

# Verfahren zur Lösung von Optimierungsproblemen mit nichtlinearen Restriktionen

Christoph Brune

Lehrstuhl Professor H. Maurer  
Institut für Numerische und Angewandte Mathematik  
Universität Münster

24.01.2006

# Inhaltsverzeichnis

- 1 Motivation
- 2 Problemstellung
- 3 Lagrange-Newton-SQP-Verfahren
- 4 Sequentielle quadratische Programmierung
  - Berechnung der Suchrichtung
  - Berechnung der Schrittweite
  - Grundversion des SQP-Verfahrens
  - Globale Konvergenz im konvexen Fall
- 5 Anwendungen
  - Optimale Steuerung einer chemischen Reaktion
  - Stabilisierung eines taumelnden Flugkörpers

# Inhaltsverzeichnis

- 1 Motivation
- 2 Problemstellung
- 3 Lagrange-Newton-SQP-Verfahren
- 4 Sequentielle quadratische Programmierung
  - Berechnung der Suchrichtung
  - Berechnung der Schrittweite
  - Grundversion des SQP-Verfahrens
  - Globale Konvergenz im konvexen Fall
- 5 Anwendungen
  - Optimale Steuerung einer chemischen Reaktion
  - Stabilisierung eines taumelnden Flugkörpers

# Motivation

- Steuerung eines Hochregallagers

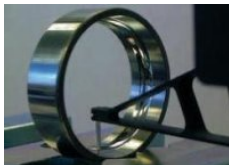


## Motivation

- Steuerung eines Hochregallagers



- Oberflächenabtastung industrieller Bauteile



# Motivation

- Optimierung des Laserstrahlschweißens zur Vermeidung von Heißbrissen



## Motivation

- Optimierung des Laserstrahlschweißens zur Vermeidung von Heißrissen



- Steuerung chemischer Reaktionen



# Inhaltsverzeichnis

- 1 Motivation
- 2 Problemstellung
- 3 Lagrange-Newton-SQP-Verfahren
- 4 Sequentielle quadratische Programmierung
  - Berechnung der Suchrichtung
  - Berechnung der Schrittweite
  - Grundversion des SQP-Verfahrens
  - Globale Konvergenz im konvexen Fall
- 5 Anwendungen
  - Optimale Steuerung einer chemischen Reaktion
  - Stabilisierung eines taumelnden Flugkörpers

# Problemstellung

- Verfahren zur Lösung von Optimierungsproblemen mit:

# Problemstellung

- Verfahren zur Lösung von Optimierungsproblemen mit:
  - nichtlinearer Zielfunktion  $f(x)$
  - nichtlinearen Restriktionen  $h(x)$  und  $g(x)$

# Problemstellung

- Verfahren zur Lösung von Optimierungsproblemen mit:
  - nichtlinearer Zielfunktion  $f(x)$
  - nichtlinearen Restriktionen  $h(x)$  und  $g(x)$

(PNU)

$$\min_{x \in D} f(x)$$

$$\text{unter } h(x) = 0, \quad g(x) \leq 0$$

# Inhaltsverzeichnis

- 1 Motivation
- 2 Problemstellung
- 3 Lagrange-Newton-SQP-Verfahren**
- 4 Sequentielle quadratische Programmierung
  - Berechnung der Suchrichtung
  - Berechnung der Schrittweite
  - Grundversion des SQP-Verfahrens
  - Globale Konvergenz im konvexen Fall
- 5 Anwendungen
  - Optimale Steuerung einer chemischen Reaktion
  - Stabilisierung eines taumelnden Flugkörpers

# Lagrange-Newton-SQP-Verfahren

- Idee:
  - Anwendung des Newton-Verfahrens auf Funktion mit Informationen aus notwendigen Optimalitätsbedingungen

# Lagrange-Newton-SQP-Verfahren

- Idee:
  - Anwendung des Newton-Verfahrens auf Funktion mit Informationen aus notwendigen Optimalitätsbedingungen
  - Lösung eines quadratischen Problems in jedem Iterationsschritt

# Lagrange-Newton-SQP-Verfahren

(PNU)

$$\min_{x \in D} f(x)$$

$$\text{unter } h(x) = 0, \quad g(x) \leq 0$$

$$\text{wobei } f : D \rightarrow \mathbb{R}, h : D \rightarrow \mathbb{R}^m, g : D \rightarrow \mathbb{R}^p$$

## Lagrange-Newton-SQP-Verfahren

(PNU)

$$\min_{x \in D} f(x)$$

$$\text{unter } h(x) = 0, \quad g(x) \leq 0$$

$$\text{wobei } f : D \rightarrow \mathbb{R}, h : D \rightarrow \mathbb{R}^m, g : D \rightarrow \mathbb{R}^p$$

- $F = \{x \in D \mid h(x) = 0, \quad g(x) \leq 0\}$  zulässige Menge

## Lagrange-Newton-SQP-Verfahren

(PNU)

$$\min_{x \in D} f(x)$$

$$\text{unter } h(x) = 0, \quad g(x) \leq 0$$

$$\text{wobei } f : D \rightarrow \mathbb{R}, h : D \rightarrow \mathbb{R}^m, g : D \rightarrow \mathbb{R}^p$$

- $F = \{x \in D \mid h(x) = 0, \quad g(x) \leq 0\}$  zulässige Menge
- $\tilde{x}$  lokale Lösung von (PNU)

## Lagrange-Newton-SQP-Verfahren

(PNU)

$$\min_{x \in D} f(x)$$

$$\text{unter } h(x) = 0, \quad g(x) \leq 0$$

$$\text{wobei } f : D \rightarrow \mathbb{R}, h : D \rightarrow \mathbb{R}^m, g : D \rightarrow \mathbb{R}^p$$

- $F = \{x \in D \mid h(x) = 0, \quad g(x) \leq 0\}$  zulässige Menge
- $\tilde{x}$  lokale Lösung von (PNU)
- $\tilde{\lambda}, \tilde{\mu}$  Lagrange-Multiplikatoren zu  $\tilde{x}$

## Lagrange-Newton-SQP-Verfahren

(PNU)

$$\min_{x \in D} f(x)$$

$$\text{unter } h(x) = 0, \quad g(x) \leq 0$$

$$\text{wobei } f : D \rightarrow \mathbb{R}, h : D \rightarrow \mathbb{R}^m, g : D \rightarrow \mathbb{R}^p$$

- $F = \{x \in D \mid h(x) = 0, \quad g(x) \leq 0\}$  zulässige Menge
- $\tilde{x}$  lokale Lösung von (PNU)
- $\tilde{\lambda}, \tilde{\mu}$  Lagrange-Multiplikatoren zu  $\tilde{x}$
- $J(x) = \{1 \leq j \leq p \mid g_j(x) = 0\}$  aktive Ungleichungen,  $p(x) := \#J(x)$

## reduziertes System

- notwendige Optimalitätsbedingungen ergeben:

## reduziertes System

- notwendige Optimalitätsbedingungen ergeben:

reduziertes System:

$$\nabla f(\tilde{x}) + h'(\tilde{x})^T \tilde{\lambda} + \tilde{g}'(\tilde{x})^T \tilde{\nu} = 0$$

$$h(\tilde{x}) = 0$$

$$\tilde{g}(\tilde{x}) = 0$$

wobei  $\tilde{g}$  mit Komponenten  $g_j, j \in J(\tilde{x})$

und  $\tilde{\nu}$  mit Komponenten  $\tilde{\mu}_j, j \in J(\tilde{x})$

## Newton Verfahren

- Sei  $\tilde{p} := p(\tilde{x})$

## Newton Verfahren

- Sei  $\tilde{p} := p(\tilde{x})$
- Definiere Abbildung  $F : \mathbb{R}^{n+m+\tilde{p}} \rightarrow \mathbb{R}^{n+m+\tilde{p}}$  durch:

$$F(z) = F(x, \lambda, \nu) = \begin{pmatrix} \nabla f(x) + h'(x)^T \lambda + \tilde{g}'(x)^T \nu \\ h(x) \\ \tilde{g}(x) \end{pmatrix}$$

## Newton Verfahren

- Sei  $\tilde{p} := p(\tilde{x})$
- Definiere Abbildung  $F : \mathbb{R}^{n+m+\tilde{p}} \rightarrow \mathbb{R}^{n+m+\tilde{p}}$  durch:

$$F(z) = F(x, \lambda, \nu) = \begin{pmatrix} \nabla f(x) + h'(x)^T \lambda + \tilde{g}'(x)^T \nu \\ h(x) \\ \tilde{g}(x) \end{pmatrix}$$

- Mit  $\tilde{z} = (\tilde{x}^T, \tilde{\lambda}^T, \tilde{\nu}^T)^T$  gilt  $F(\tilde{z}) = 0$

## Newton Verfahren

- Sei  $\tilde{p} := p(\tilde{x})$
- Definiere Abbildung  $F : \mathbb{R}^{n+m+\tilde{p}} \rightarrow \mathbb{R}^{n+m+\tilde{p}}$  durch:

$$F(z) = F(x, \lambda, \nu) = \begin{pmatrix} \nabla f(x) + h'(x)^T \lambda + \tilde{g}'(x)^T \nu \\ h(x) \\ \tilde{g}(x) \end{pmatrix}$$

- Mit  $\tilde{z} = (\tilde{x}^T, \tilde{\lambda}^T, \tilde{\nu}^T)^T$  gilt  $F(\tilde{z}) = 0$
- Anwendung des Newton-Verfahrens:  $z^{(k+1)} = z^{(k)} - F'(z^{(k)})^{-1} F(z^{(k)})$

## Voraussetzungen Newton-Verfahren

- Voraussetzungen (8.1.2) – (8.1.5) für lokal quadratische Konvergenz des Newton-Verfahrens:

## Voraussetzungen Newton-Verfahren

- Voraussetzungen (8.1.2) – (8.1.5) für lokal quadratische Konvergenz des Newton-Verfahrens:
- $f, g$  und  $h$  auf  $D = \mathbb{R}^n$  zweimal differenzierbar

## Voraussetzungen Newton-Verfahren

- Voraussetzungen (8.1.2) – (8.1.5) für lokal quadratische Konvergenz des Newton-Verfahrens:
- $f, g$  und  $h$  auf  $D = \mathbb{R}^n$  zweimal differenzierbar
- $f', g'$  und  $h'$  auf  $D = \mathbb{R}^n$  Lipschitz-stetig  
⇒  $F$  auf  $\mathbb{R}^{n+m+\tilde{p}}$  differenzierbar und Lipschitz-stetig

## Voraussetzungen Newton-Verfahren

- Voraussetzungen (8.1.2) – (8.1.5) für lokal quadratische Konvergenz des Newton-Verfahrens:
- $f, g$  und  $h$  auf  $D = \mathbb{R}^n$  zweimal differenzierbar
- $f'', g''$  und  $h''$  auf  $D = \mathbb{R}^n$  Lipschitz-stetig  
⇒  $F$  auf  $\mathbb{R}^{n+m+\tilde{p}}$  differenzierbar und Lipschitz-stetig

- $F'(\tilde{z}) = \begin{pmatrix} L_{xx}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) & h'(\tilde{x})^T & \tilde{g}'(\tilde{x})^T \\ h'(\tilde{x}) & 0 & 0 \\ \tilde{g}'(\tilde{x}) & 0 & 0 \end{pmatrix}$  regulär

## Voraussetzungen Newton-Verfahren

- Voraussetzungen (8.1.2) – (8.1.5) für lokal quadratische Konvergenz des Newton-Verfahrens:
- $f, g$  und  $h$  auf  $D = \mathbb{R}^n$  zweimal differenzierbar
- $f'', g''$  und  $h''$  auf  $D = \mathbb{R}^n$  Lipschitz-stetig  
⇒  $F$  auf  $\mathbb{R}^{n+m+\tilde{p}}$  differenzierbar und Lipschitz-stetig

- $F'(\tilde{z}) = \begin{pmatrix} L_{xx}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) & h'(\tilde{x})^T & \tilde{g}'(\tilde{x})^T \\ h'(\tilde{x}) & 0 & 0 \\ \tilde{g}'(\tilde{x}) & 0 & 0 \end{pmatrix}$  regulär

d.h. wenn:

- $\tilde{x}$  normal, d.h.  $\nabla h_i(\tilde{x}), i = 1..m, \nabla g_j(\tilde{x}), j \in J(\tilde{x})$  l.u.

## Voraussetzungen Newton-Verfahren

- Voraussetzungen (8.1.2) – (8.1.5) für lokal quadratische Konvergenz des Newton-Verfahrens:
- $f, g$  und  $h$  auf  $D = \mathbb{R}^n$  zweimal differenzierbar
- $f'', g''$  und  $h''$  auf  $D = \mathbb{R}^n$  Lipschitz-stetig  
 $\Rightarrow F$  auf  $\mathbb{R}^{n+m+\tilde{p}}$  differenzierbar und Lipschitz-stetig

- $F'(\tilde{z}) = \begin{pmatrix} L_{xx}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) & h'(\tilde{x})^T & \tilde{g}'(\tilde{x})^T \\ h'(\tilde{x}) & 0 & 0 \\ \tilde{g}'(\tilde{x}) & 0 & 0 \end{pmatrix}$  regulär

d.h. wenn:

- $\tilde{x}$  normal, d.h.  $\nabla h_i(\tilde{x}), i = 1..m, \nabla g_j(\tilde{x}), j \in J(\tilde{x})$  l.u.
- hinr. Optimalitätsbed. erfüllt, d.h.

$$d^T L_{xx}(\tilde{x}, \tilde{\lambda}, \tilde{\mu})d \geq \alpha \|d\|^2 \quad \forall d \in \mathbb{R}^n : h'(\tilde{x})d = 0, \tilde{g}'(\tilde{x})d = 0$$

## reduzierte Lagrange-Funktion

- reduzierte Lagrange-Funktion

$$\tilde{L}(x, \lambda, \nu) = f(x) + \lambda^T h(x) + \nu^T \tilde{g}(x)$$

## reduzierte Lagrange-Funktion

- reduzierte Lagrange-Funktion

$$\tilde{L}(x, \lambda, \nu) = f(x) + \lambda^T h(x) + \nu^T \tilde{g}(x)$$

$$\nabla \tilde{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) = \begin{pmatrix} \nabla_x \tilde{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) \\ \nabla_\lambda \tilde{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) \\ \nabla_\nu \tilde{L}(\tilde{x}, \tilde{\lambda}, \tilde{\nu}) \end{pmatrix} = F(\tilde{z}) = \mathbf{0}_{n+m+\tilde{p}}$$

## Lagrange-Newton-Verfahren

- in jedem Iterationsschritt lösen des LGS:

$$F'(z^{(k)})(z - z^{(k)}) = -F(z^{(k)})$$

## Lagrange-Newton-Verfahren

- in jedem Iterationsschritt lösen des LGS:

$$F'(z^{(k)})(z - z^{(k)}) = -F(z^{(k)})$$

$\iff$

$$\left\{ \begin{array}{l} \nabla f(x^{(k)}) + \tilde{L}_{xx}(x^{(k)}, \lambda^{(k)}, \nu^{(k)})(x - x^{(k)}) + h'(x^{(k)})^T \lambda + \tilde{g}'(x^{(k)})^T \nu = 0 \\ h(x^{(k)}) + h'(x^{(k)})(x - x^{(k)}) = 0 \\ \tilde{g}(x^{(k)}) + \tilde{g}'(x^{(k)})(x - x^{(k)}) = 0 \end{array} \right\}$$

wobei  $z = (x^T, \lambda^T, \nu^T)^T$

## Referenzproblem

- zeige: Lösung  $z$  des LGS ist durch Lösung  $x$ ,  $\lambda$  und  $\mu$  des quadratischen Optimierungsproblems bestimmt

(QL1)<sub>k</sub>

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \frac{1}{2}(x - x^{(k)})^T L_{xx}(x^{(k)}, \lambda^{(k)}, \nu^{(k)})(x - x^{(k)}) + \nabla f(x^{(k)})^T (x - x^{(k)}) \\ \text{unter} \quad & h(x^{(k)}) + h'(x^{(k)})(x - x^{(k)}) = 0 \\ & g(x^{(k)}) + g'(x^{(k)})(x - x^{(k)}) \leq 0 \end{aligned}$$

## Referenzproblem

- parametrisches Referenzproblem zum Parameter  
 $w := (w_x^T, w_\lambda^T, w_\mu^T)^T - \tilde{z} \in \mathbb{R}^{n+m+p}$

$(QL1)_w$

$$\min_{x \in \mathbb{R}^n} f_Q(x, w) = \frac{1}{2}(x - w_x)^T L_{xx}(w_x, w_\lambda, w_\nu)(x - w_x) + \nabla f(w_x)^T (x - w_x)$$

$$\text{unter } h_Q(x, w) = h(w_x) + h'(w_x)(x - w_x) = 0$$

$$g_Q(x, w) = g(w_x) + g'(w_x)(x - w_x) \leq 0$$

## Referenzproblem

- parametrisches Referenzproblem zum Parameter  
 $w := (w_x^T, w_\lambda^T, w_\mu^T)^T - \tilde{z} \in \mathbb{R}^{n+m+p}$

$(QL1)_w$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_Q(x, w) = \frac{1}{2}(x - w_x)^T L_{xx}(w_x, w_\lambda, w_\nu)(x - w_x) + \nabla f(w_x)^T (x - w_x) \\ \text{unter} \quad & h_Q(x, w) = h(w_x) + h'(w_x)(x - w_x) = 0 \\ & g_Q(x, w) = g(w_x) + g'(w_x)(x - w_x) \leq 0 \end{aligned}$$

- hinr. Optimalitätsbedingung für Referenzproblem mit  $\tilde{w} = 0$  nach Vor.  
 gegeben,  
 damit  $\tilde{x}$  lokales Minimum des Referenzproblems

## Referenzproblem

- parametrisches Referenzproblem zum Parameter  
 $w := (w_x^T, w_\lambda^T, w_\mu^T)^T - \tilde{z} \in \mathbb{R}^{n+m+p}$

$(QL1)_w$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_Q(x, w) = \frac{1}{2}(x - w_x)^T L_{xx}(w_x, w_\lambda, w_\nu)(x - w_x) + \nabla f(w_x)^T (x - w_x) \\ \text{unter} \quad & h_Q(x, w) = h(w_x) + h'(w_x)(x - w_x) = 0 \\ & g_Q(x, w) = g(w_x) + g'(w_x)(x - w_x) \leq 0 \end{aligned}$$

- hinr. Optimalitätsbedingung für Referenzproblem mit  $\tilde{w} = 0$  nach Vor.  
 gegeben,  
 damit  $\tilde{x}$  lokales Minimum des Referenzproblems
- Multiplikatorenregel für  $(QL1)_{\tilde{w}}$  dieselbe wie für  $(PNU)$

## Referenzproblem

- parametrisches Referenzproblem zum Parameter  
 $w := (w_x^T, w_\lambda^T, w_\mu^T)^T - \tilde{z} \in \mathbb{R}^{n+m+p}$

$(QL1)_w$

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f_Q(x, w) = \frac{1}{2}(x - w_x)^T L_{xx}(w_x, w_\lambda, w_\nu)(x - w_x) + \nabla f(w_x)^T (x - w_x) \\ \text{unter} \quad & h_Q(x, w) = h(w_x) + h'(w_x)(x - w_x) = 0 \\ & g_Q(x, w) = g(w_x) + g'(w_x)(x - w_x) \leq 0 \end{aligned}$$

- hinr. Optimalitätsbedingung für Referenzproblem mit  $\tilde{w} = 0$  nach Vor.  
 gegeben,  
 damit  $\tilde{x}$  lokales Minimum des Referenzproblems
- Multiplikatorenregel für  $(QL1)_{\tilde{w}}$  dieselbe wie für  $(PNU)$
- Eindeutigkeit der Multiplikatoren wegen Normalität

## Referenzproblem

- zusätzlich strikte Komplementarität benötigt:  $\tilde{\mu}_j > 0 \forall j \in J(\tilde{\mathbf{x}})$

## Referenzproblem

- zusätzlich strikte Komplementarität benötigt:  $\tilde{\mu}_j > 0 \forall j \in J(\tilde{x})$



Satz 7.4.2

$x(w)$  striktes lokales Minimum von  $(QL1)_w$ ,

$\lambda(w), \mu(w)$  zugehörige eindeutig bestimmte Multiplikatoren

## Referenzproblem

- zusätzlich strikte Komplementarität benötigt:  $\tilde{\mu}_j > 0 \forall j \in J(\tilde{x})$



$x(w)$  striktes lokales Minimum von  $(QL1)_w$ ,

$\lambda(w), \mu(w)$  zugehörige eindeutig bestimmte Multiplikatoren

- Lösung des LGS  $z^{(k+1)} = ((x^{(k+1)})^T, (\lambda^{(k+1)})^T, (\nu^{(k+1)})^T)^T$   
 erfüllt notw. Optimalitätsbedingung für eindeutig bestimmte Lösung  
 $z^{(k+1)} = ((x(w^{(k)}))^T, (\lambda(w^{(k)}))^T, (\mu(w^{(k)}))^T)^T$   
 von  $(QL1)_k := (QL1)_{w^{(k)}}$

## Referenzproblem

$$\bullet \implies x^{(k+1)} = x(w^{(k)}), \quad \lambda^{(k+1)} = \lambda(w^{(k)}), \quad \nu^{(k+1)} = \mu(w^{(k)}) \\ \forall j \in J(w^{(k)}) = J(\tilde{x})$$

## Referenzproblem

- $\implies x^{(k+1)} = x(w^{(k)}), \quad \lambda^{(k+1)} = \lambda(w^{(k)}), \quad \nu^{(k+1)} = \mu(w^{(k)})$   
 $\forall j \in J(w^{(k)}) = J(\tilde{x})$
- Für  $j \notin J(\tilde{x})$  ist  $\mu(w^{(k)})_j = 0$

## Referenzproblem

- $\implies x^{(k+1)} = x(w^{(k)}), \quad \lambda^{(k+1)} = \lambda(w^{(k)}), \quad \nu^{(k+1)} = \mu(w^{(k)})$   
 $\forall j \in J(w^{(k)}) = J(\tilde{x})$
- Für  $j \notin J(\tilde{x})$  ist  $\mu(w^{(k)})_j = 0$
- Also: Bestimmung  $z^{(k+1)}$  durch Lösung von  $(QL1)_k$

## quadratisches Problem

- mit Newton-Richtung  $d^{(k)} := x^{(k+1)} - x^{(k)}$  das zu  $(QL1)_k$  äquivalente Problem:

$(QL2)_k$

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T L_{xx}(x^{(k)}, \lambda^{(k)}, \nu^{(k)}) d + \nabla f(x^{(k)})^T d \\ \text{unter} \quad & h(x^{(k)}) + h'(x^{(k)})d = 0 \\ & g(x^{(k)}) + g'(x^{(k)})d \leq 0 \end{aligned}$$

## Lagrange-Newton-SQP-Verfahren

### Lagrange-Newton-SQP-Verfahren für (PNU)

1. Wähle Startpunkt  $z^{(0)} = (x^{(0)}, \lambda^{(0)}, \mu^{(0)})$  und setze  $k := 0$

## Lagrange-Newton-SQP-Verfahren

### Lagrange-Newton-SQP-Verfahren für (PNU)

1. Wähle Startpunkt  $z^{(0)} = (x^{(0)}, \lambda^{(0)}, \mu^{(0)})$  und setze  $k := 0$
2. Berechne Suchrichtung  $d^{(k)}$  und Multiplikatoren  $\lambda^{(k+1)}$  und  $\mu^{(k+1)}$  durch Lösung des quadratischen Problems  $(QL2)_k$

## Lagrange-Newton-SQP-Verfahren

### Lagrange-Newton-SQP-Verfahren für (PNU)

1. Wähle Startpunkt  $z^{(0)} = (x^{(0)}, \lambda^{(0)}, \mu^{(0)})$  und setze  $k := 0$
2. Berechne Suchrichtung  $d^{(k)}$  und Multiplikatoren  $\lambda^{(k+1)}$  und  $\mu^{(k+1)}$  durch Lösung des quadratischen Problems  $(QL2)_k$
3. Ist  $d^{(k)} = 0$ : STOP

## Lagrange-Newton-SQP-Verfahren

### Lagrange-Newton-SQP-Verfahren für (PNU)

1. Wähle Startpunkt  $z^{(0)} = (x^{(0)}, \lambda^{(0)}, \mu^{(0)})$  und setze  $k := 0$
2. Berechne Suchrichtung  $d^{(k)}$  und Multiplikatoren  $\lambda^{(k+1)}$  und  $\mu^{(k+1)}$  durch Lösung des quadratischen Problems  $(QL2)_k$
3. Ist  $d^{(k)} = 0$ : STOP
4. Setze  $x^{(k+1)} = x^{(k)} + d^{(k)}$

## Lagrange-Newton-SQP-Verfahren

### Lagrange-Newton-SQP-Verfahren für (PNU)

1. Wähle Startpunkt  $z^{(0)} = (x^{(0)}, \lambda^{(0)}, \mu^{(0)})$  und setze  $k := 0$
2. Berechne Suchrichtung  $d^{(k)}$  und Multiplikatoren  $\lambda^{(k+1)}$  und  $\mu^{(k+1)}$  durch Lösung des quadratischen Problems  $(QL2)_k$
3. Ist  $d^{(k)} = 0$ : STOP
4. Setze  $x^{(k+1)} = x^{(k)} + d^{(k)}$
5. Setze  $k := k + 1$  und gehe zu 2

## Zusammenfassung und Probleme

- Lagrange-Newton-Iteration berechnet unter Voraussetzungen dieselbe Iterationsfolge wie Lagrange-Newton-SQP-Verfahren

## Zusammenfassung und Probleme

- Lagrange-Newton-Iteration berechnet unter Voraussetzungen dieselbe Iterationsfolge wie Lagrange-Newton-SQP-Verfahren
- *lokal* quadratische Konvergenz

## Zusammenfassung und Probleme

- Lagrange-Newton-Iteration berechnet unter Voraussetzungen dieselbe Iterationsfolge wie Lagrange-Newton-SQP-Verfahren
- *lokal* quadratische Konvergenz

Probleme:

- in der Regel *keine* Folge von zulässigen Punkten

## Zusammenfassung und Probleme

- Lagrange-Newton-Iteration berechnet unter Voraussetzungen dieselbe Iterationsfolge wie Lagrange-Newton-SQP-Verfahren
- *lokal* quadratische Konvergenz

Probleme:

- in der Regel *keine* Folge von zulässigen Punkten
- Newton-Richtung  $d^{(k)} = x^{(k+1)} - x^{(k)}$  keine zulässige Abstiegsrichtung

# Inhaltsverzeichnis

- 1 Motivation
- 2 Problemstellung
- 3 Lagrange-Newton-SQP-Verfahren
- 4 Sequentielle quadratische Programmierung**
  - Berechnung der Suchrichtung
  - Berechnung der Schrittweite
  - Grundversion des SQP-Verfahrens
  - Globale Konvergenz im konvexen Fall
- 5 Anwendungen
  - Optimale Steuerung einer chemischen Reaktion
  - Stabilisierung eines taumelnden Flugkörpers

## Zielsetzung

- Ziel:  
Aus lokal konvergentem Lagrange-Newton-SQP-Verfahren  
global konvergentes Verfahren erreichen

## Zielsetzung

- Ziel:  
Aus lokal konvergentem Lagrange-Newton-SQP-Verfahren  
global konvergentes Verfahren erreichen
- Vorgehensweise:  
analog zur Konstruktion von Variable-Metrik-Verfahren

## Berechnung der Suchrichtung

$(QP)_k$

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d \\ \text{unter} \quad & h(x^{(k)}) + h'(x^{(k)})d = 0 \\ & g(x^{(k)}) + g'(x^{(k)})d \leq 0 \end{aligned}$$

## Berechnung der Suchrichtung

$(QP)_k$

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d \\ \text{unter} \quad & h(x^{(k)}) + h'(x^{(k)})d = 0 \\ & g(x^{(k)}) + g'(x^{(k)})d \leq 0 \end{aligned}$$

$$\mathcal{F}_k = \{d \in \mathbb{R}^n \mid h(x^{(k)}) + h'(x^{(k)})d = 0, g(x^{(k)}) + g'(x^{(k)})d \leq 0\}$$

## Berechnung der Suchrichtung

$(QP)_k$

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d \\ \text{unter} \quad & h(x^{(k)}) + h'(x^{(k)})d = 0 \\ & g(x^{(k)}) + g'(x^{(k)})d \leq 0 \end{aligned}$$

$$\mathcal{F}_k = \{d \in \mathbb{R}^n \mid h(x^{(k)}) + h'(x^{(k)})d = 0, g(x^{(k)}) + g'(x^{(k)})d \leq 0\}$$

- zulässige Menge  $\mathcal{F}_k = \emptyset$  möglich

## Berechnung der Suchrichtung

$(QP)_k$

$$\begin{aligned} \min_{d \in \mathbb{R}^n} \quad & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d \\ \text{unter} \quad & h(x^{(k)}) + h'(x^{(k)})d = 0 \\ & g(x^{(k)}) + g'(x^{(k)})d \leq 0 \end{aligned}$$

$$\mathcal{F}_k = \{d \in \mathbb{R}^n \mid h(x^{(k)}) + h'(x^{(k)})d = 0, g(x^{(k)}) + g'(x^{(k)})d \leq 0\}$$

- zulässige Menge  $\mathcal{F}_k = \emptyset$  möglich
- deshalb: betrachte zunächst *konvexe* Probleme vom Typ  $(PNU)$  mit *lösbaren* Problemen  $(QP)_k$

## Berechnung der Suchrichtung

### Satz 8.2.1

$f$  und  $g_j$ ,  $j = 1..p$  konvex und differenzierbar auf  $D = \mathbb{R}^n$ ,  
 $h_i$ ,  $i = 1..m$  affin-linear, d.h.  $h_i(x) = (a^i)^T x - b_i$   
 $a^i$ ,  $i = 1..m$  sind linear unabhängig  
 $\exists v \in \mathbb{R}^n$  mit  $g(v) < 0$ ,  $h(v) = 0$  } 7.2.29 jeder Punkt aus  $\mathcal{F}$  regulär  
Dann:

## Berechnung der Suchrichtung

### Satz 8.2.1

$f$  und  $g_j$ ,  $j = 1..p$  konvex und differenzierbar auf  $D = \mathbb{R}^n$ ,

$h_i$ ,  $i = 1..m$  affin-linear, d.h.  $h_i(x) = (a^i)^T x - b_i$

$a^i$ ,  $i = 1..m$  sind linear unabhängig

$\exists v \in \mathbb{R}^n$  mit  $g(v) < 0$ ,  $h(v) = 0$

} 7.2.29 jeder Punkt aus  $\mathcal{F}$  regulär

Dann:

$\tilde{x} \in \mathcal{F}$  globale Lösung von (PNU)  $\iff \exists$  Lagrange-Multiplikatoren zu  $\tilde{x}$

## Berechnung der Suchrichtung

$$\mathcal{F}(\tau) := \{x \in \mathbb{R}^n \mid |h_i(x)| \leq \tau, i = 1..m, g_j(x) \leq \tau, j = 1..p\}$$

## Berechnung der Suchrichtung

$$\mathcal{F}(\tau) := \{x \in \mathbb{R}^n \mid |h_i(x)| \leq \tau, i = 1..m, g_j(x) \leq \tau, j = 1..p\}$$

(8.2.3)  $A^{(k)}$  symmetrisch, gleichmäßig positiv definit und beschränkt

## Berechnung der Suchrichtung

$$\mathcal{F}(\tau) := \{x \in \mathbb{R}^n \mid |h_i(x)| \leq \tau, i = 1..m, g_j(x) \leq \tau, j = 1..p\}$$

(8.2.3)  $A^{(k)}$  symmetrisch, gleichmäßig positiv definit und beschränkt

### Satz 8.2.2

Vor. von Satz 8.2.1 seien erfüllt und gelte (8.2.3).

$\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt und  $\{x^{(k)}\} \subset \mathcal{F}(\tau_0)$ .

## Berechnung der Suchrichtung

$$\mathcal{F}(\tau) := \{x \in \mathbb{R}^n \mid |h_i(x)| \leq \tau, i = 1..m, g_j(x) \leq \tau, j = 1..p\}$$

(8.2.3)  $A^{(k)}$  symmetrisch, gleichmäßig positiv definit und beschränkt

### Satz 8.2.2

Vor. von Satz 8.2.1 seien erfüllt und gelte (8.2.3).

$\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt und  $\{x^{(k)}\} \subset \mathcal{F}(\tau_0)$ .

$\implies \forall k \in \mathbb{N}$  besitzt  $(QP)_k$  eindeutig bestimmte Lösung  $d^{(k)}$  mit Lagrange-Multiplikatoren  $\lambda^{(k+1)} \in \mathbb{R}^m$ ,  $\mu^{(k+1)} \in \mathbb{R}^p$  und Folgen  $\{d^{(k)}\}$ ,  $\{\lambda^{(k)}\}$  und  $\{\mu^{(k)}\}$  sind gleichmäßig beschränkt.

Beweis: nach Spellucci

## nichtkonvexe Probleme

- nichtkonvexe Probleme:

## nichtkonvexe Probleme

- nichtkonvexe Probleme:

### Satz 8.2.3

$f, g$  und  $h$  stetig differenzierbar auf  $D = \mathbb{R}^n$  und gelte (8.2.3).  
 $\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt und  $\{x^{(k)}\} \subset \mathcal{F}(\tau_0)$ .

## nichtkonvexe Probleme

- nichtkonvexe Probleme:

### Satz 8.2.3

$f$ ,  $g$  und  $h$  stetig differenzierbar auf  $D = \mathbb{R}^n$  und gelte (8.2.3).

$\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt und  $\{x^{(k)}\} \subset \mathcal{F}(\tau_0)$ .

Hat  $(QP)_k$  für jedes  $k \in \mathbb{N}$  Lösung  $d^{(k)}$  und gilt:

(8.2.4)  $\nabla h_i(x^{(k)})$ ,  $i = 1..m$ ,  $\nabla g_j(x^{(k)})$ ,  $j \in J(d^{(k)})$ , linear unabhängig,

## nichtkonvexe Probleme

- nichtkonvexe Probleme:

### Satz 8.2.3

$f, g$  und  $h$  stetig differenzierbar auf  $D = \mathbb{R}^n$  und gelte (8.2.3).

$\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt und  $\{x^{(k)}\} \subset \mathcal{F}(\tau_0)$ .

Hat  $(QP)_k$  für jedes  $k \in \mathbb{N}$  Lösung  $d^{(k)}$  und gilt:

(8.2.4)  $\nabla h_i(x^{(k)})$ ,  $i = 1..m$ ,  $\nabla g_j(x^{(k)})$ ,  $j \in J(d^{(k)})$ , linear unabhängig,

Dann  $d^{(k)}$  und Lagrange-Multiplikatoren  $\lambda^{(k+1)} \in \mathbb{R}^m$ ,  $\mu^{(k+1)} \in \mathbb{R}^p$  eindeutig bestimmt und gleichmäßig beschränkt.

## nichtkonvexe Probleme

- Forderung (8.2.4) hängt von der Lösung  $d^{(k)}$  ab

## nichtkonvexe Probleme

- Forderung (8.2.4) hängt von der Lösung  $d^{(k)}$  ab
- deshalb: Forderung einer stärkeren Bedingung bzgl.  $x$   
 $\nabla h_i(x^{(k)}), i = 1..m, \quad \nabla g_j(x^{(k)}), j \in J(x) \cup J_+(x)$ , linear unabhängig  
mit  $J_+(x) = \{1 \leq j \leq p \mid g_j(x) > 0\}$

## nichtkonvexe Probleme

- Forderung (8.2.4) hängt von der Lösung  $d^{(k)}$  ab
- deshalb: Forderung einer stärkeren Bedingung bzgl.  $x$   
 $\nabla h_i(x^{(k)})$ ,  $i = 1..m$ ,  $\nabla g_j(x^{(k)})$ ,  $j \in J(x) \cup J_+(x)$ , linear unabhängig  
mit  $J_+(x) = \{1 \leq j \leq p \mid g_j(x) > 0\}$
- $\implies$   
 $h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d = 0$ ,  $i = 1..m$   
 $g_j(x^{(k)}) + \nabla g_j(x^{(k)})^T d = 0$ ,  $j \in J(x) \cup J_+(x)$

## nichtkonvexe Probleme

- Forderung (8.2.4) hängt von der Lösung  $d^{(k)}$  ab
- deshalb: Forderung einer stärkeren Bedingung bzgl.  $x$   
 $\nabla h_i(x^{(k)})$ ,  $i = 1..m$ ,  $\nabla g_j(x^{(k)})$ ,  $j \in J(x) \cup J_+(x)$ , linear unabhängig  
 mit  $J_+(x) = \{1 \leq j \leq p \mid g_j(x) > 0\}$
- $\implies$   
 $h_i(x^{(k)}) + \nabla h_i(x^{(k)})^T d = 0$ ,  $i = 1..m$   
 $g_j(x^{(k)}) + \nabla g_j(x^{(k)})^T d = 0$ ,  $j \in J(x) \cup J_+(x)$
- restliche Nebenbedingungen  
 $g_j(x^{(k)}) + \nabla g_j(x^{(k)})^T d \leq 0$ ,  $j \in J_-(x) = \{1 \leq j \leq p \mid g_j(x) < 0\}$   
 erfüllt, wenn  $\|d\|$  hinreichend klein

## Modifikation

- Modifikation des Problems  $(QP)_k$

## Modifikation

- Modifikation des Problems  $(QP)_k$

$(QM)_k$

$$\begin{array}{ll}
 \min_{d \in \mathbb{R}^n} & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d \\
 \text{unter} & \xi h(x^{(k)}) + h'(x^{(k)})d = 0 \\
 & \xi g_j(x^{(k)}) + \nabla g_j(x^{(k)})d \leq 0, \quad j \in J(x) \cup J_-(x), \\
 & g_j(x^{(k)}) + \nabla g_j(x^{(k)})d \leq 0, \quad j \in J_-(x)
 \end{array}$$

## Modifikation

- Modifikation des Problems  $(QP)_k$

$(QM)_{k,\xi}$

$$\begin{aligned} \min_{d \in \mathbb{R}^n, \xi \in \mathbb{R}} \quad & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d + \frac{1}{2} \rho (1 - \xi)^2 \\ \text{unter} \quad & \xi h(x^{(k)}) + h'(x^{(k)}) d = 0 \\ & \xi g_j(x^{(k)}) + \nabla g_j(x^{(k)}) d \leq 0, \quad j \in J(x) \cup J_-(x), \\ & g_j(x^{(k)}) + \nabla g_j(x^{(k)}) d \leq 0, \quad j \in J_-(x) \\ & 0 \leq \xi \leq 1 \end{aligned}$$

## Modifikation

- Modifikation des Problems  $(QP)_k$

$(QM)_{k,\xi}$

$$\begin{aligned} \min_{d \in \mathbb{R}^n, \xi \in \mathbb{R}} \quad & \frac{1}{2} d^T A^{(k)} d + \nabla f(x^{(k)})^T d + \frac{1}{2} \rho (1 - \xi)^2 \\ \text{unter} \quad & \xi h(x^{(k)}) + h'(x^{(k)}) d = 0 \\ & \xi g_j(x^{(k)}) + \nabla g_j(x^{(k)}) d \leq 0, \quad j \in J(x) \cup J_-(x), \\ & g_j(x^{(k)}) + \nabla g_j(x^{(k)}) d \leq 0, \quad j \in J_-(x) \\ & 0 \leq \xi \leq 1 \end{aligned}$$

- $F_k \neq \emptyset$ , da  $\xi = 0$  und  $d = 0$  zulässig

## Berechnung der Schrittweite

Berechnung der Schrittweite:

- Schwierigkeiten:
  - $x^{(k)}$  nicht notwendig zulässig für  $(PNU)$

## Berechnung der Schrittweite

Berechnung der Schrittweite:

- Schwierigkeiten:
  - $x^{(k)}$  nicht notwendig zulässig für  $(PNU)$
  - Suchrichtung  $d^{(k)}$  in der Regel keine zulässige Abstiegs-Richtung von  $f$  in  $x^{(k)}$

## Berechnung der Schrittweite

Berechnung der Schrittweite:

- Schwierigkeiten:
  - $x^{(k)}$  nicht notwendig zulässig für  $(PNU)$
  - Suchrichtung  $d^{(k)}$  in der Regel keine zulässige Abstiegs-Richtung von  $f$  in  $x^{(k)}$
- Abhilfe: Berechnung der Schrittweite mit Merit-Funktion  $\phi$  anstelle der Zielfunktion

## Merit-Funktion

- Eigenschaften der Merit-Funktion  $\phi$ :
  - $\tilde{x} \in \mathcal{F}$  lokale Lösung von (PNU)  
 $\Rightarrow \tilde{x}$  lokales, unrestringiertes Minimum von  $\phi$ .

## Merit-Funktion

- Eigenschaften der Merit-Funktion  $\phi$ :
  - $\tilde{x} \in \mathcal{F}$  lokale Lösung von (PNU)  
 $\Rightarrow \tilde{x}$  lokales, unrestringiertes Minimum von  $\phi$ .
  - $d^{(k)}$  Abstiegsrichtung von  $\phi$  in  $x^{(k)}$

## Merit-Funktion

- Eigenschaften der Merit-Funktion  $\phi$ :
  - $\tilde{x} \in \mathcal{F}$  lokale Lösung von (PNU)  
 $\Rightarrow \tilde{x}$  lokales, unrestringiertes Minimum von  $\phi$ .
  - $d^{(k)}$  Abstiegsrichtung von  $\phi$  in  $x^{(k)}$
- erste Eigenschaft kann nicht global sichergestellt werden,

## Merit-Funktion

- Eigenschaften der Merit-Funktion  $\phi$ :
  - $\tilde{x} \in \mathcal{F}$  lokale Lösung von (PNU)  
 $\Rightarrow \tilde{x}$  lokales, unrestringiertes Minimum von  $\phi$ .
  - $d^{(k)}$  Abstiegsrichtung von  $\phi$  in  $x^{(k)}$
- erste Eigenschaft kann nicht global sichergestellt werden,  
deshalb: Anpassung der Merit-Funktion in jedem Iterationsschritt

## Merit-Funktion

- Als Merit-Funktionen praktisch bewährt:

$$\phi(x; \beta, \gamma) := f(x) + \underbrace{\sum_{j=1}^p \beta_j g_j(x)_+ + \sum_{i=1}^m \gamma_i |h_i(x)|}_{\text{Penalty-Term}},$$

$$\text{wobei } g_j(x)_+ := \max\{0, g_j(x)\}$$

Penalty-Parameter  $\beta \in \mathbb{R}^p$ ,  $\beta \geq 0$ ,  $\gamma \in \mathbb{R}^m$ ,  $\gamma \geq 0$

## Merit-Funktion

- Als Merit-Funktionen praktisch bewährt:

$$\phi(x; \beta, \gamma) := f(x) + \underbrace{\sum_{j=1}^p \beta_j g_j(x)_+ + \sum_{i=1}^m \gamma_i |h_i(x)|}_{\text{Penalty-Term}},$$

$$\text{wobei } g_j(x)_+ := \max\{0, g_j(x)\}$$

Penalty-Parameter  $\beta \in \mathbb{R}^p$ ,  $\beta \geq 0$ ,  $\gamma \in \mathbb{R}^m$ ,  $\gamma \geq 0$

- 'Bestrafung' bei Verletzung der Restriktionen

## Merit-Funktion

- Als Merit-Funktionen praktisch bewährt:

$$\phi(x; \beta, \gamma) := f(x) + \underbrace{\sum_{j=1}^p \beta_j g_j(x)_+ + \sum_{i=1}^m \gamma_i |h_i(x)|}_{\text{Penalty-Term}},$$

$$\text{wobei } g_j(x)_+ := \max\{0, g_j(x)\}$$

Penalty-Parameter  $\beta \in \mathbb{R}^p$ ,  $\beta \geq 0$ ,  $\gamma \in \mathbb{R}^m$ ,  $\gamma \geq 0$

- 'Bestrafung' bei Verletzung der Restriktionen
- $x \in \mathcal{F} \Rightarrow \phi(x; \beta, \gamma) = f(x)$

## Merit-Funktion

- Als Merit-Funktionen praktisch bewährt:

$$\phi(x; \beta, \gamma) := f(x) + \underbrace{\sum_{j=1}^p \beta_j g_j(x)_+ + \sum_{i=1}^m \gamma_i |h_i(x)|}_{\text{Penalty-Term}},$$

$$\text{wobei } g_j(x)_+ := \max\{0, g_j(x)\}$$

Penalty-Parameter  $\beta \in \mathbb{R}^p$ ,  $\beta \geq 0$ ,  $\gamma \in \mathbb{R}^m$ ,  $\gamma \geq 0$

- 'Bestrafung' bei Verletzung der Restriktionen
- $x \in \mathcal{F} \Rightarrow \phi(x; \beta, \gamma) = f(x)$
- $x \notin \mathcal{F} \Rightarrow$  Penalty-Term positiv

## Schrittweite

### Satz 8.2.5

$\tilde{x}$  lokales Minimum von  $(PNU)$  und Bedingungen (8.1.2) – (8.1.5) (13) erfüllt.

## Schrittweite

### Satz 8.2.5

$\tilde{x}$  lokales Minimum von  $(PNU)$  und Bedingungen (8.1.2) – (8.1.5) (13) erfüllt.  
Wählt man  $\beta_i > \mu_i$ ,  $i = 1..m$ ,  $\gamma_j > |\lambda_j|$ ,  $j = 1..p$   
dann  $\tilde{x}$  striktes lokales Minimum von  $\phi(\cdot; \beta, \gamma)$ .

## Berechnung der Schrittweite

- ohne Kenntnis der Multiplikatoren keine Penalty-Parameter  
deshalb: Parameter in jedem Iterationsschritt bei Bedarf vergrößern

## Berechnung der Schrittweite

- ohne Kenntnis der Multiplikatoren keine Penalty-Parameter  
deshalb: Parameter in jedem Iterationsschritt bei Bedarf vergrößern
- betrachte zur Vereinfachung nicht modifiziertes Problem  $(QP)_k$

## Berechnung der Schrittweite

- ohne Kenntnis der Multiplikatoren keine Penalty-Parameter  
deshalb: Parameter in jedem Iterationsschritt bei Bedarf vergrößern
- betrachte zur Vereinfachung nicht modifiziertes Problem  $(QP)_k$
- Falls Penalty-Parameter hinreichend groß sind, also:  
$$\beta_j \geq \mu_j^{k+1} + \epsilon, j = 1..p \quad \gamma_i \geq |\lambda_i^{k+1}| + \epsilon, i = 1..m \quad , \epsilon > 0$$

## Berechnung der Schrittweite

- ohne Kenntnis der Multiplikatoren keine Penalty-Parameter  
 deshalb: Parameter in jedem Iterationsschritt bei Bedarf vergrößern
- betrachte zur Vereinfachung nicht modifiziertes Problem  $(QP)_k$
- Falls Penalty-Parameter hinreichend groß sind, also:  
 $\beta_j \geq \mu_j^{k+1} + \epsilon, j = 1..p \quad \gamma_i \geq |\lambda_i^{k+1}| + \epsilon, i = 1..m, \epsilon > 0$
- kann man zeigen, dass dann gilt:

$$\begin{aligned} \phi(x^{(k)}; \beta, \gamma) - \phi(x^{(k)} + \sigma_k d^{(k)}; \beta, \gamma) \\ \geq \delta \sigma \left[ (d^{(k)})^T A^{(k)} d^{(k)} + \epsilon \left\| g(x^{(k)})_+ \right\|_1 + \epsilon \left\| h(x^{(k)})_+ \right\|_1 \right] \end{aligned}$$

## Berechnung der Schrittweite

- mit solchem  $\sigma$  hinreichender Abstieg der Merit-Funktion  
 $\implies$  effiziente Schrittweite
- $x^{(k+1)} := x^{(k)} + \sigma_k d^{(k)} \in \mathcal{F}_{\tau_0}$  kann erreicht werden

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$
3. Berechnung der Suchrichtung:

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$
3. Berechnung der Suchrichtung:
4. Anpassung der Penalty-Parameter:

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter  $\beta_j^0 := \epsilon$ ,  $j = 1..p$ ,  $\gamma_i^0 := \epsilon$ ,  $i = 1..m$  und setze  $k := 0$
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$
3. Berechnung der Suchrichtung:
4. Anpassung der Penalty-Parameter:

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter  $\beta_j^0 := \epsilon$ ,  $j = 1..p$ ,  $\gamma_i^0 := \epsilon$ ,  $i = 1..m$  und setze  $k := 0$
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$   
d.h. mit  $\rho_1 \|x\|^2 \leq x^T A^{(k)} x \leq \rho_2 \|x\|^2 \quad \forall x \in \mathbb{R}^n$
3. Berechnung der Suchrichtung:
4. Anpassung der Penalty-Parameter:

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter  $\beta_j^0 := \epsilon$ ,  $j = 1..p$ ,  $\gamma_i^0 := \epsilon$ ,  $i = 1..m$  und setze  $k := 0$
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$   
d.h. mit  $\rho_1 \|x\|^2 \leq x^T A^{(k)} x \leq \rho_2 \|x\|^2 \quad \forall x \in \mathbb{R}^n$
3. Berechnung der Suchrichtung: Berechne Lösung  $d^{(k)}$  von  $(QP)_k$  und Lagrange-Multiplikatoren  $\lambda^{(k+1)}$ ,  $\mu^{(k+1)}$
4. Anpassung der Penalty-Parameter:

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

1. Wähle Startpunkt  $x^{(0)} \in \mathcal{F}(\tau_0)$ , definiere Penalty-Parameter  $\beta_j^0 := \epsilon$ ,  $j = 1..p$ ,  $\gamma_i^0 := \epsilon$ ,  $i = 1..m$  und setze  $k := 0$
2. Wähle symmetrische, gleichmäßig positiv definite Matrix  $A^{(k)}$  d.h. mit  $\rho_1 \|x\|^2 \leq x^T A^{(k)} x \leq \rho_2 \|x\|^2 \quad \forall x \in \mathbb{R}^n$
3. Berechnung der Suchrichtung: Berechne Lösung  $d^{(k)}$  von  $(QP)_k$  und Lagrange-Multiplikatoren  $\lambda^{(k+1)}$ ,  $\mu^{(k+1)}$
4. Anpassung der Penalty-Parameter:

$$\text{für } j = 1..p \quad \beta_j^{k+1} := \begin{cases} \mu_j^{(k+1)} + 2\epsilon & \text{falls } \beta_j^{(k)} \leq \mu_j^{(k+1)} + \epsilon \\ \beta_j^{(k)} & \text{sonst} \end{cases}$$

$$\text{für } i = 1..m \quad \gamma_i^{k+1} := \begin{cases} |\lambda_i^{(k+1)}| + 2\epsilon & \text{falls } \gamma_i^{(k)} \leq |\lambda_i^{(k+1)}| + \epsilon \\ \gamma_i^{(k)} & \text{sonst} \end{cases}$$

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

5. Berechnung der Schrittweite:

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

5. Berechnung der Schrittweite:

6. Setze  $x^{(k+1)} = x^{(k)} + \sigma_k d^{(k)}$

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

5. Berechnung der Schrittweite:

6. Setze  $x^{(k+1)} = x^{(k)} + \sigma_k d^{(k)}$

7. Setze  $k := k + 1$  und gehe zu 2

## Grundversion des SQP-Verfahrens

SQP-Verfahren, Grundversion  $\epsilon > 0$ ,  $\tau_0 > 0$ ,  $0 < \alpha, \delta < 1$ ,  $0 < \rho_1 \ll 1 \ll \rho_2$

5. Berechnung der Schrittweite: Bestimme  $j_k \in \mathbb{N}$  minimal, so dass mit

$$\begin{aligned} \sigma_k &= \alpha^{j_k} \\ \phi(x^{(k)}; \beta^{(k+1)}, \gamma^{(k+1)}) - \phi(x^{(k)} + \sigma_k d^{(k)}; \beta^{(k+1)}, \gamma^{(k+1)}) \\ &\geq \delta \sigma_k \left[ (d^{(k)})^T A^{(k)} d^{(k)} + \epsilon \left\| g(x^{(k)})_+ \right\|_1 + \epsilon \left\| h(x^{(k)})_+ \right\|_1 \right] \end{aligned}$$

und  $x^{(k)} + \sigma_k d^{(k)} \in \mathcal{F}(\tau_0)$  erfüllt sind.

6. Setze  $x^{(k+1)} = x^{(k)} + \sigma_k d^{(k)}$   
7. Setze  $k := k + 1$  und gehe zu 2

## Globale Konvergenz im konvexen Fall

### Satz 8.2.7

Vor. von Satz 8.2.1 (3) seien erfüllt, es gelte (8.2.3) und  $\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt.

## Globale Konvergenz im konvexen Fall

### Satz 8.2.7

Vor. von Satz 8.2.1 (3) seien erfüllt, es gelte (8.2.3) und  $\mathcal{F}(\tau_0)$ ,  $\tau_0 > 0$  kompakt.

Dann berechnet SQP-Verfahren Folge  $\{x^{(k)}\} \subset \mathcal{F}(\tau_0)$  und jeder HP von  $\{x^{(k)}\}$  erfüllt notw. Optimalitätsbedingungen für (PNU)

## Bemerkungen

- Berechnung von  $A^{(k)}$  mit dem BFGS-Verfahren entspr. Updateformeln  
dann: lokal superlineare Konvergenz, falls  $\sigma_k = 1$  für hinr. großes  $k$

## Bemerkungen

- Berechnung von  $A^{(k)}$  mit dem BFGS-Verfahren entspr. Updateformeln  
dann: lokal superlineare Konvergenz, falls  $\sigma_k = 1$  für hinr. großes  $k$
- Unter Voraussetzungen für Lagrange-Newton-SQP-Verfahren  
 $A^{(k)} = L_{xx}(x^{(k)}, \lambda^{(k)}, \mu^{(k)})$   
dann: lokal quadratische Konvergenz, falls  $\sigma_k = 1$  für hinr. großes  $k$

## Bemerkungen

- Berechnung von  $A^{(k)}$  mit dem BFGS-Verfahren entspr. Updateformeln  
dann: lokal superlineare Konvergenz, falls  $\sigma_k = 1$  für hinr. großes  $k$
- Unter Voraussetzungen für Lagrange-Newton-SQP-Verfahren  
 $A^{(k)} = L_{xx}(x^{(k)}, \lambda^{(k)}, \mu^{(k)})$   
dann: lokal quadratische Konvergenz, falls  $\sigma_k = 1$  für hinr. großes  $k$
- Schrittweite  $\sigma_k = 1$  muss Abstiegskriterium nicht erfüllen  
Verfahren wählt Schrittweite  $\sigma_k = 1$  deshalb nicht automatisch

# Inhaltsverzeichnis

- 1 Motivation
- 2 Problemstellung
- 3 Lagrange-Newton-SQP-Verfahren
- 4 Sequentielle quadratische Programmierung
  - Berechnung der Suchrichtung
  - Berechnung der Schrittweite
  - Grundversion des SQP-Verfahrens
  - Globale Konvergenz im konvexen Fall
- 5 **Anwendungen**
  - **Optimale Steuerung einer chemischen Reaktion**
  - **Stabilisierung eines taumelnden Flugkörpers**

# Optimale Steuerung einer chemischen Reaktion

- Modellierung einer chemischen Reaktion:



## Optimale Steuerung einer chemischen Reaktion



- Modellierung einer chemischen Reaktion:
- aus Substanz  $S_1$  wird beim Durchlaufen eines heizbaren Zylinders Substanz  $S_2$  gewonnen

## Optimale Steuerung einer chemischen Reaktion



- Modellierung einer chemischen Reaktion:
- aus Substanz  $S_1$  wird beim Durchlaufen eines heizbaren Zylinders Substanz  $S_2$  gewonnen
- für  $t \in [0, 1]$  :  $\tau(t)$  Temperatur und  $z_i(t)$ ,  $i = 1, 2$  Konzentration von  $S_i$

## Optimale Steuerung einer chemischen Reaktion



- Modellierung einer chemischen Reaktion:
- aus Substanz  $S_1$  wird beim Durchlaufen eines heizbaren Zylinders Substanz  $S_2$  gewonnen
- für  $t \in [0, 1]$  :  $\tau(t)$  Temperatur und  $z_i(t)$ ,  $i = 1, 2$  Konzentration von  $S_i$
- Ziel: Steuerung der Temperaturverteilung, so dass  $z_2$  am rechten Zylinderende, also  $z_2(1)$ , maximal

## Steuerprozess

### Steuerprozess

$$\begin{array}{ll} \min & -z_2(1) \\ \text{unter} & 0 \leq u(t) \leq 1 \quad \forall t \in [0, 1] \\ & \dot{z}_1(t) = -z_1(t)u(t) + z_2(t)u(t)^2 \quad \forall t \in [0, 1] \\ & \dot{z}_2(t) = z_1(t)u(t) - 3z_2(t)u(t)^2 \quad \forall t \in [0, 1] \\ & z_1(0) = 1, \quad z_2(0) = 0 \end{array}$$

↓ Diskretisierung

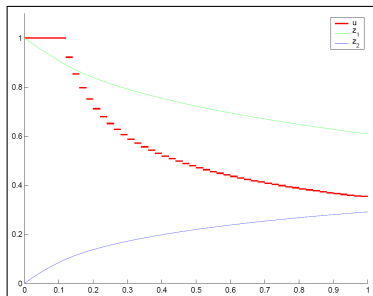
## Nichtlineares Optimierungsproblem

(PNU)

$$\begin{array}{ll}
 \min & -z_{2,N} \\
 \text{unter} & 0 \leq u_i \leq 1 \quad i=0..N-1 \\
 & z_{1,i+1} - z_{1,i} = h_N(-z_{1,i}u_i + z_{2,i}u_i^2) \quad i=0..N-1 \\
 & z_{2,i+1} - z_{2,i} = h_N(-z_{1,i}u_i - 3z_{2,i}u_i^2) \quad i=0..N-1 \\
 & z_{1,0} = 1, \quad z_{2,0} = 0 \\
 \\
 & x = (u_0, \dots, u_{N-1}, z_{1,1}, \dots, z_{1,N}, z_{2,1}, \dots, z_{2,N})^T
 \end{array}$$

## Ergebnis

- Verwendung von *fmincon*
- Implementierung des SQP-Verfahrens mit BFGS-Update von  $A^{(k)}$
- Ergebnis:  $u(t) = \exp(-\frac{c}{\tau(t)})$ ,  $t \in [0, 1]$   
optimale Steuerung der Temperaturverteilung



## Stabilisierung eines taumelnden Flugkörpers



- Taumelnder Flugkörper soll innerhalb einer vorgegebenen Zeit  $T > 0$  mit möglichst geringem Energieaufwand stabilisiert werden.

## Steuerprozess

### Steuerprozess

$$\begin{aligned} \min \quad & \frac{1}{2}(z_1(T)^2 + z_2(T)^2) + \frac{1}{2} \int_0^T [u_1(t)^2 + u_2(t)^2] dt \\ \text{unter} \quad & \dot{z}_1(t) = z_2(t) + u_1(t) \quad \forall t \in [0, T] \\ & \dot{z}_2(t) = -z_1(t) + u_2(t) \quad \forall t \in [0, T] \\ & z_1(0) = 1, \quad z_2(0) = 2 \end{aligned}$$

↓ Diskretisierung

## Nichtlineares Optimierungsproblem

(PNU)

$$\begin{aligned}
 \min \quad & \frac{1}{2}(z_{1,N}^2 + z_{2,N}^2) + \frac{1}{2}h \sum_{i=0}^{N-1} [u_{1,i}^2 + u_{2,i}^2] \\
 \text{unter} \quad & z_{1,i+1} - z_{1,i} = h_N(z_{2,i} + u_{1,i}) && i=0..N-1 \\
 & z_{2,i+1} - z_{2,i} = h_N(-z_{1,i} + u_{2,i}) && i=0..N-1 \\
 & z_{1,0} = 1, \quad z_{2,0} = 0
 \end{aligned}$$

$$x = (u_{1,0}, u_{2,0}, z_{1,1}, z_{2,1}, \dots, u_{1,N-1}, u_{2,N-1}, z_{1,N}, z_{2,N})$$

Vielen Dank für die Aufmerksamkeit!  
Noch Fragen?

