

# Inhalt

## Restringierte Optimierung: Verfahren

Strafverfahren

Augmentierte-Lagrange-Verfahren

Barriere-Verfahren

Quadratische Optimierung

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung

## 8.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

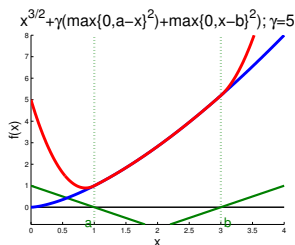
$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

Unzul. mit Strafparameter  $\gamma > 0$  bestrafen:

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\}^2 + \max\{0, g_2(x)\}^2]$$

Unrestringiertes Problem:  $\min_{x \in \mathbb{R}^n} f_\gamma(x)$

Suche Minimum  $x_\gamma^*$ , vergrößere  $\gamma$ , etc.



## 8.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

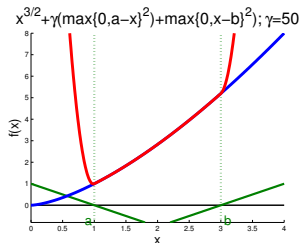
$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

Unzul. mit Strafparameter  $\gamma > 0$  bestrafen:

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\}^2 + \max\{0, g_2(x)\}^2]$$

Unrestringiertes Problem:  $\min_{x \in \mathbb{R}^n} f_\gamma(x)$

Suche Minimum  $x_\gamma^*$ , vergrößere  $\gamma$ , etc.



## 8.1 Strafverfahren (penalty methods)

Idee: Bestrafe die Verletzung von Nebenbedingungen in der Zielfunktion mit einem Strafterm, der mit einem Strafparameter gewichtet wird, und löse nun eine Folge unrestringierter Probleme mit dieser Zielfunktion für wachsenden Strafparameter, bis die Verletzung klein genug ist.

$$\begin{array}{ll} \min f(x) & \rightarrow \min_{x \in \mathbb{R}^n} f_\gamma(x) := f(x) + \gamma \left[ \underbrace{\sum_{i \in \mathcal{E}} \Psi(h_i(x)) + \sum_{i \in \mathcal{I}} \Phi(g_i(x))}_{=: \sigma(x) \dots \text{Straffunktion}} \right] \\ \text{s.t. } h_i(x) = 0 \quad i \in \mathcal{E} & \\ g_i(x) \leq 0 \quad i \in \mathcal{I} & \\ x \in \mathbb{R}^n & \end{array}$$

$\gamma \geq 0 \dots$  **Strafparameter**

mit  $\begin{cases} \Psi(y) > 0 & \text{für } y \neq 0, \\ \Psi(y) = 0 & \text{für } y = 0, \end{cases}$  und  $\begin{cases} \Phi(y) > 0 & \text{für } y > 0, \\ \Phi(y) = 0 & \text{für } y \leq 0. \end{cases}$  [Relax.!]

**Strafproblem:**  $\max_{\gamma \geq 0} \Theta(\gamma)$  mit  $\Theta(\gamma) := \inf_{x \in \mathbb{R}^n} f_\gamma(x)$

### Satz (Korrektheit von Strafverfahren)

Ist das Originalproblem zulässig und gibt es für jedes  $\gamma \geq 0$  ein globales Optimum  $x_\gamma$ , für das  $\Theta(\gamma)$  angenommen wird, und ist  $\{x_\gamma : \gamma \geq 0\}$  beschränkt und abgeschlossen, dann sind alle Häufungspunkte von  $\{x_\gamma\}$  globale Optimallösungen des Originalproblems.

## Exakte Strafverfahren und $l_1$ -Straffunktion

Gibt es für jede lokale Optimallösung  $x^*$  des Originalproblems ein endliches  $\gamma_{x^*} > 0$ , sodass  $x^*$  auch lokales Optimum von  $f_\gamma(\cdot)$  für alle  $\gamma \geq \gamma_{x^*}$  ist, heißt das Strafverfahren und die Straffunktion exakt.

Bsp: Die  $l_1$ -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

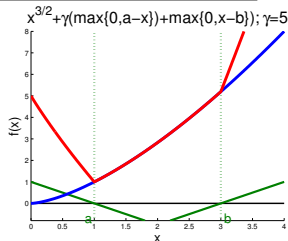
### Satz (Exaktheit der $l_1$ -Straffunktion für konvexe Probleme)

Sei  $(x^*, \mu^*, \lambda^*)$  ein KKT-Punkt eines konvexen Problems  $(P)$ , dann ist für  $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$  der Punkt  $x^*$  auch globales Optimum von  $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$ .

$$\begin{aligned} \text{Bsp: } \min \quad & f(x) := x^{\frac{3}{2}} \\ \text{s.t.} \quad & g_1(x) := a - x \leq 0 \\ & g_2(x) := x - b \leq 0 \end{aligned}$$

$$f_\gamma(x) := x^{\frac{3}{2}} + \gamma[\max\{0, g_1(x)\} + \max\{0, g_2(x)\}]$$

$$\begin{aligned} \text{In } x^* = 1 \text{ ist } g_1 \text{ aktiv mit Lagrangemult. } \lambda_1 = \frac{3}{2} \\ \text{[KKT: } 0 = \nabla f(1) + \lambda_1 \nabla g_1(1) = \frac{3}{2} - \lambda_1 \end{aligned}$$



## Exakte Strafverfahren und $l_1$ -Straffunktion

Gibt es für jede lokale Optimallösung  $x^*$  des Originalproblems ein endliches  $\gamma_{x^*} > 0$ , sodass  $x^*$  auch lokales Optimum von  $f_\gamma(\cdot)$  für alle  $\gamma \geq \gamma_{x^*}$  ist, heißt das Strafverfahren und die Straffunktion exakt.

---

Bsp: Die  $l_1$ -Straffunktion ist definiert durch

$$\Psi(y) := |y| \text{ und } \Phi(y) := \max\{0, y\}$$

und ist exakt für konvexe Probleme:

### Satz (Exaktheit der $l_1$ -Straffunktion für konvexe Probleme)

Sei  $(x^*, \mu^*, \lambda^*)$  ein KKT-Punkt eines konvexen Problems  $(P)$ , dann ist für  $\gamma \geq \max(\{|\mu_i^*| : i \in \mathcal{E}\} \cup \{\lambda_i^* : i \in \mathcal{I}\})$  der Punkt  $x^*$  auch globales Optimum von  $f_\gamma(x) := f(x) + \gamma[\sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}]$ .

---

### Was hat $\gamma$ 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  $\gamma$  macht es nur noch unattraktiver.

**Nachteil der  $l_1$ -Straffunktion:**  $f_\gamma$  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

$$f_\gamma(x) = f(x) + \frac{\gamma}{2} \left[ \sum_{i \in \mathcal{E}} h_i(x)^2 + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}^2 \right].$$

- 
- $f_\gamma$  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  $x_\gamma^*$  verwendet werden.
  - Gerade für Gleichungsnebenbedingungen führt die quadratische Straffunktion zu bananenförmigen Teilproblemen  $\rightarrow$  ungünstige Konvergenzeigenschaften.
  - Für große  $\gamma$  wird die zweite Ableitung in unzulässige Richtungen sehr groß  $\rightarrow$  numerische Probleme.
  - Die quadratische Straffunktion ist i.A. nicht exakt, selbst für sehr großes  $\gamma$  bleibt  $x_\gamma^*$  unzulässig (s. Anfangsbeispiel).

# Inhalt

## Restringierte Optimierung: Verfahren

Strafverfahren

Augmentierte-Lagrange-Verfahren

Barriere-Verfahren

Quadratische Optimierung

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung



## 8.2 Augmentierte-Lagrange-Verfahren (nur Gleichungen)

Idee: addiere die quadratische Straffunktion zur Lagrange-Funktion,

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2,$$

dann bleibt die Funktion diffbar und ist für Multiplikator  $\mu^*$  exakt.

---

Erfüllt  $x^*$  die hinreichenden Opt.-bed. mit Multiplikator  $\mu^*$ , dann gilt

$$\nabla_x \mathcal{L}(x^*, \mu^*) = 0 \text{ und } d^T \nabla_{xx} \mathcal{L}(x^*, \mu^*) d > 0 \quad \forall d \in T_P(x^*) [\Leftrightarrow J_h(x^*) d = 0].$$

In allen Richtungen orthogonal dazu [ $d = J_h(x^*)^T u$  für  $u \in \mathbb{R}^{\mathcal{E}}$ ] sorgt

$$\text{wegen } \nabla_{xx} \mathcal{L}_\gamma(x, \mu) = \nabla_{xx} \mathcal{L}(x, \mu) + \gamma \sum_{i \in \mathcal{E}} \underbrace{h_i(x)}_{=0 \text{ in } x^*} \nabla^2 h_i(x) + \gamma J_h(x)^T J_h(x)$$

der Summand  $J_h(x^*)^T J_h(x^*)$  für  $\gamma$  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  $\gamma$  groß genug erfüllt  $x^*$  die hinr. freien Opt.-Bed. für  $\mathcal{L}_\gamma(\cdot, \mu^*)$ .

---

**Geometrisch:**  $\mathcal{L}(\cdot, \mu^*)$  ist wegen  $\nabla_x \mathcal{L}_\gamma(x^*, \mu^*) = 0$  in alle Richtungen um  $x^*$  flach und in zulässige Richtungen lokal konvex. Der Strafterm

$\frac{\gamma}{2} \|h(x)\|^2$  macht die Funktion in den unzulässigen Richtungen lokal konvex.

$$\mathcal{L}_\gamma(x, \mu) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \frac{\gamma}{2} \sum_{i \in \mathcal{E}} h_i(x)^2 = \mathcal{L}(x, \mu) + \frac{\gamma}{2} \|h(x)\|^2$$

## Satz (Augmentierte-Lagrange-Verfahren bei Gleichungen)

Erfüllt  $x^*$  die hinreichenden Optimalitätsbedingungen für ein restringiertes Problem ohne Ungleichungen mit Multiplikator  $\mu^*$ , 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  $\mu^*$  bekannt  $\rightarrow$  lokale freie Optimierung. Aber  $\mu^*$  ist unbekannt!

**Wie wählt man  $\mu$  im Verfahren?** (iterativ, Schritt  $k \rightarrow k+1$ )  
 $\mu^*$  und  $x^*$  erfüllen  $0 = \nabla_x \mathcal{L}(x^*, \mu^*) = \nabla f(x^*) + \sum_{i \in \mathcal{E}} \mu_i^* \nabla h_i(x^*)$ .

Sei  $x^{(k)}$  die berechnete Lösung zu  $\min_{x \in \mathbb{R}^n} \mathcal{L}_{\gamma_k}(x, \mu^{(k)})$ , dann ist

$$0 \approx \nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)}) = \nabla f(x^{(k)}) + \sum_{i \in \mathcal{E}} \underbrace{[\mu_i^{(k)} + \gamma_k h_i(x^{(k)})]}_{\approx \mu^*} \nabla h_i(x^{(k)})$$

$$\rightarrow \mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \quad (i \in \mathcal{E})$$

# Alg. Schema für Augm.-Lagr.-Verf. mit Gleichungen

0. Wähle Startpunkt  $\bar{x}^{(0)}$ ,  $\mu^{(0)}$ ,  $\gamma_0 > 0$ , Genauigkeit  $\varepsilon_0 > 0$ ,  $k := 0$
1. Bestimme Näherungslösung  $x^{(k)}$  zu  $\min_{x \in \mathbb{R}^n} \mathcal{L}_{\gamma_k}(x, \mu^{(k)})$  mit Startpunkt  $\bar{x}^{(k)}$ , so dass  $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\| \leq \varepsilon_k$ .
2. Ist  $\|h(x^{(k)})\|$  und  $\|\nabla_x \mathcal{L}_{\gamma_k}(x^{(k)}, \mu^{(k)})\|$  klein genug, STOP.  
(u.U. Neustart, falls kein Fortschritt Richtung Zulässigkeit, etc.)
3. Aktualisiere die Multiplikatoren:  $\mu_i^{(k+1)} := \mu_i^{(k)} + \gamma_k h_i(x^{(k)}) \quad (i \in \mathcal{E})$
4. Wähle einen neuen Strafparameter  $\gamma_{k+1} \in [\gamma_k, \infty)$
5. Wähle die nächste Genauigkeit  $\varepsilon_{k+1} \in (0, \varepsilon_k]$
6. Wähle den nächsten Startpunkt, meist  $\bar{x}^{(k+1)} := x^{(k)}$
7.  $k \leftarrow k + 1$ , GOTO 1.

## Ungleichungen:

werden z.B. in LANCELOT mit Schlupfvariablen in Gln umgewandelt:

$$g_i(x) + s_i = 0, \quad s_i \geq 0 \quad (i \in \mathcal{I})$$

In Schritt 1 wird das Problem dann mit Vorzeichenbedingungen gelöst,

$$\min_{x \in \mathbb{R}^n, s \in \mathbb{R}_+^{\mathcal{I}}} \mathcal{L}_{\gamma_k}(x, s, \tilde{\mu}^{(k)})$$

(z.B. über ein quadratisches Modell mit linearen Unglgen, s. später).

# Inhalt

## Restringierte Optimierung: Verfahren

Strafverfahren

Augmentierte-Lagrange-Verfahren

**Barriere-Verfahren**

Quadratische Optimierung

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung

## 8.3 Barriere-Verfahren

Idee: Verhindere das Verlassen des zulässigen Bereiches durch einen Barriere-Term in der Zielfunktion.

→ vor allem für Ungleichungen gut geeignet! Im Folgenden  $\mathcal{E} = \emptyset$ .

Bsp: Innere-Punkte-Verfahren für Lineare Optimierung über Kegeln

Das zulässige Innere  $\overset{\circ}{\mathcal{X}} := \{x \in \mathbb{R}^n : g_i(x) < 0, i \in \mathcal{I}\}$  sei nicht leer,  $\overset{\circ}{\mathcal{X}} \neq \emptyset$ .  
Barriere-Funktionen erfüllen folgende Eigenschaften:

- Für  $x \notin \overset{\circ}{\mathcal{X}}$  haben sie den Wert  $\infty$ .
- 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  $\infty$ .

Beispiele für  $y < 0$ :  $-\frac{1}{y}$ ,  $-\log(-y)$

Der Einfluss der Barriere-Funktion wird durch einen **Barriereparameter**  $\beta$  kontrolliert und man sucht ein lokales Minimum für

$$\min_{x \in \mathbb{R}^n} f_\beta(x) := f(x) - \beta \sum_{i \in \mathcal{I}} \log(-g_i(x))$$

ausgehend von einem Startpunkt im Inneren.

## Alg. Schema für Barriere-Verf. (nur Unglgen)

0. Wähle Startpunkt  $\bar{x}^{(0)} \in \overset{\circ}{\mathcal{X}}$ ,  $\beta_0 > 0$ , Genauigkeit  $\varepsilon_0 > 0$ ,  $k := 0$
1. Bestimme Näherungslösung  $x^{(k)}$  zu  $\min_{x \in \mathbb{R}^n} f_{\beta_k}(x)$  mit Startpunkt  $\bar{x}^{(k)}$ , so dass  $\|\nabla_x f_{\beta_k}(x^{(k)})\| \leq \varepsilon_k$ .
2. Sind die KKT-Bedingungen näherungsweise erfüllt, STOP.  
(teste ob Lösungen nach unendlich gehen, etc.)
3. Wähle einen neuen Barriereparameter  $\beta_{k+1} \in (0, \beta_k)$
4. Wähle die nächste Genauigkeit  $\varepsilon_{k+1} \in (0, \varepsilon_k]$
5. Wähle den nächsten Startpunkt, meist  $\bar{x}^{(k+1)} := x^{(k)}$
6.  $k \leftarrow k + 1$ , GOTO 1.

**KKT-Bedingungen?** Sei  $x(\beta)$  lokale OL zu  $\min_{x \in \mathbb{R}^n} f_{\beta}(x)$ , dann ist

$$0 = \nabla f_{\beta}(x(\beta)) = \nabla f(x(\beta)) + \underbrace{\sum_{i \in \mathcal{I}} \frac{\beta}{-g_i(x(\beta))}}_{=: \lambda_i(\beta) \geq 0} \nabla g_i(x(\beta))$$

also ist  $\nabla f(x(\beta)) + \sum_{i \in \mathcal{I}} \lambda_i(\beta) \nabla g_i(x(\beta)) = 0$

$$g_i(x(\beta)) \leq 0 \quad (i \in \mathcal{I})$$

$$\lambda_i(\beta) \geq 0 \quad (i \in \mathcal{I})$$

$$-\lambda_i(\beta) g_i(x(\beta)) = \beta \quad (i \in \mathcal{I}) \quad [\text{pert. Kompl.}]$$

Für  $\beta \rightarrow 0$  wird bei Konvergenz ein KKT-Punkt gefunden!

## Satz (Fiacco und McCormick 1968)

Sei  $\overset{\circ}{\mathcal{X}} \neq \emptyset$  und  $x^*$  ein lokales Minimum, in dem (LICQ) und die hinreichenden Optimalitätsbed. mit streng komplementären Multiplikatoren  $\lambda^*$  erfüllt sind. Dann gilt:

- (i) Für  $\bar{\beta}$  klein genug gibt es eine diffbare Funktion  $x(\beta): (0, \bar{\beta}) \rightarrow \mathbb{R}^n$  mit  $x(\beta) \xrightarrow{\beta \rightarrow 0} x^*$  und  $x(\beta)$  ist lokales Minimum von  $f_\beta$ . [zent. Pfad]
- (ii) Für  $x(\beta)$  aus (i) konvergieren die Schätzer der Lagrange-Multiplikatoren  $\lambda_i(\beta) = \frac{\beta}{-g_i(x(\beta))} \xrightarrow{\beta \rightarrow 0} \lambda_i^*$  für  $i \in \mathcal{I}$ .
- (iii) Für  $\beta$  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.

# Inhalt

## Restringierte Optimierung: Verfahren

Strafverfahren

Augmentierte-Lagrange-Verfahren

Barriere-Verfahren

## Quadratische Optimierung

Konvexe quadratische Optimierung

Gleichungsbeschränkte Quadratische Optimierung

Aktive-Mengen-Verfahren für Quadratische Optimierung

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

Anwendungsbeispiel: Optimalsteuerung



## 8.4 Quadratische Optimierung (Quadratic Programming)

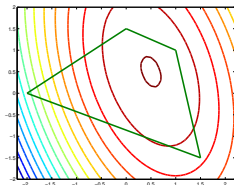
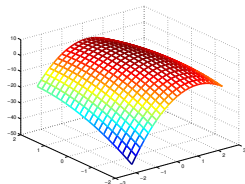
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.  
 $\Rightarrow$  Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

### Einige typische Fälle:

$f(\cdot)$  konkav ( $Q \prec 0$ ):  
 Alle OL werden in Ecken (bzw. auf minimalen Seiten) angenommen.



## 8.4 Quadratische Optimierung (Quadratic Programming)

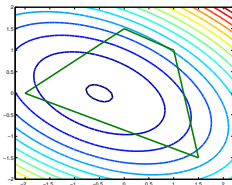
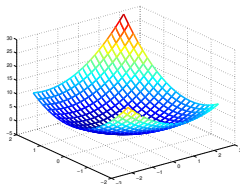
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.  
 $\Rightarrow$  Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

### Einige typische Fälle:

$f(\cdot)$  konvex ( $Q \succ 0$ ):  
 OL in der Mitte, oder



## 8.4 Quadratische Optimierung (Quadratic Programming)

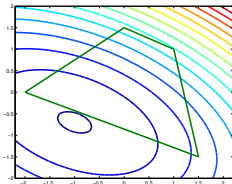
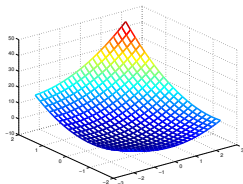
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.  
 $\Rightarrow$  Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

### Einige typische Fälle:

$f(\cdot)$  konvex ( $Q \succ 0$ ):  
 OL auf einer Seite, oder



## 8.4 Quadratische Optimierung (Quadratic Programming)

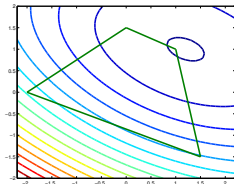
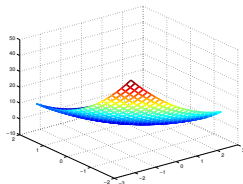
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.  
 $\Rightarrow$  Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

### Einige typische Fälle:

$f(\cdot)$  konvex ( $Q \succ 0$ ):  
 OL in einer Ecke



## 8.4 Quadratische Optimierung (Quadratic Programming)

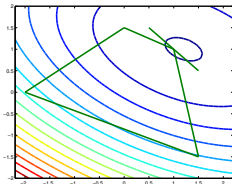
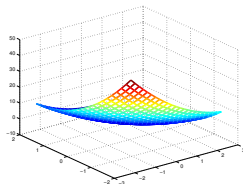
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.  
 $\Rightarrow$  Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

### Einige typische Fälle:

**degenerierte Lösung:**  
 aktive Nebenbedingungen  
 sind linear abhängig, oder



## 8.4 Quadratische Optimierung (Quadratic Programming)

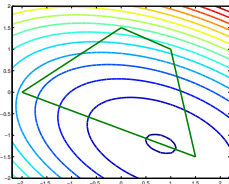
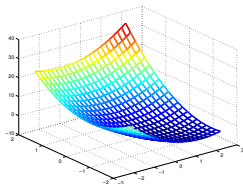
QPs sind die typischen Unterprobleme der restringierten Optimierung (wie das quadratische Modell in der freien Optimierung)

$$\begin{array}{ll} \min & f(x) := \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} & b_i - h_i^T x = 0 \quad i \in \mathcal{E} \\ & b_i - g_i^T x \leq 0 \quad i \in \mathcal{I} \\ & x \in \mathbb{R}^n \end{array} \quad \left. \begin{array}{l} [h_i \in \mathbb{R}^n] \\ [g_i \in \mathbb{R}^n] \end{array} \right\} \rightarrow Ax \geq b$$

Allgemeine quadratische Zielfunktion und lineare Nebenbedingungen.  
 $\Rightarrow$  Lagrange-Multiplikatoren existieren für jedes lokale Minimum.

### Einige typische Fälle:

**degenerierte Lösung:**  
 aktive Nebenbedingungen sind linear abhängig, oder die unrestringierte OL liegt auf einer Seite



## 8.4.1 Konvexe quadratische Optimierung

Für  $Q \succeq 0$  ist das QP ein konvexes Optimierungsproblem und es gibt ein konvexes Lagrange-Duales.

---

Herleitung für

$$\text{(PQP)} \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax \geq b$$

Lagrange-Funktion:  $\mathcal{L}(x, \lambda) = \frac{1}{2}x^T Qx + q^T x + \lambda^T (b - Ax)$ , ( $\lambda \geq 0$ )

primales Problem:  $\inf_x \sup_{\lambda \geq 0} \mathcal{L}(x, \lambda)$  [Für  $x^*$ :  $\exists \lambda^*$  mit  $(b - Ax^*)^T \lambda^* = 0$ ]

duales Problem:  $\sup_{\lambda \geq 0} \inf_x \mathcal{L}(x, \lambda) = \sup_{\lambda \geq 0} \left( b^T \lambda + \underbrace{\inf_x \left[ \frac{1}{2}x^T Qx + (q - A^T \lambda)^T x \right]}_{\nabla_x = 0 \Rightarrow Qx + q - A^T \lambda = 0, \text{ sonst } -\infty} \right)$

I.A. enthält das duale QP auch noch  $x$ -Variable(!):

$$\begin{aligned} \text{(DQP)} \quad & \max \quad \frac{1}{2}x^T Qx + q^T x + (b - Ax)^T \lambda \\ & \text{s.t.} \quad Qx + q - A^T \lambda = 0 \\ & \quad \quad x \in \mathbb{R}^n, \lambda \geq 0 \end{aligned} \quad \text{[erfüllt starke Dualität!]}$$

Für  $Q \succ 0$  ist  $x$  eindeutig bestimmt:  $x(\lambda) = Q^{-1}(A^T \lambda - q)$ ,

$$\rightarrow \max_{\lambda \geq 0} -\frac{1}{2}\lambda^T (AQ^{-1}A^T)\lambda + (b + AQ^{-1}q)^T \lambda - \frac{1}{2}q^T Q^{-1}q$$

Nur Vorzeichenbedingungen! Oft besonders effizient lösbar  $\rightarrow \lambda^*, x^* = x(\lambda^*)$

Für konvexe QPs sind, wie bei LPs, Innere-Punkte-Verfahren besonders effizient.

Bsp: Ansatz für primal-duales Pfadverfolgungsverfahren,  $A = [a_1, \dots, a_m]^T$ ,  $b \in \mathbb{R}^m$ .

$$\begin{aligned} \min \quad & \frac{1}{2}x^T Qx + q^T x \\ \text{s.t.} \quad & Ax \geq b \end{aligned} \quad \rightarrow \quad \min_{x \in \mathbb{R}^n} \quad \frac{1}{2}x^T Qx + q^T x - \gamma \sum \log(a_i^T x - b_i)$$

Stationarität des Barriere-Unterproblems für  $\gamma > 0$  ( $\nabla_x = 0$ ):

$$Qx + q - \gamma \sum a_i \frac{1}{a_i^T x - b_i} = 0$$

Führt man Schlupfvariablen  $s_i := a_i^T x - b_i \geq 0$

und Dualvariablen (Lagrange-Multiplikatoren)  $\lambda_i := \gamma \frac{1}{s_i}$

ein, erhält man ein primal-duales KKT-System:

$$\begin{aligned} Qx + q - A^T \lambda &= 0 && \text{„duale Zulässigkeit“} \\ Ax - s &= b && \text{„primale Zulässigkeit“} \\ s \circ \lambda &= \gamma \mathbf{1} && \text{„perturb. Kompl.“} \end{aligned}$$

Starte mit  $s, \lambda > 0$ , löse das System näherungsweise mit Newton, bewahre Positivität durch Line-Search, verkleinere  $\gamma$ , etc.

(wie in LP,  $\varepsilon$ -OL in gleicher polynomialer Anzahl an Iterationen).

Innere-Punkte-Verf. sind auch auf nicht-konvexe QPs recht gut anwendbar.



## 8.4.2 Gleichungsbeschränkte Quadratische Optimierung

$$(EQP) \quad \min \frac{1}{2}x^T Qx + q^T x \quad \text{s.t.} \quad Ax = b$$

$\mathcal{X} = \{x : Ax = b\}$  ist ein affiner Unterraum, auf diesem kann das Minimum jeder quadratischen Funktion explizit bestimmt werden.

### Berechnung der quadratischen Funktion über dem affinen Unterraum:

Gibt es ein  $\bar{x} \in \mathcal{X}$  ( $A\bar{x} = b$ ), ist jedes  $x \in \mathcal{X}$  darstellbar als

$$x = \bar{x} + d \quad \text{mit} \quad d \in \{d : Ad = 0\} =: \ker A \quad [\text{Kern/Nullraum von } A]$$

Der lineare Unterraum  $\ker A$  habe Dim.  $k$  und die Spalten von  $Z \in \mathbb{R}^{n \times k}$  als Basis, dann ist  $\ker A = \{Zu : u \in \mathbb{R}^k\}$ ,  $\mathcal{X} = \{\bar{x} + Zu : u \in \mathbb{R}^k\}$  und

$$(EQP) \Leftrightarrow \min_{u \in \mathbb{R}^k} \frac{1}{2}u^T Z^T QZ u + (Q\bar{x} + q)^T Z u + \frac{1}{2}\bar{x}^T Q\bar{x} + q^T \bar{x}.$$

$\Rightarrow$  endliche OL nur, falls  $Z^T QZ \succcurlyeq 0$  und  $Z^T QZ u + Z^T(Q\bar{x} + q) = 0$  lösbar, eindeutig für  $Z^T QZ \succ 0$ :  $u^* = -(Z^T QZ)^{-1} Z^T(Q\bar{x} + q)$  und  $x^* = \bar{x} + Zu$ .

### Satz (QP mit Gleichungsbedingungen)

Seien die Spalten von  $Z$  eine Basis von  $\ker A$ ,  $Z^T QZ \succ 0$ , und  $\bar{x} \in \mathcal{X}$ . Ist  $x^* = \bar{x} + d^*$  die Lösung des KKT Systems zu (EQP),

$$\begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix} \Leftrightarrow \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} -d^* \\ \mu^* \end{bmatrix} = \begin{bmatrix} q + Q\bar{x} \\ A\bar{x} - b \end{bmatrix},$$

dann ist  $x^*$  die eindeutige Optimallösung von (EQP).

## 8.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

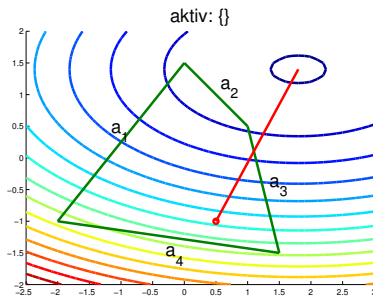
Bsp:  $\min \frac{1}{2}x^T Qx + q^T x$  s.t.  $Ax \geq b$  mit  $A = [a_1, a_2, a_3, a_4]^T$

$k = 0$ , Punkt  $x^{(0)}$ ,

aktive Menge:  $\emptyset$ ,

Richtung:  $d = -Q^{-1}q - x^{(0)}$ ,

aktiv wird ?



## 8.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

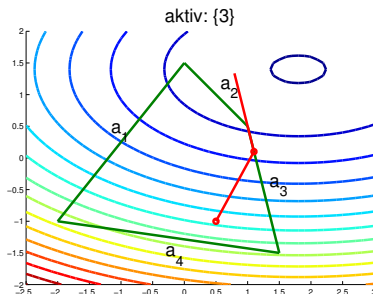
Bsp:  $\min \frac{1}{2}x^T Qx + q^T x$  s.t.  $Ax \geq b$  mit  $A = [a_1, a_2, a_3, a_4]^T$

$k = 1$ , Punkt  $x^{(1)}$ ,  
aktive Menge:  $\{3\}$ ,

Richtung:

$$\begin{bmatrix} Q & a_3 \\ a_3^T & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} q + Qx^{(1)} \\ a_3^T x^{(1)} - b_3 \end{bmatrix},$$

aktiv wird ?



## 8.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

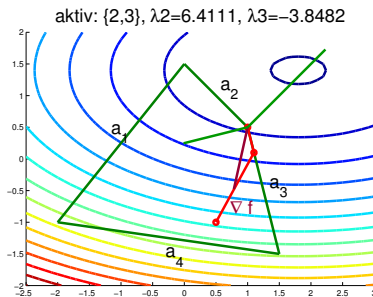
Bsp:  $\min \frac{1}{2}x^T Qx + q^T x$  s.t.  $Ax \geq b$  mit  $A = [a_1, a_2, a_3, a_4]^T$

$k = 2$ , Punkt  $x^{(2)}$ ,  
aktive Menge:  $\{2, 3\}$ ,  
Richtung:

$$\begin{bmatrix} Q & a_2 & a_3 \\ a_2^T & 0 & 0 \\ a_3^T & 0 & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} q + Qx^{(2)} \\ a_2^T x^{(2)} - b_2 \\ a_3^T x^{(2)} - b_3 \end{bmatrix}$$

→  $d = 0$ ,  $\lambda_3 < 0$  (!),

$x^{(3)} := x^{(2)}$ , aktiv wird ?



## 8.4.3 Aktive-Mengen-Verfahren für Quadratische Optimierung

Idee wie bei Simplex: Ausgehend von einem zulässigen Punkt,

- Fixiere eine Teilmenge der Nebenbedingungen als aktive Gleichungen (alle Gleichungen und einige Ungleichungen) → **active set**,
- bestimme das Minimum dieses gleichungsbeschränkten QPs,
- gehe in diese Richtung, bis die nächste Ungleichung aktiv wird.
- Ist keine Bewegung möglich, teste, ob eine Ungleichung zu unrecht fixiert wurde.

Bsp:  $\min \frac{1}{2}x^T Qx + q^T x$  s.t.  $Ax \geq b$  mit  $A = [a_1, a_2, a_3, a_4]^T$

$k = 3$ , Punkt  $x^{(3)}$ ,

aktive Menge:  $\{2\}$ ,

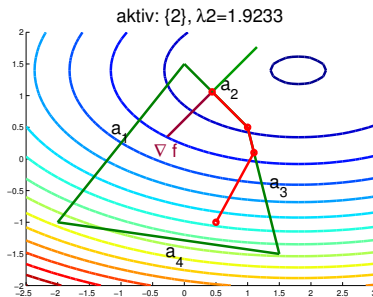
Richtung:

$$\begin{bmatrix} Q & a_2 \\ a_2^T & 0 \end{bmatrix} \begin{bmatrix} -d \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} q + Qx^{(2)} \\ a_2^T x^{(3)} - b_2 \end{bmatrix},$$

$$x^{(4)} := x^{(3)} + d,$$

$$\nabla_x f(x^{(4)}) + \lambda_2(-a_2) = 0, \lambda_2 \geq 0,$$

KKT erfüllt!



# Algorithmisches Active-Set-Schema für konvexes QP

$Q \succeq 0$  und o.B.d.A. nur Unglgen:  $\min x^T Q x + q^T x \quad \text{s.t.} \quad a_i^T x \geq b_i \quad (i \in \mathcal{I})$

---

0. Bestimme zulässigen Startpunkt  $x^{(0)} \in \{x : Ax \geq b\}$  (Simplex/IP), wähle aktive Menge  $\mathcal{A}_0 \subseteq \mathcal{A}(x)$ , setze  $k := 0$ .
  1. Bestimme  $d, \lambda$  für (EQP) mit Gleichungsbedingungen  $i \in \mathcal{A}_k$
  2. Falls  $d = 0$ :
    - a) Ist  $\lambda \geq 0$  (KKT erfüllt), STOP mit  $x^* = x^{(k)}$ .
    - b) Wähle  $\hat{i} \in \mathcal{A}_k$  mit  $\lambda_{\hat{i}} < 0$ , setze  $\mathcal{A}_{k+1} := \mathcal{A}_k \setminus \{\hat{i}\}$ ,  $x^{(k+1)} := x^{(k)}$ .
 Sonst ( $d \neq 0$ )
    - a) bestimme  $\alpha_k := \min \left\{ \frac{b_i - a_i^T x^{(k)}}{a_i^T d} : i \in \mathcal{I} \setminus \mathcal{A}_k, a_i^T d < 0 \right\}$ . [ $+\infty$  für  $\emptyset$ ]
    - b) Ist  $\alpha_k > 1$ , setze  $x^{(k+1)} := x^{(k)} + d$ ,  $\mathcal{A}_{k+1} := \mathcal{A}_k$ ,  
sonst  $x^{(k+1)} := x^{(k)} + \alpha_k d$  und  $\mathcal{A}_{k+1} := \mathcal{A}_k \cup \{\hat{i}\}$  mit  $\hat{i}$  erzeugt  $\alpha_k$ .
  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.

# Inhalt

## Restringierte Optimierung: Verfahren

Strafverfahren

Augmentierte-Lagrange-Verfahren

Barriere-Verfahren

Quadratische Optimierung

**SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)**

Lokale quadratische Konvergenz über Newton

Globalisierung mit Merit-Funktionen

Globalisierung mit Trust-Region-Ansätzen

Globalisierung mit einem Filter-Ansatz

Anwendungsbeispiel: Optimalsteuerung



## 8.5 SQP-Verfahren (Sequential Quadratic Programming)

Derzeit der erfolgreichste Ansatz für nichtlineare restringierte Optimierung.

**Idee:** Finde den nächsten Punkt durch Lösung eines QP-Modells, das aus den linearisierten Nebenbedingungen und einem quadratischen Modell der Lagrange-Funktion als Zielfunktion aufgebaut wird.

---

In 2 Schritten:

1. Lokale Konvergenz:

Das QP-Modell leitet sich aus dem Newton-Verfahren zur Bestimmung eines stationären Punktes der Lagrange-Funktion (= KKT-Bed.) her  
→ Newton führt zu lokal quadratischer Konvergenz von SQP-Verf.

2. Globalisierungs-Ansätze:

Fortschritt bezüglich Erfüllung der Nebenbedingungen und Verbesserung der Zielfunktion kann durch Merit-Funktion, Trust-Region-Ansatz oder Filter-Verfahren kontrolliert werden.

## 8.5.1 Herleitung des QPs aus KKT und Newton

Mit Gleichungsbedingungen:  $\min f(x)$  s.t.  $h(x) = 0$ . [ $h : \mathbb{R}^n \rightarrow \mathbb{R}^{\mathcal{E}}$ ]

Löse die KKT-Bed. für  $\mathcal{L}(x, \mu) = f(x) + \mu^T h(x)$  mit Newton:

$$\text{KKT: } F(x, \mu) := \begin{bmatrix} \nabla_x \mathcal{L} \\ \nabla_{\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla f(x) + J_h(x)^T \mu \\ h(x) \end{bmatrix} = 0$$

$$\text{Newton: } \begin{bmatrix} x^{(k+1)} \\ \mu^{(k+1)} \end{bmatrix} := \begin{bmatrix} x^{(k)} \\ \mu^{(k)} \end{bmatrix} + \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} \text{ mit } F(x^{(k)}, \mu^{(k)}) + J_F(x^{(k)}, \mu^{(k)}) \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = 0$$

$$J_F(x, \mu) = \begin{bmatrix} \nabla_{xx} \mathcal{L} & \nabla_{x\mu} \mathcal{L} \\ \nabla_{\mu x} \mathcal{L} & \nabla_{\mu\mu} \mathcal{L} \end{bmatrix} = \begin{bmatrix} \nabla_{xx} \mathcal{L}(x, \mu) & J_h(x)^T \\ J_h(x) & 0 \end{bmatrix} =: \begin{bmatrix} Q & A^T \\ A & 0 \end{bmatrix}$$

$$\text{Iteration } k: \text{ Löse } \begin{bmatrix} Q_k & A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x^N \\ d_{\mu}^N \end{bmatrix} = \begin{bmatrix} -\nabla f_k - A_k^T \mu^{(k)} \\ -h^{(k)} \end{bmatrix}$$

mit  $Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)})$ ,  $A_k := J_h(x^{(k)})$ ,  $h^{(k)} := [h_1(x^{(k)}), \dots, h_{|\mathcal{E}|}(x^{(k)})]^T$ .

Vergleiche dazu das folgende gleichungsbeschränkte QP:

$$\min \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x$$

$$\text{s.t. } A_k d_x = -h^{(k)} \quad [\text{Zeile } i: h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0]$$

Dafür bestimmt sich eine Optimallösung mit Multiplikatoren  $y$  durch

$$\begin{bmatrix} Q_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} d_x \\ y \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -h^{(k)} \end{bmatrix}, \text{ setze } y = -(\mu^{(k)} + d_{\mu}) \Leftrightarrow \text{Newton-System}$$

# Das QP des SQP-Verfahrens

$$\begin{aligned} \min & f(x) \\ (P) \text{ s.t. } & h_i(x) = 0 \quad i \in \mathcal{E}, \quad \mathcal{L}(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x), \\ & g_i(x) \leq 0 \quad i \in \mathcal{I} \end{aligned}$$

bestimme  $x^{(k+1)} := x^{(k)} + d_x^{(k)}$  für geg.  $\mu^{(k)}, \lambda^{(k)}, Q_k := \nabla_{xx} \mathcal{L}(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$  aus

$$\begin{aligned} \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x & \min & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\ (QP_k) \text{ s.t. } & h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \rightarrow \text{s.t. } A_k^h d_x = -h^{(k)} \quad [y_h \in \mathbb{R}^{\mathcal{E}}] \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} & & A_k^g d_x \leq -g^{(k)} \quad [y_g \in \mathbb{R}^{\mathcal{I}}] \end{aligned}$$

wobei  $A_k^h = J_h(x^{(k)})$ ,  $A_k^g = J_g(x^{(k)})$ , setze  $\mu^{(k+1)} := -y_h^{(k)}$ ,  $\lambda^{(k+1)} := -y_g^{(k)}$ .

## Bemerkungen:

- Ist  $(x^{(k)}, \mu^{(k)}, \lambda^{(k)})$  nahe genug an  $(x^*, \mu^*, \lambda^*)$ , das die hinr. Opt.-Bed. und strenge Kompl. erfüllt, sind die aktiven Mengen gleich und Newton garantiert **lokal quadratische Konvergenz**.
- $Q_k$  enthält Krümmung von  $f$  **und** den aktiven Nebenbedingungen,
 
$$Q_k = \nabla^2 f(x^{(k)}) + \sum_{i \in \mathcal{E}} \mu_i^{(k)} \nabla^2 h_i(x^{(k)}) + \sum_{i \in \mathcal{I}} \lambda_i^{(k)} \nabla^2 g_i(x^{(k)}),$$
 → kurze Schritte in Richtungen mit starker Änderung wichtiger Funktionen.
- Die Hessematrizen in  $Q_k$  sind durch BFGS, etc. approximierbar.

## 8.5.2 Globalisierung mit Merit-Funktionen

Verwende die Lösung  $d_x^{(k)}$  von  $(QP_k)$  als Suchrichtung und bestimme  $x^{(k+1)} = x^{(k)} + \alpha_k d_x^{(k)}$  durch Line-Search bezüglich einer Merit-Funktion, die Verbesserung in Zielfunktion und Zulässigkeit gemeinsam bewertet.

---

**$I_1$ -Merit-Funktion:** Für geg. Strafparameter  $\gamma > 0$  verwende

$$f_\gamma(x) := f(x) + \gamma \left[ \sum_{i \in \mathcal{E}} |h_i(x)| + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\} \right]$$

- $f_\gamma$  ist nicht diffbar, aber Richtungsabl. existiert und genügt für Line-Search.
  - $d_x^{(k)}$  ist Abstiegsrichtung für  $f_\gamma$  für  $\gamma > \max(\{|\mu_i| : i \in \mathcal{E}\} \cup \{\lambda_i : i \in \mathcal{I}\})$ .
- 

**Augmentierte-Lagrange-Merit-Funktion von Fletcher:** Für geg.  $\gamma > 0$ ,

$$\mathcal{L}_\gamma(x, \mu, \lambda) := f(x) + \sum_{i \in \mathcal{E}} \mu_i h_i(x) + \sum_{i \in \mathcal{I}} \lambda_i g_i(x) + \gamma \left[ \sum_{i \in \mathcal{E}} h_i(x)^2 + \sum_{i \in \mathcal{I}} \max\{0, g_i(x)\}^2 \right]$$

- differenzierbar und für  $\gamma$  groß genug ist  $d_x^{(k)}$  Abstiegsrichtung
  - Die Wahl von  $\gamma$  hat starken Einfluss auf das Konvergenzverhalten ...
- 

Generelles Problem: **Maratos-Effekt**

Line-Search bzgl. Merit-Funktion verbietet oft Schrittweite 1 im Gebiet der lokalen quadratischen Konvergenz  $\rightarrow$  schlechte lokale Konvergenz.

Es gibt heuristische Gegenstrategien ...

## 8.5.3 Globalisierung mit Trust-Region-Ansätzen

### Klassischer Ansatz:

- Fordere zusätzlich  $\|d_x\| \leq \Delta$  oder, damit über lineare Nebenbed. darstellbar,  $\|d_x\|_\infty := \max\{|[d_x]_i| : i = 1, \dots, n\} \leq \Delta$ .
- Schwierigkeit: Ist  $x^{(k)}$  unzulässig, kann durch  $\|d_x\|_\infty \leq \Delta$  auch (QP<sub>k</sub>) unzulässig werden. Lösungsvorschläge sind z.B. erst ein Zulässigkeitsproblem zu lösen und dann  $\Delta$  zu wählen, etc.

### Kombinierter Strafansatz: (im open-source-Paket IPOPT implementiert)

Idee:  $x$  ist zulässig, wenn  $\max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\}) = 0$ .

→ Löse als Trust-Region-Unterproblem für Strafparameter  $\gamma > 0$

$$\begin{aligned} \min \quad & \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x + \gamma s \\ \text{s.t.} \quad & -s \leq h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x \leq s \quad i \in \mathcal{E} \\ & g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq s \quad i \in \mathcal{I} \\ & -\Delta \leq [d_x]_i \leq \Delta \quad (i = 1, \dots, n), s \geq 0 \end{aligned}$$

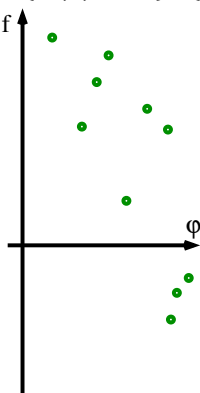
Der Trust-Region Algorithmus vergleicht Fortschritt der Funktion  $f_\gamma(d) := f(x^{(k)} + d) + \gamma \max(\{|h_i(x^{(k)} + d)| : i \in \mathcal{E}\} \cup \{g_i(x^{(k)} + d) : i \in \mathcal{I}\} \cup \{0\})$  zum Fortschritt im Modell und passt  $\Delta$  entsprechend an. Strafparameter  $\gamma$  wird nur bei Verletzung der Nebenbed. vergrößert, wenn der Fortschritt Richtung Zulässigkeit im Verhältnis zur Schrittlänge zu klein ist.

## 8.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$  ... Verletzungsmaß, z.B.  $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$ .

Betrachte  $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$  als Punkt in  $\mathbb{R}^2$ .



## 8.5.4 Globalisierung mit einem Filter-Ansatz

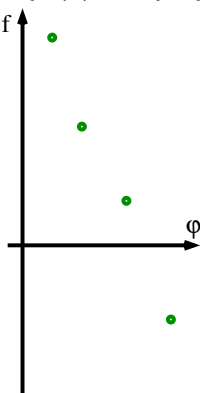
Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$  ... Verletzungsmaß, z.B.  $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$ .

Betrachte  $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$  als Punkt in  $\mathbb{R}^2$ .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$



## 8.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$  ... Verletzungsmaß, z.B.  $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$ .

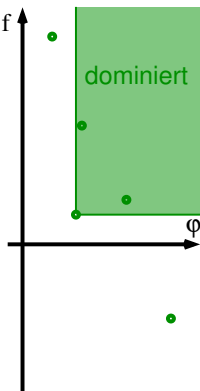
Betrachte  $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$  als Punkt in  $\mathbb{R}^2$ .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein  $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$  **dominiert**  $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$ , falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$





## 8.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$  ... Verletzungsmaß, z.B.  $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$ .

Betrachte  $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$  als Punkt in  $\mathbb{R}^2$ .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein  $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$  **dominiert**  $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$ , falls

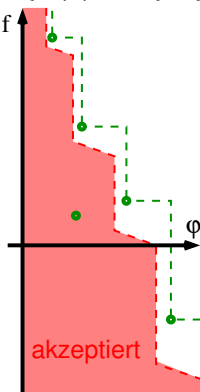
$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein  $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$  wird von  $\mathcal{F}_k$  für gegebene

$0 < \gamma_1 < \gamma_2 < 1$  **akzeptiert**, falls

$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]



## 8.5.4 Globalisierung mit einem Filter-Ansatz

Idee: Ersetze die Merit-Funktion durch ein Akzeptanzschema, das neue Punkte akzeptiert, wenn sie Verbesserung in Zielfunktion oder Verletzung bringen (also bikriteriell besser sind).

$\varphi(x)$  ... Verletzungsmaß, z.B.  $\varphi(x) := \max(\{|h_i(x)| : i \in \mathcal{E}\} \cup \{g_i(x) : i \in \mathcal{I}\} \cup \{0\})$ .

Betrachte  $(f_k, \varphi_k) := (f(x^{(k)}), \varphi(x^{(k)}))$  als Punkt in  $\mathbb{R}^2$ .

Ein **Filter** enthält derzeit akzeptierte Punkte,

$$\mathcal{F}_k := \{(f_j, \varphi_j) : j \in J_k \subseteq \{1, \dots, k\}\}.$$

Ein  $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$  **dominiert**  $(\bar{f}, \bar{\varphi}) \in \mathbb{R}^2$ , falls

$$\hat{f} \leq \bar{f} \text{ und } \hat{\varphi} \leq \bar{\varphi}.$$

Ein  $(\hat{f}, \hat{\varphi}) \in \mathbb{R}^2$  wird von  $\mathcal{F}_k$  für gegebene

$0 < \gamma_1 < \gamma_2 < 1$  **akzeptiert**, falls

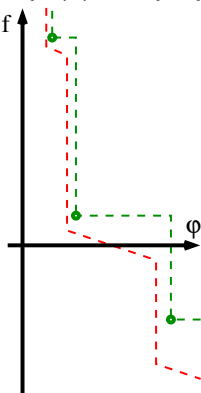
$$\forall j \in J_k : (\hat{\varphi} \leq \gamma_2 \varphi_j) \text{ oder } (\hat{f} + \gamma_1 \hat{\varphi} \leq f_j)$$

[Hinreichende Verbesserung in Verletzung oder Wert]

Wird  $(f_{k+1}, \varphi_{k+1})$  von  $\mathcal{F}_k$  akzeptiert, setze

$$\mathcal{F}_{k+1} := \{(f_{k+1}, \varphi_{k+1})\} \cup \{(\bar{f}, \bar{\varphi}) \in \mathcal{F}_k \text{ nicht von } (f_{k+1}, \varphi_{k+1}) \text{ dominiert}\}$$

[Der Filter sichert monotone Verbesserung und sammelt Erfahrung über gute Werte.]



## Trust-Region-Variante eines Filter-Ansatzes

Schritt  $d_x$  löst approximativ ein polyedrisches Trust-Region-Unterproblem:

$$\begin{aligned}
 (QP_k^\Delta) \quad & \min \quad \frac{1}{2} d_x^T Q_k d_x + \nabla f_k^T d_x \\
 & \text{s.t.} \quad h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d_x = 0 \quad i \in \mathcal{E} \\
 & \quad \quad g_i(x^{(k)}) + \nabla g_i(x^{(k)})^T d_x \leq 0 \quad i \in \mathcal{I} \\
 & \quad \quad -\Delta \leq [d_x]_i \leq \Delta \quad i \in \{1, \dots, n\}
 \end{aligned}$$

Falls  $(QP_k^\Delta)$  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  $\Delta$ , 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  $\varphi$  im Filter
- $\sigma \in (0, 1)$  ... Modellqualität für Reduktion von  $\Delta$

# 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  $(f_k, \varphi_k)$  wird von  $\mathcal{F}_{k-1}$  akzeptiert und  $(QP_k^\Delta)$  ist zulässig.
2. SQP-Schritt: Löse  $(QP_k^\Delta) \rightarrow d_x$ .  
Falls  $(QP_k^\Delta)$  unzul., setze  $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$ ,  $k \leftarrow k + 1$ , GOTO 1.
3. Ist  $d_x = 0$ , STOP,  $x^{(k)}$  ist stationärer Punkt.
4. Wird  $(f(x^{(k)} + d_x), \varphi(x^{(k)} + d_x))$  nicht vom erweiterten Filter  $\mathcal{F}_{k-1} \cup \{(f_k, \varphi_k)\}$  akzeptiert, setze  $\Delta \leftarrow \Delta/2$ , GOTO 2.
5. Setze  $\Delta q := -\frac{1}{2}d_x^T Q_k d_x - \nabla f_k^T d_x$  und  $\Delta f := f_k - f(x^{(k)} + d_x)$ .  
Ist  $\Delta q > 0$  und  $\Delta f < \sigma \Delta q$ , setze  $\Delta \leftarrow \Delta/2$  und GOTO 2.  
[Der Fortschritt in  $f$  ist im Vergleich zum Modell zu gering.]
6. Ist  $\Delta q < 0$ , [ $\Rightarrow x^{(k)}$  ist unzulässig, sonst wäre  $d_x = 0$  besser]  
setze  $\mathcal{F}_k := \{(f_k, \varphi_k)\} \cup \mathcal{F}_{k-1}$ , sonst  $\mathcal{F}_k := \mathcal{F}_{k-1}$
7. Setze  $x^{(k+1)} := x^{(k)} + d_x$ ,  $k \leftarrow k + 1$ , wähle  $\Delta \geq \underline{\Delta}$ , GOTO 2.

---

Unter geeigneten Voraussetzungen kann Konvergenz gegen einen stationären Punkt gezeigt werden.

# Inhalt

## Restringierte Optimierung: Verfahren

Strafverfahren

Augmentierte-Lagrange-Verfahren

Barriere-Verfahren

Quadratische Optimierung

SQP-Verfahren (Sequentielle quadratische Opt.-Verfahren)

## Anwendungsbeispiel: Optimalsteuerung

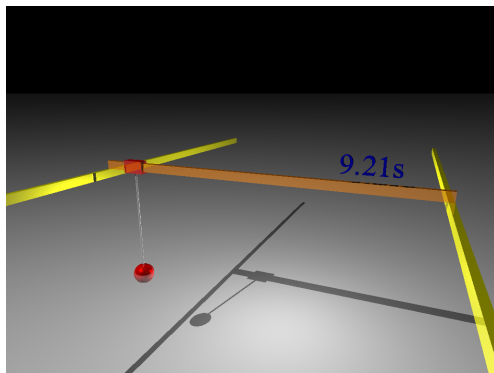
Mathematisches Modell

Zeitoptimale Steuerung

Diskretisierung

## 8.6 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?



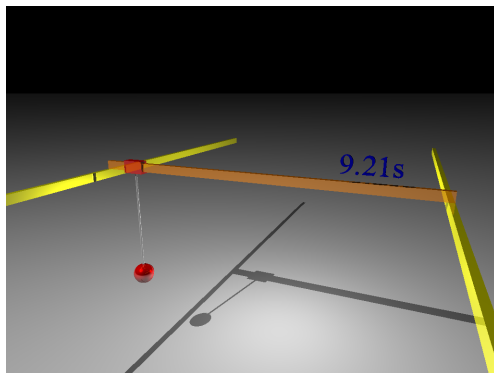
---

Die erste Idee – bis zur Hälfte voll beschleunigen, dann voll bremsen – ist nicht richtig.

Illustration

## 8.6 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)

## 8.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$ : horizontale Position der Laufkatze

$d(t)$ : horizontale Position der Last

$M$ : die Masse der Laufkatze

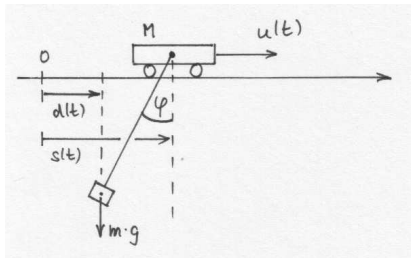
$m$ : die Masse der Last

$u(t)$ : **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$ : Winkel der Last zur Vertikalen

$g$ : die Erdbeschleunigung

$L$ : die Länge des Seiles



Der **Zustand**  $y(t) \in \mathbb{R}^4$  wird durch Position  $s(t)$  und Geschwindigkeit  $\dot{s}(t) := \frac{\partial s}{\partial t}$  der Laufkatze sowie Relativposition  $z(t) := s(t) - d(t)$  und Relativgeschwindigkeit  $\dot{z}(t)$  der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \quad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel  $\varphi$  ist

$$\ddot{s}(t) = -\frac{m}{M} \frac{g}{L} z + \frac{u(t)}{M}$$

$$\ddot{z}(t) = -\frac{(m+M)}{M} \frac{g}{L} z + \frac{u(t)}{M}$$

$$\Leftrightarrow \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$



## 8.6.1 (eindimensionale) Laufkatze: Mathematisches Modell

$s(t)$ : horizontale Position der Laufkatze

$d(t)$ : horizontale Position der Last

$M$ : die Masse der Laufkatze

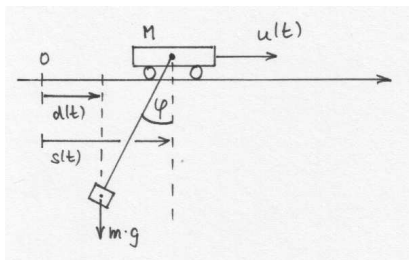
$m$ : die Masse der Last

$u(t)$ : **Steuerung**, hier die auf die Laufkatze einwirkende Kraft (E-Motor)

$\varphi(t)$ : Winkel der Last zur Vertikalen

$g$ : die Erdbeschleunigung

$L$ : die Länge des Seiles



Der **Zustand**  $y(t) \in \mathbb{R}^4$  wird durch Position  $s(t)$  und Geschwindigkeit  $\dot{s}(t) := \frac{\partial s}{\partial t}$  der Laufkatze sowie Relativposition  $z(t) := s(t) - d(t)$  und Relativgeschwindigkeit  $\dot{z}(t)$  der Last zur Laufkatze beschrieben,

$$y = (s, \dot{s}, z, \dot{z})^T. \quad [t \text{ wird meist weggelassen}]$$

Bewegungsgleichungen beschreiben die Beschleunigung und damit die Veränderung des Zustands über die Zeit. Für kleine Winkel  $\varphi$  ist

lineare gewöhnliche  
Differentialgleichung

$$\dot{y} = Ay + Bu \Leftrightarrow \dot{y} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{ML} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & -\frac{(m+M)g}{ML} & 0 \end{bmatrix} y + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ \frac{1}{M} \end{bmatrix} u$$

## Optimalstearaufgaben für gew. Diffgl.

Ein System gewöhnlicher Differentialgleichungen

$$\dot{y}(t) = f(t, y(t), u(t))$$

beschreibe die Entwicklung des Zustands  $y(t)$  in Abhängigkeit von Zeit  $t$  und Steuerung  $u(t)$ .

Für  $t \in [0, T]$  ist eine Steuerfunktion  $u(t)$  gesucht, die den Zustand  $y(t)$  von einem Anfangszustand  $y(0) = y_0$  in einen Endzustand  $y(T) = y_T$  überführt, dabei Zulässigkeitsbedingungen  $h(t, y(t), u(t)) \leq 0$  erfüllt und zugleich eine Zielfunktion/Gütekriterium/Performance-Maß optimiert.

$$\begin{array}{ll} \min_u & g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt & \text{Zielfunktion(al)} \\ \text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\ & y(0) = y_0 & \text{Anfangsbedingungen} \\ & y(T) = y_T & \text{Endbedingungen} \\ & h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.} \end{array}$$

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.

## 8.6.2 Zeitoptimale Steuerung der Laufkatze

Die Laufkatze soll mit Last in kürzest möglicher Zeit  $T$  vom Stillstand am Punkt  $P$  zum Stillstand an einem Punkt  $Q$  gebracht werden.

→ Zielfunktion:  $\min \int_0^T 1 dt [= T]$

→  $y(0) = \begin{pmatrix} P \\ 0 \\ 0 \\ 0 \end{pmatrix}$  Position der Laufkatze  
 Relativposition der Last zur Laufkatze  
 Geschwindigkeit der Laufkatze  
 Relativgeschwindigkeit der Last zur Laufkatze

→  $y(T) = (Q, 0, 0, 0)^T$

**Steuerbeschränkungen:**  $-1 \leq u(t) \leq 1$  [keine unendliche Beschleunigung]

Es könnte auch **Zustandsbeschränkungen** für  $y(t)$  (z.B. Hindernisse) geben.

Hauptproblem noch:

$T$  ist eine Variable und gibt zugleich den Definitionsbereich von  $u$  an!

# Transformation von freier Endzeit auf feste Endzeit

$$\begin{array}{ll}
 \min_{u, T} & g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt & \text{Zielfunktion(al)} \\
 \text{s.t.} & \dot{y}(t) = f(t, y(t), u(t)) & \text{Differentialgleichung} \\
 & y(0) = y_0 & \text{Anfangsbedingungen} \\
 & y(T) = y_T & \text{Endbedingungen} \\
 & h(t, y(t), u(t)) \leq 0 & \text{Beschränkungen.}
 \end{array}$$


---

Führe eine neue Zeitvariable  $\tau \in [0, 1]$  ein:  $t = \tau T, \tau \in [0, 1]$

Definiere neue Zustände/Steuerung für  $\tau$ :  $\tilde{y}(\tau) := y(\tau T), \tilde{u}(\tau) := u(\tau T)$

Die Differentialgleichung bezüglich  $\tau$  ergibt sich aus der Kettenregel,

$$\frac{d\tilde{y}}{d\tau}(\tau) = T \frac{dy}{dt}(\tau T) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)).$$

Die neue Zielfunktion lässt sich über die Substitutionsregel berechnen,

$$g(y(T)) + \int_0^T f_0(t, y(t), u(t)) dt \rightarrow g(\tilde{y}(1)) + T \int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) d\tau.$$


---

$$\begin{array}{ll}
 \min_{\tilde{u}, T} & g(\tilde{y}(1)) + T \int_0^1 f_0(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) d\tau \\
 \text{s.t.} & \dot{\tilde{y}}(\tau) = Tf(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) \\
 & \tilde{y}(0) = y_0 \\
 & \tilde{y}(1) = y_T \\
 & h(\tau T, \tilde{y}(\tau), \tilde{u}(\tau)) \leq 0.
 \end{array}$$

[Def-Bereich von  $\tilde{u}$  unabhängig von  $T$ ,  $T$  ist nun „normale“ Variable!]

## 8.6.3 Das Laufkatzenproblem nach Transformation

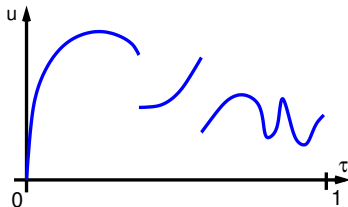
Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne  $\sim$  aber mit neuer Zeit  $\tau \in [0, 1]$ )

$$\begin{aligned}
 & \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} && T \\
 & \text{s.t.} && \dot{y}(\tau) = T A y(\tau) + T B u(\tau) && \text{[nichtlineare Nebenbed.!]} \\
 & && y(0) = (P, 0, 0, 0)^T \\
 & && y(1) = (Q, 0, 0, 0)^T \\
 & && -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{aligned}$$

Variablen:  $T, u(\tau)$  für  $\tau \in [0, 1]$   $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.



## 8.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne  $\sim$  aber mit neuer Zeit  $\tau \in [0, 1]$ )

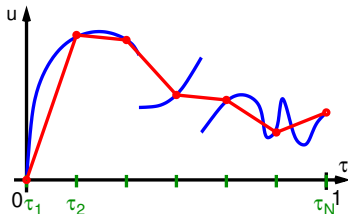
$$\begin{aligned} \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} \quad & T \\ \text{s.t.} \quad & \dot{y}(\tau) = T Ay(\tau) + T Bu(\tau) \quad \text{[nichtlineare Nebenbed.!]} \\ & y(0) = (P, 0, 0, 0)^T \\ & y(1) = (Q, 0, 0, 0)^T \\ & -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1]. \end{aligned}$$

Variablen:  $T, u(\tau)$  für  $\tau \in [0, 1]$   $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.

$\rightarrow$  Diskretisierung und, z.B., lineare Interpolation  $\rightarrow$  Diskretisierungsfehler



## 8.6.3 Das Laufkatzenproblem nach Transformation

Eine zeitoptimale Steuerung findet man durch Lösen des transformierten Optimierungsproblems (zur besseren Lesbarkeit ohne  $\sim$  aber mit neuer Zeit  $\tau \in [0, 1]$ )

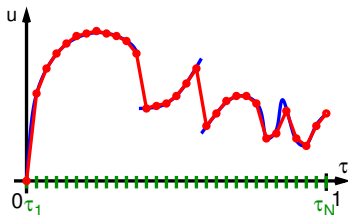
$$\begin{aligned}
 & \min_{u: [0,1] \rightarrow \mathbb{R}, T \in \mathbb{R}} && T \\
 & \text{s.t.} && \dot{y}(\tau) = T A y(\tau) + T B u(\tau) && \text{[nichtlineare Nebenbed.!]} \\
 & && y(0) = (P, 0, 0, 0)^T \\
 & && y(1) = (Q, 0, 0, 0)^T \\
 & && -1 \leq u(\tau) \leq 1 \quad \text{für alle } \tau \in [0, 1].
 \end{aligned}$$

Variablen:  $T, u(\tau)$  für  $\tau \in [0, 1]$   $\rightarrow \infty$ -dimensional

Wie optimiert man über Funktionen?

Wenn die Funktion nicht zu oft springen darf (z.B. stetig oder stückweise stetig), reichen die Funktionswerte an endlich vielen Stellen aus, um sie gut zu approximieren.

$\rightarrow$  Diskretisierung und, z.B., lineare Interpolation  $\rightarrow$  Diskretisierungsfehler



# Laufkatze: Diskretisierung des Optimierungsproblems

Ersetze die Funktionsvariable  $u : [0, 1] \rightarrow \mathbb{R}$  über die Wahl einer Zeitdiskretisierung  $0 = \tau_1 < \tau_2 < \dots < \tau_N = 1$ ,  $N \in \mathbb{N}$ , durch einen Variablenvektor  $u \in \mathbb{R}^N$  mit Interpretation  $u_i = u(\tau_i)$  und (bei linearer Interpolation)  $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$  für  $\tau \in (\tau_i, \tau_{i+1})$ .

min  $T$

$$\text{s.t. } \dot{y}(\tau) = T A y(\tau) + T B u(\tau) \quad \tau \in [0, 1]$$

$$y(0) = (P, 0, 0, 0)^T$$

$$y(1) = (Q, 0, 0, 0)^T$$

$$-1 \leq u_i \leq 1 \quad (i = 1, \dots, N), T \in \mathbb{R}_+.$$

Was geschieht mit der Differentialgleichung?

Berechne

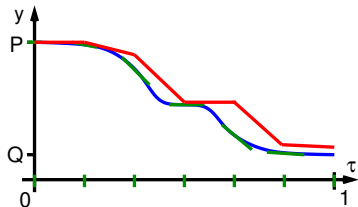
$$y(\tau) = y(0) + \int_0^\tau T A y(t) + T B u(t) dt$$

über ein numerisches Integrationsverfahren, das das Integral wieder durch Diskretisierung über Summen approximiert (z.B. Euler- oder Runge-Kutta-Verfahren).

→ Funktionsorakel  $y(\tau) = R(u, \tau)$ , das  $y(0) = (P, 0, 0, 0)^T$  erfüllt.

→ Es bleibt nur eine Gls-Nebenbedingung:  $R(u, 1) - (Q, 0, 0, 0)^T = 0$

Funktionsauswertung mit Gradienten über automatisches Differenzieren!





# Laufkatze: Diskretisierte Optimierungsaufgabe

$$\begin{aligned} \min \quad & T \\ \text{s.t.} \quad & R(u, 1) - (Q, 0, 0, 0)^T = 0 \\ & -1 \leq u_i \leq 1 \quad (i = 1, \dots, N), \quad T \in \mathbb{R}_+. \end{aligned}$$

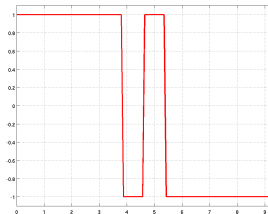
Dieses kann z.B. mit einem SQP-Verfahren gelöst werden, wobei die Gradienten zum Runge-Kutta-Verfahren  $R(\cdot)$  über automatisches Differenzieren und die Hessematrix über BFGS approximiert wird.

→ Approximation der zeitoptimalen Steuerung  
 $u(\tau) = u_i + \frac{\tau - \tau_i}{\tau_{i+1} - \tau_i} (u_{i+1} - u_i)$  für  $\tau \in (\tau_i, \tau_{i+1})$

Illustration

Ganz analog sind zeitoptimale 2D-Steuerungen berechenbar,

2D-Illustration



Die Qualität der Lösung und der notwendige Aufwand zur Bestimmung der Lösung sind stark von der Wahl der Diskretisierung abhängig!