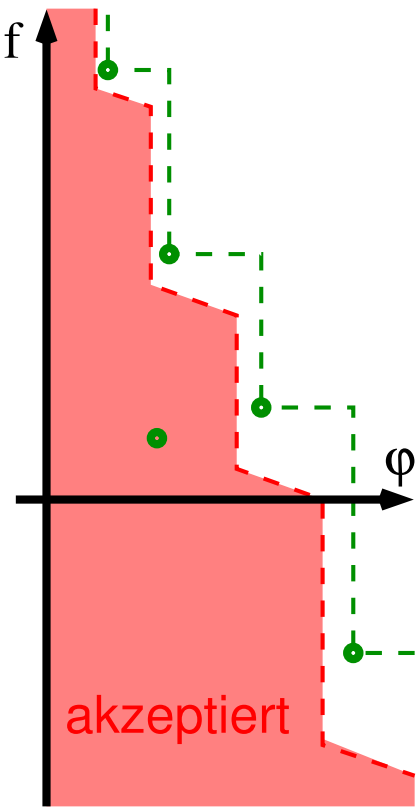
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

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]



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

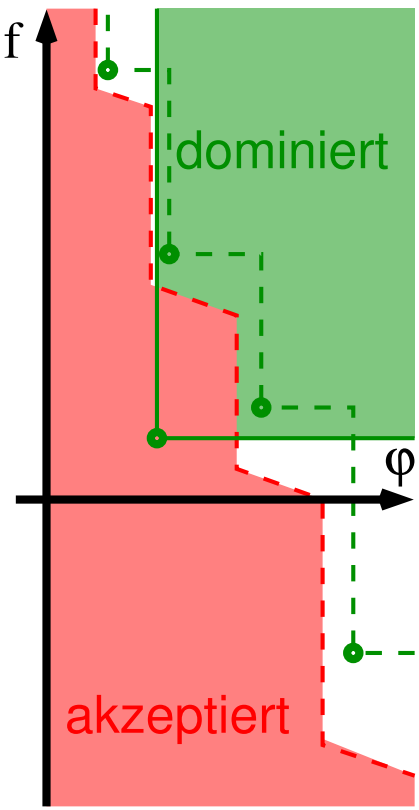
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von $\mathcal{F}_k$ akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

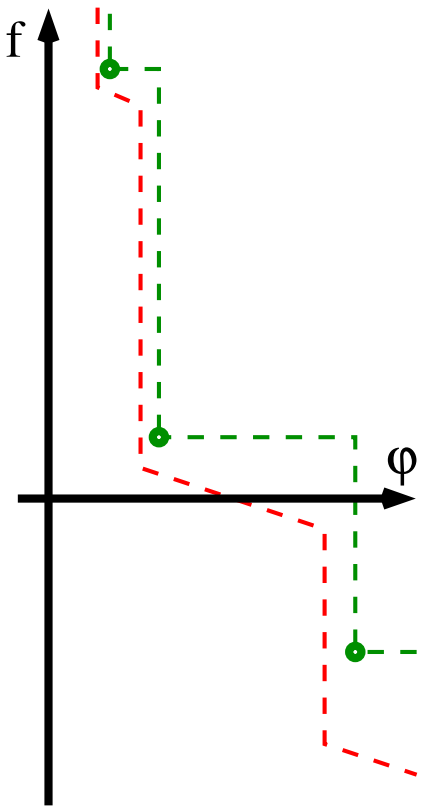
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird (f_{k+1}, φ_{k+1}) von $\mathcal{F}_k$ akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$



7.5.4 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 $(f_k, \varphi_k) := \left(f(x^{(k)}), \varphi(x^{(k)})\right)$ als Punkt in $\mathbb{R}^2$.

Ein Filter enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \left\{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \ldots, k\}\right\}.$$

Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ dominiert $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$, falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

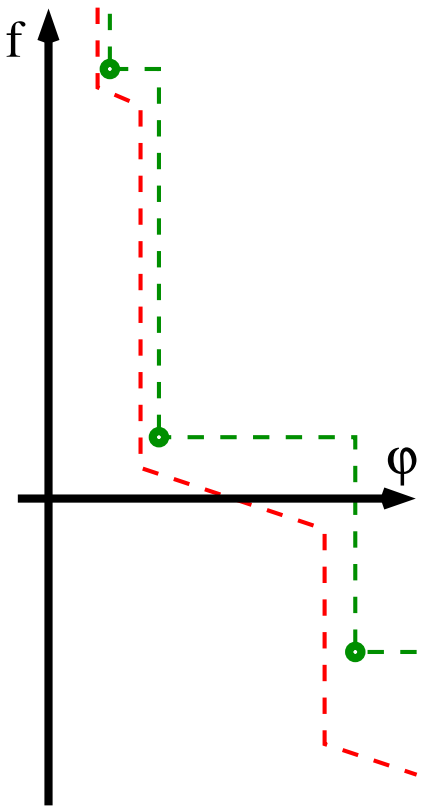
Ein $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$ wird von $\mathcal{F}_k$ für gegebene $0 < \gamma_1 < \gamma_2 < 1$ akzeptiert, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[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:

$$
(QP_k^{\Delta}) \qquad
\begin{array}{lll}
\min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\
\text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\
& g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\
& -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \ldots, n\}
\end{array}
$$

Trust-Region-Variante eines Filter-Ansatzes

Schritt d_x löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \ldots, n\} \end{array}$$

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:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \ldots, n\} \end{array}$$

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:

$$(QP_k^\Delta) \qquad \begin{array}{lll} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \\ \text{s.t.} & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 & i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 & i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta & i \in \{1, \ldots, n\} \end{array}$$

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 $(QP_k^\Delta) \to d_x$.
   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 $(QP_k^\Delta) \to d_x$.
   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 $(QP_k^\Delta) \to d_x$.
   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 $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ 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 $(QP_k^\Delta) \to d_x$.
   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 $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ 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 $(QP_k^\Delta) \to d_x$.
   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 $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ 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 $(QP_k^\Delta) \to d_x$.
   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 $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ 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

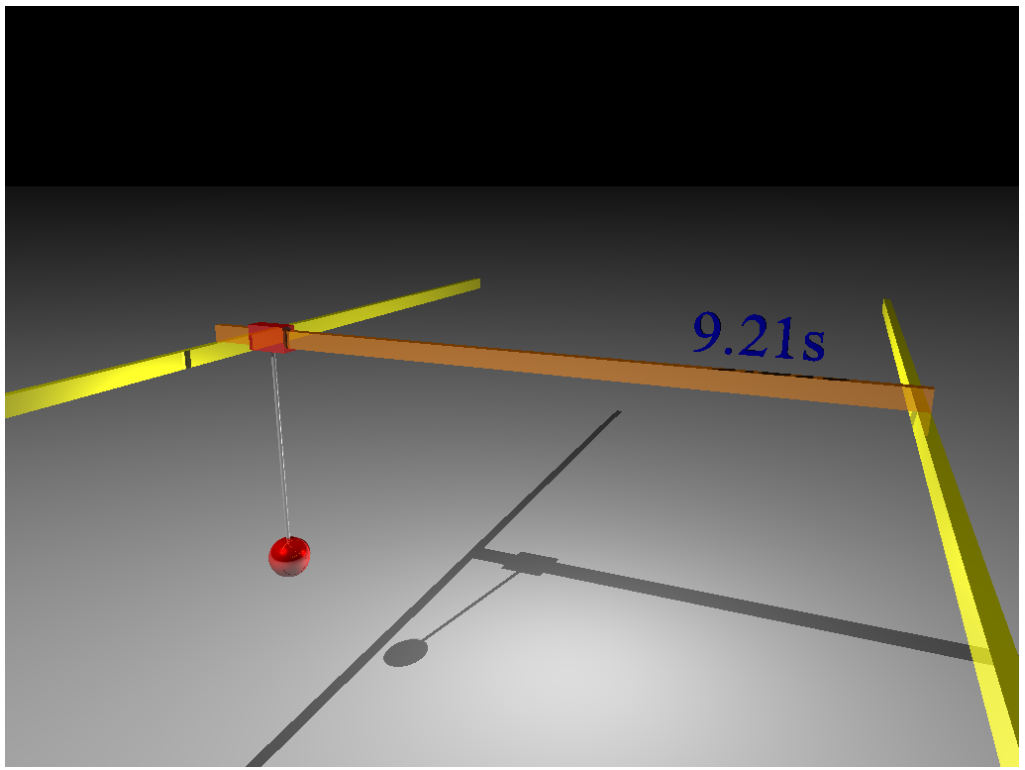
0. Wähle $x^{(0)}$, $u \geq \varphi(x^{(0)})$, $\underline{\Delta} > 0$, $\sigma \in (0,1)$, setze $\mathcal{F}_0 := \{(-\infty, u)\}$, $k := 1$
1. Wiederherstellung: Ausgehend von $x^{(k-1)}$ finde $x^{(k)}$ und $\Delta \geq \underline{\Delta}$ mit (f_k, φ_k) wird von $\mathcal{F}_{k-1}$ akzeptiert und (QP_k^Δ) ist zulässig.
2. SQP-Schritt: Löse $(QP_k^\Delta) \to d_x$.
   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 $\left(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x)\right)$ 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

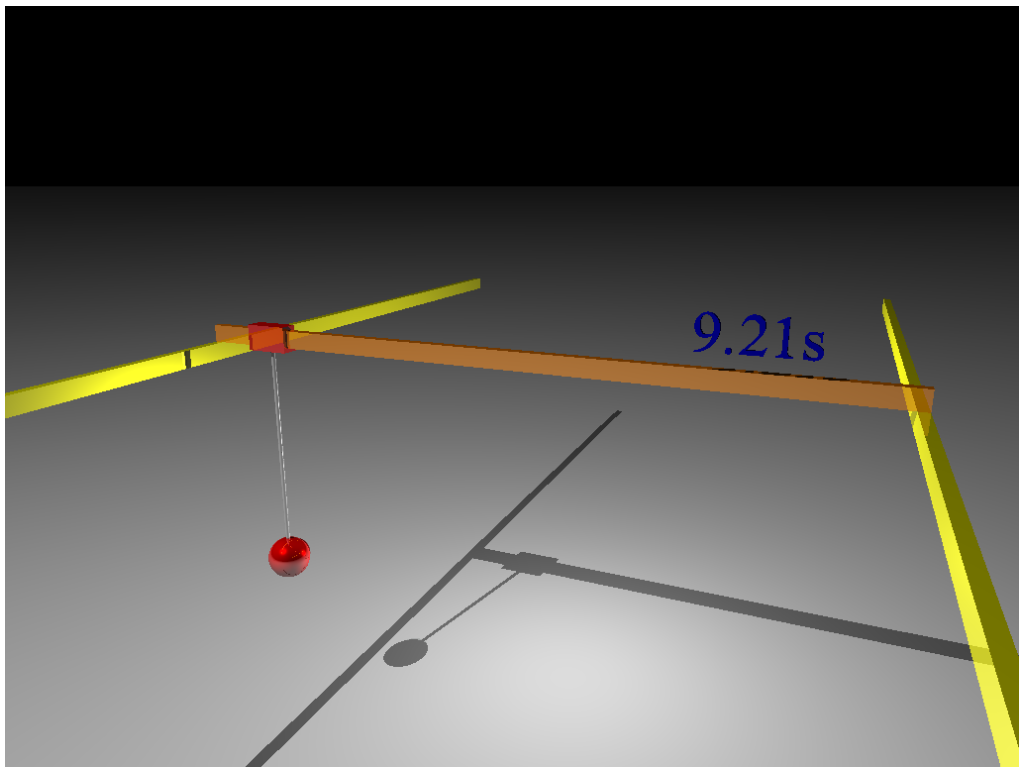
7.6 Anwendungsbeispiel: Optimalsteuerung

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?



7.6 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?

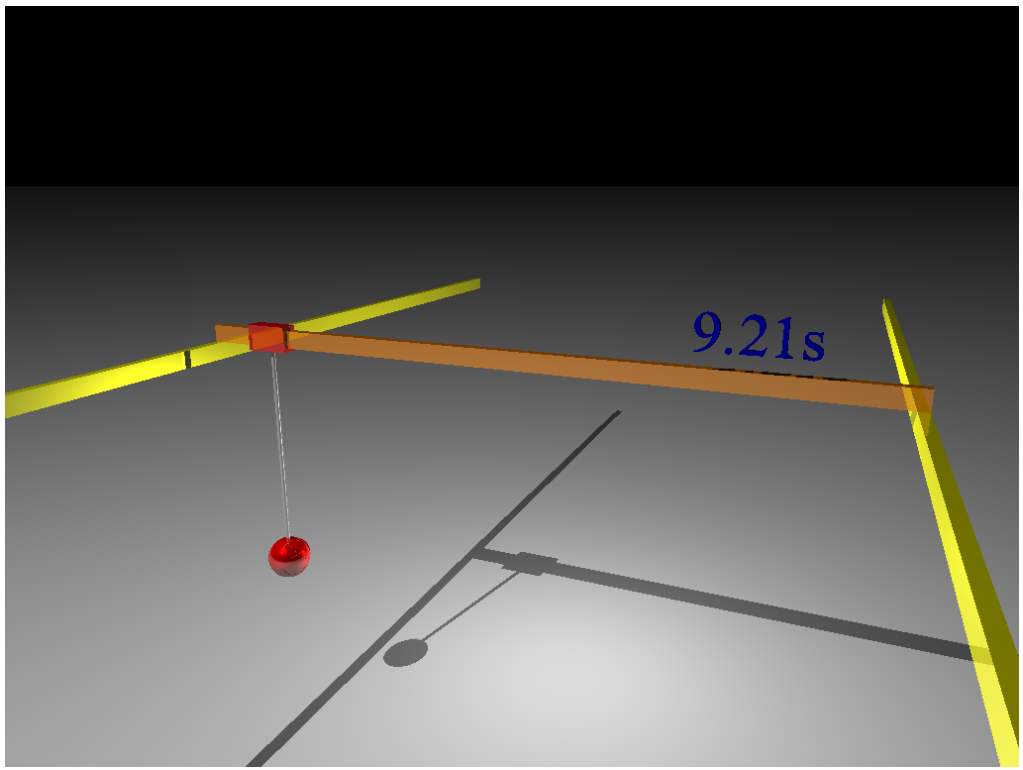


Die erste Idee – bis zur Hälfte voll beschleunigen, dann voll bremsen – ist nicht richtig.

Illustration

7.6 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?

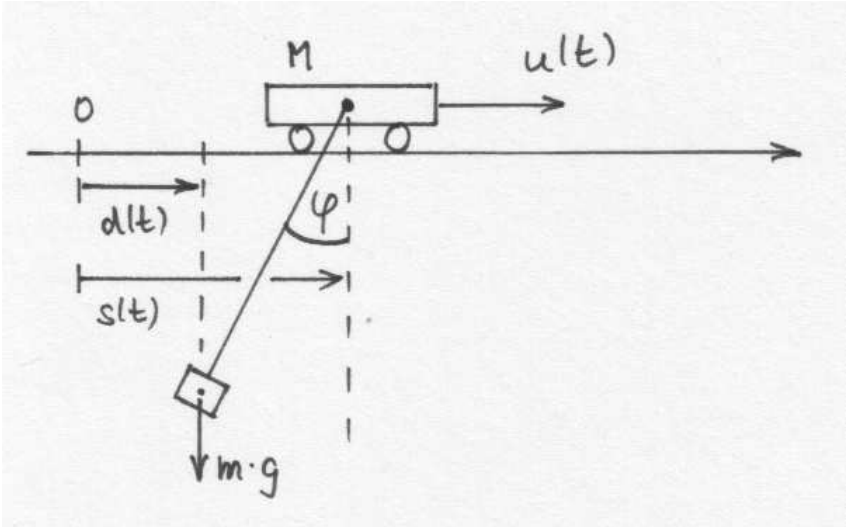


Wie lässt sich das mathematisch modellieren?

(Material und Illustrationen von Prof. Dr. Roland Herzog)

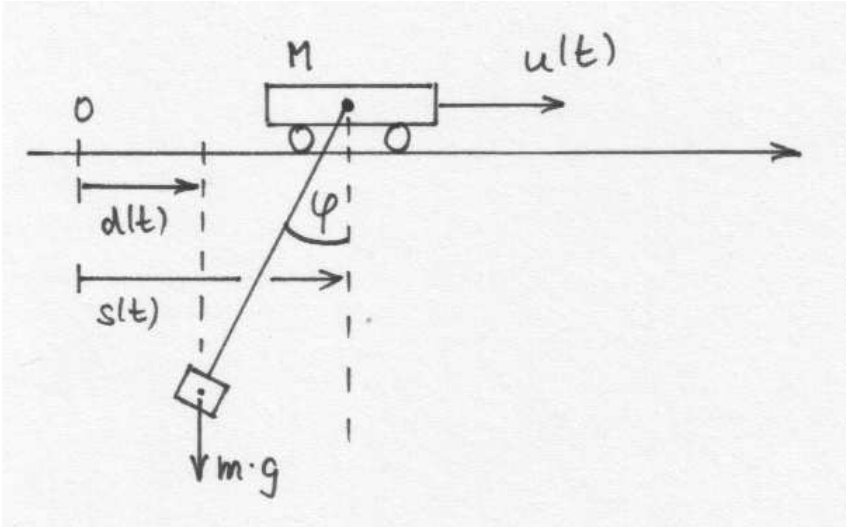
7.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$: 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



7.6.1 (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

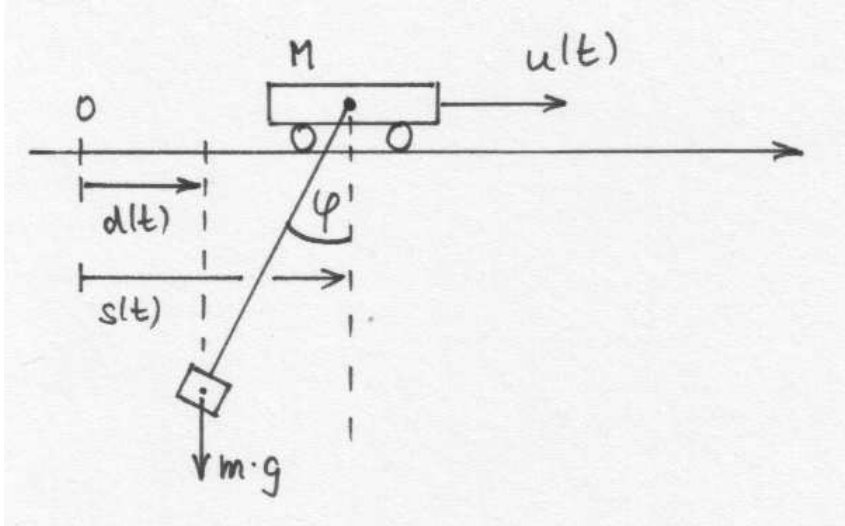


Der Zustand $y(t) \in \mathbb{R}^4$ wird durch Position $s(t)$ und Geschwindigkeit $\dot{s}(t) := \frac{\partial s}{\partial t}$ der Laufkatze sowie Relativposition $z(t) := s(t) - d(t)$ und Relativgeschwindigkeit $\dot{z}(t)$ der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

7.6.1 (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,

$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

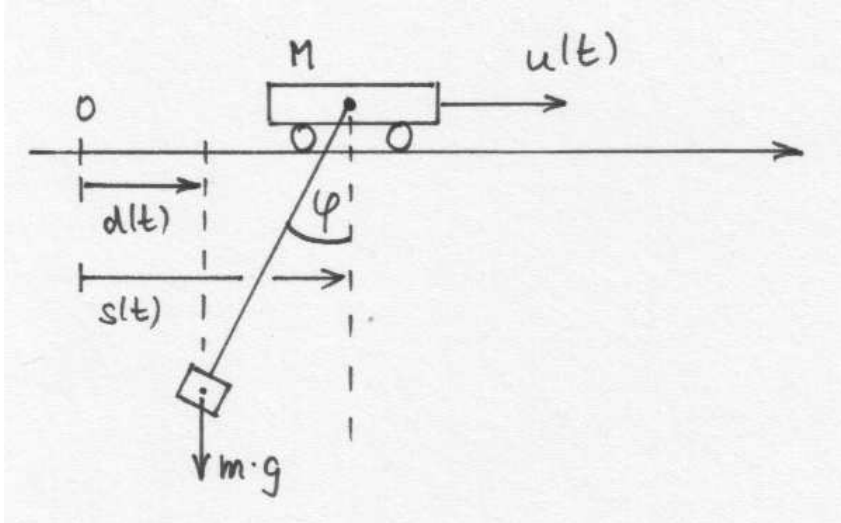
Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

$$\ddot{s}(t) = -\frac{m}{M}\frac{g}{L} z + \frac{u(t)}{M}$$

$$\ddot{z}(t) = -\frac{(m+M)}{M}\frac{g}{L} z + \frac{u(t)}{M}$$

7.6.1 (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,

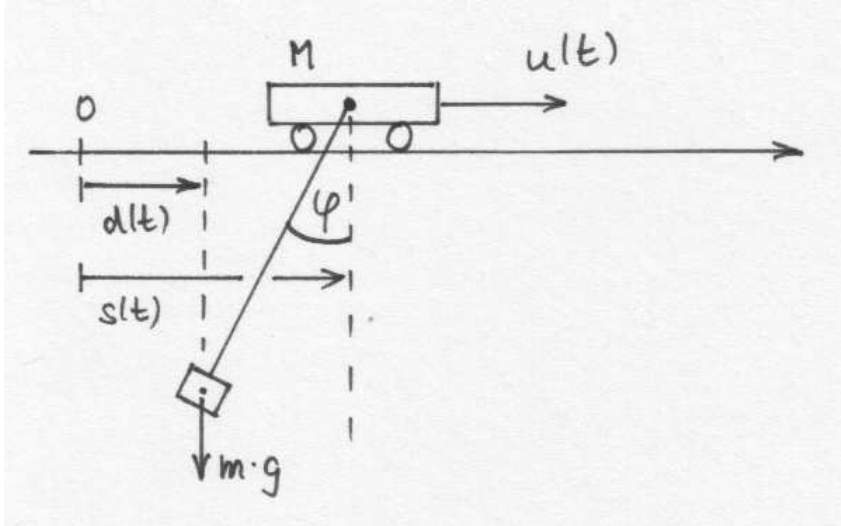
$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

$$\begin{aligned} \ddot{s}(t) &= -\frac{m}{M}\frac{g}{L} z + \frac{u(t)}{M} \\ \ddot{z}(t) &= -\frac{(m+M)}{M}\frac{g}{L} z + \frac{u(t)}{M} \end{aligned} \quad \leftrightarrow \quad \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

7.6.1 (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,

$$y = (s, \dot{s}, z, \dot{z})^T. \qquad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel φ ist

lineare gewöhnliche Differentialgleichung

$$\dot{y} = Ay + Bu \quad \leftrightarrow \quad \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

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.

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.

$$\begin{array}{lll}
\min\limits_{u} & g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt & \text{Zielfunktion(al)} \\
\text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
& y(0) = y_0 & \text{Anfangsbedingungen} \\
& y(T) = y_T & \text{Endbedingungen} \\
& h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.}
\end{array}$$

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.

$$\begin{array}{lll}
\min\limits_{u} & g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt & \text{Zielfunktion(al)} \\
\text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
& y(0) = y_0 & \text{Anfangsbedingungen} \\
& y(T) = y_T & \text{Endbedingungen} \\
& h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.}
\end{array}$$

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.

7.6.2 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.

$\longrightarrow$ Zielfunktion:

7.6.2 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.
$\rightarrow$ Zielfunktion: min $\int_0^T 1dt [= T]$

$\rightarrow y(0) =$

7.6.2 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.

$\rightarrow$ Zielfunktion: min $\int_0^T 1 dt [= T]$

$\rightarrow y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Position der Laufkatze
Relativposition der Last zur Laufkatze
Geschwindigkeit der Laufkatze
Relativgeschwindigkeit der Last zur Laufkatze

$\rightarrow y(T) =$

7.6.2 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.

$\rightarrow$ Zielfunktion: min $\int_0^T 1 dt [= T]$

$\rightarrow$ $y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Position der Laufkatze
Relativposition der Last zur Laufkatze
Geschwindigkeit der Laufkatze
Relativgeschwindigkeit der Last zur Laufkatze

$\rightarrow$ $y(T) = (Q, 0, 0, 0)^T$

Steuerbeschränkungen: $-1 \leq u(t) \leq 1$ [keine unendliche Beschleunigung]

Es könnte auch Zustandsbeschränkungen für $y(t)$ (z.B. Hindernisse) geben.

7.6.2 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.

$\rightarrow$ Zielfunktion: min $\int_0^T 1dt [= T]$

$\rightarrow$ $y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$ Position der Laufkatze
Relativposition der Last zur Laufkatze
Geschwindigkeit der Laufkatze
Relativgeschwindigkeit der Last zur Laufkatze

$\rightarrow$ $y(T) = (Q, 0, 0, 0)^T$

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

$\min_{u,T}$	$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt$	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.

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: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

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: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\widetilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = T f(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau)).$$

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: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\widetilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = T f(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt \quad \rightarrow \quad g(\widetilde{y}(1)) + T \int_0^1 f_0(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau))\, d\tau.$$

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: $\widetilde{y}(\tau) := y(\tau T)$, $\widetilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich τ ergibt sich aus der Kettenregel,

$$\frac{d\widetilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = T f(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t))\, dt \quad \rightarrow \quad g(\widetilde{y}(1)) + T \int_0^1 f_0(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau))\, d\tau.$$

$\min_{\widetilde{u},T}$	$g(\widetilde{y}(1)) + T \int_0^1 f_0(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau))\, d\tau$	
s.t.	$\dot{\widetilde{y}}(\tau) = T f(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau))$	[Def-Bereich von $\widetilde{u}$ unabhängig von T, T ist nun „normale“ Variable!]
	$\widetilde{y}(0) = y_0$	
	$\widetilde{y}(1) = y_T$	
	$h(\tau T, \widetilde{y}(\tau), \widetilde{u}(\tau)) \leq 0.$	

7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$
\begin{aligned}
\min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\
\text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1].
\end{aligned}
$$

7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$
\begin{aligned}
\min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\
\text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1].
\end{aligned}
$$

Variablen:

7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$
\begin{aligned}
\min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\
\text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1].
\end{aligned}
$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$
\begin{aligned}
\min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\
\text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1].
\end{aligned}
$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-dimensional

Wie optimiert man über Funktionen?

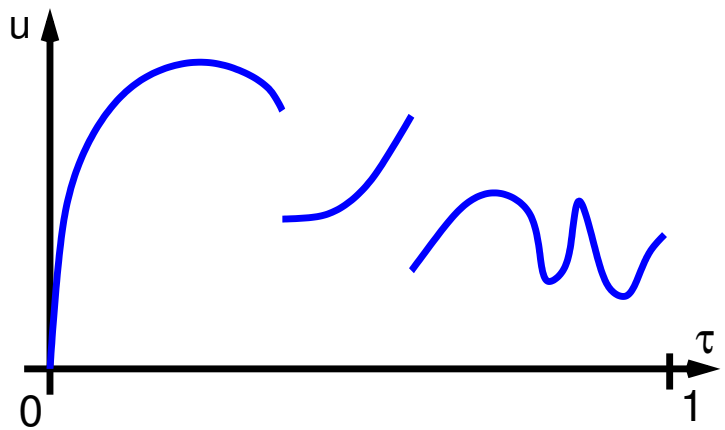
7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$
\begin{aligned}
\min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\
\text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1].
\end{aligned}
$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-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.



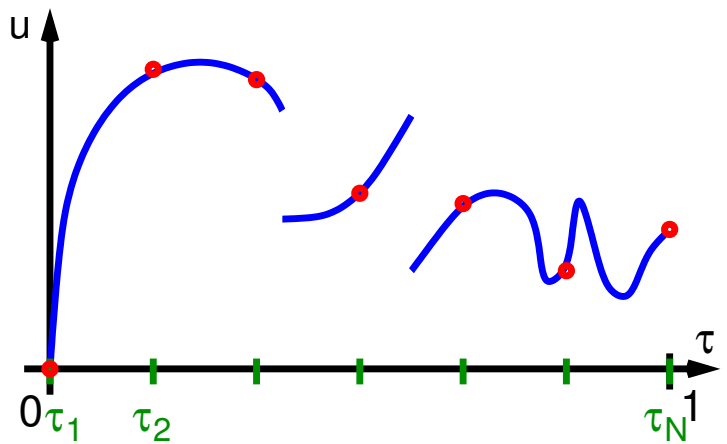
7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{aligned} \min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{aligned}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-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



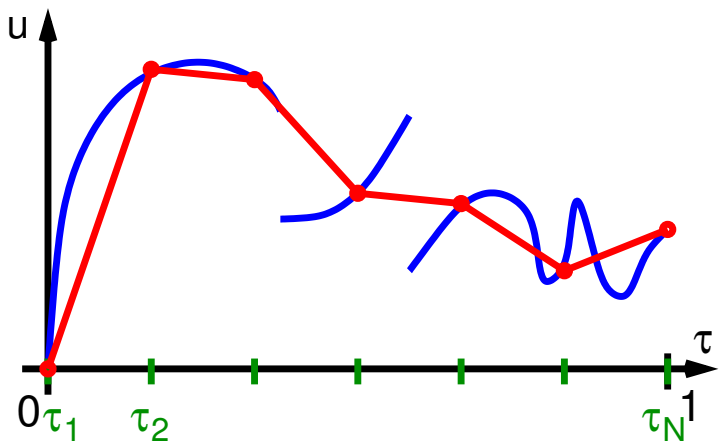
7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{aligned} \min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{aligned}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-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



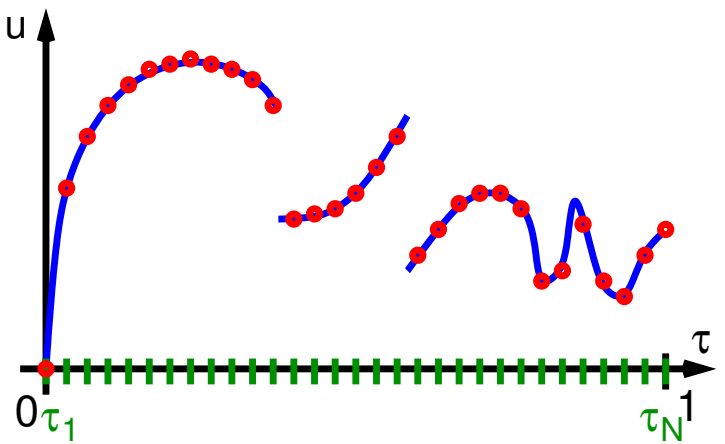
7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{aligned} \min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{aligned}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-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



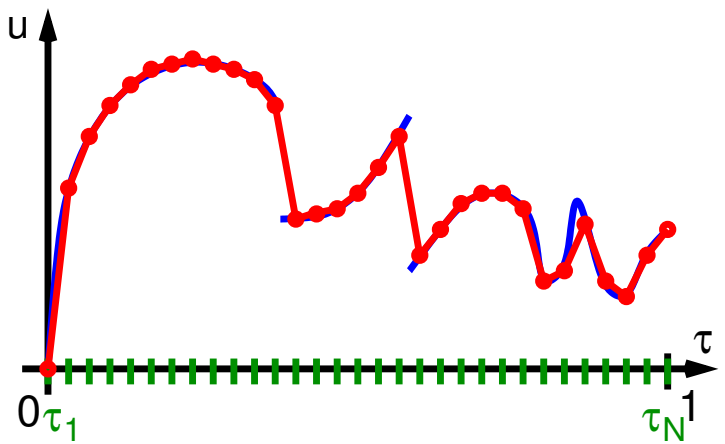
7.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne $\sim$ aber mit neuer Zeit $\tau \in [0,1]$)

$$\begin{aligned} \min_{u:[0,1]\to\mathbb{R},\ T\in\mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \qquad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P,0,0,0)^T \\ & y(1) = (Q,0,0,0)^T \\ & -1 \le u(\tau) \le 1 \quad \text{für alle } \tau \in [0,1]. \end{aligned}$$

Variablen: T, $u(\tau)$ für $\tau \in [0,1]$ $\quad\rightarrow\quad$ ∞-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 $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, 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 $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, 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 $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung?

Laufkatze: Diskretisierung des Optimierungsproblems

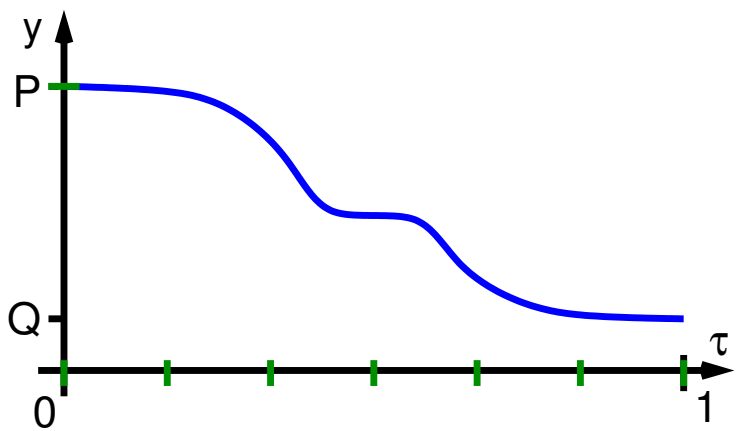
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

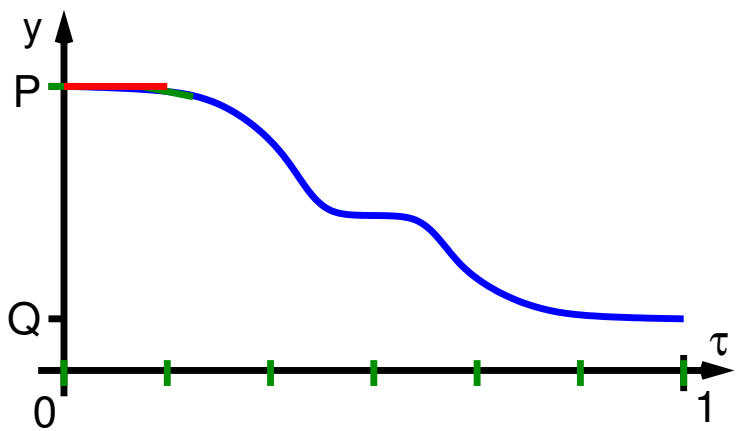
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

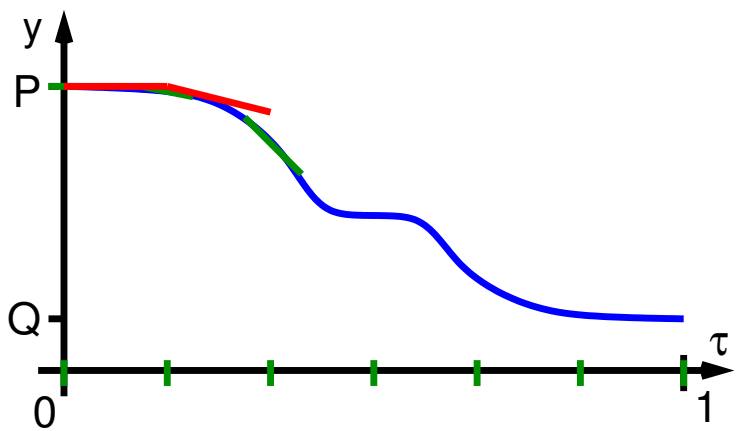
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

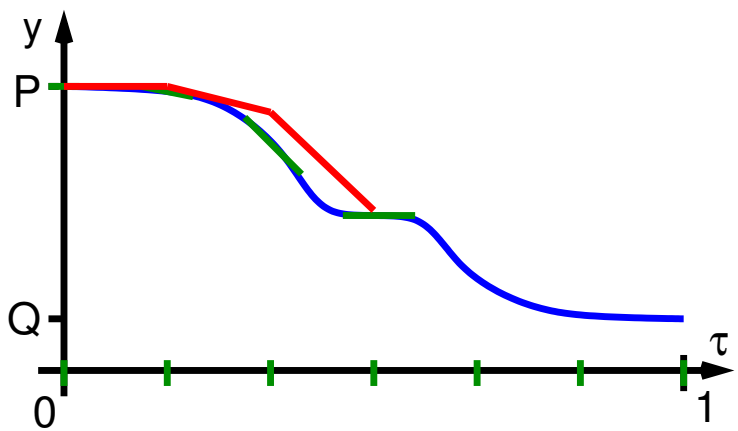
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{lll}
\min & T & \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) & \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T & \\
& y(1) = (Q,0,0,0)^T & \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. &
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

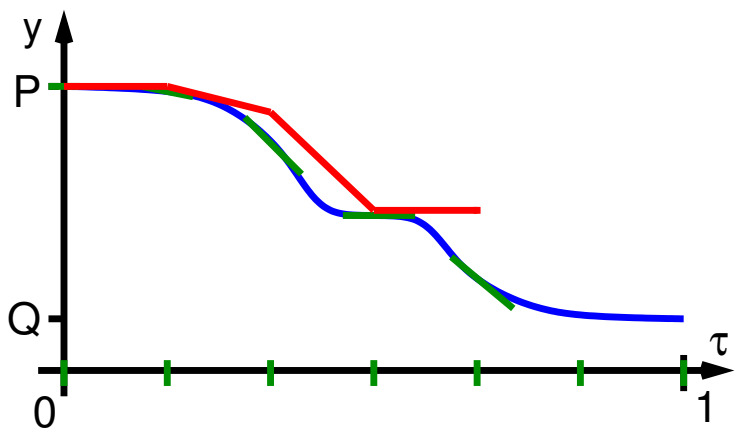
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

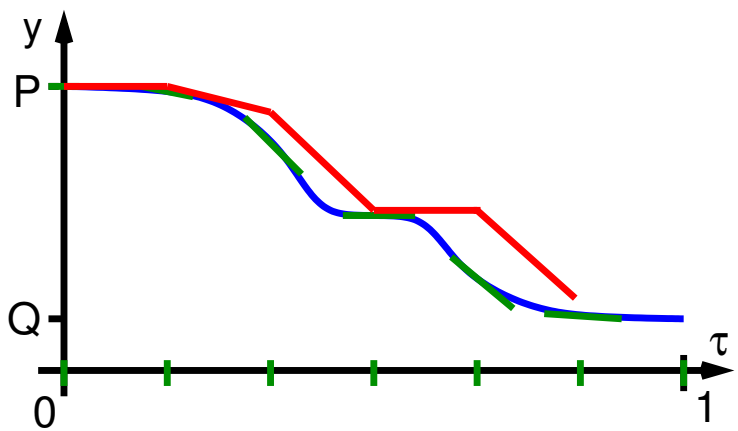
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{lll}
\min & T & \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) & \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T & \\
& y(1) = (Q,0,0,0)^T & \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. &
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



Laufkatze: Diskretisierung des Optimierungsproblems

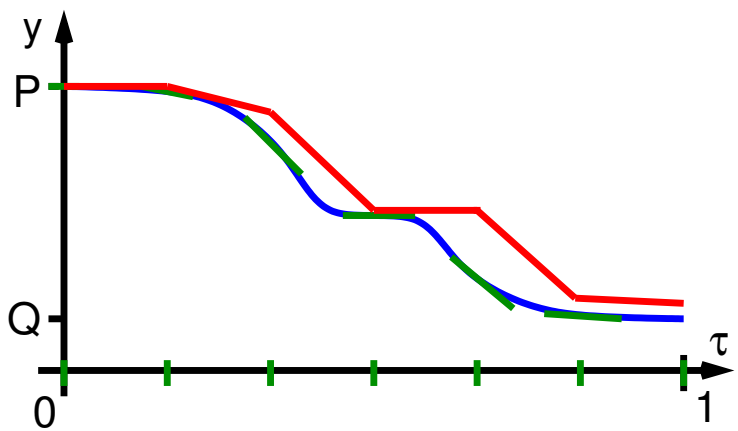
Ersetze die Funktionsvariable $u : [0,1] \to \mathbb{R}$ über die Wahl einer Zeitdiskretisierung $0 = \tau_1 < \tau_2 < \cdots < \tau_N = 1$, $N \in \mathbb{N}$, durch einen Variablenvektor $u \in \mathbb{R}^N$ mit Interpretation $u_i = u(\tau_i)$ und (bei linearer Interpolation) $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i}(u_{i+1} - u_i)$ für $\tau \in (\tau_i, \tau_{i+1})$.

$$
\begin{array}{ll}
\min & T \\
\text{s.t.} & \dot{y}(\tau) = TAy(\tau) + TBu(\tau) \quad \tau \in [0,1] \\
& y(0) = (P,0,0,0)^T \\
& y(1) = (Q,0,0,0)^T \\
& -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+.
\end{array}
$$

Was geschieht mit der Differentialgleichung? Berechne

$$y(\tau) = y(0) + \int_0^\tau TAy(t) + TBu(t)dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).



$\to$ Funktionsorakel $y(\tau) = R(u,\tau)$, das $y(0) = (P,0,0,0)^T$ erfüllt.

$\to$ Es bleibt nur eine Glgs-Nebenbedingung: $R(u,1) - (Q,0,0,0)^T = 0$

Funktionsauswertung mit Gradienten über automatisches Differenzieren!

Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{array}{ll} \min & T \\ \text{s.t.} & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \leq u_i \leq 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{array}$$

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.

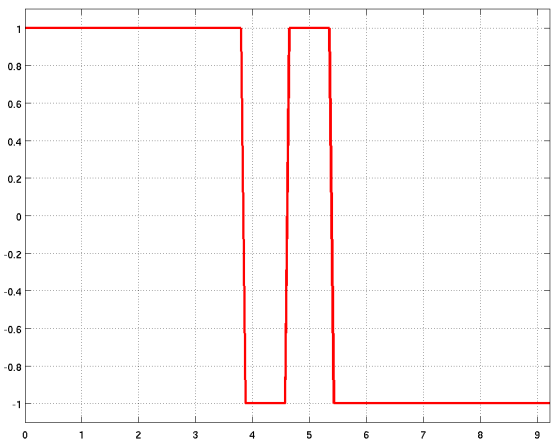
Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{aligned}$$

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.

$\longrightarrow$ 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



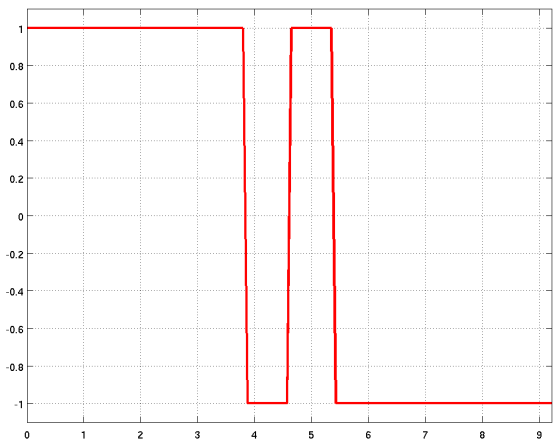
Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{aligned}$$

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.

$\longrightarrow$ 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

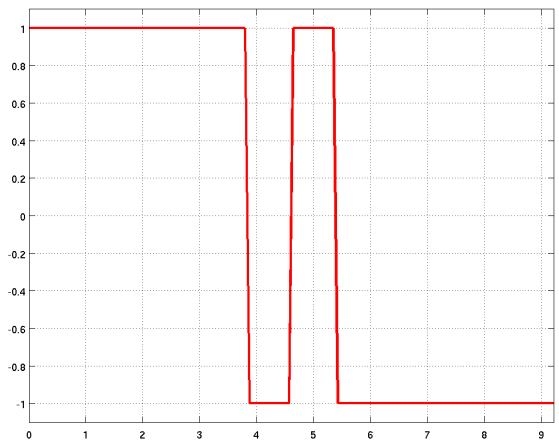
Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u,1) - (Q,0,0,0)^T = 0 \\ & -1 \le u_i \le 1 \ (i = 1,\ldots,N),\ T \in \mathbb{R}_+. \end{aligned}$$

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.

$\longrightarrow$ 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

8 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!).

8 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.

8 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 nichtglatten 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.

8 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/ Subdifferentiale 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

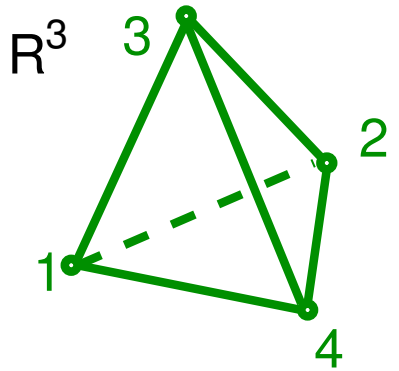
8.1 Das Simplex-Verfahren von Nelder und Mead

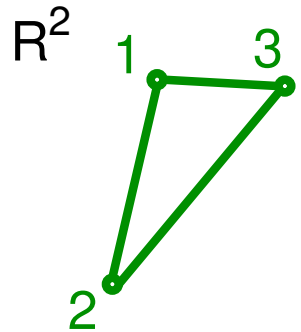
. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

8.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.

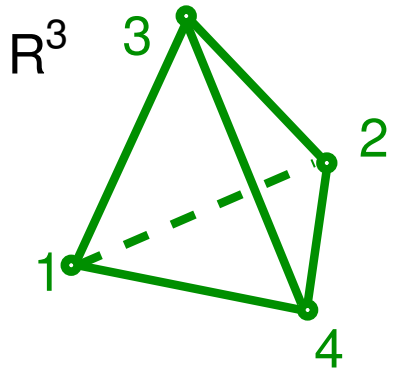


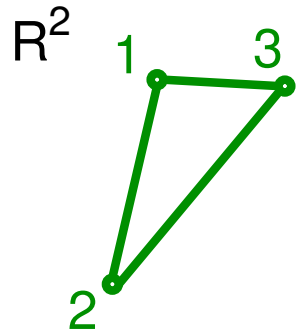


8.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat <u>nichts</u> mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.

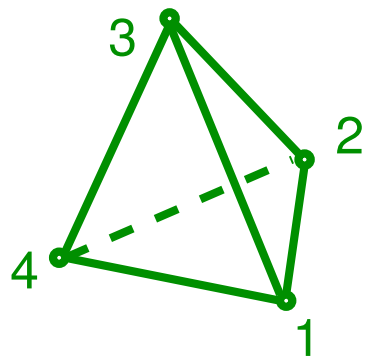




Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)}).$$

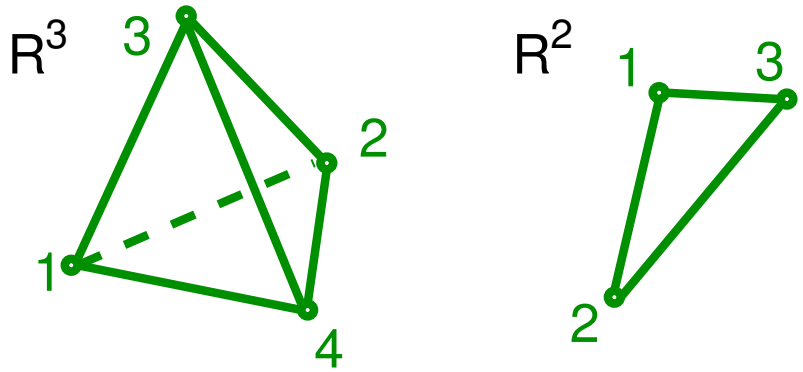
Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \ldots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.



8.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat <u>nichts</u> mit dem Simplex-Verf. der Linearen Optimierung zu tun!

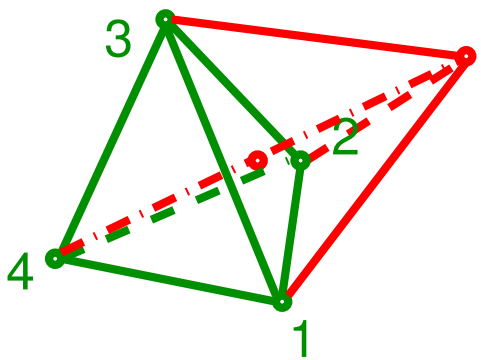
Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.



Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)}).$$

Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \ldots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.
$\to$ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).



8.1 Das Simplex-Verfahren von Nelder und Mead

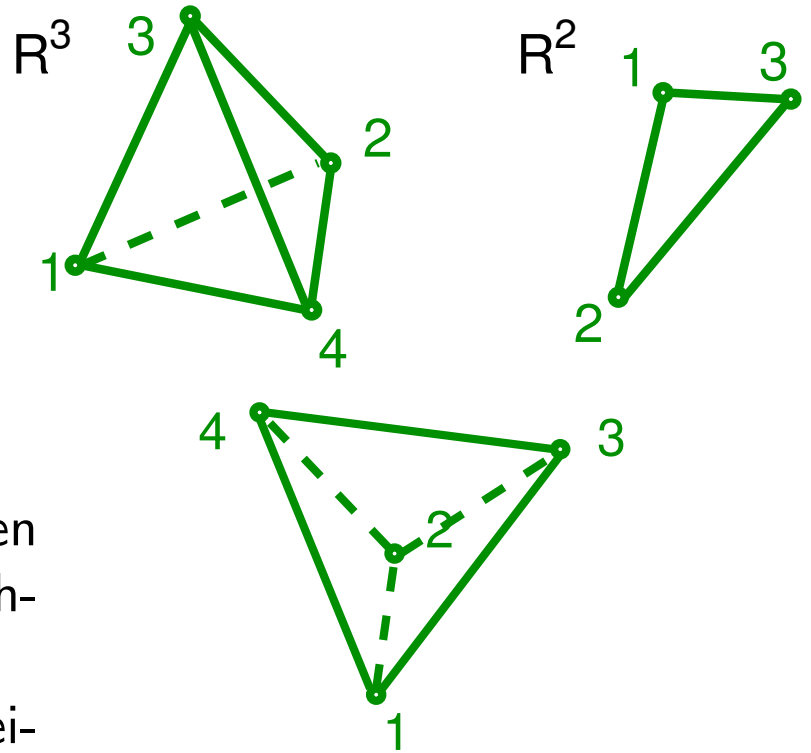
. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.

Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)}).$$

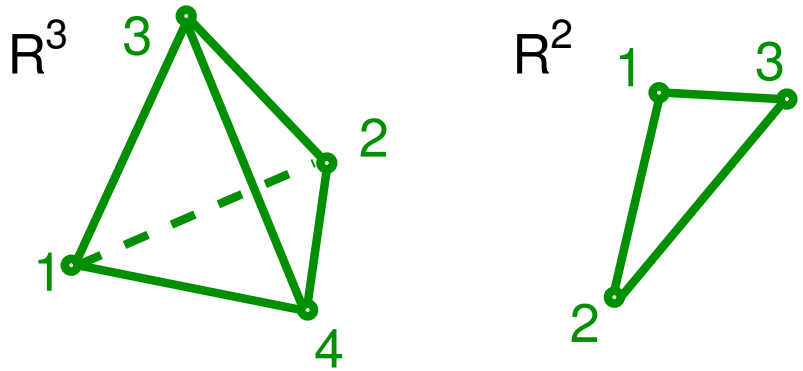
Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \ldots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.
$\to$ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).



8.1 Das Simplex-Verfahren von Nelder und Mead

. . . hat nichts mit dem Simplex-Verf. der Linearen Optimierung zu tun!

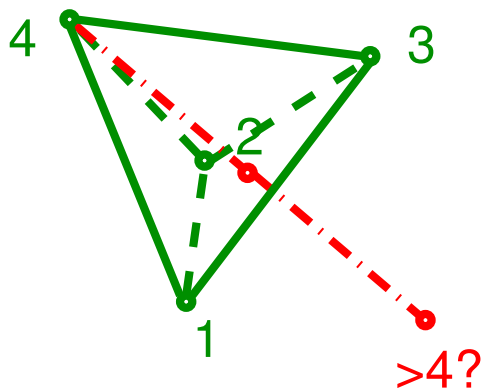
Idee: Um $f : \mathbb{R}^n \to \mathbb{R}$ zu minimieren, wähle $n+1$ affin unabhängige Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$, diese bilden die Ecken eines n-dimensionalen Simplex.



Sortiere die Punkte so, dass

$$f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)}).$$

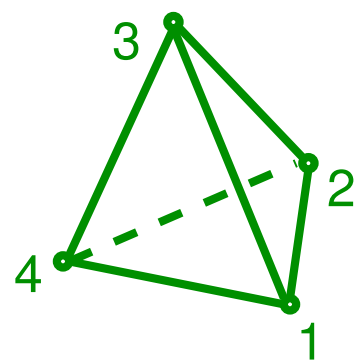
Vorstellung: Die Ebene durch die besseren Punkte $x^{(1)}, \ldots, x^{(n)}$ trennt den schlechten Punkt $x^{(n+1)}$ von den guten.
$\to$ „Klappe“ den Punkt auf die andere Seite (am Baryzentrum spiegeln).



Ist der Punkt immer noch am schlechtesten, schrumpfe den Simplex in dieser Richtung, ist er sehr gut, strecke die Richtung.

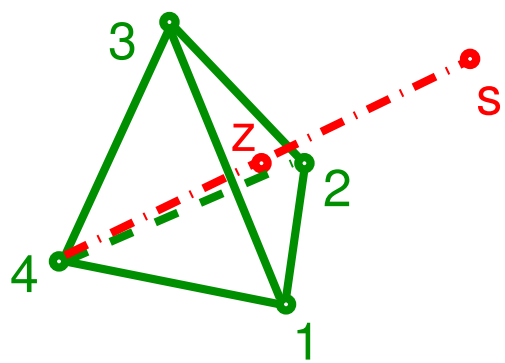
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, wähle $\alpha > 0$, $\beta > \max\{1, \alpha\}$, $\gamma \in (0, 1)$ [z.B., $\alpha = 1$, $\beta = 2$, $\gamma = \frac{1}{2}$]



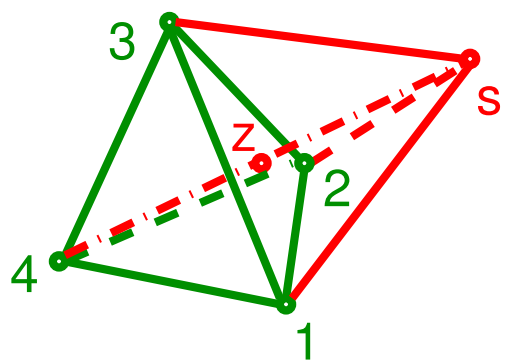
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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).



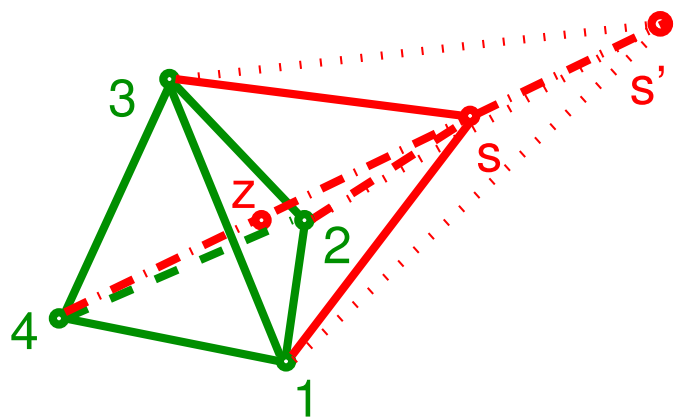
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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.



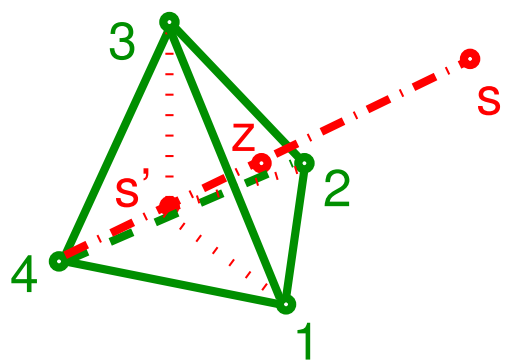
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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.



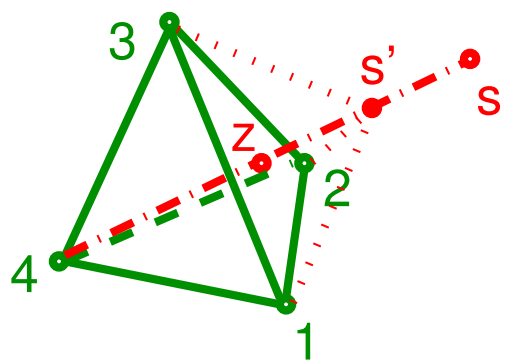
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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 $x^{(n+1)} \leftarrow s'$,



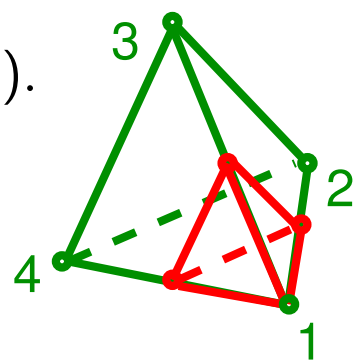
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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 $x^{(n+1)} \leftarrow s'$,



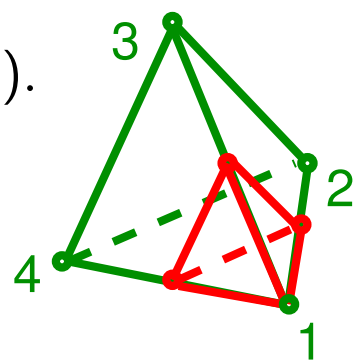
Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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 $x^{(1)}$:
   $x^{(i)} \leftarrow \frac{1}{2}(x^{(1)} + x^{(i)})$, $(i = 2, \ldots, n+1)$.
   Sortiere neu, GOTO 1.



Algorithmisches Schema des Nelder-Mead-Verfahrens:

0. Wähle affin unabh. Punkte $x^{(1)}, \ldots, x^{(n+1)} \in \mathbb{R}^n$ und sortiere sie, $f(x^{(1)}) \leq \cdots \leq f(x^{(n+1)})$, 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 $x^{(1)}$:
   $x^{(i)} \leftarrow \frac{1}{2}(x^{(1)} + x^{(i)})$, $(i = 2, \ldots, n+1)$.
   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 $\longrightarrow$ 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

8.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen $\longrightarrow$ wird gerne verwendet, wenn jede Funktionsauswertung einem realen Experiment oder einer aufwendigen Simulation entspricht.

8.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen $\longrightarrow$ wird gerne verwendet, wenn jede Funktionsauswertung einem realen Experiment oder einer aufwendigen Simulation entspricht.

Idee: Beachte folgende Leitlinien bei Entwurf von Modell und Verfahren:

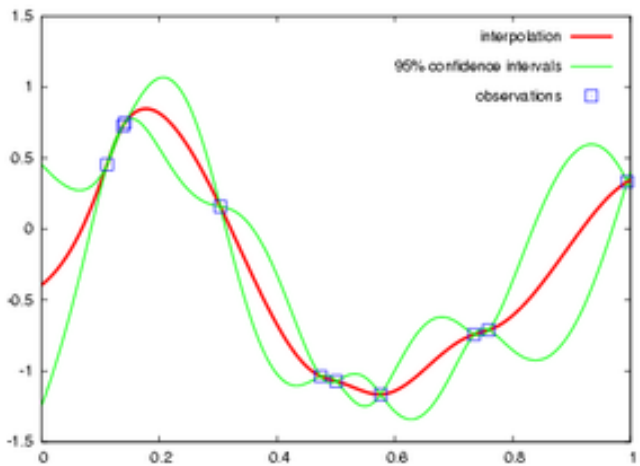
- Das Modell sollte die Funktion in bekannten Punkten interpolieren.
- Oszillationen sollten im Modell vermieden werden.
- In der Nähe bekannter Punkte sind Modellpunkte vertrauenswürdiger.
- Optimierungsaufwand ist vernachlässigbar gegenüber Auswertungen.

8.2 Das Kriging-Verfahren

Krige war Montaningenieur in Südafrika. Um mit möglichst wenigen Bohrungen Bodenschätze zu finden, orientierte er sich an statistischen Modellen $\longrightarrow$ 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 $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.

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 $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.

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.

$\rightarrow$ 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, $\hat{f}(x^{(i)}) = f_i$ $(i = 1, \ldots, k)$.

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.

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.

$\rightarrow$ 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, $\hat{f}(x^{(i)}) = f_i$ $(i = 1, \ldots, k)$.

$\rightarrow$ Löse
$$\begin{bmatrix} b_1(x^{(1)}) & \ldots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \ldots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Das Modell der Funktion:

Gegeben seien k Punkte $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.
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.

$\rightarrow$ 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, $\hat{f}(x^{(i)}) = f_i$ $(i = 1, \ldots, k)$.

$\rightarrow$ Löse
$$\begin{bmatrix} b_1(x^{(1)}) & \ldots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \ldots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.
$\rightarrow$ **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 $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.
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.

$\rightarrow$ 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, $\hat{f}(x^{(i)}) = f_i$ $(i = 1, \ldots, k)$.

$\rightarrow$ Löse
$$\begin{bmatrix} b_1(x^{(1)}) & \ldots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \ldots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.
$\rightarrow$ **Skalierung:** Bestimme eine Norm $\|d\|_A^2 := d^T A d$ mit $A \succ 0$ wie folgt:
$\|\cdot\|_A$ ist gut, wenn für jedes $j \in \{1, \ldots, k\}$ 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 $x^{(1)}, \ldots, x^{(k)} \in \mathbb{R}^n$ mit Werten $f_1, \ldots, f_k \in \mathbb{R}$.
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.

$\rightarrow$ 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, $\hat{f}(x^{(i)}) = f_i$ $(i = 1, \ldots, k)$.

$\rightarrow$ Löse
$$\begin{bmatrix} b_1(x^{(1)}) & \ldots & b_k(x^{(1)}) \\ \vdots & \ddots & \vdots \\ b_1(x^{(k)}) & \ldots & b_k(x^{(k)}) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_k \end{bmatrix} = \begin{bmatrix} f_1 - \mu \\ \vdots \\ f_k - \mu \end{bmatrix}.$$

Aber: Die Koordinaten der $x^{(i)}$ gehören oft zu unterschiedlichen physikalischen Größen, die Euklidische Norm $\|x - x^{(i)}\|$ macht wenig Sinn.

$\rightarrow$ **Skalierung:** Bestimme eine Norm $\|d\|_A^2 := d^T A d$ mit $A \succ 0$ wie folgt:
$\|\cdot\|_A$ ist gut, wenn für jedes $j \in \{1, \ldots, k\}$ 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.

$\rightarrow$ Löse $\min_{A \succ 0} \max_{1 \le j \le k} e_A^j$ (ein nichtglattes Problem) $\rightarrow A_*$, b_i^*, $\hat{f}_*$

Man hofft nun, dass die Fehler $e^j_{A_*}$ auch gute **Fehlerschätzer** für die Abweichung des Modells $\hat{f}_*$ von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}^j_{A^*}$ über die Distanz $d^*_j := \min_{1 \le i \le k, i \neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$ einen Fehler von $e^j_{A_*}$.

Man hofft nun, dass die Fehler $e^j_{A_*}$ auch gute **Fehlerschätzer** für die Abweichung des Modells $\hat{f}_*$ von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}^j_{A^*}$ über die Distanz $d^*_j := \min_{1\le i\le k, i\neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$ einen Fehler von $e^j_{A_*}$. Pro Distanzeinheit von $\|\cdot\|_{A_*}$ weicht damit $\hat{f}_*$ von f hoffentlich nur um

$$\bar{s} := \max_{1\le j\le k} \frac{e^j_{A_*}}{d^*_j},$$

ab, vielleicht noch multipliziert mit einer Konstanten $\nu > 1$, also

hoffentlich $f(x) \ge \hat{f}_*(x) - \nu\bar{s} \min_{0\le i\le k} \|x - x^{(i)}\|_{A_*} =: g(x)$ für $x \in \Omega$,

wobei die Grundmenge $\Omega \subset \mathbb{R}^n$ als kompakt angenommen wird.

Man hofft nun, dass die Fehler $e^j_{A_*}$ auch gute **Fehlerschätzer** für die Abweichung des Modells $\hat{f}_*$ von der Funktion f liefern:

Für j zeigt das Modell $\hat{f}^j_{A^*}$ über die Distanz $d^*_j := \min\limits_{1\le i\le k, i\neq j} \|x^{(j)} - x^{(i)}\|_{A_*}$ einen Fehler von $e^j_{A_*}$. Pro Distanzeinheit von $\|\cdot\|_{A_*}$ weicht damit $\hat{f}_*$ von f hoffentlich nur um

$$\bar{s} := \max_{1\le j\le k} \frac{e^j_{A_*}}{d^*_j},$$

ab, vielleicht noch multipliziert mit einer Konstanten $\nu > 1$, also

hoffentlich $f(x) \ge \hat{f}_*(x) - \nu\bar{s} \min\limits_{0\le i\le k} \|x - x^{(i)}\|_{A_*} =: g(x)$ 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 $\min\limits_{x\in\Omega} g(x)$.

Eine Optimallösung dieses Problems wird als nächster Auswertungspunkt gewählt. Dann wird für die nun $k+1$ Punkte das Modell erneut berechnet, etc.

Inhaltsübersicht

8.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ 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.

8.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ 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 Lösungskoordinaten sei.

8.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ 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 Lösungskoordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \tfrac{1}{2} x^T Q^{(k)} x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, $(i = 0, \ldots, 2n)$ die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert.

8.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ 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 Lösungskoordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \tfrac{1}{2} x^T Q^{(k)} x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, $(i = 0, \ldots, 2n)$ die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \qquad \begin{array}{ll} \min & \sum_{1 \le i,j \le n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2 \\ \text{s.t.} & \frac{1}{2}(x^{(i)})^T Q^{(k)} x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \ldots, 2n, \\ & Q^{(k)} \in S^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}. \end{array}$$

8.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ 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 Lösungskoordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \tfrac{1}{2} x^T Q^{(k)} x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, $(i = 0, \ldots, 2n)$ die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \qquad \begin{array}{ll} \min & \sum_{1 \le i,j \le n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2 \\ \text{s.t.} & \frac{1}{2}(x^{(i)})^T Q^{(k)} x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \ldots, 2n, \\ & Q^{(k)} \in S^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}. \end{array}$$

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.

8.3 NEWUOA (NEW Unconstrained Opt. Alg.)

Interpoliert die „letzten“ $2n+1$ Funktionswerte von $f : \mathbb{R}^n \to \mathbb{R}$ 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 Lösungskoordinaten sei.

In Iteration k bestimmt NEWUOA ein quadratisches Modell

$$m^{(k)}(x) = \tfrac{1}{2} x^T Q^{(k)} x + (q^{(k)})^T x + c^{(k)},$$

das in Punkten $x^{(i)} \in \mathbb{R}^n$, $(i = 0, \ldots, 2n)$ die Werte f_i annimmt und, für $k > 0$, $\|\nabla^2 m^{(k-1)} - \nabla^2 m^{(k)}\|$ minimiert. Dazu löst es

$$(QP_m^k) \qquad \begin{array}{ll} \min & \sum_{1 \le i,j \le n} (Q_{ij}^{(k-1)} - Q_{ij}^{(k)})^2 \\ \text{s.t.} & \frac{1}{2}(x^{(i)})^T Q^{(k)} x^{(i)} + (q^{(k)})^T x^{(i)} + c^{(k)} = f_i, \quad i = 0, \ldots, 2n, \\ & Q^{(k)} \in S^n, q^{(k)} \in \mathbb{R}^n, c^{(k)} \in \mathbb{R}. \end{array}$$

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.

Die Initialisierung von $m^{(0)}$ verwendet die Punkte $x^{(0)}$, $x^{(2i-1)} := x^{(0)} - \bar{\Delta} e_i$, $x^{(2i)} := x^{(0)} + \bar{\Delta} e_i$ $(i = 1, \ldots, n) \to c^{(0)} = f_0$, $q^{(0)}$, 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 $(i = 1, \ldots, 2n)$ 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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $\|x^{(j)} - x^{(\hat{\imath})}\|$ 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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $\|x^{(j)} - x^{(\hat{\imath})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{\imath}} - f(x^{(\hat{\imath})} + d)}{f_{\hat{\imath}} - m^{(k)}(x^{(\hat{\imath})} + d)}$, passe Δ $(\geq \delta)$ entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{\imath})} + d$ 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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $\|x^{(j)} - x^{(\hat{\imath})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{\imath}} - f(x^{(\hat{\imath})} + d)}{f_{\hat{\imath}} - m^{(k)}(x^{(\hat{\imath})} + d)}$, passe Δ $(\geq \delta)$ entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{\imath})} + d$ 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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $\|x^{(j)} - x^{(\hat{\imath})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{\imath}} - f(x^{(\hat{\imath})} + d)}{f_{\hat{\imath}} - m^{(k)}(x^{(\hat{\imath})} + d)}$, passe Δ $(\geq \delta)$ entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{\imath})} + d$ 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 NEWUOA:

0. Initialisierung: Wähle $x^{(0)}$, $[\underline{\Delta}, \bar{\Delta}]$, $\delta \leftarrow \bar{\Delta}$, $\Delta \leftarrow \bar{\Delta}$
   Bestimme $x^{(i)}$, f_i $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $\|x^{(j)} - x^{(\hat{\imath})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{\imath}} - f(x^{(\hat{\imath})} + d)}{f_{\hat{\imath}} - m^{(k)}(x^{(\hat{\imath})} + d)}$, passe Δ $(\geq \delta)$ entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{\imath})} + d$ 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 $(i = 1, \ldots, 2n)$ und $m^{(0)}$, setze $k = 0$.
1. Finde $\hat{\imath} \in \{0, \ldots, 2n\}$ mit $f_{\hat{\imath}} \leq f_i$ $(\forall i)$ und löse das Trustregionproblem $\min_{\|d\| \leq \Delta} m^{(k)}(x^{(\hat{\imath})} + d)$ (mit inexaktem konjugierten Gradientenverf.)
2. Falls $\|d\| \leq \frac{1}{2}\rho$: [Modellumgebung verkleinern?]
   Ist zum 3. Mal $|f(x^{(\hat{\imath})} + d) - m^{(k)}(x^{(\hat{\imath})} + d)|$ 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 $\|x^{(j)} - x^{(\hat{\imath})}\|$ maximal und GOTO 4.
3. Für $\|d\| \geq \frac{1}{2}\delta$ berechne $\rho = \frac{f_{\hat{\imath}} - f(x^{(\hat{\imath})} + d)}{f_{\hat{\imath}} - m^{(k)}(x^{(\hat{\imath})} + d)}$, passe Δ $(\geq \delta)$ entsprechend an und setze $x^{(j)} \leftarrow x^{(\hat{\imath})} + d$ 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.