

Neue Methoden für die Optimierung verfahrenstechnischer Prozesse

Von der Fakultät Mathematik und Physik der Universität Stuttgart
zur Erlangung der Würde eines
Doktors der Naturwissenschaften (Dr. rer. nat.)
genehmigte Abhandlung

Vorgelegt von

Dipl.-Math. Stephania Irmengard Johanna Barbara
Hokenmaier

aus Schwäbisch Gmünd

Hauptberichter:	Prof. Dr. Barbara Kaltenbacher
Mitberichter:	Prof. Dr. Boris Vexler
	Prof. Dr. Bastian von Harrach

Tag der mündlichen Prüfung: 21. Juni 2013

Institut für Stochastik und Anwendungen
Universität Stuttgart
2013

Vorwort

Die vorliegende Arbeit entstand während meiner Zeit als Doktorand in der Abteilung Information Technology for Process Design and Control (ITP) der Engineering Division der Linde AG. Dort beschäftigte ich mich seitdem mit den Optimierungsverfahren des hauseigenen Prozesssimulators OPTISIM®.

Bedanken möchte ich mich bei der Linde AG, Engineering Division für die Ermöglichung dieser Arbeit, sowie meinen Kollegen der Abteilung ITP für die Motivation und Unterstützung, die vielfältigen Ratschläge, die Hilfe bei allen möglichen Problemen und das freundliche Arbeitsklima. Besonders danke ich meinen Betreuern seitens der Linde AG, Frau Dr. Gabriele Engl, die diese Arbeit überhaupt erst in die Wege geleitet hat, sowie Herrn Dr. Andreas Kröner für die jahrelange Motivation und Unterstützung.

Ganz besonders bedanke ich mich bei meiner Doktormutter Frau Prof. Dr. Barbara Kaltenbacher für die unglaubliche Geduld, die fachliche Unterstützung, die vielfältigen Hilfestellungen sowie ihre „Erreichbarkeit in allen Lebenslagen“.

Prof. Dr. Boris Vexler und Prof. Dr. Bastian von Harrach danke ich ganz herzlich für die teils sehr spontane Übernahme des Amtes als Mitberichter.

Schließlich möchte ich meinem Ehemann Harald danken, der mir stets den Rücken gestärkt und das Ende dieser Arbeit herbeigesehnt hat.

Inhaltsverzeichnis

Symbolverzeichnis	7
Zusammenfassung	13
Abstract	15
1 Einführung	17
1.1 Modellierung	18
1.2 Stationäre Simulation	21
1.3 Stationäre Optimierung in OPTISIM®	23
2 Aufgabenstellung	27
2.1 Verfahrenstechnische Herausforderungen in der Optimierung	27
2.2 Neue Ansätze für die Optimierung	32
3 Innere-Punkte-Methode und IPOPT	37
3.1 Einige Definitionen vorab	37
3.2 Das Innere-Punkte-Verfahren	39
3.3 Konvergenz der Innere-Punkte-Verfahren	57
3.4 Konvergenz des Filter-Verfahrens bei Störungen in den Nebenbedingungen	66
4 Numerische Ergebnisse	85
4.1 Einbinden von IPOPT in OPTISIM®	85
4.2 Testergebnisse	90
5 Globale Optimierung	97
5.1 Wegfindung	97
5.2 Globaler Ansatz in OPTISIM®	105
Literaturverzeichnis	111

Symbolverzeichnis

Schreibweise

Zur Vereinfachung der Schreibweise wird die Konvention $(x, y) = (x^T, y^T)^T$ für Vektoren x und y verwendet. Die i . Komponente eines Vektors x wird mit x_i bezeichnet. Innerhalb eines iterativen Verfahrens werden bei Skalaren und Matrizen die Iterationsindizes tiefgestellt (z.B. α_k, A_k), bei Vektoren jedoch hochgestellt (z.B. x^k). Für Anlagenkomponenten sowie deren Modelle wird der Index hochgestellt in runden Klammern angegeben, z.B. $U^{(j)}$.

Relationen der Art $x > 0$ gelten grundsätzlich komponentenweise. Mehrmals wird mit großem Buchstaben $X = \text{diag}(x) \in \mathbb{R}^{n \times n}$ eine Diagonalmatrix bezeichnet, deren Einträge auf der Diagonalen die Komponenten des Vektors mit dem entsprechenden kleinen Buchstaben $x \in \mathbb{R}^n$ sind. $e = (1, \dots, 1)^T$ ist ein Vektor in der jeweils benötigten Dimension, dessen Koordinaten alle 1 sind.

Falls nicht genauer bezeichnet, so ist $\|\cdot\|$ eine feste Vektornorm und für Matrizen ihre kompatible Matrixnorm.

Für eine Matrix $A \in \mathbb{R}^{n \times m}$ wird mit $\text{range}(A)$ das Bild der Abbildung, die durch $\mathbb{R}^m \rightarrow \mathbb{R}^n : x \mapsto Ax$ definiert ist, bezeichnet. Des Weiteren ist $\sigma_i(A)$ der i . größte Singulärwert der Matrix A . $\sigma_{\min}(A)$ bezeichnet den kleinsten und $\sigma_{\max}(A)$ den größten Singulärwert von A . Für eine quadratische, symmetrische und positiv definite Matrix A bezeichnet $\lambda_{\min}(A)$ den kleinsten Eigenwert.

Bezeichnungen

A	transponierte Jacobimatrix der Gleichungsnebenbedingungen
A^δ	gestörte transponierte Jacobimatrix der Gleichungsnebenbedingungen
\mathcal{A}	Indexmenge der aktiven Ungleichungsnebenbedingungen
α	primale Schrittweite im Innere-Punkte-Verfahren
α^z	duale Schrittweite im Innere-Punkte-Verfahren

α^{max}	maximale primale Schrittweite im Innere-Punkte-Verfahren
α^{min}	minimale primale Schrittweite im Innere-Punkte-Verfahren (bei Unterschreitung wird die <i>Feasibility Restoration Phase</i> gestartet)
α^{SOC}	Schrittweite, die von der <i>Second-Order Correction</i> berechnet wird
$\bar{\alpha}_1, \bar{\alpha}_2$	Schranken an die Schrittweite für die Gültigkeit des Konvergenzbeweises der Filter-Liniensuche für gestörte Gleichungsnebenbedingungen
B	Barrierefunktion
c	(reelle) Nebenbedingungen des Optimierungsproblems
c^δ	gestörte (Gleichungs-)Nebenbedingungen des Optimierungsproblems
c^{int}	Nebenbedingungen mit ganzzahligen Werten
c^E	Gleichungsnebenbedingungen
c^U	Ungleichungsnebenbedingungen
c^L, c_{min}	untere Schranke für c
c^U, c_{max}	obere Schranke für c
c_{min}^{int}	untere Schranke für c^{int}
c_{max}^{int}	obere Schranke für c^{int}
C	offene Menge im Konvergenzbeweis der Filter-Liniensuche
C_f, C_θ	Konstanten in der Taylorabschätzung für f bzw. θ
c_G	Wegkosten in einem Graphen
δ_A	Störung der Jacobimatrix der Gleichungsnebenbedingungen
δ_c	Störung der Gleichungsnebenbedingungen
δ_{ij}	Kroneckersymbol
$\Delta\lambda, \Delta z^L, \Delta z^U$	Suchrichtungen der dualen Variablen während des Innere-Punkte-Verfahrens
Δx	Suchrichtung der primalen Variablen während des Innere-Punkte-Verfahrens
$\Delta x_Y, \Delta x_Z$	Vektoren zur Berechnung der Suchrichtung der primalen Variablen durch den reduzierten Ansatz
$\Delta x_Y^\delta, \Delta x_Z^\delta$	Vektoren zur Berechnung der Suchrichtung für das gestörte Optimierungsproblem durch den reduzierten Ansatz
Δx^{SOC}	korrigierte Suchrichtung, die von der <i>Second-Order Correction</i> berechnet wird
e	Vektor, dessen Komponenten alle 1 sind
e_i	i . kanonischer Einheitsvektor
E_μ	Fehler in den Optimalitätsbedingungen des Barriereproblems
ε_{tol}	Abbruchtoleranz im Innere-Punkte-Verfahren

ϵ_x	Beschränkung der Iterierten im Konvergenzbeweis der Filter-Liniensuche für Innere-Punkte-Verfahren
$\varepsilon_1, \varepsilon_2$	Abschätzungen im Konvergenzbeweis der Filter-Liniensuche für gestörte Gleichungsnebenbedingungen
η_b	Konstante für die Armijo-artige Bedingung in der Filter-Liniensuche des Innere-Punkte-Verfahrens
f	Zielfunktion des Optimierungsproblems
$f^{(j)}$	Modellfunktion einer Unit $U^{(j)}$
F	Modellfunktion des gesamten Anlagenmodells
\mathcal{F}	Filter bei der Filter-Liniensuche des Innere-Punkte-Verfahrens
g_G	Wegkosten vom Startpunkt aus in einem Graphen
Γ_B	Konstante für die Schaltbedingung in der Filter-Liniensuche
γ_α	Konstante zur Bestimmung von α^{min}
γ_b, γ_θ	Konstanten für die Erweiterung des Filters bzw. die Akzeptanz eines neuen Punktes durch den Filter
H	Hessematrix (oder deren Approximation) der Lagrange-Funktion
$H^{(j)}$	Projektionsmatrix für die Ausgangsgrößen einer Unit $U^{(j)}$
h^{max}	maximale Anzahl der Iterationen der <i>Second-Order Correction</i>
h_G	Heuristikfunktion auf einem Graphen
Id	Einheitsmatrix, Identität
$K^{(jl)}$	Koppelmatrix der Ein- und Ausgangsströme von Unit $U^{(j)}$ bzw. $U^{(l)}$ des Anlagenmodells
K^δ	Iterationsindex ab dem die Voraussetzungen an die Störungen gelten
κ_{resto}	Faktor um den die Verletzung der Nebenbedingungen durch die <i>Feasibility Restoration Phase</i> verringert werden soll
κ^{SOC}	Faktor um den die Verletzung der Nebenbedingungen durch die <i>Second-Order Correction</i> verringert werden soll
κ_ε	Konstante zur Berechnung der Abbruchtoleranz der inneren Schleife des Innere-Punkte-Verfahrens
κ_μ	Konstante zur Berechnung des nächsten Barriereparameters
κ_Σ	Faktor zur Anpassung der dualen Variablen
κ_1, κ_2	Konstanten zur Berechnung des Filterrandes Ψ
\mathcal{L}	Lagrange-Funktion
λ	Lagrangemultiplikatoren
λ_+	Kurzform von $\lambda + \Delta\lambda$

m	Anzahl der Nebenbedingungen des Optimierungsproblems
M_d	Schranke für $\ \Delta x\ $ im Konvergenzbeweis der Filter-Liniensuche
M_λ	Schranke für $\ \lambda_+\ $ im Konvergenzbeweis der Filter-Liniensuche
M_m	Schranke für den Betrag des Abstiegs der Zielfunktion im Konvergenzbeweis der Filter-Liniensuche
M_A	Abschätzung für $\sigma_{\min}(A)$ im Konvergenzbeweis der Filter-Liniensuche
M_A^δ	Abschätzung für $\sigma_{\min}(A^\delta)$ im Konvergenzbeweis der Filter-Liniensuche
M_H	Abschätzung für die Hessematrix im Konvergenzbeweis der Filter-Liniensuche
M_H^δ	Abschätzung für die Hessematrix des gestörten Optimierungsproblems im Konvergenzbeweis der Filter-Liniensuche
μ	Barriereparameter
n	Anzahl der Optimierungsvariablen
n_p	Anzahl der Designvariablen des Anlagenmodells
n_{p2}	Anzahl der ganzzahligen Designvariablen des Anlagenmodells
$n_u^{(j)}$	Anzahl der Eingangsgrößen einer Unit $U^{(j)}$
$n_y^{(j)}$	Anzahl der Ausgangsgrößen einer Unit $U^{(j)}$
ν_1, ν_2	Konstanten zur Bestimmung der nächsten Schrittweite in der Liniensuche
\mathcal{O}, o	Landausymbole
p	Designvariablen eines Anlagenmodells
p^{int}	ganzzahlige Designvariablen
p_{\min}	untere Schranke für p
p_{\max}	obere Schranke für p
p_{\min}^{int}	untere Schranke für p^{int}
p_{\max}^{int}	obere Schranke für p^{int}
p_{Init}	Startvektor der Designvariablen für eine Optimierung in OPTISIM®
$\mathcal{P}_{\text{Init}}$	Menge der Startvektoren p_{Init} für die globale Optimierung in OPTISIM®
q	Schaltfunktionen eines Anlagenmodells
\mathcal{R}	Menge der Iterationsindizes der Iterationen, in denen die <i>Feasibility Restoration Phase</i> aufgerufen wurde
\mathcal{R}_{inc}	Menge der Iterationsindizes der Iterationen, in denen die <i>Feasibility Restoration Phase</i> während der Berechnung der Suchrichtung aufgerufen wurde

s	Anzahl der Zustandsvariablen des Anlagenmodells
s_b, s_θ	Konstanten für die Schaltbedingung in der Filter-Liniensuche
$\sigma(i)$	Index der Modellfunktion für die Parameterquelle der Designvariable p_i
$\Sigma, \Sigma_L, \Sigma_U$	duale Hessematrizen
τ	Abstandsfaktor zum Rand bei der Innere-Punkte-Methode
τ_{min}	minimaler Abstandsfaktor
θ	Maß für die Verletzung der Gleichungsnebenbedingungen
θ^δ	Maß für die Verletzung der gestörten Gleichungsnebenbedingungen
θ_{inc}	Schranke für $\theta(x)$ im Konvergenzbeweis der Filter-Liniensuche
θ^{max}	obere Schranke für $\theta(x)$ zur Initialisierung des Filters
θ^{min}	Schranke für $\theta(x)$ während der Filter-Liniensuche
θ_μ	Konstante zur Berechnung des nächsten Barriereparameters
Θ	minimaler Wert von θ im Filter \mathcal{F}
$u^{(j)}$	Eingangsgrößen einer Unit $U^{(j)}$
u	Kanten eines Graphen
$U^{(j)}$	j . Unit einer verfahrenstechnischen Anlage
U	Kantenmenge eines Graphen
v	Knoten eines Graphen
V	Knotenmenge eines Graphen
$x = (\xi, p)$	Optimierungsvariablen, primale Variablen
x^L	untere Schranke für x
x^U	obere Schranke für x
x_{min}	untere Schranke für x
x_{max}	obere Schranke für x
x^{SOC}	Testpunkt während der <i>Second-Order Correction</i>
ξ	Zustandsvariablen
ξ_{p_i}	die zur Designvariable p_i zugehörige Zustandsvariable
ξ_{min}	untere Schranke für ξ
ξ_{max}	obere Schranke für ξ
$y^{(j)}$	Ausgangsgrößen einer Unit $U^{(j)}$
\tilde{Y}	Matrix zur Berechnung der Suchrichtung der primalen Variablen durch den reduzierten Ansatz
\tilde{Y}^δ	Matrix zur Berechnung der Suchrichtung durch den reduzierten Ansatz für das gestörte Optimierungsproblem

z^L	duale Variable bzw. Lagrangemultiplikatoren der unteren Beschränkung der Optimierungsvariablen x
z^U	duale Variable bzw. Lagrangemultiplikatoren der oberen Beschränkung der Optimierungsvariablen x
\tilde{Z}	Matrix zur Berechnung der Suchrichtung durch den reduzierten Ansatz
\tilde{Z}^δ	Matrix zur Berechnung der Suchrichtung durch den reduzierten Ansatz für das gestörte Optimierungsproblem
ζ_θ, ζ_x $\zeta_s, \overline{\zeta_s}$	Schranken für $\theta(x)$ bzw. x im Konvergenzbeweis der Filter-Liniensuche Schranken für die „fast aktiven“ Nebenbedingungen im Konvergenzbeweis der Filter-Liniensuche
Φ	Zielfunktion des Optimierungsproblems in OPTISIM®
χ	Optimalitätsmaß
χ^δ	gestörtes Optimalitätsmaß
ψ_c, ψ_w	Faktoren zur Anpassung der Matrix des zu lösenden Gleichungssystems im Innere-Punkte-Verfahren, sodass die Matrix nicht singulär wird.
Ψ	Filterrand im adaptiven Algorithmus des Innere-Punkte-Verfahrens

Zusammenfassung

Die Verfahrenstechnik beschäftigt sich mit der Trennung, Umwandlung oder Verflüssigung chemischer Stoffe, wie zum Beispiel die Zerlegung von Luft in ihre Hauptbestandteile Sauerstoff, Stickstoff und Argon oder die Reinigung und Verflüssigung von Erdgas. Mit Hilfe von Prozesssimulatoren können verfahrenstechnische Anlagen, oder Teile davon, modelliert, simuliert und anschließend optimiert werden.

Die Engineering Division der Linde AG, eines der weltweit führenden Unternehmen für die Planung und den Bau von verfahrenstechnischen Anlagen, betreibt und entwickelt den hauseigenen Prozesssimulator OPTISIM[®]. Steigende Anforderungen an die Optimierung im Bezug auf Problemgröße, Effizienz und Robustheit, insbesondere bei auftretenden Unstetigkeiten und der Verwendung von Approximationen während der Simulation und Optimierung, erfordern den Einsatz neuer Optimierungsverfahren.

In dieser Arbeit wurde ein simultaner Optimierungsansatz implementiert, bei dem die Modellgleichungen als Gleichungsnebenbedingungen zu dem Optimierungsproblem hinzugefügt werden. Zur Lösung dieser komplexen Optimierungsaufgaben wurde der Optimierer IPOPT in OPTISIM[®] eingebunden. Hierbei wurde die globale Konvergenz der Filter-Liniensuche von Biegler und Wächter, die in IPOPT verwendet wird, unter der Annahme von Störungen in den Gleichungsnebenbedingungen und deren Ableitungen untersucht. Die Störungen modellieren hierbei die verwendeten Approximationen sowie bis zu einem gewissen Maße auch auftretende Unstetigkeiten.

Des Weiteren wurde ein globaler Optimierungsalgorithmus mit einer Methode des mehrfachen Starts umgesetzt. Hierzu wurde eine spezielle Behandlung der Startwerte entwickelt, sodass die Optimierung von unterschiedlichen Punkten aus gestartet werden kann.

Abstract

Chemical Engineering deals with separation, chemical reactions and liquefaction of different chemical compounds, for example separation of air into its main components oxygen, nitrogen and argon or cleaning and liquefying of natural gas. Entire Chemical plants or parts of them can be modelled, simulated and also optimized with the help of process simulation tools.

The Engineering Division of The Linde Group, one of the world leading companies for plant engineering and construction, has been developing the in-house process simulation program OPTISIM[®]. Increasing demands on the optimizer concerning problem size, efficiency and robustness, especially with the occurrence of discontinuities and the use of approximations during simulation and optimization, give rise to a closer look towards new optimization methods.

In this thesis a simultaneous optimization approach was implemented where the model equations are added to the optimization problem as equality constraints. For this large-scaled problem the optimizer IPOPT was included into OPTISIM[®]. In this context the global convergence of the filter line search of Biegler and Wächter, used in the optimizer IPOPT, was investigated under the assumption of perturbed equality constraints and derivatives, which models their approximate evaluation as well as the discontinuities to some extent.

In addition a global optimization approach with a multiple start method was adopted. Therefore a special handling of the starting points was developed in order to facilitate the start of the optimization from different points.

Kapitel 1

Einführung

Prozesssimulatoren sind ein wichtiges Werkzeug der Verfahrenstechnik und aus der täglichen Arbeit von Chemieingenieuren nicht mehr wegzudenken. Sie werden zur Planung, Auslegung oder auch während des Betriebs von verfahrenstechnischen Anlagen verwendet. Heute steht eine Vielzahl kommerzieller Softwarepakete zur Verfügung, die auf die jeweiligen Bedürfnisse zugeschnitten sind. Während manche Simulatoren ganze chemische Produktionsanlagen abbilden können, sind andere auf einzelne Apparatetypen wie z.B. Destillationskolonnen zur Stofftrennung oder Reaktoren zur Stoffumwandlung spezialisiert. Die Modellierung und stationäre Simulation zählen zur Grundfunktionalität aller verfahrenstechnischen Simulatoren. Sie unterscheiden sich in Funktionen wie stationäre Optimierung, dynamische Simulation, dynamische Optimierung oder auch Betriebsoptimierung.

Die Engineering Division der Linde AG plant und baut schlüsselfertige verfahrenstechnische Anlagen und zählt mit über 1000 verfahrenstechnischen Patenten und 4000 abgeschlossenen Projekten zu den weltweit führenden Anbietern von Anlagen zur Wasserstoff- und Synthesegaserzeugung, Petrochemie, Erdgasverflüssigung und Luftzerlegung. Neben der optimalen Auslegung dieser Anlagen entsprechend der geforderten Prozessbedingungen, spielen Effizienz und Reduzierung der Energiekosten eine immer größere Rolle. Hierfür sind in der Planung robuste Simulationswerkzeuge und Optimierungsverfahren notwendig.

Seit über 25 Jahren kommt in der Engineering Division der Linde AG der hauseigene Prozesssimulator OPTISIM[®] zum Einsatz. In einem kontinuierlichen Weiterentwicklungsprozess wird er an die ständig wachsenden Anforderungen angepasst [25]. OPTISIM[®] ist ein gleichungsorientierter Prozesssimulator zur stationären und dynamischen Simulation sowie zur Optimierung von verfahrenstechnischen Anlagen. Er

wird sowohl zur Planung und Auslegung von neuen Anlagen, als auch zur Analyse bestehender Produktionsanlagen verwendet.

In dieser Arbeit werden neue Ansätze zur Optimierung von verfahrenstechnischen Anlagen für den Prozesssimulator OPTISIM[®] untersucht und getestet. In den folgenden Abschnitten dieses Kapitels wird die Modellierung und stationäre Simulation verfahrenstechnischer Prozesse in OPTISIM[®] skizziert und der seitherige Stand der Optimierung erläutert. Anschließend werden in Kapitel 2 die Anforderungen der Verfahrenstechnik an die Optimierung beschrieben und die in dieser Arbeit untersuchten Möglichkeiten erläutert. In Kapitel 3 wird die Innere-Punkte-Methode mit Filter-Liniensuche vorgestellt, sowie einige Konvergenzresultate für dieses Optimierungsverfahren aus der Literatur zusammengefasst. Anschließend wird der Beweis der globalen Konvergenz des Filterverfahrens für Innere-Punkte-Verfahren bei Störungen in den Gleichungsnebenbedingungen und deren Ableitungen betrachtet. In Kapitel 4 wird auf die Anbindung des Softwarepakets IPOPT an den Prozesssimulator OPTISIM[®] eingegangen, sowie numerische Ergebnisse präsentiert. Kapitel 5 zeigt die Ergebnisse zweier Bachelorarbeiten, die im Rahmen dieses Promotionsvorhabens durchgeführt wurden. Dabei wurde in OPTISIM[®] erstmals ein Ansatz zur globalen Optimierung implementiert und getestet.

1.1 Modellierung

Modelle von verfahrenstechnischen Anlagen oder Teilprozessen werden in **Flowsheets** beschrieben. Darin werden für alle relevanten Komponenten (Units) der Anlage Simulationsmodelle benannt und spezifiziert [24]. **Units** sind Apparate wie z.B. Wärmetauscher, Pumpen oder Ventile, aber auch Prozessschritte wie Phasentrennung oder einzelne thermodynamische Vorgänge. Im Flowsheet werden zudem **Verbindungsströme** für Material, Energie, Impuls und Information zwischen einzelnen Units angegeben und benannt, als auch Startwerte für die stationäre Simulation gesetzt. Durch die Definition der Verbindungsströme wird auch die Anordnung (**Topologie**) der Units im Flowsheet festgelegt.

1.1.1 Unitmodelle

Für jede Art von Unit ist ein verallgemeinertes Modell in der OPTISIM[®]-Modellbibliothek hinterlegt. Das Modell einer Unit $U^{(j)}$ besteht aus

- den Eingangsgrößen $u^{(j)} \in \mathbb{R}^{n_u^{(j)}}$ mit $n_u^{(j)} \in \mathbb{N}$,
- den internen Zustandsgrößen $\xi^{(j)} \in \mathbb{R}^{s^{(j)}}$ mit $s^{(j)} \in \mathbb{N}$ und

- den Ausgangsgrößen $y^{(j)} \in \mathbb{R}^{n_y^{(j)}}$ mit $n_y^{(j)} \in \mathbb{N}$, sowie
- der Modellfunktion $f^{(j)} : \mathbb{R}^{s^{(j)}+n_u^{(j)}} \rightarrow \mathbb{R}^{s^{(j)}}$.

Die Modellgleichungen

$$f^{(j)}(\xi^{(j)}, u^{(j)}) = 0, \quad (1.1)$$

formulieren die Massen-, Impuls- und Energiebilanzen sowie thermodynamische Gleichgewichte. Um die Modellgleichungen mit einem Newtonverfahren lösen zu können, muss das Gleichungssystem quadratisch sein. Das heißt die Anzahl der Gleichungen muss mit der der internen Zustandsvariablen übereinstimmen.

Die **Ausgangsgrößen** $y^{(j)}$ sind ein Teil der internen Zustandsvariablen $\xi^{(j)}$. Dieser Zusammenhang lässt sich auch mit Hilfe einer geeigneten Projektionsmatrix $H^{(j)} \in \mathbb{R}^{n_y^{(j)} \times s^{(j)}}$ formulieren (vgl. [35]).

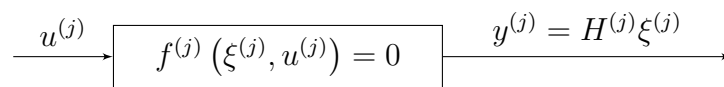


Abbildung 1.1 Aufbau des Modells der Unit $U^{(j)}$

Die **Eingangsgrößen** $u^{(j)}$ bestehen aus Eingangsströmen, Spezifikationen, konstanten Parametern und Designvariablen:

Spezifikationen sind Auswahlparameter von Modellvarianten, beispielsweise ob eine Unit durch Vorgabe des Austrittsdrucks oder des Druckverlusts spezifiziert wird. Je nachdem werden die Modellgleichungen dann entsprechend ausgewählt.

Konstante Parameter sind beispielsweise die Anzahl an Kolonnenböden oder die Anzahl der Diskretisierungspunkte in einem Wärmetauschermodell. Die konstanten Parameter und die Spezifikationen beeinflussen die Struktur der Modellgleichungen, d.h. sowohl die Anzahl der Gleichungen als auch die Funktionsvorschriften der Modellfunktion. Beide sind während der Simulation konstant.

Designvariablen sind Anlagenparameter wie Wärmetauscherfläche, Enddruck eines Verdichters oder Behältervolumen, die während der Laufzeit durch externe Berechnungsvorschriften oder Optimierungsverfahren verändert werden können, während der Auswertung der Modellfunktion jedoch konstant bleiben.

1.1.2 Stoffdaten

Für die Formulierung sowie die Wahl der geeigneten Modellgleichungen in einem Prozesssimulator werden Stoffdaten benötigt. Damit bezeichnet man thermodynamische Zustandsgrößen und Eigenschaften chemischer Stoffe wie z.B. Enthalpie, Molekulargewicht, Dichte, Siedepunkt, Viskosität, Dampfanteil oder Entropie. Zur Berechnung solcher Größen wird in OPTISIM® das Linde-eigene Stoffdatenprogramm GMPS (General Multi-phase Property System) verwendet. GMPS ist ein Stoffwertberechnungssystem, das seit ca. 1978 für die Anwendungen der Linde AG maßgeschneidert und ständig weiterentwickelt wird [36]. Das Programm enthält eine Vielzahl von Berechnungsmethoden zur Bestimmung von Phasengleichgewichten, Reinstoff- und Gemischeigenschaften, sowie Datenbanken, in denen Stoffwerte und Parameter für die einzelnen Berechnungsmethoden hinterlegt sind. Für jede Stoffeigenschaft stehen unterschiedliche Berechnungsmethoden zur Verfügung, die anwendungsspezifisch verwendet werden.

Bei mehrphasigen Gemischen muss zunächst das Phasengleichgewicht berechnet werden. Dieses besteht dann, wenn alle Phasen die gleiche Temperatur und den gleichen Druck haben und für jede Komponente die chemischen Potentiale in allen Phasen übereinstimmen [13]. Dabei ist in jeder Komponente das gesamte chemische Potenzial minimal. Das Phasengleichgewicht liefert für jede Komponente die Verteilung in die einzelnen Phasen. Für bestimmte Bedingungen formulieren Zustandsgleichungen den funktionalen Zusammenhang zwischen den thermodynamischen Zustandsgrößen. Aus der Lösung dieser meist nichtlinearen Gleichungen können dann die einzelnen Stoffwerte bestimmt werden.

1.1.3 Anlagenmodelle

Anhand der Topologie des Flowsheets sind die Verbindungsströme zwischen den einzelnen Units bekannt. Daraus lässt sich die Zuordnung von Ein- und Ausgangsströmen formal mit Hilfe von Koppelmatrizen $K^{(jl)}$ ausdrücken (vgl. [35]):

$$u_{stream}^{(j)} = K^{(jl)} y^{(l)} = K^{(jl)} H^{(l)} \xi^{(l)} \quad (1.2)$$

Die Verknüpfung der entsprechenden Ströme kann auf unterschiedliche Arten realisiert werden: Beispielsweise werden in Dymola [5] sogenannte Koppelgleichungen erzeugt, die dann zusammen mit den Modellgleichungen gelöst werden [41].

In OPTISIM® hingegen werden die Eingangsströme einer Unit durch die Ausgangsströme der topologisch vorgeschalteten Unit(s) substituiert. Für Eingangsströme, die von außerhalb der Modellgrenzen kommen (z.B. Feedströme für Energie und Material), gibt es in der OPTISIM®-Modellbibliothek eigene Quellunits [24].

Damit können die Eingangsströme aller Units durch Ausgangsströme einer jeweils anderen Unit und somit als Zustandsvariablen ausgedrückt werden.

Die Modellfunktion $f^{(j)}$ einer Unit $U^{(j)}$ kann in OPTISIM[®] folglich allein in Abhängigkeit von internen Zustandsgrößen $\xi^{(j)}$, $\xi^{(l)}$, usw. und den Designvariablen, die im Folgenden mit $p^{(j)} \in \mathbb{R}^{n_p^{(j)}}$ ($n_p^{(j)} \in \mathbb{N}$) bezeichnet werden, ausgedrückt werden.

Fasst man nun die Zustandsvariablen aller Units in $\xi \in \mathbb{R}^s$, die Designvariablen aller Units in $p \in \mathbb{R}^{n_p}$ und die Modellfunktionen aller Units in der Anlagenmodellfunktion $F : \mathbb{R}^s \times \mathbb{R}^{n_p} \rightarrow \mathbb{R}^s$ mit $s = \sum_j s^{(j)}$ und $n_p = \sum_j n_p^{(j)}$ zusammen, so erhält man aus den einzelnen Unit-Modellgleichungen (1.1) das Modellgleichungssystem

$$F(\xi, p) = 0 \quad (1.3)$$

für das gesamte Anlagenmodell. Dieses Gleichungssystem ist in der Regel nichtlinear und besitzt stets genauso viele Gleichungen wie Zustandsvariablen. Für jede Zustandsvariable gibt es also eine Modellgleichung, die ihr über eine Adressverwaltung σ zugeordnet werden kann. Um später, beispielsweise für die Optimierung, die Abhängigkeit der Modellfunktion von den Designvariablen p bestimmen zu können, wird (ähnlich wie für die Feedströme) für jede Designvariable p_i in einer Quellunit eine Zustandsvariable ξ_{pi} sowie die dazugehörige Modellgleichung

$$F_{\sigma(i)} = \xi_{pi} - p_i = 0 \quad (1.4)$$

erzeugt. In den übrigen Modellgleichungen, die von dieser Designvariable abhängen, wird anstelle von p_i dann ξ_{pi} verwendet, sodass jede Designvariable nur genau einmal im Modellgleichungssystem vorkommt.

1.2 Stationäre Simulation

Während der stationären Simulation werden die Modellgleichungen für fest gewählte Designvariablen p nach den Zustandsvariablen ξ gelöst. Hierfür gibt es hauptsächlich zwei Strategien:

1.2.1 Gleichungsorientierte Prozesssimulatoren

In einem gleichungsorientierten Prozesssimulator wie z.B. OPTISIM[®], DIVA [35], Dymola [5] oder gPROMS [7] wird das gesamte nichtlineare Anlagengleichungssystem (1.3) als Ganzes nach den Zustandsvariablen ξ gelöst.

Die Berechnung erfolgt in OPTISIM[®] durch ein inexaktes Newtonverfahren, wobei in jeder Iteration ein Newtonschritt $\Delta\xi$ durch Lösen des linearen Gleichungssystems

$$\frac{\partial F(\xi, p)}{\partial \xi} \Delta\xi = -F(\xi, p) \quad (1.5)$$

berechnet wird [24]. Die analytischen Ableitungen der Modellfunktion nach den Zustandsvariablen sind, soweit möglich, in den einzelnen Units hinterlegt. Die Jacobimatrix ist dünn besetzt, da die Modellgleichungen einer Unit nur von den internen Zustandsvariablen und den Eingangsströmen dieser Unit abhängen. Daher können zur Lösung von (1.5) effiziente Lösungsverfahren für Systeme mit dünnbesetzte Matrizen wie z.B. GMRES oder BiCG eingesetzt werden [24].

Die Modellfunktion F und folglich auch die Lösung ξ des Gleichungssystems (1.3) hängen von der Wahl der Designvariablen p ab. Die Zustandsvariablen, die die Modellgleichungen erfüllen, können daher auch als Parametrisierung $\xi(p)$ aufgefasst werden.

1.2.2 Sequentiell-modulare Prozesssimulatoren

Bei sequentiell-modularen Prozesssimulatoren wie z.B. UNISIM[®] Design [11], ProSimPlus[®] [10], Aspen Plus[®] [2], Aspen HYSYS[®] [1], CHEMCAD [3] oder Pro/II [9] werden die Modellgleichungen (1.1) jeder Unit einzeln gelöst. In manchen Fällen werden auch hier die Modellgleichungen mehrerer Units zu einem Subflowsheet zusammengefasst und gemeinsam gelöst, der Gesamtansatz bleibt jedoch sequentiell. Die Reihenfolge, in der die Modellgleichungen der einzelnen Units gelöst werden, entspricht bei der Mehrzahl der Simulatoren der Stromrichtung. Dadurch können die berechneten Ausgangsgrößen einer Unit entsprechend der Flowsheettopologie wie in (1.2) direkt als Eingangsgrößen an die nächste Unit weitergegeben werden.

Vorteil der sequentiell-modularen Methode ist zum einen, dass durch die Ergebnisse der vorhergehenden Unit stets Startwerte zur Lösung der Modellgleichungen der betrachteten Unit gegeben sind. Bei der gleichungsorientierten Methode hingegen müssen zu Beginn der Simulation erst geeigneten Startwerte gesucht werden. Zum anderen können für jeden Modelltyp die Eigenschaften der zugehörigen Modellgleichungen ausgenutzt und speziell angepasste Lösungsverfahren verwendet werden (z.B. [45]).

Nachteil des sequentiellen Ansatzes ist jedoch, dass Stromrückführungen (Recycles) iterativ berechnet werden müssen, was besonders bei großen Anlagen mit großen Recycles sehr aufwendig ist und enorme Rechenzeiten zur Folge hat.

Zudem können bei der gleichungsorientierten Methode analytische Ableitungen der Designvariablen nach den Zustandsvariablen berechnet werden (vgl. Abschnitt 1.3.1).

OPTISIM[®] ist ein gleichungsorientierter Simulator, da er für große Flowsheets mit vielen Units und großen Kreisläufen entwickelt wurde. Für spezielle Aufgabenstellungen ist jedoch auch ein Verfahren zur sequentiellen stationären Simulation implementiert.

1.3 Stationäre Optimierung in OPTISIM[®]

Nachdem in der stationären Simulation die Zustandsvariablen ξ für fest gewähltes p gelöst wurden, können nun innerhalb einer stationären Optimierung die Designvariablen p freigegeben werden. Diese Freiheitsgrade werden dann so bestimmt, dass eine Zielfunktion Φ minimiert und gegebene Nebenbedingungen erfüllt werden. Die Zielfunktion kann beispielsweise die Gesamtenergie oder den spezifischen Verbrauch einer Anlage, Investitionskosten oder auch eine Fehlerquadratsumme zur Anpassung eines Modells an Messdaten enthalten. Nebenbedingungen sind z.B. Produktreinheiten, ein Mindestabstand zwischen der kalten und der warmen Summenkurve eines Wärmetauschers zur Einhaltung des zweiten Hauptsatzes der Thermodynamik, oder die Begrenzung von Temperatur und Druck, um Feststoffausfall oder das Durchlaufen des kritischen Punktes zu verhindern.

1.3.1 Reelle Optimierung

Während einer reellen (kontinuierlichen) Optimierung werden Probleme der Form

$$\begin{aligned} \min_p \quad & \Phi(\xi(p)) \\ \text{u.d.N.} \quad & p_{\min} \leq p \leq p_{\max} \\ & \xi_{\min} \leq \xi_J(p) \leq \xi_{\max} \\ & c_{\min} \leq c(\xi(p)) \leq c_{\max} \end{aligned} \tag{1.6}$$

mit $\Phi : \mathbb{R}^{n_p} \rightarrow \mathbb{R}$, $c : \mathbb{R}^{n_p} \rightarrow \mathbb{R}^{m_2}$, betrachtet.

Die Auswertung von Zielfunktion und Nebenbedingungen erfolgt nur für stationäre Zustände, die die Modellgleichungen (1.3) erfüllen, daher wird ξ in Abhängigkeit von p angegeben.

Außer den Zustands- und Designvariablen werden in OPTISIM[®] während der Optimierung noch **Zwischengrößen** verwendet, die meist Kenngrößen eines Wärmetauschers

wie z.B. seine Übertragungsfähigkeit repräsentieren. Diese Größen hängen von den Zustandsvariablen $\xi(p)$ für einen bestimmten Wert von p ab und werden erst nach erfolgreicher Lösung des Anlagengleichungssystems (1.3) berechnet.

Die Zielfunktion Φ wird in Abhängigkeit von den Zustandsvariablen ξ und evtl. den eben beschriebenen Zwischengrößen formuliert. Selbst wenn die Zielfunktion eine Designvariable p_i enthalten soll, so wird die zugehörige Zustandsvariable ξ_{p_i} aus (1.4) verwendet. Die Nebenbedingungen sind Beschränkungen einerseits der Optimierungsvariablen p durch p_{\min} und $p_{\max} \in \mathbb{R}^{n_p}$ und einzelner Zustandsvariablen $\xi_J = (\xi_i)_{i \in J}$ mit $J \subset \{1, \dots, s\}$, $|J| = m_1$ durch ξ_{\min} und $\xi_{\max} \in \mathbb{R}^{m_1}$. Zudem können die oben beschriebenen Zwischengrößen oder auch Summen aus Zustandsvariablen als zusätzliche Nebenbedingungen, die hier mit c bezeichnet werden, durch c_{\min} und $c_{\max} \in \mathbb{R}^{m_2}$ beschränkt werden.

Die Lösung solcher Optimierungsprobleme erfolgt in OPTISIM® bislang durch SQP- oder SLP-Verfahren. Diese iterativen Verfahren benötigen in jeder Iteration die Auswertung von Zielfunktion und Nebenbedingungen sowie deren Ableitungen nach den Optimierungsvariablen p . Dadurch ist in jeder Iteration das Lösen des Anlagengleichungssystems (1.3) erforderlich. Aus dem Ergebnis dieser stationären Rechnung für einen aktuellen Wert der Optimierungsvariablen lassen sich dann die aktuellen Werte der Funktionen Φ und c sowie deren Ableitungen bestimmen.

Die Ableitungen können über den impliziten Funktionensatz berechnet werden. Dieser besagt, dass die Gleichung $F(\xi, p) = 0$ in einer Umgebung von (ξ, p) die Zustandsvariablen ξ als eindeutige, stetig differenzierbare Funktion von den Designvariablen p definiert. Die Abhängigkeit der Zustandsvariablen von p wurde bereits in Abschnitt 1.2 erwähnt und in der Formulierung des Optimierungsproblems verwendet. Für die Ableitungen folgt aus Gleichung (1.3)

$$\frac{\partial F(\xi, p)}{\partial \xi} \cdot \frac{d\xi}{dp} + \frac{\partial F(\xi, p)}{\partial p} = 0 \quad (1.7)$$

Die partiellen Ableitungen der Modellfunktionen nach den Designvariablen sind bekannt, da p_j nur in der ihr zugeordneten Gleichung (1.4) enthalten ist, daher gilt mit dem Kroneckersymbol δ_{ij}

$$\frac{\partial F_i(\xi, p)}{\partial p_j} = -\delta_{i\sigma(j)}. \quad (1.8)$$

Die partiellen Ableitungen der Modellfunktionen nach den Zustandsvariablen sind bereits aus Gleichung (1.5) bekannt. Folglich lassen sich die Ableitungen der Zustandsva-

riablen nach einer Optimierungsvariablen p_j aus folgendem linearem Gleichungssystem berechnen, wobei mit e_i der i . kanonische Einheitsvektor bezeichnet wird:

$$\frac{\partial F(\xi, p)}{\partial \xi} \cdot \frac{d\xi}{dp_j} = e_{\sigma(j)} \quad (1.9)$$

Dieses System muss für jede Optimierungsvariable p_j gelöst werden. Hierbei kann die Matrix und v.a. ihre LU-Zerlegung aus der Berechnung des letzten Newtonschrittes der vorangegangenen stationären Simulation (1.5) weiter verwendet werden.

Die Ableitung der Zielfunktion Φ nach den Designvariablen erhält man durch die Kettenregel als

$$\frac{d\Phi(\xi(p))}{dp} = \sum_i \frac{\partial \Phi(\xi)}{\partial \xi_i} \cdot \frac{\partial \xi_i}{\partial p}. \quad (1.10)$$

Die Ableitungen der Nebenbedingungen c nach den Optimierungsvariablen ergeben sich genauso aus den partiellen Ableitungen nach den Zustandsvariablen und den Ableitungen der Zustandsvariablen nach den Designvariablen.

$$\frac{dc(\xi(p))}{dp} = \sum_i \frac{\partial c(\xi)}{\partial \xi_i} \cdot \frac{d\xi_i}{dp}. \quad (1.11)$$

Die in OPTISIM[®] implementierten SQP-Verfahren enthalten Regularisierungsverfahren nach Powell [42], Tone [49] und Spellucci [47]. Diese Verfahren ermöglichen das Lösen des Optimierungsproblems, auch wenn die Nebenbedingungen anfangs verletzt sind oder durch die Linearisierung der Nebenbedingungen kein zulässiger Punkt mehr gefunden werden kann.

Außerdem ist eine Schrittweitenbeschränkung implementiert. Damit soll verhindert werden, dass die Auswertungspunkte für die stationäre Simulation zu weit vom aktuellen Punkt entfernt sind, was aufgrund der lokalen Konvergenz des Newtonverfahrens zu Konvergenzproblemen führen kann. Darauf wird in Kapitel 2 weiter eingegangen.

1.3.2 Gemischt-ganzzahlige Optimierung

In manchen Fällen sollen auch ganzzahlige Designvariablen $p^{\text{int}} \in \mathbb{Z}^{n_{p^2}}$, wie z.B. die Bodennummer eines Zulaufs oder Abzugs an einer Destillationskolonne, sowie ganzzahlige Nebenbedingungen, z.B. der Mindestbodenabstand zwischen zwei Kolonnenzuläufen, in die Optimierung einbezogen werden. In diesem Fall liegt dann folgendes nichtlineares gemischt-ganzzahliges Problem (MINLP) vor:

$$\begin{aligned}
& \min_{(p, p^{\text{int}})} && \Phi(\xi(p, p^{\text{int}})) \\
\text{u.d.N.} & p_{\min} \leq & p & \leq p_{\max} \\
& p_{\min}^{\text{int}} \leq & p^{\text{int}} & \leq p_{\max}^{\text{int}} \\
& \xi_{\min} \leq & \xi_J(p, p^{\text{int}}) & \leq \xi_{\max} \\
& c_{\min} \leq & c(\xi(p, p^{\text{int}})) & \leq c_{\max} \\
& c_{\min}^{\text{int}} \leq & c^{\text{int}}(\xi(p, p^{\text{int}})) & \leq c_{\max}^{\text{int}}
\end{aligned} \tag{1.12}$$

wobei nun die Zielfunktion $\Phi : \mathbb{R}^{n_p} \times \mathbb{Z}^{n_{p2}} \rightarrow \mathbb{R}$, die reellen Nebenbedingungen $c : \mathbb{R}^{n_p} \times \mathbb{Z}^{n_{p2}} \rightarrow \mathbb{R}^{m_2}$ und zusätzlich die ganzzahligen Nebenbedingungen $c^{\text{int}} : \mathbb{R}^{n_p} \times \mathbb{Z}^{n_{p2}} \rightarrow \mathbb{Z}^{m_3}$, sowie die ganzzahligen Schranken $p_{\min}^{\text{int}}, p_{\max}^{\text{int}} \in \mathbb{Z}^{n_{p2}}$ und $c_{\min}^{\text{int}}, c_{\max}^{\text{int}} \in \mathbb{Z}^{m_3}$ ($n_{p2}, m_3 \in \mathbb{N}$) gegeben sind.

Solche Optimierungsprobleme werden in OPTISIM® durch ein Outer-Approximation Verfahren nach Grossmann et al. [22, 50] gelöst. Aufgrund von Stabilitätsproblemen beim Umschalten zwischen verschiedenen Konfigurationen einer Destillationskolonne wurden spezielle Stabilisierungsmaßnahmen entwickelt [48]. In diesem Outer-Approximation Verfahren wird abwechselnd ein reelles Optimierungsproblem der Form (1.6) für feste Werte in den ganzzahligen Parametern p^{int} und ein linearisiertes gemischt-ganzzahliges Problem (MILP) gelöst.

Die Ableitungen der Zielfunktion und der Nebenbedingungen nach den ganzzahligen Optimierungsvariablen p^{int} lassen sich genauso bestimmen wie für die reellen Optimierungsvariablen p (siehe Abschnitt 1.3.1).

Die Lösung der linearen gemischt-ganzzahligen Probleme erfolgt über den GNU-Solver glpk [6].

Kapitel 2

Aufgabenstellung

2.1 Verfahrenstechnische Herausforderungen in der Optimierung

Viele Prozesssimulatoren betrachten nur einzelne Bauteile bzw. chemische Teilprozesse einer Anlage. Die Besonderheit von OPTISIM[®] ist, dass hiermit ganze Anlagen mit vielen unterschiedlichen Apparaten modelliert, simuliert und optimiert werden können. Dabei entstehen sehr große Systeme mit mehreren tausend Gleichungen und Zustandsvariablen. Die Lösung dieser Systeme beinhaltet aufgrund der Eigenschaften der Modellfunktionen Herausforderungen, welche sich oftmals als Konvergenzprobleme bei der stationären Simulation äußern. Können die Modellgleichungen (1.3) während einer Optimierung nicht gelöst werden, so kann die Optimierung, wie sie in Abschnitt 1.3 beschrieben wurde, nicht fortgeführt werden, da die aktuellen Werte für die Zielfunktion, die Nebenbedingungen sowie deren Ableitungen nicht vorliegen.

Im Folgenden werden mögliche Gründe für die Konvergenzprobleme bei der Berechnung des stationären Modells betrachtet:

2.1.1 Nichtlinearität

Die Modellfunktion F ist hochgradig nichtlinear und das Newtonverfahren besitzt nur lokale Konvergenzeigenschaften. Der Erfolg einer stationären Simulation hängt daher sehr stark von der Wahl des Startwerts ab. Aus diesem Grund wurden in OPTISIM[®] an einigen Stellen Homotopieverfahren (vgl. z.B. [20, Abschnitt 4.4.2]) implementiert. Bei einer Änderung der Designvariablen p beispielsweise wird diese Änderung in mehrere Schritte unterteilt und das Modellgleichungssystem so nur langsam verändert. Die Simulation wird in jedem Schritt durchgeführt, wobei die Lösung einer Simulation als Startwert für die nächste verwendet wird, solange bis man schließlich zur Lösung des Systems für die

gewünschten Designvariablen gelangt. Ähnlich wirkt die Schrittweitenbeschränkung in der Optimierung (vgl. Seite 25). Dabei wird die maximale Schrittweite in der Liniensuche beschränkt, um die Auswertung der Zielfunktion und Nebenbedingungen in den Testpunkten der Liniensuche zu erreichen.

2.1.2 Unstetigkeiten

Die Struktur und Funktionsvorschrift der Modellfunktion F hängt nicht nur, wie in Abschnitt 1.1.1 beschrieben, von einigen Konstanten ab, sondern auch von den Vorzeichen der Werte $q(\xi, p)$ sogenannter Schaltfunktionen $q : \mathbb{R}^{s+n_p} \rightarrow \mathbb{R}$ [24]. Die Schaltfunktionen dienen dazu Größen, die nicht über die gesamte Rechnung hinweg konstant sind, zu berücksichtigen und zwischen unterschiedlichen Zuständen des Systems, die durch unterschiedliche Gleichungen modelliert werden, zu wechseln. Damit können beispielsweise

- unstetige Kennlinien eines Reglers,
- das Auftreten oder Verschwinden einer Phase (z.B. wenn beim Erhitzen der Siedepunkt eines Gemisches erreicht wird und das Gemisch nicht mehr nur flüssig, sondern teilweise flüssig und teilweise gasförmig ist),
- abschnittsweise stetige physikalische Stoffeigenschaften (z.B. die Dichte an Phasengrenzflächen mehrphasiger Strömungen oder die Enthalpie eines Reinstoffs als Funktion in der Temperatur am Tau- oder Siedepunkt, wie in den Abbildungen 2.1 und 2.2 dargestellt),
- die geometrische Beschränkung eines Simulationsmodells (z.B. beim Überlaufen oder Leerlaufen eines Behälters) oder
- der Übergang von laminarer in turbulente Rohrströmung beim Überschreiten der kritischen Reynoldszahl

modelliert werden.

Jedes Mal wenn eine Schaltfunktion ihr Vorzeichen wechselt, d.h. eine Modellfunktion ausgetauscht wird, entsteht eine Unstetigkeit. Die Unstetigkeitsstellen können jedoch nicht a priori bestimmt werden. Die Unstetigkeiten in den Modellfunktionen führen meist auch zu Unstetigkeiten in den Ableitungen.

In OPTISIM[®] wird zwischen folgenden Arten von Unstetigkeiten unterschieden [24]:

1. Änderungen in der Modellfunktion F ohne Auswirkung auf die Jacobimatrix $\frac{\partial F(\xi, p)}{\partial \xi}$
2. Änderungen in den Werten der dünnbesetzten Jacobimatrix der Modellfunktion unter Beibehaltung ihrer Belegungsstruktur

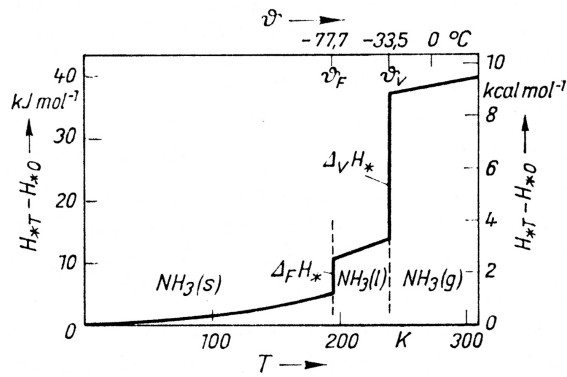


Abbildung 2.1 Molare Enthalpiezunahme von Ammoniak in Abhängigkeit von der Temperatur, aus: [37, S.107]

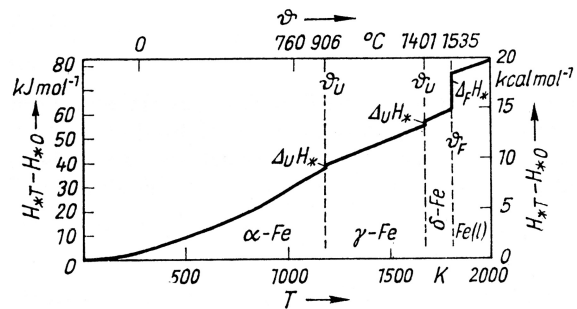


Abbildung 2.2 Molare Enthalpiezunahme von Eisen in Abhängigkeit von der Temperatur, aus: [37, S.107]

3. Änderungen in den Abhängigkeiten der Modellfunktion und damit in der Belegungsstruktur der Jacobimatrix der Modellfunktion
4. Änderungen in der Anzahl der Gleichungen und Zustandsvariablen

Beispiel 2.1 (Phasenwechsel innerhalb eines Wärmetauschers)

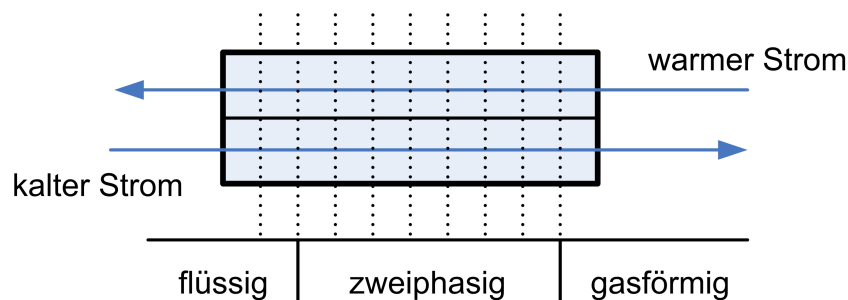


Abbildung 2.3 Modell eines Wärmetauschers

Betrachtet wird das detaillierte Modell eines Gegenstromwärmetauschers mit zwei Strömen (siehe Abb. 2.3). Die Eintrittstemperaturen der beiden Ströme seien vorgegeben, wobei der kalte Strom am Eintritt flüssig und am Austritt gasförmig sei. Die Diskretisierung des Wärmetauschers erfolgt entlang seiner Länge [24]. An einem dieser Diskretisierungspunkte überschreitet die Temperatur des kalten Stroms erstmals die Siedetemperatur und an einem Punkt die Tautemperatur. Vor dem ersten Phasenwchelpunkt wird der kalte Strom mit den Eigenschaften eines flüssigen Stroms behandelt, zwischen den beiden Punkten (falls diese nicht zusammenfallen) als zweiphasiger Strom und danach

als gasförmiger Strom. Verändert man nun die Strommenge des warmen Stroms, so verschieben sich diese Phasenwechsellpunkte. Dadurch ändert sich dann auch die Struktur der Jacobimatrix der Modellfunktion.

2.1.3 Approximation von Ableitungen

Wie in Abschnitt 1.1.2 erwähnt, beschreiben Zustandsgleichungen den funktionalen Zusammenhang von Stoffeigenschaften unter bestimmten Bedingungen. Für manche Stoffwerte steht jedoch unter gewissen Bedingungen keine Zustandsgleichung in geschlossener Form zur Verfügung. In diesem Fall müssen die Stoffwerte heuristisch bestimmt werden. Dadurch liegen keine analytischen Ableitungen vor. Diese werden dann beispielsweise durch finite Differenzen approximiert.

2.1.4 Singularitäten

In manchen Fällen kann eine Änderung der Designvariablen p dazu führen, dass die Jacobimatrix der Modellfunktion singular wird. Dann ist die Lösung des linearen Gleichungssystems (1.5) nicht mehr möglich.

Beispiel 2.2 (Druck spezifizierter Kreislauf)

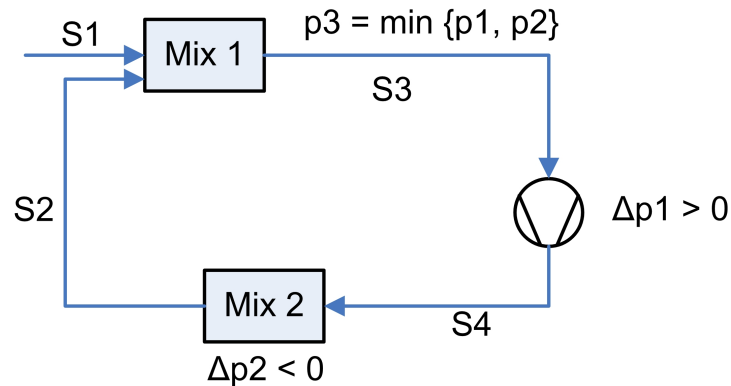


Abbildung 2.4 Beispiel für Singularität

Der Kreislauf aus Abbildung 2.4 kann in der Designrechnung über Druckangaben spezifiziert werden. Der Feedstrom $S1$ wird in der ersten Mixunit mit dem Strom $S2$, der aus einer zweiten Mixunit stammt, gemischt. $S1$ habe den Druck $p1$ und $S2$ den Druck $p2$. Der erste Mix sei so spezifiziert, dass der Austrittsstrom $S3$ den Druck $p3 = \min\{p1, p2\}$ hat. Der Verdichter erhöht den Druck anschließend auf $p4 = p3 + \Delta p1$ mit $\Delta p1 > 0$ und

der zweite Mix verringert den Druck wieder auf $p_2 = p_4 + \Delta p_2$ mit $\Delta p_2 < 0$. Dazwischen können noch weitere Prozessschritte liegen, die den Druck jedoch nicht verändern. Eine typische Optimierungsaufgabe lautet dann, die Verdichterleistung Δp_1 unter Freigabe des Drucks p_1 zu minimieren. Wird dabei der Druck p_1 so stark erhöht bzw. Δp_1 so stark verringert, dass $p_2 < p_1$ ist, so gilt $p_3 = p_2$ und das System ist unterbestimmt, d.h. die Jacobimatrix der Modellgleichungen ist singulär und p_1 hat keinen Einfluss mehr auf das System.

2.1.5 Grenzen des Modells

Ein gegebenes Modell ist nur innerhalb bestimmter Grenzen gültig. Daher gibt es nicht zu jeder Wahl der Designvariablen p Zustandsvariablen ξ , die die entsprechenden Modellgleichungen lösen.

Beispiel 2.3 (Überlaufender Kreislauf)

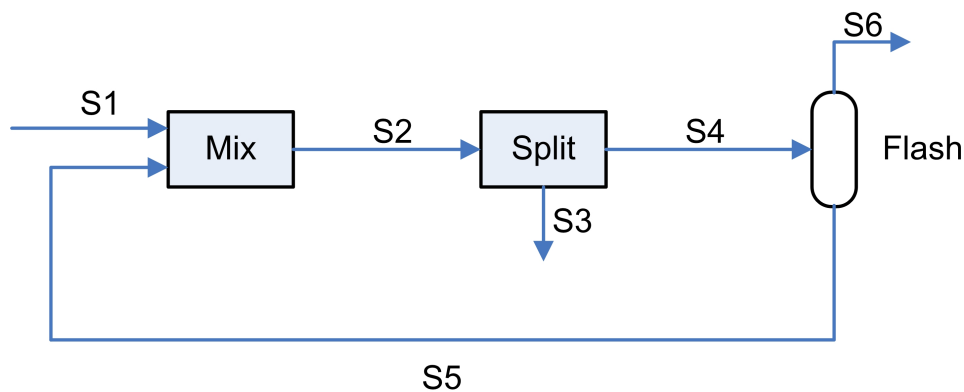


Abbildung 2.5 Beispiel für Grenze der Modellierung

In den Kreislauf, der in Abbildung 2.5 dargestellt ist, wird ein Strom S_1 mit zwei Komponenten x_1 und x_2 eingespeist. Der Mix mischt den Feedstrom mit dem Rückfluss S_5 eines Flashes. Im Split kann überschüssiges Material bis zu einer bestimmten Menge abfließen. Der Reststrom S_4 wird in den Flash, der auf Druck und Dampfanteil spezifiziert wurde, geleitet. Dort wird oben der Strom S_6 abgezogen, während der untere Strom S_5 in den ersten Mix zurückgeführt wird. Nun kann es passieren, dass aufgrund der gegebenen Temperatur und des Drucks, Strom S_6 nur die Komponente x_1 enthält. Ist nun der Ausfluss S_3 im Split zu klein, so „läuft der Kreislauf über“, d.h. der Anteil an der Komponente x_2 wird groß, der Strom S_6 wird klein und die Materialbilanz kann für Komponente x_2 nicht mehr erfüllt werden. Dem System wird dann mehr Material

zugeführt als entnommen. Die thermodynamische Formulierung der Austrittsmenge der zweiten Komponente im oberen Strom des Flashs $x_2(S6) = f(x_1, x_2, p, T)$ kann den zur Erfüllung der Materialbilanz notwendigen Wert dann nicht mehr erreichen.

2.1.6 Physikalische Grenzen

Werden physikalische Grenzen überschritten, so kann die Modellfunktion F nicht mehr ausgewertet werden. Beispielsweise können die Stoffdatenfunktionen für negative Werte des Drucks nicht ausgewertet werden. Das Anlagengleichungssystem (1.3) kann an einem solchen Punkt nicht gelöst werden.

2.2 Neue Ansätze für die Optimierung

2.2.1 Simultane Optimierung

Um während einer Optimierung die aktuellen Werte der Zielfunktion und der Nebenbedingungen für einen bestimmten Wert der Designvariablen p zu erhalten, müssen, wie in Abschnitt 1.3 beschrieben, jedes Mal das stationäre Modellgleichungssystem (1.3) nach den Zustandsvariablen ξ gelöst, sowie die benötigten Zwischengrößen berechnet werden. Folglich wird bei diesem Ansatz, der auch als sequentielle (oder verschachtelte [18]) Optimierung bezeichnet wird, in jeder Iteration des Optimierungsverfahrens das Gleichungssystem mindestens einmal gelöst. Falls die Änderungen in p zu groß sind, muss das Gleichungssystem über das in Abschnitt 2.1.1 vorgestellte Homotopieverfahren mehrfach gelöst werden. Ebenso sind während einer Liniensuche meist mehrere Auswertungen für verschiedene Schrittweiten innerhalb einer Optimierungsiteration nötig. Dies führt einerseits zu einem großen Rechenaufwand, andererseits kann die Optimierung nur fortgeführt werden, wenn die stationäre Simulation ein Ergebnis liefert.

Um die Konvergenzprobleme der stationären Simulation während der Optimierung zu umgehen, wird in dieser Arbeit der simultane Ansatz für die in Abschnitt 1.3 vorgestellte Optimierungsaufgabe betrachtet. Bei der simultanen Optimierung werden die Modellgleichungen (1.3) nicht in jeder Optimierungsiteration vollständig gelöst, sondern als Nebenbedingungen in die Optimierung mit eingebunden. Die Optimierungsvariablen bestehen dann aus den Zustandsvariablen ξ und den Designvariablen p . Lässt man nun für unbeschränkte Variablen die Grenzen $\pm\infty$ zu, so erhält man anstelle des in Abschnitt 1.3.1 beschriebenen Optimierungsproblems (1.6) folgendes, ungleich größeres Optimierungsproblem:

$$\begin{aligned}
 & \min_{(\xi, p)} \quad \Phi(\xi, p) \\
 \text{u.d.N.} \quad & p_{\min} \leq p \leq p_{\max} \\
 & \xi_{\min} \leq \xi \leq \xi_{\max} \\
 & c_{\min} \leq c(\xi, p) \leq c_{\max} \\
 & F(\xi, p) = 0
 \end{aligned} \tag{2.1}$$

Hierbei sind während der Optimierung nur Funktionsauswertungen von $F(\xi, p)$ sowie deren Ableitungen notwendig, die Gleichungen (1.3) müssen jedoch erst am Ende der Rechnung erfüllt sein. Die Zielfunktion Φ und die Nebenbedingungen c sind nun in Abhängigkeit von ξ und p formuliert, da der Zusammenhang $\xi(p)$ nicht mehr gegeben ist.

Die in Abschnitt 1.3.1 beschriebenen Zwischengrößen werden bei der sequentiellen Optimierung erst nach Lösung des stationären Gleichungssystems (1.3) berechnet. Dies ist während einer simultanen Optimierung nicht möglich, da die Modellgleichungen nicht in jeder Iteration erfüllt sind. Allerdings steht in der OPTISIM[®]-Modellbibliothek eine Unit zur Verfügung, die die meisten dieser in der Optimierung verwendeten Größen berechnet und als Ausgangsgrößen bereit stellt. Damit sind diese Größen Zustandsvariablen und müssen nicht länger separat betrachtet werden.

Die Zielfunktion Φ ist entweder eine gewichtete Summe von Zustandsvariablen oder eine gewichtete Fehlerquadratsumme aus Zustandsvariablen und Konstanten. Als zusätzliche Nebenbedingungen c können zudem Summen aus Zustandsvariablen beschränkt werden.

In diesem Abschnitt sowie auch den beiden folgenden Kapiteln wird nur das kontinuierliche nichtlineare Problem aus Abschnitt 1.3.1 mit reellen Optimierungsvariablen betrachtet. Liegt ein gemischt-ganzzahliges Optimierungsproblem vor, so wird weiterhin das Outer-Approximation Verfahren verwendet, währenddessen für verschiedene Werte der ganzzahligen Variablen wiederum reelle Probleme gelöst werden. Damit können auch in diesem Fall die hier gemachten Überlegungen eingesetzt werden.

Fasst man nun die Optimierungsvariablen zu $x = \begin{pmatrix} \xi \\ p \end{pmatrix}$ zusammen und setzt entsprechend $x_{\min} = \begin{pmatrix} \xi_{\min} \\ p_{\min} \end{pmatrix}$ und $x_{\max} = \begin{pmatrix} \xi_{\max} \\ p_{\max} \end{pmatrix}$, so erhält man das Optimierungsproblem

$$\begin{aligned} \min_x \quad & \Phi(x) \\ \text{u.d.N.} \quad & x_{\min} \leq x \leq x_{\max} \\ & c_{\min} \leq c(x) \leq c_{\max} \\ & F(x) = 0 \end{aligned} \tag{2.2}$$

Der simultane Optimierungsansatz wurde bereits unter verschiedenen Bezeichnungen in unterschiedlichen Bereichen angewendet: Biegler und Hughes [14] nutzten ihn beispielsweise unter der Bezeichnung „infeasible path“ für sequentiell-modulare Prozesssimulatoren, Schmid und Biegler [45] in Verbindung mit dem Prozesssimulator SEPSIM [12], Haftka [32] für die Optimierung von Modellen zur Fachwerk- und Strukturberechnung, Biegler, Cervantes und Wächter [15] für die dynamische Optimierung einzelner verfahrenstechnischer Prozesse oder Biegler und Wächter [18] unter der Bezeichnung „SAND“ ebenfalls für verfahrenstechnische Anwendungen.

Bei der Optimierung mit partiellen Differentialgleichungen als Nebenbedingungen wird dieser Ansatz auch als „one-shot“ Methode bezeichnet.

Zur Lösung des Optimierungsproblems (2.2) ist ein robustes und v.a. für große Systeme und dünnbesetzte Matrizen geeignetes Optimierungsverfahren notwendig. Biegler verwendete hierfür in [45, 18] das rSQP-Verfahren. Dies eignet sich besonders bei bestehenden Simulatoren, die zur Lösung der Modellgleichungen das Newtonverfahren verwenden, da die vorhandenen Routinen zur Berechnung eines Newtonschritts weiterverwendet werden können. Dieses Verfahren wurde später durch die Innere-Punkte-Methode und ihre Implementierung IPOPT abgelöst (vgl. [18, 15]). Dadurch muss kein quadratisches Teilproblem gelöst werden, das mit seiner Strategie der aktiven Menge besonders bei vielen Ungleichungsnebenbedingungen doch sehr rechenzeitaufwändig werden kann.

Die Innere-Punkte-Methode soll nun auch bei der simultanen Optimierung in OPTISIM[®] zum Einsatz kommen.

Betrachtet man die Formulierung des Optimierungsproblems des simultanen Ansatzes (2.2) unter den in Abschnitt 2.1.2 und 2.1.3 beschriebenen Unstetigkeiten und Approximationen, so erhält man folgende Eigenschaften: In der Modellfunktion $F(\xi, p)$ und folglich auch in ihren Ableitungen $\frac{\partial F}{\partial \xi}$ treten an manchen Stellen Unstetigkeiten bzw. Störungen auf. Dies hat zur Folge, dass im hier betrachteten Optimierungsproblem die beschreibende Funktion der Gleichungsnebenbedingung sowie deren Ableitungen nicht überall exakt auswertbar sind. Die Zielfunktion hingegen kann immer exakt bestimmt werden, da sie eine Summe oder Fehlerquadratsumme von Optimierungsvariablen ist.

In Kapitel 3 werden zunächst die Innere-Punkte-Methode sowie einige Konvergenzresultate dieses Verfahrens, wie es in IPOPT implementiert wurde, vorgestellt. Anschließend wird die Konvergenz der dort verwendeten Filter-Liniensuche für gestörte Gleichungsnebenbedingungen untersucht. Dabei werden bestimmte Forderungen an die Störungen in den Nebenbedingungen und deren Ableitungen gestellt, damit die Konvergenz noch gezeigt werden kann. Diese Störungen sollen einen Teil der Unstetigkeiten sowie den durch die Approximationen verursachten Fehler modellieren, wie sie sich zum Beispiel durch Sprünge in der Modellfunktion oder auch nur in deren Ableitungen beim Wechsel einer Modellfunktion ergeben.

2.2.2 Globaler Ansatz

Bei der optimalen Auslegung einer verfahrenstechnischen Anlage ist es wichtig, das globale Minimum des betrachteten Optimierungsproblems (1.6) bzw. (1.12) zu finden. Die gängigen Verfahren zur Optimierung unter Nebenbedingungen, wie z.B. das SQP- oder das Innere-Punkte-Verfahren, liefern unter bestimmten Voraussetzungen globale Konvergenz gegen Kandidaten für optimale Punkte. Ob eine gefundene Lösung jedoch wirklich die globale Lösung unter allen zulässigen Punkten ist, wird dabei nicht geprüft [40, Kapitel 1]. Da die globale Konvergenz zudem nicht für alle in Abschnitt 2.1.2 beschriebenen Unstetigkeiten erwartet werden kann, ist es sinnvoll, einen globalen Ansatz zu betrachten.

Im Übersichtsartikel von Neumaier [38] werden verschiedene globale Optimierungsverfahren vorgestellt. Die Gittermethode ist die einfachste Form der vollständigen globalen Suche (vgl. [38, Abschnitt 1], das heißt bei exakter Rechnung wird das globale Minimum sicher gefunden. Dabei wird der zulässige Bereich der Optimierungsvariablen durch ein Gitter unterteilt, dessen Zellen wiederum in ein feineres Gitter unterteilt werden. In jedem Punkt des feinen Gitters wird die Zielfunktion ausgewertet und der Punkt mit dem besten Wert der Zielfunktion innerhalb einer groben Zelle bestimmt. In diesen Punkten wird dann jeweils eine lokale Optimierung gestartet. Das globale Minimum wird zwar so garantiert gefunden, die Rechenzeit wächst jedoch exponentiell mit der Anzahl der Optimierungsvariablen und ist daher nur für ein oder zwei Variablen sinnvoll einsetzbar [38]. Beschleunigt wird dieses Verfahren beispielsweise durch wenige, zufällig gewählte Startpunkte. Dadurch wird das Verfahren, das in [38, Abschnitt 7] als „multiple random start“ bezeichnet wurde, zwar unvollständig, das heißt es ist nicht sichergestellt, dass wirklich das globale Minimum gefunden wird. Dennoch ist diese Vorgehensweise bei Optimierungen in der Prozesssimulation durchaus gängig [23].

Anstelle der zufällig gewählten Startwerte wurde es hier als sinnvoll erachtet, die Kenntnisse der Anwender über die Zusammenhänge der Optimierungsparameter innerhalb der Anlage auszunutzen und vorgegebene Startwerte zu verwenden. Auf die Umsetzung dieses Verfahrens wird in Kapitel 5 eingegangen.

Kapitel 3

Innere-Punkte-Methode und IPOPT

3.1 Einige Definitionen vorab

Bevor in den nächsten Abschnitten das Innere-Punkte-Verfahren vorgestellt wird, werden der Vollständigkeit halber hier noch einige Grundwerkzeuge der Optimierung definiert, wie sie beispielweise in [29, 40] eingeführt werden.

Für ein restringiertes nichtlineares Optimierungsproblem der Form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{u.d.N.} \quad & c^{\mathcal{E}}(x) = 0 \\ & c^{\mathcal{I}}(x) \geq 0 \end{aligned} \tag{3.1}$$

mit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $c^{\mathcal{E}} : \mathbb{R}^n \rightarrow \mathbb{R}^{m_{\mathcal{E}}}$ und $c^{\mathcal{I}} : \mathbb{R}^n \rightarrow \mathbb{R}^{m_{\mathcal{I}}}$, werden folgende Funktionen und Bedingungen definiert:

Definition 3.1 (Lagrange-Funktion)

Die Funktion $\mathcal{L} : \mathbb{R}^n \times \mathbb{R}^{m_{\mathcal{E}}} \times \mathbb{R}^{m_{\mathcal{I}}} \rightarrow \mathbb{R}$,

$$\mathcal{L}(x, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = f(x) + \sum_{i=1}^{m_{\mathcal{E}}} \lambda_i^{\mathcal{E}} c_i^{\mathcal{E}}(x) - \sum_{i=1}^{m_{\mathcal{I}}} \lambda_i^{\mathcal{I}} c_i^{\mathcal{I}}(x)$$

heißt *Lagrange-Funktion des restringierten Optimierungsproblems* (3.1). Die Komponenten von $\lambda^{\mathcal{E}}$ und $\lambda^{\mathcal{I}}$ heißen *Lagrange-Multiplikatoren*.

Definition 3.2 (Logarithmische Barrierefunktion)

Die Funktion $B : \mathbb{R}^n \times (0, \infty) \rightarrow \mathbb{R}$,

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^{m_{\mathcal{I}}} \ln c_i^{\mathcal{I}}(x) \tag{3.2}$$

heißt *logarithmische Barrierefunktion des ungleichungsrestringierten Optimierungsproblems* (3.1) mit dem Barriereparameter $\mu > 0$.

Eine Barrierefunktion ist nur im zulässigen Bereich der Ungleichungsbedingungen definiert. Ein rein ungleichungsrestringiertes Optimierungsproblem lässt sich als unrestringiertes Problem mit einer Barrierefunktion $B(x, \mu)$ als Zielfunktion ausdrücken. Genauso lässt sich ein ungleichungs- und gleichungsrestringiertes Problem als rein gleichungsrestringiertes Problem formulieren:

Definition 3.3 (Barriereproblem)

Sind eine Barrierefunktion $B(x, \mu)$, die die Ungleichungsrestriktionen $c^{\mathcal{I}}$ berücksichtigt, und ein Barriereparameter $\mu > 0$ gegeben, so lautet das zu (3.1) gehörige Barriereproblem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & B(x, \mu) \\ \text{u.d.N.} \quad & c^{\mathcal{E}}(x) = 0. \end{aligned} \tag{3.3}$$

Definition 3.4 (KKT-Bedingungen und KKT-Punkt)

Die *K(arush-)K(uhn-)T(ucker)-Bedingungen* des Optimierungsproblems (3.1) lauten

$$\begin{aligned} (i) \quad & \nabla_x \mathcal{L}(x, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) = \nabla f(x) + \sum_{i=1}^{m_{\mathcal{E}}} \lambda_i^{\mathcal{E}} \nabla c_i^{\mathcal{E}}(x) - \sum_{i=1}^{m_{\mathcal{I}}} \lambda_i^{\mathcal{I}} \nabla c_i^{\mathcal{I}}(x) = 0 \\ (ii) \quad & c^{\mathcal{E}}(x) = 0 \\ (iii) \quad & \forall_{i=1, \dots, m_{\mathcal{I}}} \quad \lambda_i^{\mathcal{I}} \geq 0 \quad \wedge \quad c_i^{\mathcal{I}}(x) \geq 0 \quad \wedge \quad \lambda_i^{\mathcal{I}} c_i^{\mathcal{I}}(x) = 0. \end{aligned} \tag{3.4}$$

Jeder Vektor $(x, \lambda^{\mathcal{E}}, \lambda^{\mathcal{I}}) \in \mathbb{R}^{n+m_{\mathcal{E}}+m_{\mathcal{I}}}$, der die KKT-Bedingungen erfüllt, heißt *KKT-Punkt*.

Definition 3.5 (Regularitätsbedingung von Mangasarian-Fromovitz)

Sei $x \in \mathbb{R}^n$ ein zulässiger Punkt von (3.1), d.h. alle Nebenbedingungen sind erfüllt, und sei $\mathcal{A} = \{i \mid c_i^{\mathcal{I}}(x) = 0\}$ die Indexmenge der aktiven Ungleichungsnebenbedingungen. Der Punkt x genügt den Regularitätsbedingung von Mangasarian-Fromovitz (MFCQ), wenn gilt:

(i) Die Gradienten der Gleichungsrestriktionen $\nabla c_i^{\mathcal{E}}(x)$, $i \in \{1, \dots, m_{\mathcal{E}}\}$ in x sind linear unabhängig.

(ii) Es existiert ein Vektor $d \in \mathbb{R}^n$ mit

$$\nabla c_i^{\mathcal{E}}(x)^T d = 0 \quad \text{für } i \in \{1, \dots, m_{\mathcal{E}}\} \quad \text{und} \quad \nabla c_i^{\mathcal{I}}(x)^T d > 0 \quad \text{für } i \in \mathcal{A}.$$

Zwischen den KKT-Bedingungen und den MFCQ-Bedingungen lässt sich folgender Zusammenhang herleiten (vgl. z.B. [29, Satz 2.39]):

Satz 3.1 *Sei x^* ein lokales Minimum von (3.1), das die MFCQ-Bedingungen erfüllt. Dann existieren Lagrange-Multiplikatoren $\lambda^{\mathcal{E}^*} \in \mathbb{R}^{m_{\mathcal{E}}}$ und $\lambda^{\mathcal{I}^*} \in \mathbb{R}^{m_{\mathcal{I}}}$ sodass $(x^*, \lambda^{\mathcal{E}^*}, \lambda^{\mathcal{I}^*})$ ein KKT-Punkt von (3.1) ist.*

3.2 Das Innere-Punkte-Verfahren

Die Innere-Punkte-Methode erzielte in den 1980er Jahren ihren Durchbruch, nachdem der indische Mathematiker Narendra Karmarkar in [33] die polynomiale Laufzeit für lineare Probleme bewiesen hatte. Das Verfahren wurde dann für nichtlineare Probleme erweitert. Hierbei sei auf den Artikel von Forsgren, Gill und Wright [27] verwiesen, in dem die historische Entwicklung, sowie verschiedene Ansätze für die Innere-Punkte-Methode und die Konvergenz des Barriere-Verfahrens zusammengefasst wurden. In den letzten Jahren wurden verschiedene Optimierungssoftwarepakete, die ein Innere-Punkte-Verfahren enthalten, entwickelt. Das wohl bekannteste und erfolgreichste ist IPOPT [8, 54], das von Wächter und Biegler entwickelt wurde. Dieses Paket ist im Internet über COIN-OR [4] unter der „Eclipse Public License“ (EPL) auch für kommerzielle Anwendungen frei verfügbar und wurde für unzähligen Problemstellungen verwendet. Eine Liste der Anwendungen ist unter der genannten Homepage verfügbar. Man kann das Innere-Punkte-Verfahren daher als „state of the art“ Lösungsmethode für reelle, nichtlineare Optimierungsprobleme bezeichnen.

Im Folgenden wird die Innere-Punkte-Methode, insbesondere wie sie in IPOPT implementiert wurde, an Hand von [27, 40, 51, 52, 53, 54] vorgestellt.

Gegeben sei ein Optimierungsproblem der Form

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{u.d.N.} \quad & x^L \leq x \leq x^U \\ & c^L \leq c(x) \leq c^U \end{aligned} \tag{3.5}$$

mit $f : \mathbb{R}^n \rightarrow \mathbb{R}$, $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$, $x^L \in [-\infty, \infty)^n$, $x^U \in (-\infty, \infty]^n$, $c^L \in [-\infty, \infty)^m$, $c^U \in (-\infty, \infty]^m$, das durch die Setzung von $c_i^L = c_i^U$ auch Gleichungsnebenbedingungen enthalten kann. Da Ungleichungsnebenbedingungen mit Hilfe von Schlupfvariablen durch Gleichungsnebenbedingungen unter Beschränkung der Schlupfvariablen auf nicht-negative Werte ersetzt werden können, wird im Folgenden das Optimierungsproblem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{u.d.N.} \quad & x^L \leq x \leq x^U \\ & c(x) = 0 \end{aligned} \tag{3.6}$$

betrachtet. Des Weiteren werden $I_L = \{i \mid x_i^L \neq -\infty\}$ und $I_U = \{i \mid x_i^U \neq \infty\}$ mit $I_L, I_U \subset \{1, \dots, n\}$ als die Indextmengen der beschränkten Optimierungsvariablen bezeichnet.

$A(x) = (\nabla c_1(x), \dots, \nabla c_m(x)) = \left(\frac{\partial c_i(x)}{\partial x_j} \right)_{ji}$ ist die transponierte Jacobimatrix der Gleichungsnebenbedingungen $c(x)$.

3.2.1 Motivation

Die Innere-Punkte-Methode kann auf zwei verschiedene Arten motiviert werden (vgl. [27, 40, 51]): zum einen durch das Lösen einer Folge von Barriereproblemen mit kleiner werdendem Barriereparameter, zum anderen durch wiederholtes Lösen von gestörten KKT-Bedingungen für kleiner werdende Störung in der primal-dualen Formulierung.

Mit der Barrierefunktion

$$B(x, \mu) = f(x) - \mu \sum_{i \in I_L} \ln(x_i - x_i^L) - \mu \sum_{i \in I_U} \ln(x_i^U - x_i) \tag{3.7}$$

und einem Barriereparameter $\mu > 0$ erhält man aus (3.6) das Barriereproblem

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & B(x, \mu) \\ \text{u.d.N.} \quad & c(x) = 0 \end{aligned} \tag{3.8}$$

Die KKT-Bedingungen dieses Barriereproblems lauten dann:

$$\begin{aligned} \nabla_x B(x, \mu) + A(x)\lambda &= 0 \\ c(x) &= 0 \end{aligned} \tag{3.9}$$

Das System (3.9) ist jedoch nicht wohldefiniert, da der Gradient der Barrierefunktion an der Lösung x^* nicht definiert ist, falls dort eine Beschränkung von x , beispielsweise $x_i^* = x_i^L$, aktiv ist (vgl. z.B. [51]).

Definiert man

$$z_i^L = \begin{cases} \frac{\mu}{x_i - x_i^L}, & \text{falls } i \in I_L \\ 0, & \text{sonst} \end{cases} \quad \text{und} \quad z_i^U = \begin{cases} \frac{\mu}{x_i^U - x_i}, & \text{falls } i \in I_U \\ 0, & \text{sonst,} \end{cases} \quad (3.10)$$

so erhält man den Gradienten der Barrierefunktion $\nabla_x B(x, \mu) = \nabla f(x) - z^L + z^U$.

Betrachtet man nun die **primale Variablen** x und die **dualen Variablen** λ , z^L und z^U als voneinander unabhängig, so ergibt sich die **primal-duale Formulierung** der KKT-Bedingungen (3.9) als

$$\begin{aligned} \nabla f(x) + A(x)\lambda - z^L + z^U &= 0 \\ c(x) &= 0 \\ z_i^L(x_i - x_i^L) - \mu &= 0, \quad i \in I_L \\ z_i^U(x_i^U - x_i) - \mu &= 0, \quad i \in I_U. \end{aligned} \quad (3.11)$$

Andererseits lautet die Lagrangefunktion von (3.6)

$$\mathcal{L}(x, \lambda, z^L, z^U) = f(x) + \sum_{i=1}^m \lambda_i c_i(x) - \sum_{i \in I_L} z_i^L (x_i - x_i^L) - \sum_{i \in I_U} z_i^U (x_i^U - x_i) \quad (3.12)$$

mit den Lagrange-Multiplikatoren λ , z^L und z^U , wobei zur Vervollständigung $z_i^L = 0$ für $i \notin I_L$ und $z_i^U = 0$ für $i \notin I_U$ gesetzt wird.

Betrachtet man dann die mit dem Faktor $\mu > 0$ gestörten KKT-Bedingungen für (3.6), so erhält man ebenfalls das Gleichungssystem (3.11).

Die Bedingungen in (3.11) werden auch **gestörte Optimalitätsbedingungen** des Problems (3.6) genannt.

Für $\mu \rightarrow 0$ wird der Einfluss des Barriereterms bzw. die Störung der KKT-Bedingungen immer kleiner. Das Innere-Punkte-Verfahren besteht nun darin, das lineare Gleichungssystem (3.11) für eine Folge von Barriereparametern $\{\mu_j\}_j$ mit $\mu_j \xrightarrow{j \rightarrow \infty} 0$ zu lösen. Das Verfahren gliedert sich folglich in eine äußere und eine innere Schleife: die innere Schleife löst die primal-dualen Gleichungen für einen festen Barriereparameter μ , die äußere Schleife prüft den Fehler in den Optimalitätsbedingungen und passt den Barriereparameter geeignet an.

Die Bezeichnung Innere-Punkte-Methode leitet sich daraus ab, dass die Ungleichungen

$$\begin{aligned} x_i - x_i^L &\geq 0 \quad \text{und} \quad z_i^L \geq 0 \quad \text{für} \quad i \in I_L \quad \text{sowie} \\ x_i^U - x_i &\geq 0 \quad \text{und} \quad z_i^U \geq 0 \quad \text{für} \quad i \in I_U \end{aligned} \quad (3.13)$$

für $\mu > 0$ stets strikt erfüllt sein müssen [51].

3.2.2 Fehler in den Optimalitätsbedingungen

Um den Fortschritt des Verfahrens beurteilen zu können, werden die primale und duale Unzulässigkeit sowie der Komplementaritätsfehler der Optimalitätsbedingungen bestimmt.

$$\begin{aligned} E_{pr}(x) &= \|c(x)\|_\infty \\ E_{du}(x, \lambda, z^L, z^U) &= \|\nabla f(x) + A(x)\lambda - z^L + z^U\|_\infty \\ E_{com}(x, z^L, z^U) &= \max \left\{ \|(X - X^L)z^L - \mu e\|_\infty, \|(X^U - X)z^U - \mu e\|_\infty \right\} \end{aligned} \quad (3.14)$$

Der Fehler in den Optimalitätsbedingungen des Barriereproblems ist dann

$$E_\mu(x, \lambda, z^L, z^U) = \max \left\{ \frac{E_{du}(x, \lambda, z^L, z^U)}{s_d}, E_{pr}(x), \frac{E_{com}(x, z^L, z^U)}{s_c} \right\}, \quad (3.15)$$

wobei die Wahl der Skalierungsfaktoren $s_d, s_c \geq 1$ in [54] beschrieben wurde.

Das Innere-Punkte-Verfahren wird dann beendet, wenn ein Punkt $(\tilde{x}, \tilde{\lambda}, \tilde{z}^L, \tilde{z}^U)$ gefunden wurde, für den der Fehler in den ungestörten Optimalitätsbedingungen klein ist, d.h.

$$E_0(\tilde{x}, \tilde{\lambda}, \tilde{z}^L, \tilde{z}^U) \leq \varepsilon_{tol}. \quad (3.16)$$

3.2.3 Wahl des Barriereparameters

In IPOPT wurden sowohl eine monotone als auch eine adaptive Strategie zur Wahl des Barriereparameters μ implementiert, die adaptive Strategie wird jedoch erst später in Abschnitt 3.2.8 behandelt.

Die monotone Strategie nach Fiacco und McCormick wurde in [54] beschrieben. Das Barriereproblem (3.8) wird für ein fest gewähltes μ_j gelöst und anschließend der Barriereparameter verringert. Der neue Barriereparameter wird mit $\kappa_\mu \in (0, 1)$ und $\theta_\mu \in (1, 2)$ folgendermaßen berechnet:

$$\mu_{j+1} = \max \left\{ \frac{\varepsilon_{tol}}{10}, \min \{ \kappa_\mu \mu_j, \mu_j^{\theta_\mu} \} \right\}. \quad (3.17)$$

Um die Konvergenz zu beschleunigen, wird jedoch nicht jedes Barriereproblem für μ_j bis zur Toleranz ε_{tol} gelöst, sondern nur solange bis eine Näherungslösung $(\tilde{x}^j, \tilde{\lambda}^j, \tilde{z}^{L,j}, \tilde{z}^{U,j})$ gefunden wurde, für die mit $\kappa_\varepsilon > 0$ gilt

$$E_{\mu_j}(\tilde{x}^j, \tilde{\lambda}^j, \tilde{z}^{L,j}, \tilde{z}^{U,j}) \leq \kappa_\varepsilon \mu_j. \quad (3.18)$$

3.2.4 Lösen der primal-dualen Gleichungen

Zur Vereinfachung der Notation seien im Folgenden alle primalen Variablen x beschränkt, d.h. $I_L = I_U = \{1, \dots, n\}$.

Das System der primal-dualen Gleichungen (3.11) soll durch ein Newtonverfahren iterativ nach den primalen und dualen Variablen gelöst werden. Dazu wird in jeder Iteration im aktuellen Punkt $(x^k, \lambda^k, z^{L,k}, z^{U,k})$ das lineare Gleichungssystem

$$\begin{pmatrix} H_k & A(x^k) & -Id & Id \\ A(x^k)^T & 0 & 0 & 0 \\ Z_k^L & 0 & X_k - X^L & 0 \\ -Z_k^U & 0 & 0 & X^U - X_k \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta z^{L,k} \\ \Delta z^{U,k} \end{pmatrix} = - \begin{pmatrix} \nabla f(x^k) + A(x^k)\lambda^k - z^{L,k} + z^{U,k} \\ c(x^k) \\ Z_k^L(x^k - x^L) - \mu_j e \\ Z_k^U(x^U - x^k) - \mu_j e \end{pmatrix}$$

nach dem Newtonschritt $(\Delta x^k, \Delta \lambda^k, \Delta z^{L,k}, \Delta z^{U,k})$ gelöst. Hierbei bezeichnet H_k die exakte oder approximierte Hessematrix der Lagrangefunktion.

Dieses lineare Gleichungssystem lässt sich mithilfe von $\Sigma_k^L = (X_k - X^L)^{-1} Z_k^L$ sowie $\Sigma_k^U = (X^U - X_k)^{-1} Z_k^U$ und $\Sigma_k = \Sigma_k^L + \Sigma_k^U$ schreiben als

$$\begin{pmatrix} H_k + \Sigma_k & A(x^k) & 0 & 0 \\ A(x^k)^T & 0 & 0 & 0 \\ \Sigma_k^L & 0 & Id & 0 \\ -\Sigma_k^U & 0 & 0 & Id \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta z^{L,k} \\ \Delta z^{U,k} \end{pmatrix} = - \begin{pmatrix} \nabla_x B(x^k, \mu_j) + A(x^k)\lambda^k \\ c(x^k) \\ z^{L,k} - \mu_j (X_k - X^L)^{-1} e \\ z^{U,k} - \mu_j (X^U - X_k)^{-1} e \end{pmatrix}$$

Anstatt dieses großen Systems genügt es, das wesentlich kleinere Teilsystem mit symmetrischer Matrix

$$\begin{pmatrix} H_k + \Sigma_k & A(x^k) \\ A(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta \lambda^k \end{pmatrix} = - \begin{pmatrix} \nabla_x B(x^k, \mu_j) + A(x^k)\lambda^k \\ c(x^k) \end{pmatrix} \quad (3.19)$$

nach $(\Delta x^k, \Delta \lambda^k)$ zu lösen und anschließend die entsprechenden Änderungen in den Lagrange-Multiplikatoren zu berechnen

$$\begin{aligned}\Delta z^{L,k} &= -z^{L,k} + \mu_j(X_k - X^L)^{-1}e - \Sigma_k^L \Delta x^k \\ \Delta z^{U,k} &= -z^{U,k} + \mu_j(X^U - X_k)^{-1}e + \Sigma_k^U \Delta x^k.\end{aligned}\tag{3.20}$$

Um später bestimmte Eigenschaften der Filter-Liniensuche zu garantieren, ist es notwendig, dass die Projektion des oberen linken Blocks der Matrix aus (3.19) auf dem Nullraum von $A(x^k)^T$ positiv definit ist. Außerdem wird die ganze Matrix singular, falls $A(x^k)^T$ nicht vollen Rang hat. Daher wird die Matrix mithilfe von $\psi_w, \psi_c \geq 0$, die in jeder Iteration geeignet gewählt werden (vgl. [54, Abschnitt 3.1]), leicht modifiziert und das folgende System gelöst (vgl. [51], [54]):

$$\begin{pmatrix} H_k + \Sigma_k + \psi_w Id & A(x^k) \\ A(x^k)^T & -\psi_c Id \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \Delta \lambda^k \end{pmatrix} = - \begin{pmatrix} \nabla_x B(x^k, \mu_j) + A(x^k) \lambda^k \\ c(x^k) \end{pmatrix}\tag{3.21}$$

Wichtig für die Verwendung in OPTISIM[®] ist, dass eine Suchrichtung auch dann berechnet werden kann, wenn die Jacobimatrix der Nebenbedingungen numerisch singular wird.

Zur Lösung des Teilproblems (3.19) der primal-dualen Gleichungen wird laut [51] ein reduzierter Lösungsansatz verwendet. Dieser Ansatz war der wesentliche Bestandteil der reduzierten SQP-Verfahren, die in [16], [17], [45], etc. vorgestellt wurden.

Dafür werden Matrizen $\tilde{Y}_k \in \mathbb{R}^{n \times m}$ und $\tilde{Z}_k \in \mathbb{R}^{n \times (n-m)}$ gewählt, sodass die Spalten von $\begin{pmatrix} \tilde{Y}_k \\ \tilde{Z}_k \end{pmatrix}$ eine Orthonormalbasis von \mathbb{R}^n bilden und außerdem $A(x^k)^T \tilde{Z}_k = 0$ gilt, d.h. die Spalten von \tilde{Z}_k sind eine Basis des Nullraums von $A(x^k)$.

Die Suchrichtung Δx^k wird dann zerlegt in

$$\Delta x^k = \tilde{Y}_k \Delta x_Y^k + \tilde{Z}_k \Delta x_Z^k\tag{3.22}$$

Aus Gleichung (3.19) erhält man dann

$$\begin{aligned}\Delta x_Y^k &= - \left(A(x^k)^T \tilde{Y}_k \right)^{-1} c(x^k) \\ \Delta x_Z^k &= - \left(\tilde{Z}_k^T (H_k + \Sigma_k) \tilde{Z}_k \right)^{-1} \tilde{Z}_k^T \left(\nabla_x B(x^k, \mu_j) + (H_k + \Sigma_k) \tilde{Y}_k \Delta x_Y^k \right)\end{aligned}\tag{3.23}$$

3.2.5 Liniensuche

Nachdem für den aktuellen Punkt $(x^k, \lambda^k, z^{L,k}, z^{U,k})$ eine Suchrichtung $(\Delta x^k, \Delta \lambda^k, \Delta z^{L,k}, \Delta z^{U,k})$ berechnet wurde, muss zur Bestimmung der nächsten Iterierten $(x^{k+1}, \lambda^{k+1}, z^{L,k+1}, z^{U,k+1})$ eine Schrittweite α bestimmt werden.

Bei dem in IPOPT implementierten Innere-Punkte-Verfahren wird zunächst zwischen der **primalen Schrittweite** $\alpha_k \in (0, 1]$ für die primalen Variablen x sowie die Lagrangemultiplikatoren der Gleichungsnebenbedingungen λ und der **dualen Schrittweite** $\alpha_k^z \in (0, 1]$ für die Lagrangemultiplikatoren der Ungleichungsnebenbedingungen z^L, z^U unterschieden (vgl.[54]). Die nächste Iterierte ergibt sich dann als

$$\begin{aligned} x^{k+1} &= x^k + \alpha_k \Delta x^k \\ \lambda^{k+1} &= \lambda^k + \alpha_k \Delta \lambda^k \\ z^{L,k+1} &= z^{L,k} + \alpha_k^z \Delta z^{L,k} \\ z^{U,k+1} &= z^{U,k} + \alpha_k^z \Delta z^{U,k} \end{aligned} \tag{3.24}$$

Für die Berechnung der dualen Schrittweite sowie der maximalen primalen Schrittweite wird in IPOPT der Abstandsfaktor

$$\tau_j = \max \{ \tau_{min}, 1 - \mu_j \} \quad \text{mit } \tau_{min} \in (0, 1) \tag{3.25}$$

verwendet. Durch die Anwendung der sog. **Fraction-to-the-boundary Regel** (dt.: Randabstandsregel) wird verhindert, dass die Zielfunktion und die Nebenbedingungen an Stellen außerhalb des Intervalls $[x^L, x^U]$ ausgewertet werden, in denen die Funktionen eventuell gar nicht definiert sind.

Damit erhält man unter Berücksichtigung der beidseitigen Beschränkungen von x analog zu [54, Gl.(15)] die duale Schrittweite als

$$\alpha_k^z = \max \{ \alpha \in (0, 1] \mid z^{L,k} + \alpha \Delta z^{L,k} \geq (1 - \tau_j) z^{L,k} \wedge z^{U,k} + \alpha \Delta z^{U,k} \geq (1 - \tau_j) z^{U,k} \}. \tag{3.26}$$

Für die primale Schrittweite ist das Maximum durch

$$\alpha_k^{max} = \max \{ \alpha \in (0, 1] \mid x^k + \alpha \Delta x^k - x^L \geq (1 - \tau_j)(x^k - x^L) \wedge x^U - (x^k + \alpha \Delta x^k) \geq (1 - \tau_j)(x^U - x^k) \} \tag{3.27}$$

gegeben. Der genaue Wert für α_k wird nun in einer Filter-Liniensuche bestimmt.

3.2.5.1 Filter-Liniensuche

Die Filter-Liniensuche wurde von Wächter [51] bzw. Wächter und Biegler [53, 54] vorgestellt.

Bei einer herkömmlichen Liniensuche (siehe z.B. [28, 29]) wird eine Meritfunktion der Art

$$M_\rho(\alpha) = B(x^k + \alpha \Delta x^k, \mu_j) + \rho \|c(x^k + \alpha \Delta x^k)\|$$

mit Penaltyparameter ρ betrachtet und die Schrittweite α sukzessive verringert, bis eine hinreichende Verbesserung der Meritfunktion, beispielsweise durch die Erfüllung der Armijo-Bedingung

$$M_\rho(\alpha) \leq M_\rho(0) + \eta \alpha M'_\rho(0) \quad \text{mit} \quad \eta \in (0, \frac{1}{2}),$$

gewährleistet ist. Um diese Bedingung zu erfüllen, müssen sowohl der Wert der Barrierefunktion als auch die Verletzung der Gleichungsnebenbedingungen zeitgleich verringert werden. Der Penaltyparameter ρ nimmt hierbei entscheidenden Einfluss darauf, ob die Bedingung erfüllt werden kann oder nicht, und damit auch auf das Verhalten des Optimierungsverfahrens. Der Penaltyparameter muss in jeder Iteration geeignet angepasst werden, wofür es verschiedene Heuristiken, aber keine beste Wahl gibt.

Um die Schwierigkeit der Wahl von ρ zu umgehen, wurden Filtermethoden entwickelt [53]. Dabei wird das gleichungsrestringierte Problem (3.8) als Optimierungsproblem mit zwei Zielfunktionen betrachtet. Einerseits soll der Wert der Barrierefunktion $B(x, \mu_j)$ und damit auch der Wert der ursprünglichen Zielfunktion $f(x)$ minimiert werden, andererseits die **Verletzung der Gleichungsnebenbedingungen**, die im folgenden mit

$$\theta(x) = \|c(x)\| \tag{3.28}$$

bezeichnet wird.

Definition 3.6 (Filter)

Ein Filter $\mathcal{F}_k \subset [0, \infty) \times \mathbb{R}$ ist die Menge aller Paare (θ, b) , die in der k . Iteration verboten sind. Ein neuer Punkt x heißt durch den Filter akzeptabel, falls gilt

$$(\theta(x), B(x, \mu_j)) \notin \mathcal{F}_k.$$

Der Filter wird so gefüllt, dass die Paare $(\theta(x^k), B(x^k, \mu_j))$ einiger vorangegangener Iterierten x^k sowie alle Paare mit größeren Werten in beiden Komponenten, im Filter enthalten sind. Damit sind nur Punkte akzeptabel, für die sich mindestens eines der beiden Maße gegenüber dem Filter verbessert.

Beim Innere-Punkte-Verfahren mit Filter-Liniensuche wird der Filter zu Beginn der Optimierung am Startpunkt x^0 als

$$\mathcal{F}_0 = \{(\theta, b) \in [0, \infty) \times \mathbb{R} \mid \theta \geq \theta^{max}\} \quad (3.29)$$

mit $\theta^{max} \geq \theta(x^0)$ initialisiert. In bestimmten Iterationen wird der Filter dann durch die Vorschrift

$$\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{(\theta, b) \in \mathbb{R}^2 \mid \theta > (1 - \gamma_\theta)\theta(x^k) \wedge b > B(x^k, \mu_j) - \gamma_b\theta(x^k)\} \quad (3.30)$$

erweitert, wobei $\gamma_\theta, \gamma_b \in (0, 1)$ klein gewählt werden. In den restlichen Iterationen bleibt der Filter $\mathcal{F}_{k+1} = \mathcal{F}_k$ unverändert.

Die Filter-Liniensuche erfolgt über eine sukzessive Verkleinerung der **Testschrittweite** $\alpha_{k,l}$ so lange bis der Testpunkt

$$x^{k,l} = x^k + \alpha_{k,l} \Delta x^k$$

als neue Iterierte akzeptiert wird. Letzteres erfolgt dann, wenn $x^{k,l}$ erstens durch den Filter \mathcal{F}_k akzeptabel ist und zweitens eine von zwei Bedingungen zur hinreichenden Verringerung von $B(x, \mu_j)$ oder $\theta(x)$ erfüllt ist. Welche der beiden Bedingungen erfüllt werden soll, hängt von der **Schaltbedingung**

$$\nabla_x B(x^k, \mu_j)^T \Delta x^k < 0 \quad \wedge \quad \alpha_{k,l} [-\nabla_x B(x^k, \mu_j)^T \Delta x^k]^{s_b} > \Gamma_B [\theta(x^k)]^{s_\theta} \quad (3.31)$$

mit den Konstanten $\Gamma_B > 0$, $s_\theta > 1$ und $s_b \geq 1$ ab.

Ist (3.31) erfüllt und $\theta(x^k) \leq \theta^{min}$ mit $\theta^{min} > 0$, so muss die Armijo-artige Bedingung

$$B(x^{k,l}, \mu_j) \leq B(x^k, \mu_j) + \eta_b \alpha_{k,l} \nabla_x B(x^k, \mu_j)^T \Delta x^k \quad (3.32)$$

mit $\eta_b \in (0, \frac{1}{2})$ erfüllt sein, um eine hinreichende Verbesserung der Barrierefunktion zu erhalten.

Ist (3.31) hingegen nicht erfüllt oder $\theta(x^k) > \theta^{min}$, so muss die Bedingung

$$\theta(x^{k,l}) \leq (1 - \gamma_\theta)\theta(x^k) \quad \vee \quad B(x^{k,l}, \mu_j) \leq B(x^k, \mu_j) - \gamma_b\theta(x^k) \quad (3.33)$$

erfüllt sein, d.h. mindestens ein Wert muss sich hinreichend verbessern.

Wird die Testschrittweite $\alpha_{k,l}$ akzeptiert, so wird $\alpha_k = \alpha_{k,l}$ gesetzt. Falls die Schaltbedingung (3.31) erfüllt ist, heißt die Schrittweite **b-Schrittweite**. Wird die Testschrittweite $\alpha_{k,l}$ jedoch nicht akzeptiert, so wird eine neue Testschrittweite

$$\alpha_{k,l+1} \in [\nu_1 \alpha_{k,l}, \nu_2 \alpha_{k,l}] \quad \text{mit} \quad 0 < \nu_1 \leq \nu_2 < 1 \quad (3.34)$$

gewählt. Die initiale Testschrittweite ist stets $\alpha_{k,0} = \alpha_k^{max}$.

Falls eine Testschrittweite $\alpha_{k,l}$ kleiner wird als

$$\alpha_k^{min} = \gamma_\alpha \cdot \begin{cases} \min \left\{ \gamma_\theta, \frac{\gamma_b \theta(x^k)}{-\nabla_x B(x^k, \mu_j)^T \Delta x^k}, \frac{\Gamma_B[\theta(x^k)]^{s_\theta}}{[-\nabla_x B(x^k, \mu_j)^T \Delta x^k]^{s_b}} \right\}, & (*) \\ \min \left\{ \gamma_\theta, \frac{\gamma_b \theta(x^k)}{-\nabla_x B(x^k, \mu_j)^T \Delta x^k} \right\}, & (**) \\ \gamma_\theta, & \text{sonst,} \end{cases} \quad (3.35)$$

$$\text{wobei } (*) \Leftrightarrow \nabla_x B(x^k, \mu_j)^T \Delta x^k < 0 \quad \wedge \quad \theta(x^k) \leq \theta^{min},$$

$$(**) \Leftrightarrow \nabla_x B(x^k, \mu_j)^T \Delta x^k < 0 \quad \wedge \quad \theta(x^k) > \theta^{min}$$

und $\gamma_\alpha \in (0, 1]$ ist, so wird die Liniensuche verlassen und die *Feasibility Restoration Phase* (siehe Abschnitt 3.2.6) gestartet.

Die Erweiterung des Filters erfolgt nach der Formel (3.30), falls α_k keine b -Schrittweite ist oder Bedingung (3.32) nicht erfüllt wurde [53, Abschnitt 2.5, Remark 4], ansonsten wird der Filter beibehalten. Durch dieses Vorgehen können Zyklen in den Iterierten verhindert werden [53].

Anmerkung 3.1 In [53], [51], [54], etc. wird die Vorschrift zur Filtererweiterung durch

$$\mathcal{F}_{k+1} = \mathcal{F}_k \cup \{ (\theta, b) \in \mathbb{R}^2 \mid \theta \geq (1 - \gamma_\theta) \theta(x^k) \wedge b \geq B(x^k, \mu_j) - \gamma_b \theta(x^k) \} \quad (3.36)$$

angegeben. Um die für den Konvergenzbeweis (siehe [53] und Abschnitt 3.4) notwendige Forderung $x^{k+1} \notin \mathcal{F}_{k+1}$ für alle $k \in \mathbb{N}$ zu erfüllen, müssen meines Erachtens jedoch in dieser Vorschrift beide ' \geq ' durch strikte '>' ersetzt werden (vgl. Gleichung (3.30)).

Eine neue Iterierte $x^{k+1} \notin \mathcal{F}_k$ kann dadurch akzeptiert werden, dass sie nur eine der Ungleichungen in Bedingung (3.33) mit Gleichheit erfüllt. Aufgrund der obigen Regelung wird daraufhin der Filter erweitert. Geschieht dies nach der Vorschrift (3.36), so liegt x^{k+1} jedoch im neuen Filter \mathcal{F}_{k+1} .

Ein Beispiel hierfür wurde in Abbildung 3.1 veranschaulicht. Sei $\mathcal{F}_0 = [\theta(x^0), \infty) \times \mathbb{R}$, im Bild als rot schraffierte Fläche einschließlich der roten Linie gekennzeichnet. Im ersten Fall erfüllt x^1 die erste Ungleichung aus (3.33) mit Gleichheit und die zweite Ungleichung nicht. Im zweiten Fall ist es genau umgekehrt. Beide Male liegt der Punkt $(\theta(x^1), B(x^1))$ auf dem Rand des neuen Filters \mathcal{F}_1 , der im Bild als blaue Linie dargestellt ist. Nach Vor-

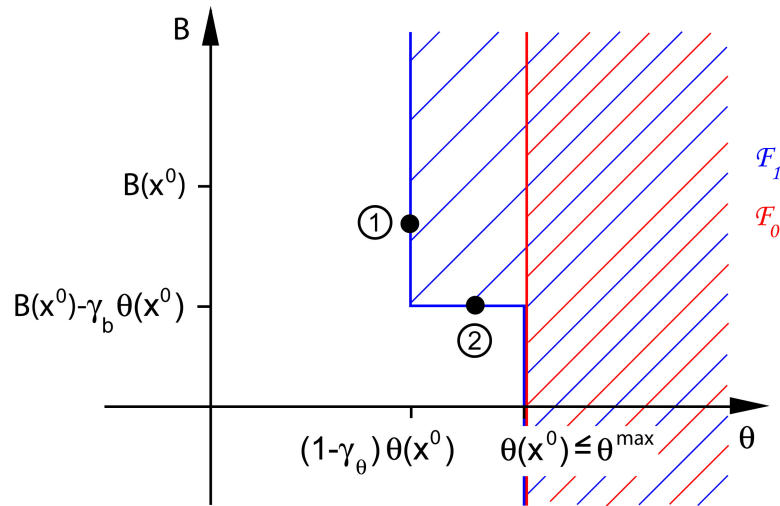


Abbildung 3.1 Beispiel für eine Filtererweiterung

schrift (3.36) gehört der Rand zum neuen Filter \mathcal{F}_1 , nach der hier verwendeten Vorschrift (3.30) jedoch nicht.

Anmerkung 3.2 In den Konvergenzbeweisen für das Filterverfahren (siehe Abschnitt 3.3.2 und 3.4) spielt die Logik, nach welchen Kriterien ein Punkt x^k während der Liniensuche akzeptiert bzw. wann der Filter erweitert wird, eine entscheidende Rolle. Daher wird sie hier nochmals zusammengefasst:

- Ist die Schaltbedingung (3.31) erfüllt und gilt $\theta(x^k) \leq \theta^{min}$, so muss die Armijo-artige Bedingung (3.32) erfüllt und $x^{k,l}$ durch den Filter \mathcal{F}_k akzeptabel sein, damit $x^{k,l}$ als neue Iterierte akzeptiert wird.
- Ist die Schaltbedingung (3.31) nicht erfüllt oder gilt $\theta(x^k) > \theta^{min}$, so muss Bedingung (3.33) erfüllt und $x^{k,l}$ durch den Filter \mathcal{F}_k akzeptabel sein, damit $x^{k,l}$ als neue Iterierte akzeptiert wird.
- Sind die Schaltbedingung (3.31) und die Armijo-artige Bedingung (3.32) bei der Akzeptanz der neuen Iterierten erfüllt, so wird der Filter nicht erweitert.
- Ist die Schaltbedingung (3.31) oder die Armijo-artige Bedingung (3.32) bei der Akzeptanz der neuen Iterierten nicht erfüllt, so wird der Filter erweitert.

3.2.5.2 Second-Order Correction

Auch bei der Filter-Liniensuche kann es zum sog. Maratos-Effekt kommen, d.h. in der Nähe der Lösung x^* wird der volle Schritt Δx^k nicht akzeptiert, da sich die Barrierefunktion und die Verletzung der Nebenbedingungen verschlechtern, obwohl der Schritt eigentlich in die richtige Richtung geht [52]. Um dies zu verhindern, wurde in

IPOPT eine *Second-Order Correction* (dt.: Korrektur zweiter Ordnung) implementiert, die in [54] vorgestellt wurde. Bei der *Second-Order Correction* wird eine Korrekturrichtung berechnet und zu der Suchrichtung Δx^k addiert. Die Korrekturrichtung wird so bestimmt, dass die Verletzung der Nebenbedingungen verringert wird und dadurch die Akzeptanz dieser neuen Suchrichtung vereinfacht wird. Für die Filter-Liniensuche mit *Second-Order Correction* wurde in [52] unter bestimmten Voraussetzungen lokale superlineare Konvergenz bewiesen.

Wird nun die erste Testschrittweite $\alpha_{k,0}$ nicht akzeptiert, so wird eine korrigierte primale Suchrichtung Δx^{cor} mit Hilfe von

$$c^{SOC} = \alpha_{k,0}c(x^k) + c(x^k + \alpha_{k,0}\Delta x^k) \quad (3.37)$$

aus dem folgenden Gleichungssystem berechnet

$$\begin{pmatrix} H_k + \Sigma_k + \psi_w Id & A(x^k) \\ A(x^k)^T & -\psi_c Id \end{pmatrix} \begin{pmatrix} \Delta x^{cor} \\ \Delta \lambda^k \end{pmatrix} = - \begin{pmatrix} \nabla_x B(x^k, \mu_j) + A(x^k)\lambda^k \\ c^{SOC} \end{pmatrix} \quad (3.38)$$

Um keine weitere Matrixzerlegung vornehmen zu müssen, wird hierbei die Matrix aus dem System in Gleichung (3.21) verwendet. Dadurch erhält man mit Δx^{cor} bereits die Summe aus erster Suchrichtung und Korrekturrichtung.

Anschließend wird wie in (3.27) die *Fraction-to-the-boundary* Regel angewandt und

$$\alpha^{SOC} = \max \{ \alpha \in (0, 1] \mid \begin{aligned} x^k + \alpha \Delta x^{cor} - x^L &\geq (1 - \tau_j)(x^k - x^L) \quad \wedge \\ x^U - x^k - \alpha \Delta x^{cor} &\geq (1 - \tau_j)(x^U - x^k) \end{aligned} \} \quad (3.39)$$

bestimmt. Falls der neue Testpunkt

$$x^{SOC} = x^k + \alpha^{SOC} \Delta x^{cor}$$

die entsprechenden Bedingungen der Filter-Liniensuche erfüllt, wird $x^{k+1} = x^{SOC}$ und $\alpha_k = \alpha^{SOC}$ gesetzt. Falls nicht, können bis zu h^{max} weitere Korrekturen berechnet werden, wobei die Verletzung der Nebenbedingungen jedes Mal mindestens um den Faktor $\kappa^{SOC} \in (0, 1)$ verringert werden soll. Falls die *Second-Order Correction* dann immer noch nicht erfolgreich ist, wird die Korrekturrichtung Δx^{cor} verworfen und die Filter-Liniensuche mit der ursprünglichen Suchrichtung Δx^k und der nächstkleineren Testschrittweite $\alpha_{k,1}$ fortgesetzt.

3.2.5.3 Anpassen der dualen Variablen

Um die Voraussetzungen für die globale Konvergenz der Filter-Liniensuche (siehe Abschnitt 3.3.2) zu erfüllen, sollte die primal-duale Hessematrix

$$H_k + \Sigma_k = H_k + (X_k - X^L)^{-1} Z_k^L + (X^U - X_k)^{-1} Z_k^U$$

nicht zu stark von der primalen Hessematrix

$$\nabla_{xx}^2 \mathcal{L}(x^k, \lambda^k, z^{L,k}, z^{U,k}) + \mu_j (X_k - X^L)^{-2} + \mu_j (X^U - X_k)^{-2}$$

abweichen. Um dies zu erreichen werden die aus (3.24) neu berechneten dualen Variablen $z^{L,k+1}$ und $z^{U,k+1}$ geeignet angepasst [54, Gl.(16)], [53, Abschnitt 4.3]. Die Komponenten $z_i^{L,k+1}$ und $z_i^{U,k+1}$ werden dabei für ein $\kappa_\Sigma > 1$ in das Intervall

$$\left[\frac{\mu_j}{\kappa_\Sigma (x_i^{k+1} - x_i^L)}, \frac{\mu_j \kappa_\Sigma}{x_i^{k+1} - x_i^L} \right] \quad \text{bzw.} \quad \left[\frac{\mu_j}{\kappa_\Sigma (x_i^U - x_i^{k+1})}, \frac{\mu_j \kappa_\Sigma}{x_i^U - x_i^{k+1}} \right]$$

projiziert

$$\begin{aligned} z_i^{L,k+1} &\leftarrow \max \left\{ \min \left\{ z_i^{L,k+1}, \frac{\mu_j \kappa_\Sigma}{x_i^{k+1} - x_i^L} \right\}, \frac{\mu_j}{\kappa_\Sigma (x_i^{k+1} - x_i^L)} \right\} \\ z_i^{U,k+1} &\leftarrow \max \left\{ \min \left\{ z_i^{U,k+1}, \frac{\mu_j \kappa_\Sigma}{x_i^U - x_i^{k+1}} \right\}, \frac{\mu_j}{\kappa_\Sigma (x_i^U - x_i^{k+1})} \right\}, \end{aligned} \quad (3.40)$$

wodurch gilt

$$\begin{aligned} \frac{1}{\kappa_\Sigma} \mu_j (x_i^{k+1} - x_i^L)^{-2} &\leq (x_i^{k+1} - x_i^L)^{-1} z_i^{L,k+1} \leq \kappa_\Sigma \mu_j (x_i^{k+1} - x_i^L)^{-2} \\ \frac{1}{\kappa_\Sigma} \mu_j (x_i^U - x_i^{k+1})^{-2} &\leq (x_i^U - x_i^{k+1})^{-1} z_i^{U,k+1} \leq \kappa_\Sigma \mu_j (x_i^U - x_i^{k+1})^{-2}. \end{aligned} \quad (3.41)$$

3.2.6 Feasibility Restoration Phase

Die *Feasibility Restoration Phase* (dt.: Zulässigkeits-Wiederherstellungs-Phase) wird gestartet, wenn die primale Schrittweite $\alpha_{k,l}$ während der Liniensuche in Iteration k kleiner als α_k^{\min} wird oder falls die Matrix in Gleichung (3.21) sehr schlecht konditioniert ist und keine neue Suchrichtung berechnet werden kann. Während der *Feasibility Restoration Phase* soll eine neue Iterierte $(x^{k+1}, \lambda^{k+1}, z^{L,k+1}, z^{U,k+1})$ berechnet werden, die vom Filter \mathcal{F}_{k+1} akzeptiert wird und die Verletzung der Nebenbedingungen hinreichend verringert.

Die *Feasibility Restoration Phase*, die in IPOPT implementiert wurde, besteht aus zwei Teilen. In Teil A soll die Verletzung der Nebenbedingungen minimiert werden. Dazu wird für einen Referenzpunkt x^R mit Hilfe der Matrix

$$D_R = \text{diag} \left(\min \left\{ 1, \frac{1}{|x_i^R|} \right\} \right)$$

und dem Gewichtungsfaktor $\zeta > 0$ das Optimierungsproblem

$$\begin{aligned} \min_{\bar{x} \in \mathbb{R}^n, \bar{p}, \bar{n} \in \mathbb{R}^m} \quad & \frac{\zeta}{2} \|D_R(\bar{x} - x^R)\|_2^2 + \sum_{i=1}^m \bar{p}_i \bar{n}_i \\ \text{u.d.N.} \quad & c(\bar{x}) - \bar{p} + \bar{n} = 0 \\ & x^L \leq \bar{x} \leq x^U \\ & \bar{p}, \bar{n} \geq 0 \end{aligned} \tag{3.42}$$

wiederum durch die Innere-Punkte-Methode mit Filter-Liniensuche gelöst. Der Iterationsindex für \bar{x} sei t . Die Matrix des zu lösenden Systems hat dieselbe Struktur wie die Matrix in (3.21) mit $\bar{\psi}_c = 0$.

Eine *Second-Order Correction* wurde für die *Feasibility Restoration Phase* nicht implementiert. Falls die Filter-Liniensuche, angewendet auf Problem (3.42), selbst eine *Feasibility Restoration Phase* starten möchte, so wird das Optimierungsproblem (3.42) für fixes $\bar{x} = \bar{x}^t$ nur in \bar{p} und \bar{n} gelöst und dieses Ergebnis als Lösung der *Feasibility Restoration Phase* innerhalb der *Feasibility Restoration Phase* verwendet.

Teil A der *Feasibility Restoration Phase* wird dann verlassen, wenn die aktuelle Iterierte \bar{x}^t vom Filter \mathcal{F}_{k+1} akzeptiert wird und die Verletzung der Nebenbedingungen $\theta(x)$ mindestens um den Faktor $\kappa_{resto} \in (0, 1)$ verringert wurde. Dann werden $x^{k+1} = \bar{x}^t$ und $\Delta x^k = x^{k+1} - x^k$ gesetzt und die dualen Variablen $z^{L,k+1}$ und $z^{U,k+1}$ aus (3.20) und (3.40) berechnet, während λ reinitialisiert wird.

Teil B der *Feasibility Restoration Phase* versucht einen neuen Punkt zu berechnen, der den KKT-Fehler $\|G_\mu(x, \lambda, z^L, z^U)\|_1$ um einen Faktor $\kappa_G \in (0, 1)$ verringert und vom Filter \mathcal{F}_{k+1} akzeptiert wird, wobei $G_\mu(x, \lambda, z^L, z^U)$ hier die linke Seite der primal-dualen Gleichungen (3.11) bezeichnet. Dabei werden Punkte $(\bar{x}^t, \bar{\lambda}^t, \bar{z}^{L,t}, \bar{z}^{U,t})$ generiert, wobei auch hier t der Iterationsindex ist.

Wenn die *Feasibility Restoration Phase* von der Liniensuche aus gestartet wird, so beginnt diese zunächst mit Teil B. Verläuft dies erfolgreich, so wird $(x^{k+1}, \lambda^{k+1}, z^{L,k+1}, z^{U,k+1}) = (\bar{x}^t, \bar{\lambda}^t, \bar{z}^{L,t}, \bar{z}^{U,t})$ gesetzt und zur regulären Methode zurückgekehrt. Verläuft Teil B hingegen nicht erfolgreich, so wird Teil A der *Feasibility Restoration Phase* mit \bar{x}^t als Referenzpunkt gestartet.

Wird die *Feasibility Restoration Phase* hingegen gestartet, weil keine Suchrichtung berechnet werden konnte, so wird gleich Teil A mit der aktuellen Iterierten x^k als Referenzpunkt ausgeführt.

Ist die Verletzung der Nebenbedingungen zu Beginn einer *Feasibility Restoration Phase* sehr klein, so wird ein Fehler zurückgegeben, da davon ausgegangen werden kann, dass das Optimierungsproblem (3.6) gar keinen zulässigen Punkt hat [54].

3.2.7 Algorithmus

Das Innere-Punkte-Verfahren IPOPT lässt sich schematisch durch den folgenden Algorithmus darstellen (vgl. [54]):

Algorithmus 3.1 Innere-Punkte-Verfahren

S0 Wähle Startwerte $(x^0, \lambda^0, z^{L,0}, z^{U,0})$ mit $x^0, z^{L,0}, z^{U,0} > 0$ und $\mu_0 > 0$, sowie $\varepsilon_{tol} > 0$, $\kappa_\mu \in (0, 1)$, $\theta_\mu \in (1, 2)$, $\kappa_\varepsilon > 0$, $\tau_{min} \in (0, 1)$, $\theta^{max} \geq \theta(x^0)$, $\gamma_\theta, \gamma_b \in (0, 1)$, $\Gamma_B > 0$, $s_\theta > 1$, $s_b \geq 1$, $\theta^{min} > 0$, $\eta_b \in (0, \frac{1}{2})$, $0 < \nu_1 \leq \nu_2 < 1$, $\gamma_\alpha \in (0, 1]$, $h^{max} \in \mathbb{N}$, $\kappa^{SOC} \in (0, 1)$, $\kappa_\Sigma > 1$ und $\kappa_{resto} \in (0, 1)$.

S1 Setze $j \leftarrow 0$, $k \leftarrow 0$, initialisiere \mathcal{F}_0 wie in Gleichung (3.29) und τ_0 aus (3.25).

S2 Falls $E_0(x^k, \lambda^k, z^{L,k}, z^{U,k}) \leq \varepsilon_{tol}$: STOP

S3 Falls $E_{\mu_j}(x^k, \lambda^k, z^{L,k}, z^{U,k}) \leq \kappa_\varepsilon \mu_j$, dann

S3.1 Berechne μ_{j+1} und τ_{j+1} aus (3.17), (3.25) und setze $j \leftarrow j + 1$.

S3.2 Reinitialisiere Filter $\mathcal{F}_k = \mathcal{F}_0$

S3.3 Falls $k=0$, gehe zu S3.

S4 Berechne eine Suchrichtung $(\Delta x^k, \Delta \lambda^k, \Delta z^{L,k}, \Delta z^{U,k})$ aus den Gleichungen (3.19) und (3.20).

Falls das System jedoch zu schlecht konditioniert ist, gehe zu S9.

S5 Liniensuche

S5.1 Initialisierung: Setze $\alpha_{k,0} = \alpha_k^{max}$ aus Gleichung (3.27) und $l \leftarrow 0$.

S5.2 Berechne Testpunkt $x^{k,l} = x^k + \alpha_{k,l} \Delta x^k$.

S5.3 Falls $x^{k,l}$ nicht akzeptabel durch den Filter \mathcal{F}_k ist, gehe zu S5.5.

S5.4 Fall I: $\theta(x^k) \leq \theta^{min}$ und Bedingung (3.31) erfüllt:

Falls Bedingung (3.32) gilt, setze $x^{k+1} = x^{k,l}$, $\alpha_k = \alpha_{k,l}$ und gehe zu S6.

Sonst gehe zu S5.5.

Fall II: $\theta(x^k) > \theta^{min}$ oder Bedingung (3.31) nicht erfüllt:

Falls Bedingung (3.33) gilt, setze $x^{k+1} = x^{k,l}$, $\alpha_k = \alpha_{k,l}$ und gehe zu S6.

Sonst gehe zu S5.5.

S5.5 Falls $l > 0$ oder $\theta(x^{k,0}) < \theta(x^k)$, gehe zu S5.10.

Sonst initialisiere SOC: Setze $h \leftarrow 1$, $\theta_{old}^{SOC} = \theta(x^k)$ und c^{SOC} aus Gl. (3.37).

S5.6 Berechne $(\Delta x^{cor,h}, \Delta \lambda^k)$ aus Gleichung (3.38), α^{SOC} aus Gleichung (3.39) und $x^{SOC} = x^k + \alpha^{SOC} \Delta x^{cor,h}$.

S5.7 Falls x^{SOC} vom Filter \mathcal{F}_k nicht akzeptiert wird, verwerfe x^{SOC} und gehe zu S5.10.

S5.8 Fall I: $\theta(x^k) \leq \theta^{min}$ und Bedingung (3.31) erfüllt:

Falls Bedingung (3.32) mit x^{SOC} statt $x^{k,l}$ gilt, setze $x^{k+1} = x^{SOC}$, $\alpha_k = \alpha^{SOC}$ und gehe zu S6.

Sonst gehe zu S5.9.

Fall II: $\theta(x^k) > \theta^{min}$ oder Bedingung (3.31) nicht erfüllt:

Falls Bedingung (3.33) mit x^{SOC} statt $x^{k,l}$ gilt, setze $x^{k+1} = x^{SOC}$, $\alpha_k = \alpha^{SOC}$ und gehe zu S6.

Sonst gehe zu S5.9.

S5.9 Falls $h = h^{max}$ oder $\theta(x^{SOC}) > \kappa^{SOC} \theta_{old}^{SOC}$, gehe zu S5.10.

Sonst setze $h \leftarrow h + 1$, $c^{SOC} \leftarrow \alpha^{SOC} c^{SOC} + c(x^{SOC})$, $\theta_{old}^{SOC} \leftarrow \theta(x^{SOC})$ und gehe zu S5.6.

S5.10 Wähle $\alpha_{k,l+1} = \frac{1}{2} \alpha_{k,l}$, $l \leftarrow l + 1$.

Falls $\alpha_{k,l} < \alpha_k^{min}$, gehe zu S9.

Sonst gehe zu S5.2.

S6 Berechne α^z aus Gleichung (3.26) und λ^{k+1} , $z^{L,k+1}$, $z^{U,k+1}$ aus Gleichung (3.24) und korrigiere mit Gleichung (3.40).

S7 Falls Bedingung (3.31) oder (3.32) für α_k nicht erfüllt ist, erweitere Filter \mathcal{F}_{k+1} nach Formel (3.30).

Sonst setze $\mathcal{F}_{k+1} = \mathcal{F}_k$.

S8 Setze $k \leftarrow k + 1$, gehe zu S2.

S9 *Feasibility Restoration Phase:* Erweitere Filter \mathcal{F}_{k+1} wie in Gleichung (3.30) und berechne x^{k+1} , sodass $\theta(x)$ um Faktor κ_{resto} verringert und x^{k+1} von \mathcal{F}_{k+1} akzeptiert wird, gehe zu S8.

3.2.8 Adaptive Wahl des Barriereparameters

Wie bereits in Abschnitt 3.2.3 erwähnt, wurde in IPOPT neben der monotonen Strategie nach Fiacco und McCormick auch eine adaptive Strategie zur Wahl des Barriereparameters μ implementiert. Die zweite Strategie wurde von Nocedal, Wächter und Waltz in [39] vorgestellt. Dabei wird die Struktur der inneren und äußeren Schleife aufgehoben. Anstatt den Barriereparameter konstant zu halten, bis das Barriereproblem bis zu einer bestimmten Genauigkeit gelöst wurde, kann er hier während der Lösung des Barriereproblems verändert werden. Das Verfahren wird so robuster, da auf Änderungen in der Skalierung des Problems sofort reagiert und eine zu schnelle Verkleinerung von μ verhindert werden kann [39].

Zur Berechnung eines neuen Barriereparameters μ_{k+1} innerhalb der adaptiven Strategie wurden in [39] verschiedene Methoden vorgestellt. Dabei können feste Formeln wie die von Fiacco und McCormick (vgl. (3.17)), Formeln, die von dem Wert der Komplementarität $(x - x^L)^T z^L + (x^U - x)^T z^U$ abhängen, oder eine Berechnungsmethode mit Hilfe einer sog. *quality function* angewendet werden. Die *quality function*, die beispielsweise dem nach dem Barriereparameter linearisierten Fehler der Optimalitätsbedingung entspricht, soll durch den neuen Barriereparameter minimiert werden.

In IPOPT wurde die adaptive Strategie für das globalisierte Filter-Verfahren implementiert, welche in [39, Algorithmus B] vorgestellt wurde. Dieser Algorithmus kann folgendermaßen skizziert werden:

Algorithmus 3.2 Innere-Punkte-Verfahren mit adaptiver Strategie

- S0 Wähle $(x^0, \lambda^0, z^{L,0}, z^{U,0})$ mit $x^0, z^{L,0}, z^{U,0} > 0$, $\kappa_1, \kappa_2 > 0$, setze $k \leftarrow 0$ und initialisiere \mathcal{F}_0 .
- S1 Freier Modus:
 - S1.1 Wähle μ_k nach beliebiger Regel.
 - S1.2 Berechne eine Suchrichtung $(\Delta x^k, \Delta \lambda^k, \Delta z^{L,k}, \Delta z^{U,k})$ im aktuellen Punkt $(x^k, \lambda^k, z^{L,k}, z^{U,k})$ aus Gleichung (3.11) bzw. (3.19) und (3.20).
 - S1.3 Berechne eine primale Schrittweite α_k mit Hilfe der Filter-Liniensuche und eine duale Schrittweite α_k^z .
 - S1.4 Berechne daraus den neuen Testpunkt $(\bar{x}, \bar{\lambda}, \bar{z}^L, \bar{z}^U)$.
 - S1.5 Berechne den Filterrand $\Psi_k = \kappa_1 \min\{\kappa_2, E_0(x^k, \lambda^k, z^{L,k}, z^{U,k})\}$.

S1.6 Falls $(B(\bar{x}, \mu_k) + \Psi_k, \theta(\bar{x}) + \Psi_k) \notin \mathcal{F}_k$,
 setze $(x^{k+1}, \lambda^{k+1}, z^{L,k+1}, z^{U,k+1}) = (\bar{x}, \bar{\lambda}, \bar{z}^L, \bar{z}^U)$, erweitere Filter \mathcal{F}_{k+1} nach
 Formel (3.30), setze $k \leftarrow k + 1$ und gehe zu S1.1.
Sonst gehe zu S2.

S2 Monotoner Modus:

S2.1 Setze Startwert $(\bar{x}^0, \bar{\lambda}^0, \bar{z}^{L,0}, \bar{z}^{U,0}) = (\bar{x}, \bar{\lambda}, \bar{z}^L, \bar{z}^U)$, $t \leftarrow 0$ und wähle ein geeig-
 netes $\bar{\mu}_0$.

S2.2 Berechne eine Suchrichtung $(\Delta \bar{x}^t, \Delta \bar{\lambda}^t, \Delta \bar{z}^{L,t}, \Delta \bar{z}^{U,t})$ im aktuellen Punkt
 $(\bar{x}^t, \bar{\lambda}^t, \bar{z}^{L,t}, \bar{z}^{U,t})$ aus Gleichung (3.11) bzw. (3.19) und (3.20).

S2.3 Berechne eine primale Schrittweite $\bar{\alpha}_t$ mit Hilfe der Filter-Liniensuche und
 eine duale Schrittweite $\bar{\alpha}_t^z$.

S2.4 Berechne daraus den neuen Testpunkt $(\bar{x}^{t+1}, \bar{\lambda}^{t+1}, \bar{z}^{L,t+1}, \bar{z}^{U,t+1})$.

S2.5 Falls $(B(\bar{x}^{t+1}, \bar{\mu}_t) + \Psi_k, \theta(\bar{x}^{t+1}) + \Psi_k) \notin \mathcal{F}_k$,
 setze $(x^{k+1}, \lambda^{k+1}, z^{L,k+1}, z^{U,k+1}) = (\bar{x}^{t+1}, \bar{\lambda}^{t+1}, \bar{z}^{L,t+1}, \bar{z}^{U,t+1})$, erweitere Filter
 \mathcal{F}_{k+1} nach Formel (3.30), setze $k \leftarrow k + 1$ und gehe zu S1.1.

Sonst: Falls $E_{\bar{\mu}_t}(\bar{x}^{t+1}, \bar{\lambda}^{t+1}, \bar{z}^{L,t+1}, \bar{z}^{U,t+1})$ klein genug,
 berechne neuen Barriereparameter $\bar{\mu}_{t+1} < \bar{\mu}_t$, setze $t \leftarrow t + 1$ und
 gehe zu S2.2.

Sonst setze $\bar{\mu}_{t+1} = \bar{\mu}_t$, $t \leftarrow t + 1$ und gehe zu S2.2.

3.3 Konvergenz der Innere-Punkte-Verfahren

In diesem Abschnitt werden einige Ergebnisse des Konvergenzverhaltens der Innere-Punkte-Methode anhand von [27, 39, 53] zusammengefasst.

3.3.1 Gesamtkonvergenz des Innere-Punkte-Verfahrens

In Forsgren, Gill und Wright [27, Theorem 3.12] wird für ungleichungsrestringierte Optimierungprobleme der Form

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u.d.N.} \quad c(x) \geq 0 \quad (3.43)$$

mit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ folgendes gezeigt: Die unrestringierten Minimierer der zu (3.43) gehörenden Barrierefunktion

$$B(x, \mu) = f(x) - \mu \sum_{i=1}^m \ln c_i(x)$$

konvergieren für $\mu \rightarrow 0$ gegen ein Minimum von (3.43). Zur Formulierung des Satzes werden folgende Definitionen benötigt:

Definition 3.7 (restringierte und unrestringierte Minimierer)

- a) Sei $\mathcal{F} = \{x \in \mathbb{R}^n \mid c(x) \geq 0\}$ die Menge der zulässigen Punkte und $\text{strict}(\mathcal{F}) = \{x \in \mathbb{R}^n \mid c(x) > 0\}$ die Menge der strikt zulässigen Punkte.
- b) $x^* \in \mathbb{R}^n$ heißt lokal restringierter Minimierer des Optimierungsproblems (3.43)

$$\Leftrightarrow \quad x^* \in \mathcal{F} \quad \wedge \quad \exists_{S \subset \mathbb{R}^n, \text{ kompakt}} \quad x^* \in \text{int}(S) \quad \wedge \quad f(x^*) = \min_{x \in S \cap \mathcal{F}} f(x).$$

- c) $x^* \in \mathbb{R}^n$ heißt lokal unrestringierter Minimierer von f

$$\Leftrightarrow \quad \exists_{S \subset \mathbb{R}^n, \text{ kompakt}} \quad x^* \in \text{int}(S) \quad \wedge \quad f(x^*) = \min_{x \in S} f(x).$$

Satz 3.2 (Zentraler Pfad) Seien f und c_i , $i = 1, \dots, m$ zweifach stetig differenzierbar, $\text{strict}(\mathcal{F}) \neq \emptyset$ und x^* ein lokal restringierter Minimierer von (3.43), für den gilt:

- a) x^* ist KKT-Punkt, d.h. es existiert eine nichtleere Menge M_λ von Lagrangemultiplikatoren λ , die die KKT-Bedingungen von (3.43) erfüllen,
- b) in x^* gelten die MFCQ-Bedingungen und

c) es existiert ein $W > 0$ sodass für alle $\lambda \in M_\lambda$ und für alle $w \in \mathbb{R}^n \setminus \{0\}$ mit $\nabla f(x^*)^T w = 0$ und $\nabla c_A(x^*)^T w \geq 0$ gilt

$$w^T H(x^*, \lambda) w \geq W \|w\|_2^2,$$

wobei H die Hessematrix der Lagrangefunktion von (3.43) und c_A die in x^* aktiven Nebenbedingungen sind.

Weiter sei $\{\mu_k\}_k$ eine Folge von positiven Barriereparametern, die für $k \rightarrow \infty$ monoton gegen 0 konvergiert und x^k sei jeweils der unrestringierte Minimierer von $B(\cdot, \mu_k)$.

Dann gilt:

- (i) Es existiert mindestens eine Teilfolge $\{x^{k_j}\}_{k_j}$ von $\{x^k\}_k$, die gegen x^* konvergiert.
- (ii) Die entsprechende Folge von Lagrangemultiplikatoren $\{\lambda^{k_j}\}_{k_j}$ mit $\lambda_i^{k_j} = \frac{\mu_{k_j}}{c_i(x^{k_j})}$ ist beschränkt.
- (iii) $\lim_{j \rightarrow \infty} \lambda^{k_j} = \lambda^* \in M_\lambda$.

Gilt zudem strikte Komplementarität, d.h. existiert ein $\lambda \in M_\lambda$ mit $\lambda_A > 0$, so gilt außerdem:

- (iv) $\lambda_A^* > 0$.
- (v) Für j groß genug ist die Hessematrix $\nabla_{xx}^2 B(x^{k_j}, \mu_{k_j})$ positiv definit.
- (vi) Es existiert eine eindeutige, stetig differenzierbare Funktion, die in einer Umgebung von $\mu = 0$ jedem positiven μ den unrestringierten Minimierer $x(\mu)$ von $B(\cdot, \mu)$ zuordnet.
- (vii) $\lim_{\mu \rightarrow 0_+} x(\mu) = x^*$.

3.3.2 Globale Konvergenz des Filter-Verfahrens

Wächter und Biegler zeigten in [53] die globale Konvergenz des Innere-Punkte-Verfahrens mit Filter-Liniensuche. Der Beweis gliedert sich in mehrere Abschnitte:

3.3.2.1 Globale Konvergenz für gleichungsrestringierte Probleme

In [53, Abschnitt 3] wird die Konvergenz des Filter-Verfahrens zunächst für rein gleichungsrestringierte Probleme der Form

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u.d.N.} \quad c(x) = 0 \quad (3.44)$$

mit $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ bewiesen.

Während des Verfahrens wird in jeder Iteration die Gleichung

$$\begin{pmatrix} H_k & A(x^k) \\ A(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \lambda_+^k \end{pmatrix} = - \begin{pmatrix} \nabla f(x^k) \\ c(x^k) \end{pmatrix} \quad (3.45)$$

gelöst, wobei die Hessematrix der zu (3.44) gehörigen Lagrangefunktion durch $H_k \approx \nabla^2 f(x^k) + \sum_{i=1}^m \lambda_i^k \nabla^2 c_i(x^k)$ approximiert wird. Die Lösung von (3.45) erfolgt durch den reduzierten Ansatz, der in Abschnitt 3.2.4 vorgestellt wurde, hier jedoch mit der Matrix H anstelle von $H + \Sigma$ und der Zielfunktion $f(x)$ statt $B(x, \mu)$. Das Gleichungssystem in (3.45) wurde mit Hilfe von $\lambda_+^k = \Delta \lambda^k + \lambda^k$ formuliert, da so der Einfluss von A besser abgeschätzt werden kann (vgl. [53, Lemma 1]). Die Filter-Liniensuche erfolgt wie in Abschnitt 3.2.5.1 beschrieben, jedoch mit der Zielfunktion f aus (3.44) anstelle der Barrierefunktion B , der maximalen Schrittweite $\alpha_k^{max} = 1$, da für x keine Schranken gegeben sind, sowie ohne *Second-Order Correction*.

Definition 3.8 (Iterationen mit *Feasibility Restoration Phase*)

Mit $\mathcal{R} \subset \mathbb{N}$ wird die Menge aller Iterationsindices der Iterationen bezeichnet, in denen die *Feasibility Restoration Phase* aufgerufen wird.

Mit $\mathcal{R}_{inc} \subset \mathcal{R}$ wird die Menge aller Iterationsindices der Iterationen bezeichnet, in denen die *Feasibility Restoration Phase* während der Suche nach einer neuen Suchrichtung aufgerufen wird.

Um die globale Konvergenz dieses Filter-Verfahrens für gleichungsrestringierte Probleme zu erhalten, werden folgende Voraussetzungen angenommen:

Voraussetzungen 3.1 Sei $\{x^k\}_k$ die von dem eben beschriebenen Filter-Verfahren erzeugte Folge, x^0 der Startpunkt und $s_b \geq 1$. Die *Feasibility Restoration Phase* werde

stets erfolgreich beendet und der Algorithmus ende nicht an einem KKT-Punkt. Außerdem seien folgende Bedingungen erfüllt:

(V1) Es existiere eine offene Menge $C \subset \mathbb{R}^n$ mit $[x^k, x^k + \Delta x^k] \subset C \quad \forall k \notin \mathcal{R}_{inc}$ sodass f und c differenzierbar sind auf C und $f, \nabla f, c$ sowie ∇c auf C beschränkt und Lipschitz-stetig sind.

(V2) Die Approximationen H_k der Hessematrix der zu (3.44) gehörigen Lagrangefunktion seien für alle $k \notin \mathcal{R}_{inc}$ gleichmäßig beschränkt.

(V3) Die Matrizen H_k seien außerdem gleichmäßig positiv definit auf dem Nullraum von $A(x^k)^T$, d.h. es existiert eine Konstante $M_H > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\lambda_{\min} \left(\tilde{Z}_k^T H_k \tilde{Z}_k \right) \geq M_H.$$

(V4) Es existiere eine Konstante $M_A > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\sigma_{\min} (A(x^k)) \geq M_A.$$

(V5) Die Iterierten, für die die Feasibility Restoration Phase während der Berechnung einer neuen Suchrichtung gestartet wird, seien nicht beliebig nahe am zulässigen Bereich, d.h. es gibt eine Konstante $\theta_{inc} > 0$ sodass $k \notin \mathcal{R}_{inc}$ ist, wenn $\theta(x^k) \leq \theta_{inc}$ gilt.

Folgende Zuordnung wird in [53] definiert und deren Eigenschaft als Optimalitätsmaß erster Ordnung gezeigt:

Definition 3.9 (Optimalitätsmaß)

$$\chi : \{x^k\}_k \rightarrow [0, \infty], \quad \chi(x^k) = \begin{cases} \|\Delta x_Z^k\|_2 = \|\tilde{Z}_k \Delta x_Z^k\|_2, & \text{falls } k \notin \mathcal{R}_{inc} \\ \infty, & \text{sonst,} \end{cases} \quad (3.46)$$

wobei $\tilde{Z}_k \Delta x_Z^k$ ein Teil der Suchrichtung, der sich aus dem reduzierten Ansatz (vgl. (3.22) bzw. (3.23)) ergibt, ist.

Lemma 3.1 (Optimalitätsmaß)

Seien die Voraussetzungen 3.1 erfüllt. Dann ist χ ein Maß für die Optimalität erster Ordnung, d.h. falls eine Teilfolge $\{x^{k_j}\}_j$ der Iterierten $\{x^k\}_k$ mit $\chi(x^{k_j}) \rightarrow 0$ gegen einen zulässigen Häufungspunkt x^* konvergiert, dann ist x^* ein KKT-Punkt von (3.44).

Der Beweis aus [53] wird hier etwas detaillierter wiedergegeben, da später nochmals darauf eingegangen wird.

Beweis: Man betrachtet eine Teilfolge $\{x^{k_j}\}_j$ mit $\lim_{j \rightarrow \infty} \chi(x^{k_j}) = 0$ und $\lim_{j \rightarrow \infty} x^{k_j} = x^*$, wobei x^* zulässig ist. Aus $\chi(x^k) = \infty$ für $k \in \mathcal{R}_{inc}$ folgt dann, dass $k_j \notin \mathcal{R}_{inc}$ für j groß genug. Aus der Stetigkeit von c , Voraussetzung 3.1 (V4) und der Konstruktion von Δx_Y folgt $\lim_{j \rightarrow \infty} \Delta x_Y^{k_j} = 0$. Aus $\lim_{j \rightarrow \infty} \chi(x^{k_j}) = 0$, Voraussetzung 3.1 (V3) und der Konstruktion von Δx_Z (siehe Gleichung (3.23)) folgt dann $\lim_{j \rightarrow \infty} \tilde{Z}_{k_j}^T \nabla f(x^{k_j}) = 0$. Daraus folgt wiederum $\nabla f(x^*) \in \text{range}(A(x^*))$. Damit existiert ein λ^* mit $\nabla f(x^*) = -A(x^*)\lambda^*$. Außerdem ist x^* zulässig, sodass die KKT-Bedingungen des Problems (3.44) erfüllt sind. \square

Als nächstes wird der folgende Satz [53, Theorem 1] gezeigt.

Satz 3.3 (Zulässigkeit) *Seien die Voraussetzungen 3.1 erfüllt. Dann gilt*

$$\lim_{k \rightarrow \infty} \theta(x^k) = 0,$$

d.h. alle Häufungspunkte der Folge $\{x^k\}_k$ sind zulässig.

Der Beweis dieses Satzes verwendet die Voraussetzung, dass die Folge $\{f(x^k)\}_k$ nach unten beschränkt ist, sowie die Aussagen von [53, Lemma 5] und [53, Lemma 6]. Der Beweis von [53, Lemma 6] benutzt wiederum die Beschränkung der Folge $\{f(x^k)\}_k$ nach unten, sowie die Beschränkung der Nebenbedingungen. Hierbei spielt der in Gleichung (3.31) benötigte Parameter s_b eine entscheidende Rolle: Für den Fall $s_b = 1$ kann [53, Lemma 5] direkt aus der Konstruktion des Algorithmus und der Beschränkung von $\{f(x^k)\}_k$ nach unten gefolgert werden. Für den Fall $s_b > 1$ wird [53, Lemma 1] benutzt, welches jedoch die beidseitige Beschränkung von Zielfunktion und deren Gradienten voraussetzt.

Wie in [53, Anmerkung 6] beschrieben, gilt dieses erste Theorem also für den Fall $s_b = 1$ auch dann, wenn die Folgen $\{f(x^k)\}_k$ und $\{\|\nabla f(x^k)\|\}_k$ nach oben unbeschränkt sind.

Anschließend wird der zweite Satz [53, Theorem 1] mit Hilfe verschiedener Lemmata bewiesen.

Satz 3.4 (Optimalität) *Seien die Voraussetzungen 3.1 erfüllt. Dann gilt*

$$\liminf_{k \rightarrow \infty} \chi(x^k) = 0,$$

d.h. wenn die Folge der Iterierten $\{x^k\}_k$ beschränkt ist, so gibt es einen Häufungspunkt dieser Folge, der KKT-Punkt des betrachteten Optimierungsproblems (3.44) ist.

3.3.2.2 Globale Konvergenz für zusätzlich ungleichungsrestringierte Probleme

In [53, Abschnitt 4.3] wird die Konvergenz der Filter-Liniensuche dann auf Innere-Punkte-Verfahren erweitert. Gleichungsrestringierten und beschränkten Problemen der Form

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u.d.N.} \quad c(x) = 0, \quad x \geq 0 \quad (3.47)$$

werden (gleichungsrestringierte) Barriereprobleme der Form

$$\min_{x \in \mathbb{R}^n} B(x, \mu) = f(x) - \mu \sum_{i=1}^n \ln(x_i) \quad \text{u.d.N.} \quad c(x) = 0 \quad (3.48)$$

zugeordnet. Darauf wird wieder das in Abschnitt 3.2.5.1 beschriebene Filter-Verfahren ohne *Second-Order Correction* angewandt. Im Vergleich zum vorhergehenden Abschnitt wird nun wieder die Barrierefunktion B anstelle der Zielfunktion f minimiert und als initiale primale Schrittweite während der Filter-Liniensuche α^{max} aus der *Fraction-to-the-boundary* Regel (vgl. Gleichung (3.27)) verwendet. Allerdings sind die Barrierefunktion sowie die *Fraction-to-the-boundary* Regel hier nur für einseitige Beschränkungen formuliert. In jeder Iteration wird die Gleichung

$$\begin{pmatrix} H_k + \Sigma_k & A(x^k) \\ A(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \lambda_+^k \end{pmatrix} = - \begin{pmatrix} \nabla_x B(x^k, \mu) \\ c(x^k) \end{pmatrix} \quad (3.49)$$

gelöst. Dieses lineare Gleichungssystem entspricht Gleichung (3.19), allerdings ist, wiederum aufgrund der einseitigen Beschränkung, $\Sigma_k \approx \mu X_k^{-2}$.

Für den Konvergenzbeweis wird statt (3.49) die Gleichung

$$\begin{pmatrix} H_k + \mu X_k^{-2} & A(x^k) \\ A(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \lambda_+^k \end{pmatrix} = - \begin{pmatrix} \nabla_x B(x^k, \mu) \\ c(x^k) \end{pmatrix} \quad (3.50)$$

mit der „primale Hessematrix“ $H_k + \mu X_k^{-2}$ der zum Barriereproblem (3.48) gehörigen Lagrangefunktion betrachtet. Die Argumentation gilt jedoch auch für das Verfahren mit der „primal-dualen Hessematrix“ $H_k + \Sigma_k$, wenn die Abweichung nicht zu groß wird. Dies wird erreicht, indem Σ_k analog zu der in Abschnitt 3.2.5.3 beschriebenen Korrektur angepasst wird [53, S.26].

Durch die Verwendung der Barrierefunktion anstelle der Zielfunktion ergibt sich für den globalen Konvergenzbeweis folgende Problematik: Erstens ist die Funktion selbst,

wie auch ihr Gradient, nur dann definiert, wenn x in der strikt zulässigen Menge liegt. Zweitens werden die Werte von Funktion und Ableitung unbeschränkt, sobald sich x der Grenze, also dem Rand des zulässigen Bereichs, beliebig nähert. Aus diesem Grund wurde in [53] gezeigt, dass die Folge der Iterierten $\{x^k\}_k$ unter bestimmten Voraussetzungen von 0 weg beschränkt ist. Hierbei wurde μ fest gewählt und so eine feste Schranke für x^k gefunden.

Die Voraussetzungen 3.1 aus dem vorherigen Abschnitt werden nun für ungleichungsstringierte Probleme erweitert:

Voraussetzungen 3.2 Sei $\{x^k\}_k$ die Folge, die von dem auf (3.47) angewandten Filter-Verfahren erzeugt wurde, der Startpunkt $x^0 > 0$ und $\mathbf{s}_b = \mathbf{1}$. Die Feasibility Restoration Phase werde stets erfolgreich mit $x^{k+1} > 0$ beendet und der Algorithmus ende nicht an einem KKT-Punkt. Außerdem seien folgende Bedingungen erfüllt:

(V1) Es existiere eine offene Menge $C \subset \mathbb{R}^n$ mit $[x^k, x^k + \alpha_k^{\max} \Delta x^k] \subset C \quad \forall k \notin \mathcal{R}_{inc}$ sodass f und c differenzierbar sind auf C und $f, \nabla f, c$ sowie ∇c auf C beschränkt und Lipschitz-stetig sind.

(V2) Die Approximationen H_k der Hessematrix der zu (3.47) gehörigen Lagrangefunktion seien für alle $k \notin \mathcal{R}_{inc}$ gleichmäßig beschränkt.

(V3) Die Matrizen $H_k + \mu X_k^{-2}$ seien gleichmäßig positiv definit auf dem Nullraum von $A(x^k)^T$, d.h. es existiert eine Konstante $M_H > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\lambda_{\min} \left(\tilde{Z}_k^T (H_k + \mu X_k^{-2}) \tilde{Z}_k \right) \geq M_H.$$

(V4) Es existiere eine Konstante $M_A > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\sigma_{\min} (A(x^k)) \geq M_A.$$

(V5) Die Iterierten, für die die Feasibility Restoration Phase während der Berechnung einer neuen Suchrichtung gestartet wird, seien nicht beliebig nahe am zulässigen Bereich, d.h. es gibt eine Konstante $\theta_{inc} > 0$ sodass $k \notin \mathcal{R}_{inc}$ ist, wenn $\theta(x^k) \leq \theta_{inc}$ gilt.

(V6) Die Iterierten $\{x^k\}_k$ seien beschränkt.

(V7) An allen zulässigen Häufungspunkten \bar{x} von $\{x^k\}_k$ seien die Gradienten der aktiven Nebenbedingungen

$$\nabla c_1(\bar{x}), \dots, \nabla c_m(\bar{x}) \quad \text{und} \quad \{e_i \mid \bar{x}_i = 0\}$$

linear unabhängig.

(V8) Es existieren Konstanten $\zeta_\theta, \zeta_x > 0$ sodass in allen Iterationen, in denen die Feasibility Restoration Phase mit $\theta(x^k) \leq \zeta_\theta$ gestartet wird, eine neue Iterierte x^{k+1} berechnet wird, für die gelte

$$x_i^{k+1} \geq x_i^k \quad \text{für alle } i \text{ mit } x_i^k \leq \zeta_x.$$

Die Voraussetzungen 3.2 (V1) - (V5) entsprechen den Voraussetzungen 3.1. Während sich Voraussetzung 3.2 (V1) und (V2) auf die Funktionen des eigentlichen Optimierungsproblems (3.47) beziehen, betrachtet 3.2 (V3) die Hessematrix der Lagrangefunktion des Barriereproblems (3.48). Aus Voraussetzung (V3) und der Korrektur der Matrix Σ_k folgt die Beschränktheit des minimalen Eigenwerts von $H_k + \Sigma_k$.

Aus Voraussetzung 3.2 (V6) folgt, dass die Folge $\{B(x^k, \mu)\}_k$ nach unten beschränkt ist, eine Beschränkung nach oben ist jedoch nicht möglich.

In [53] wird empfohlen, die *Feasibility Restoration Phase* aufzurufen, sobald $\sigma_{\min}(A(x^k)) \leq b_3\theta(x^k)$ für ein fixes $b_3 > 0$ gilt. Voraussetzung 3.2 (V7) impliziert in diesem Fall (V4) und (V5).

In [53, S. 27] wird auch gezeigt, dass die Forderung von Voraussetzung 3.2 (V8) durchaus erfüllbar ist. Durch die Bedingung an den Startvektor x^0 , die *Fraction-to-the-boundary* Regel und die Voraussetzung 3.2 (V8) liegen alle Iterierten im strikt zulässigen Bereich. Zusätzlich werden hier im Vergleich zu den Voraussetzungen für rein gleichungsrestringierte Probleme $s_b = 1$ sowie $x^{k+1} > 0$ für die Ergebnisse einer *Feasibility Restoration Phase* gefordert.

Satz 3.5 (Beschränkung der Iterierten)

Seien die Voraussetzungen 3.2 erfüllt. Dann existiert eine Konstante $\varepsilon_x > 0$ sodass

$$x^k \geq \varepsilon_x e$$

für alle Iterationen k gilt.

Der Beweis dieses Satzes [53, Theorem 3] verwendet neben einem Lemma [53, Lemma 11], das aus den Voraussetzungen 3.2 folgt, auch Satz 3.3 aus dem vorangehenden Abschnitt. Da die Beschränkung der Werte der Barrierefunktion nach unten bereits bekannt ist und in den Voraussetzungen 3.2 $s_b = 1$ gefordert wurde, ist die Verwendung von Satz 3.3 hier ohne einen Zirkelschluss möglich.

Aus Satz 3.5 folgt die Beschränkung der Barrierefunktion und deren Ableitung in den Punkten x^k , wodurch die Voraussetzungen 3.1 für das Barriereproblem (3.48) erfüllt sind. Damit gilt die zuvor gezeigte globale Konvergenz auch für die Filter-Liniensuche innerhalb des Innere-Punkte-Verfahrens mit festem Barriereparameter.

In der Praxis wird das Verfahren, wie in Abschnitt 3.2 beschrieben, mit unterschiedlichen Barriereparametern μ_j verwendet. Das in Satz 3.5 verwendete ε_x ist jedoch proportional zu $(1 - \tau)$. Nach Gleichung (3.25) gilt $\tau \xrightarrow{\mu \rightarrow 0} 1$, sodass die Konvergenz zu möglichen Lösungen am Rand des Zulässigkeitsbereichs gewährleistet ist.

3.3.3 Globale Konvergenz der adaptiven Strategie

In [39] wurde die globale Konvergenz des Innere-Punkte-Verfahren mit Filter-Liniensuche und adaptiver Strategie für die Wahl des Barriereparameters (siehe Algorithmus 3.2) gezeigt. Für den Fall, dass der monotone Modus stets wieder verlassen wird, wurde folgender Satz bewiesen:

Satz 3.6 (Konvergenz des adaptiven Filter-Verfahrens)

Sei $\{(x^k, \lambda^k, z^{L,k}, z^{U,k})\}_k$ die von Algorithmus 3.2 erzeugte Folge, der monotone Modus werde stets erfolgreich verlassen, $\{B(x^k, \mu_k)\}_k$ sei nach unten und $\{\theta(x^k)\}_k$ nach oben beschränkt.

Dann konvergiert der KKT-Fehler gegen 0, d.h. $\lim_{k \rightarrow \infty} E_0(x^k, \lambda^k, z^{L,k}, z^{U,k}) = 0$.

Verbleibt der Algorithmus jedoch im monotonen Modus, so gilt die Aussage nach Fiacco und McCormick auch.

3.4 Konvergenz des Filter-Verfahrens bei Störungen in den Nebenbedingungen

Wie in Abschnitt 2.2.1 beschrieben, können die Nebenbedingungen eines Optimierungsproblems sowie deren Ableitungen Störungen unterliegen. Dabei ist es interessant, zu untersuchen, welche Voraussetzungen diese Störungen erfüllen müssen, um die globale Konvergenz des Innere-Punkte-Verfahrens mit Filter-Liniensuche noch beweisen zu können.

In den Gleichungssystemen und Funktionsvorschriften des Algorithmus des Innere-Punkte-Verfahrens werden nun alle Auswertungen der Funktion $c : \mathbb{R}^n \rightarrow \mathbb{R}^m$ an einer Stelle x durch $c^\delta(x)$ und die Auswertung der transponierten Jacobimatrix A an einer Stelle x durch $A^\delta(x)$ ersetzt.

Für die Störungen in den Nebenbedingungen c und deren Ableitungen A soll folgende Fehlerabschätzungen gelten:

$$\|c^\delta(x^k) - c(x^k)\| \leq \delta_c^k \quad \text{und} \quad \|A^\delta(x^k) - A(x^k)\| \leq \delta_A^k \quad \text{mit} \quad \delta_c^k, \delta_A^k \geq 0 \quad (3.51)$$

Da die Zielfunktion f nicht von Störungen betroffen ist (vgl. Abschnitt 2.2.1), gilt dies auch für die Barrierefunktion B .

Die Verletzung der Nebenbedingungen ist dann $\theta^\delta(x) := \|c^\delta(x)\|$, wobei gilt

$$|\theta^\delta(x^k) - \theta(x^k)| = \left| \|c^\delta(x^k)\| - \|c(x^k)\| \right| \leq \|c^\delta(x^k) - c(x^k)\| \leq \delta_c^k \quad (3.52)$$

Im Folgenden wird nun die globale Konvergenz des Innere-Punkte-Verfahrens mit Filter-Liniensuche für gestörte Nebenbedingungen bewiesen. Dies geschieht entlang des Beweises in [53], dessen Ergebnisse in Abschnitt 3.3.2 zusammengefasst wurden. Auch hier wird wieder zuerst die Konvergenz für rein gleichungsrestringierte Probleme gezeigt und dann für Probleme mit Beschränkung der Optimierungsvariablen erweitert.

3.4.1 Globale Konvergenz für gleichungsrestringierte Probleme mit gestörten Gleichungsnebenbedingungen

Betrachtet werden Probleme der Form

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u.d.N.} \quad c^\delta(x) = 0 \quad (3.53)$$

mit den Funktionen $f : \mathbb{R}^n \rightarrow \mathbb{R}$ und $c^\delta : \mathbb{R}^n \rightarrow \mathbb{R}^m$. Es wird das in Abschnitt 3.3.2.1

beschriebene Filter-Verfahren angewendet und dabei in jeder Iteration das lineare Gleichungssystem

$$\begin{pmatrix} H_k & A^\delta(x^k) \\ A^\delta(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \lambda_+^k \end{pmatrix} = - \begin{pmatrix} \nabla f(x^k) \\ c^\delta(x^k) \end{pmatrix} \quad (3.54)$$

mit Hilfe des reduzierten Ansatzes, wie in Abschnitt 3.2.4 beschrieben, gelöst. Dabei werden Matrizen $\tilde{Y}^\delta \in \mathbb{R}^{n \times m}$ und $\tilde{Z}^\delta \in \mathbb{R}^{n \times (n-m)}$ gewählt, deren Spalten eine Orthonormalbasis des \mathbb{R}^n bilden und außerdem $A^\delta(x)^T \tilde{Z}^\delta = 0$ gilt. Daraus werden dann wie in (3.22) und (3.23) Δx^k , $\Delta x_{\tilde{Y}}^{\delta k}$ und $\Delta x_{\tilde{Z}}^{\delta k}$ berechnet.

Definition 3.10 (Optimalitätsmaß des gestörten Problems)

Es sei die folgende Abbildung definiert:

$$\chi^\delta : \{x^k\}_k \rightarrow [0, \infty], \quad \chi^\delta(x^k) = \begin{cases} \|\Delta x_{\tilde{Z}}^{\delta k}\|_2 = \|\tilde{Z}_k^\delta \Delta x_{\tilde{Z}}^{\delta k}\|_2, & \text{falls } k \notin \mathcal{R}_{inc} \\ \infty, & \text{sonst} \end{cases} \quad (3.55)$$

Aufgrund der Störungen in den Nebenbedingungen müssen die Voraussetzungen 3.1 für die globale Konvergenz des Filterverfahrens nun erweitert werden. Für den ersten Teil des Beweises lauten sie:

Voraussetzungen 3.3 Sei $\{x^k\}_k$ die Folge, die von dem auf (3.53) angewandten Filter-Verfahren erzeugt wurde, x^0 der Startvektor und $s_b \geq 1$. Die Feasibility Restoration Phase werde stets erfolgreich beendet und der Algorithmus ende nicht an einem KKT-Punkt. Außerdem seien folgende Bedingungen erfüllt:

(V1) Es existiere eine offene Menge $C \subset \mathbb{R}^n$ mit $[x^k, x^k + \Delta x^k] \subset C \quad \forall k \notin \mathcal{R}_{inc}$ sodass f und c differenzierbar sind auf C und f , ∇f , c sowie ∇c auf C beschränkt und Lipschitz-stetig sind.

(V2) Die Approximationen H_k der Hessematrix der zu (3.53) gehörigen Lagrangefunktion seien für alle $k \notin \mathcal{R}_{inc}$ gleichmäßig beschränkt.

(V3) Die Matrizen H_k seien gleichmäßig positiv definit auf dem Nullraum von $A^\delta(x^k)^T$, d.h. es existiert eine Konstante $M_H^\delta > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\lambda_{\min} \left(\tilde{Z}_k^{\delta T} H_k \tilde{Z}_k^\delta \right) \geq M_H^\delta.$$

(V4) Es existiere eine Konstante $M_A^\delta > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\sigma_{\min} \left(A^\delta(x^k) \right) \geq M_A^\delta.$$

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

(V5) Die Iterierten, für die die Feasibility Restoration Phase während der Berechnung einer neuen Suchrichtung gestartet wird, seien nicht beliebig nahe am zulässigen Bereich, d.h. es gibt eine Konstante $\theta_{inc} > 0$ sodass $k \notin \mathcal{R}_{inc}$ ist, wenn $\theta^\delta(x^k) \leq \theta_{inc}$ gilt.

(V9) Für die Störung der Nebenbedingungen gelte Gleichung (3.51) und außerdem existiere eine Schranke C_δ sodass für alle $k \in \mathbb{N}$ gilt

$$0 \leq \delta_c^k, \delta_A^k \leq C_\delta < \infty.$$

Analog zur Vorgehensweise in [53] werden zunächst die vorbereitenden Lemmata (vgl. [53, Lemma 1-4]), sowie die Zulässigkeit der Häufungspunkte der Iterierten (vgl. [53, Theorem 1]) bewiesen. Sind bestimmte Voraussetzungen an die Störungen erfüllt, so bleiben die Aussagen größtenteils erhalten.

Als erstes wird [53, Lemma 1] betrachtet:

Lemma 3.2 Seien die Voraussetzungen 3.3 erfüllt. Dann existieren Konstanten $M_d, M_\lambda, M_m > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\|\Delta x^k\| \leq M_d, \quad \|\lambda_+^k\| \leq M_\lambda, \quad |\nabla f(x^k)^T \Delta x^k| \leq M_m.$$

Der Beweis verläuft analog zu dem aus [53, Lemma 1], die Aussagen folgen direkt aus den Voraussetzungen 3.3 (V1)-(V4), (V9) und der Gleichung (3.54).

Lemma 3.3 Seien die Voraussetzungen 3.3 erfüllt und $\{x^{k_i}\}_i$ eine Teilfolge der Iterierten, wobei $\chi^\delta(x^{k_i}) \geq \varepsilon$ für ein $\varepsilon > 0$ unabhängig von i gilt.

Dann gibt es Konstanten $\varepsilon_1, \varepsilon_2 > 0$ sodass für alle i gilt

$$\theta^\delta(x^{k_i}) \leq \varepsilon_1 \quad \Rightarrow \quad \nabla f(x^{k_i})^T \Delta x^{k_i} \leq -\varepsilon_2.$$

Dieses Lemma entspricht [53, Lemma 2] und folgt aus den Voraussetzungen 3.3 (V1)-(V3) und den Berechnungsvorschriften für Δx^{k_i} .

Die Abschätzung aus [53, Lemma 3] wird ersetzt durch

Lemma 3.4 Es gelte Voraussetzung 3.3 (V1). Dann existieren Konstanten $C_\theta, C_f > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ und $\alpha \leq 1$ gilt

$$|f(x^k + \alpha \Delta x^k) - f(x^k) - \alpha \nabla f(x^k)^T \Delta x^k| \leq C_f \alpha^2 \|\Delta x^k\|^2 \quad (3.56)$$

$$\begin{aligned} |\theta^\delta(x^k + \alpha \Delta x^k) - (1 - \alpha) \theta^\delta(x^k)| &\leq \delta_c^k + \|c^\delta(x^k + \alpha \Delta x^k) - c(x^k + \alpha \Delta x^k)\| \\ &\quad + \delta_A^k \alpha \|\Delta x^k\| + C_\theta \alpha^2 \|\Delta x^k\|^2 \end{aligned} \quad (3.57)$$

Beweis: Die Abschätzung für die Zielfunktion ändert sich gegenüber [53, Lemma 3] nicht, da diese keiner Störung unterliegt. Für die gestörten Nebenbedingungen hingegen gilt:

$$\begin{aligned}
 c^\delta(x^k + \alpha\Delta x^k) &= c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k) + c(x^k + \alpha\Delta x^k) \\
 &= c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k) + c(x^k) + \alpha A(x^k)^T \Delta x^k \\
 &\quad + \int_0^1 \alpha [A(x^k + t\alpha\Delta x^k) - A(x^k)]^T \Delta x^k dt \\
 &= c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k) + c(x^k) + \alpha [A(x^k) - A^\delta(x^k)]^T \Delta x^k + \alpha A^\delta(x^k)^T \Delta x^k \\
 &\quad + \int_0^1 \alpha [A(x^k + t\alpha\Delta x^k) - A(x^k)]^T \Delta x^k dt \\
 &\stackrel{(3.54)}{=} c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k) + c(x^k) - c^\delta(x^k) + c^\delta(x^k) \\
 &\quad + \alpha [A(x^k) - A^\delta(x^k)]^T \Delta x^k - \alpha c^\delta(x^k) + \int_0^1 \alpha [A(x^k + t\alpha\Delta x^k) - A(x^k)]^T \Delta x^k dt
 \end{aligned}$$

Damit gilt

$$\begin{aligned}
 \|c^\delta(x^k + \alpha\Delta x^k) - (1 - \alpha)c^\delta(x^k)\| &\leq \|c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k)\| \\
 &\quad + \|c^\delta(x^k) - c(x^k)\| + \alpha \|A^\delta(x^k) - A(x^k)\| \|\Delta x^k\| \\
 &\quad + \int_0^1 \alpha \|A(x^k + t\alpha\Delta x^k) - A(x^k)\| \|\Delta x^k\| dt
 \end{aligned}$$

und damit

$$\begin{aligned}
 |\theta^\delta(x^k + \alpha\Delta x^k) - (1 - \alpha)\theta^\delta(x^k)| &\leq \|c^\delta(x^k + \alpha\Delta x^k) - (1 - \alpha)c^\delta(x^k)\| \\
 &\leq \|c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k)\| + \delta_c^k + \alpha \delta_A^k \|\Delta x^k\| + C_\theta \alpha^2 \|\Delta x^k\|^2
 \end{aligned}$$

□

Lemma 3.5 *Seien die Voraussetzungen 3.3 erfüllt und der Filter werde nur endlich oft erweitert. Dann gilt $\lim_{k \rightarrow \infty} \theta^\delta(x^k) = 0$.*

Dieses Lemma folgt aus der Konstruktion des Verfahrens, der Voraussetzung 3.3 (V1), sowie Lemma 3.2. Der Beweis läuft analog zu [53, Lemma 5].

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

Lemma 3.6 *Sei $\{x^{k_i}\}_{k_i}$ eine Teilfolge der Iterierten, wobei der Filter in jeder Iteration k_i erweitert wurde. Außerdem existieren Konstanten $c_f \in \mathbb{R}$ und $c_\theta > 0$ sodass für alle $i \in \mathbb{N}$ gilt $f(x^{k_i}) \geq c_f$ und $\theta^\delta(x^{k_i}) \leq c_\theta$ (z.B. durch Voraussetzung 3.3 (V1)). Dann folgt*

$$\lim_{i \rightarrow \infty} \theta^\delta(x^{k_i}) = 0.$$

Dieses Lemma entspricht [53, Lemma 6]. Dieses wurde in [26] aus der Konstruktion des Filters sowie der Beschränkung der Zielfunktion und der Verletzung der Nebenbedingungen nach unten bewiesen. Dieser Beweis kann genauso für gestörte Nebenbedingungen geführt werden.

Satz 3.7 *Seien die Voraussetzungen 3.3 erfüllt. Dann gilt*

$$\lim_{k \rightarrow \infty} \theta^\delta(x^k) = 0.$$

Dieser Satz folgt aus Lemma 3.5, Lemma 3.6, der Beschränkung von $\{f(x^k)\}_k$ nach unten sowie der Konstruktion des Verfahrens. Der Beweis verläuft genauso wie der von [53, Theorem 1]. Auch Anmerkung [53, Remark 6] bleibt für gestörte Nebenbedingungen bestehen: Satz 3.7 gilt für den Fall $s_b = 1$ auch dann, wenn $\{f(x^k)\}_k$ und $\{\|\nabla f(x^k)\|\}_k$ nach oben unbeschränkt sind.

Mit den nun bekannten Abschätzungen können die Forderungen an die Störungen δ_c und δ_A , die für die globale Konvergenz des Filter-Verfahrens nötig sind, formuliert werden:

Voraussetzungen 3.4 *Es existiere ein $K_\delta \in \mathbb{N}$, sodass für die Störungen der Nebenbedingungen und deren Ableitungen für alle $k \geq K_\delta$ gelte*

$$(V10) \text{ Für alle } \alpha \in [0, 1] \text{ sei } \|c^\delta(x^k + \alpha\Delta x^k) - c(x^k + \alpha\Delta x^k)\| \leq \delta_c^k.$$

$$(V11) \text{ Falls } \theta^\delta(x^k) > 0 \text{ ist, dann gelte } \delta_c^k < \frac{\theta^\delta(x^k)^2}{8C_\theta M_d^2};$$

falls $\theta^\delta(x^k) = 0$ ist, dann gelte $\delta_c^k = 0$.

$$(V12) \text{ Falls } \theta^\delta(x^k) > 0 \text{ ist, dann gelte}$$

$$\delta_A^k < \frac{\theta^\delta(x^k)}{M_d} - \sqrt{8C_\theta \delta_c^k}, \quad \text{d.h.} \quad \theta^\delta(x^k) - \delta_A^k M_d > \sqrt{8C_\theta \delta_c^k M_d^2};$$

falls $\theta^\delta(x^k) = 0$ ist, dann gelte $\delta_A^k = 0$.

(V13) Falls

$$\theta^\delta(x^k) - \delta_A^k M_d > \alpha_k^{\min} \theta^\delta(x^k)$$

mit α_k^{\min} aus Gleichung (3.35) - hier jedoch mit der Zielfunktion f anstelle von B - gilt, dann sei auch

$$8 C_\theta M_d^2 \delta_c^k \leq (\alpha_k^{\min} \theta^\delta(x^k))^2$$

erfüllt.

Lemma 3.7 Gelten beide Voraussetzungen 3.3 und 3.4, so ist

$$\lim_{k \rightarrow \infty} \delta_c^k = 0 \quad \text{und} \quad \lim_{k \rightarrow \infty} \delta_A^k = 0.$$

Beweis: Aus der Voraussetzung 3.3 (V9) und Voraussetzung 3.4 (V11) folgt

$$0 \leq \delta_c^k \leq \frac{\theta^\delta(x^k)^2}{8 C_\theta M_d^2}$$

und daraus mit Satz 3.7 die erste Aussage. Aus Voraussetzung 3.3 (V9) und Voraussetzung 3.4 (V12) folgt

$$0 \leq \delta_A^k \leq \frac{\theta^\delta(x^k)}{M_d} - \sqrt{8 C_\theta \delta_c^k}$$

und mit $\lim_{k \rightarrow \infty} \delta_c^k = 0$ und Satz 3.7 die zweite Aussage. □

Aus Lemma 3.4 und Voraussetzung 3.4 (V10) folgt dann:

Lemma 3.8 Es gelten die Voraussetzungen 3.3 (V1) und 3.4 (V10). Dann existieren Konstanten $C_\theta, C_f > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ und $\alpha \leq 1$ gilt

$$|f(x^k + \alpha \Delta x^k) - f(x^k) - \alpha \nabla f(x^k)^T \Delta x^k| \leq C_f \alpha^2 \|\Delta x^k\|^2 \quad (3.56)$$

$$|\theta^\delta(x^k + \alpha \Delta x^k) - (1 - \alpha) \theta^\delta(x^k)| \leq 2 \delta_c^k + \delta_A^k \alpha \|\Delta x^k\| + C_\theta \alpha^2 \|\Delta x^k\|^2 \quad (3.58)$$

Unter den Voraussetzungen 3.4 lässt sich nun zeigen, dass die Abbildung χ^δ ein Optimalitätsmaß für das gestörte Problem (3.53) ist:

Lemma 3.9 (gestörtes Optimalitätsmaß)

Seien die Voraussetzungen 3.3 und 3.4 erfüllt. Dann ist χ^δ ein Optimalitätsmaß erster Ordnung für (3.53).

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

Beweis: Der Beweis erfolgt parallel zu dem von Lemma 3.1:

Sei $\{x^{k_j}\}_j$ die Teilfolge mit $\lim_{j \rightarrow \infty} \chi^\delta(x^{k_j}) = 0$ und $\lim_{j \rightarrow \infty} x^{k_j} = x^*$, wobei x^* zulässig bzgl. c^δ ist. Da $\lim_{k \rightarrow \infty} \delta_c^k = 0$ ist, gilt $c^\delta(x^*) = c(x^*)$ und damit ist x^* auch zulässig bzgl. c .

Aus $\chi^\delta(x^k) = \infty$ für $k \in \mathcal{R}_{inc}$ folgt, dass $k_j \notin \mathcal{R}_{inc}$ für j groß genug.

Aus der Stetigkeit von c und Voraussetzung 3.3 (V9) folgt außerdem

$$\begin{aligned} \|c^\delta(x^{k_j})\| &= \|c^\delta(x^{k_j}) - c^\delta(x^*)\| \leq \|c^\delta(x^{k_j}) - c(x^{k_j})\| + \|c(x^{k_j}) - c(x^*)\| \\ &\leq \delta_c^{k_j} + \|c(x^{k_j}) - c(x^*)\| \xrightarrow{j \rightarrow \infty} 0. \end{aligned}$$

Wegen Voraussetzung 3.3 (V4) und der Konstruktion von Δx_Y^δ gilt dann $\lim_{j \rightarrow \infty} \Delta x_Y^{\delta k_j} = 0$.

Aus $\lim_{j \rightarrow \infty} \chi^\delta(x^{k_j}) = 0$, Voraussetzung 3.3 (V3) und der Konstruktion von Δx_Z^δ folgt dann

$$\lim_{j \rightarrow \infty} \tilde{Z}_{k_j}^{\delta T} \nabla f(x^{k_j}) = 0.$$

Da $\lim_{k \rightarrow \infty} \delta_A^k = 0$, ist in x^* keine Störung vorhanden und damit $\tilde{Z}_*^\delta = \tilde{Z}_*$ und $A^\delta(x^*) = A(x^*)$. Damit gilt wie in Lemma 3.1 $\tilde{Z}_*^T \nabla f(x^*) = 0$, woraus folgt, dass x^* die KKT-Bedingungen des inzwischen ungestörten, gleichungsrestringierten Problems (3.53) und damit auch von (3.44) erfüllt. \square

Damit kann nun der Beweis der globalen Konvergenz mit Hilfe von χ^δ fortgesetzt werden.

Lemma 3.10 *Seien die Voraussetzungen 3.3 und 3.4 erfüllt. Dann gilt für alle k*

$$\theta^\delta(x^k) = 0 \quad \Rightarrow \quad \nabla f(x^k)^T \Delta x^k < 0$$

und

$$\Theta_k := \min \{ \theta \mid (\theta, f) \in \mathcal{F}_k \} > 0.$$

Die Aussage folgt analog wie bei [53, Lemma 4] aus den Voraussetzungen 3.3, der Konstruktion des Verfahrens, sowie der Eigenschaft von χ^δ als Optimalitätsmaß.

Lemma 3.11 *Seien die Voraussetzungen 3.3 erfüllt, $\{x^{k_i}\}_i$ eine Teilfolge der Iterierten mit $k_i \notin \mathcal{R}_{inc}$ und $\nabla f(x^{k_i})^T \Delta x^{k_i} \leq -\varepsilon_2$ für ein $\varepsilon_2 > 0$ unabhängig von k_i .*

Dann gilt mit η_b aus Gleichung (3.32) für alle k_i und alle $\alpha \leq \bar{\alpha} = \frac{(1 - \eta_b)\varepsilon_2}{C_f M_d^2}$

$$f(x^{k_i} + \alpha \Delta x^{k_i}) - f(x^{k_i}) \leq \eta_b \alpha \nabla f(x^{k_i})^T \Delta x^{k_i}.$$

Dieses Lemma folgt aus Lemma 3.2 und Lemma 3.4. Der Beweis läuft genauso wie in [53, Lemma 7], da von Lemma 3.4 nur die erste Aussage über die Zielfunktion verwendet wird.

Lemma 3.12 *Seien die Voraussetzungen 3.3 und 3.4 erfüllt und der Filter werde nur endlich oft erweitert. Dann gilt $\lim_{k \rightarrow \infty} \chi^\delta(x^k) = 0$.*

Beweis: Analog zu [53, Lemma 8] wird wieder die Gegenannahme, dass eine Teilfolge $\{x^{k_i}\}_i$ mit $\chi^\delta(x^{k_i}) \geq \epsilon$ existiert, getroffen. Aus Lemma 3.5 folgt $\lim_{k \rightarrow \infty} \theta^\delta(x^k) = 0$. Sei $\tilde{K} \in \mathbb{N}$ so gewählt, dass für alle $k_i \geq \tilde{K}$ der Filter nicht mehr erweitert werde und $\theta^\delta(x^{k_i}) \leq \epsilon_1$ gilt. Damit folgt aus Lemma 3.3 $\nabla f(x^{k_i})^T \Delta x^{k_i} \leq -\epsilon_2 < 0$. Da der Filter nicht erweitert wird, müssen für alle $k \geq \tilde{K}$ die Schaltbedingung (3.31) und die Armijo-artige Bedingung (3.32), jedoch mit f anstelle von B , erfüllt sein (siehe Anmerkung 3.2). Damit ergibt sich (ähnlich wie im Beweis von [53, Lemma 5])

$$\begin{aligned}
 f(x^{\tilde{K}+j}) &= f(x^{\tilde{K}}) + \sum_{l=\tilde{K}}^{\tilde{K}+j-1} (f(x^{l+1}) - f(x^l)) \\
 &= f(x^{\tilde{K}}) + \sum_{\substack{l=\tilde{K} \\ x^l \text{ aus Teilfolge}}}^{\tilde{K}+j-1} (f(x^{l+1}) - f(x^l)) + \sum_{\substack{l=\tilde{K} \\ x^l \text{ nicht aus Teilfolge}}}^{\tilde{K}+j-1} (f(x^{l+1}) - f(x^l)) \\
 &\stackrel{(3.32)}{\leq} f(x^{\tilde{K}}) + \sum_{\substack{l=\tilde{K} \\ x^l \text{ aus Teilfolge}}}^{\tilde{K}+j-1} \eta_b \alpha_l \nabla f(x^l)^T \Delta x^l + \sum_{\substack{l=\tilde{K} \\ x^l \text{ nicht aus Teilfolge}}}^{\tilde{K}+j-1} \eta_b \alpha_l \nabla f(x^l)^T \Delta x^l \\
 &\stackrel{(3.31)}{<} f(x^{\tilde{K}}) + \sum_{\substack{l=\tilde{K} \\ x^l \text{ aus Teilfolge}}}^{\tilde{K}+j-1} \eta_b \alpha_l \nabla f(x^l)^T \Delta x^l \\
 &\stackrel{\text{Lemma 3.3}}{\leq} f(x^{\tilde{K}}) + \sum_{\substack{l=\tilde{K} \\ x^l \text{ aus Teilfolge}}}^{\tilde{K}+j-1} -\eta_b \alpha_l \epsilon_2
 \end{aligned}$$

Da $f(x^{\tilde{K}+j})$ nach unten beschränkt ist für $j \rightarrow \infty$, so muss die Reihe auf der rechten Seite ebenfalls beschränkt sein, woraus $\lim_{k_i \rightarrow \infty} \alpha_{k_i} = 0$ folgt. Für alle $k_i \geq K \geq \max\{\tilde{K}, K_\delta\}$ sei $\alpha_{k_i} < 1$, d.h. die erste Testschrittweite $\alpha_{k_i,0} = 1$ wurde nicht akzeptiert. Sei $\alpha_{k_i,l_i} \in [\frac{1}{\nu_2} \alpha_{k_i}, \frac{1}{\nu_1} \alpha_{k_i}]$ mit ν_1, ν_2 aus der Berechnung der Testschrittweiten (3.34), die letzte nicht akzeptierte Testschrittweite. Da in der Iteration k_i der Filter nicht mehr erweitert wird, muss die Schaltbedingung (3.31) mit f statt B erfüllt sein. Folglich muss der Testpunkt die Armijo-artige Bedingung (3.32) verletzen, d.h.

$$f(x^{k_i} + \alpha_{k_i,l_i} \Delta x^{k_i}) - f(x^{k_i}) > \eta_b \alpha_{k_i,l_i} \nabla f(x^{k_i})^T \Delta x^{k_i} \quad (3.59)$$

oder er wird vom Filter nicht akzeptiert, d.h.

$$(\theta^\delta(x^{k_i} + \alpha_{k_i,l_i} \Delta x^{k_i}), f(x^{k_i} + \alpha_{k_i,l_i} \Delta x^{k_i})) \in \mathcal{F}_{k_i} = \mathcal{F}_K. \quad (3.60)$$

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

Aus $\lim_{i \rightarrow \infty} \alpha_{k_i} = 0$ und Lemma 3.11 folgt jedoch, dass (3.59) irgendwann für k_i groß genug nicht mehr erfüllt sein kann.

Sei $\Theta_K := \min\{\theta | (\theta, f) \in \mathcal{F}_K\}$ für das aus Lemma 3.10 $\Theta_K > 0$ folgt. Aus Lemma 3.2 und 3.8 gilt dann

$$\theta^\delta(x^{k_i} + \alpha_{k_i, l_i} \Delta x^{k_i}) \leq (1 - \alpha_{k_i, l_i}) \theta^\delta(x^{k_i}) + 2\delta_c^{k_i} + \delta_A^{k_i} \alpha_{k_i, l_i} M_d + C_\theta \alpha_{k_i, l_i}^2 M_d^2.$$

Aus $\lim_{i \rightarrow \infty} \alpha_{k_i, l_i} = 0$, $\lim_{i \rightarrow \infty} \theta^\delta(x^{k_i}) = 0$ und $\lim_{i \rightarrow \infty} \delta_c^{k_i} = 0$ folgt dann $\theta^\delta(x^{k_i} + \alpha_{k_i, l_i} \Delta x^{k_i}) < \Theta_K$ für i groß genug, was (3.60) widerspricht.

Damit muss die Gegenannahme falsch gewesen sein und das Lemma ist bewiesen. \square

Entsprechend [53, Lemma 9] wird folgendes Lemma, jedoch nur für den Fall $\theta^\delta(x) > 0$ für welchen dieses Lemma später verwendet wird, formuliert:

Lemma 3.13 *Seien die Voraussetzungen 3.3 und 3.4 erfüllt, $\{x^{k_i}\}_i$ eine Teilfolge der Iterierten mit $k_i \notin \mathcal{R}_{inc}$, $\theta^\delta(x^{k_i}) > 0$ und $\nabla f(x^{k_i})^T \Delta x^{k_i} \leq -\varepsilon_2$ für ein $\varepsilon_2 > 0$ unabhängig von k_i .*

Dann existieren ein $K \in \mathbb{N}$ und $\bar{\alpha}_1, \bar{\alpha}_2 \in [0, 1]$ mit $\bar{\alpha}_1 > \bar{\alpha}_2$, sodass für alle $k_i \geq K$ und $\alpha \in [\bar{\alpha}_2, \bar{\alpha}_1]$ gilt

$$(\theta^\delta(x^{k_i} + \alpha \Delta x^{k_i}), f(x^{k_i} + \alpha \Delta x^{k_i})) \notin \mathcal{F}_{k_i}.$$

Beweis: Aus der Konstruktion des Filters folgt $(\theta^\delta(x^{k_i}), f(x^{k_i})) \notin \mathcal{F}_{k_i}$. Nun gilt es Grenzen für α zu finden, sodass gilt

$$f(x^{k_i} + \alpha \Delta x^{k_i}) \leq f(x^{k_i}) \quad \text{und} \quad (3.61)$$

$$\theta^\delta(x^{k_i} + \alpha \Delta x^{k_i}) \leq \theta^\delta(x^{k_i}). \quad (3.62)$$

Aus den Lemmata 3.2 und 3.4 folgt

$$f(x^{k_i} + \alpha \Delta x^{k_i}) - f(x^{k_i}) \leq \alpha \nabla f(x^{k_i})^T \Delta x^{k_i} + C_f \alpha^2 M_d^2$$

und damit ist (3.61) für $\alpha \leq \min\left\{1, \frac{\varepsilon_2}{M_d^2 C_f}\right\}$ erfüllt.

Für alle $k_i \geq K_\delta$ folgt aus Lemma 3.2 und 3.8 für die Verletzung der Nebenbedingungen

$$\theta^\delta(x^{k_i} + \alpha \Delta x^{k_i}) - \theta^\delta(x^{k_i}) \leq -\alpha \theta^\delta(x^{k_i}) + 2\delta_c^{k_i} + \delta_A^{k_i} \alpha M_d + C_\theta \alpha^2 M_d^2.$$

Die Ungleichung (3.62) ist folglich für alle $k_i \geq K_\delta$ und $\alpha_2 \leq \alpha \leq \alpha_1$ mit

$$\alpha_{1,2} = \frac{1}{2C_\theta M_d^2} \left(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d \pm \sqrt{(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i}} \right)$$

erfüllt.

Sei $K \in \mathbb{N}$ so gewählt, dass $K \geq K_\delta$ und für alle $k_i \geq K$ gilt

$$\theta^\delta(x^{k_i}) < \min \left\{ 2C_\theta M_d^2, 2\frac{\varepsilon_2 C_\theta}{C_f} \right\}. \quad (3.63)$$

Die Existenz von K folgt aus Satz 3.7. Da $\theta^\delta(x^{k_i}) > 0$ ist, folgt aus den Voraussetzungen 3.4(V11) und (V12) für α_1 und α_2

- $(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i} > 0$, d.h. α_1 und α_2 existieren und sind nicht gleich.
- $\alpha_1 = \frac{1}{2C_\theta M_d^2} \left(\underbrace{\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d}_{\geq 0} + \underbrace{\sqrt{(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i}}}_{> 0} \right) > 0$
- $\alpha_2 = \frac{1}{2C_\theta M_d^2} \left(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d - \underbrace{\sqrt{(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i}}}_{\geq 0} \right) \geq 0$
- $\alpha_2 < \frac{\theta^\delta(x^{k_i})}{2C_\theta M_d^2} < 1$, da $\theta^\delta(x^{k_i}) < 2C_\theta M_d^2$
- $\alpha_2 < \frac{\theta^\delta(x^{k_i})}{2C_\theta M_d^2} < \frac{\varepsilon_2}{M_d^2 C_f}$, da $\theta^\delta(x^{k_i}) < 2\frac{\varepsilon_2 C_\theta}{C_f}$

Mit $\bar{\alpha}_1 = \min \left\{ 1, \frac{\varepsilon_2}{M_d^2 C_f}, \alpha_1 \right\}$ und $\bar{\alpha}_2 = \alpha_2$ ist die Aussage des Lemmas bewiesen. \square

Lemma 3.14 *Seien die Voraussetzungen 3.3 und 3.4 erfüllt, $\{x^{k_i}\}_{k_i}$ eine Teilfolge der Iterierten mit $\chi^\delta(x^{k_i}) \geq \varepsilon$ für ein $\varepsilon > 0$ unabhängig von k_i .*

Dann existiert ein $K \in \mathbb{N}$ sodass für alle Iterationen $k_i \geq K$ der Filter \mathcal{F}_K nicht mehr erweitert wird.

Beweis: Der Beweis verläuft analog zum Beweis von [53, Lemma 10]. Aus Satz 3.7 folgt $\lim_{i \rightarrow \infty} \theta^\delta(x^{k_i}) = 0$, aus Lemma 3.3 folgt $\exists_{K_1 \in \mathbb{N}, \varepsilon_2 > 0} \forall_{k_i \geq K_1} \nabla f(x^{k_i})^T \Delta x^{k_i} \leq -\varepsilon_2$.

Damit erhält man aus Lemma 3.11, dass für alle k_i und alle $\alpha \leq \bar{\alpha} = \frac{(1 - \eta_b)\varepsilon_2}{C_f M_d^2}$ gilt

$$f(x^{k_i} + \alpha \Delta x^{k_i}) - f(x^{k_i}) \leq \eta_b \alpha \nabla f(x^{k_i})^T \Delta x^{k_i},$$

d.h. die Armijo-artige Bedingung (3.32) mit f statt B ist erfüllt.

Für den Fall $\theta^\delta(x^{k_i}) = 0$ ist die Schaltbedingung (3.31) stets erfüllt, d.h. der Filter wird nicht erweitert.

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

Im Fall $\theta^\delta(x^{k_i}) > 0$ erhält man aus Lemma 3.13 ein $K_2 \in \mathbb{N}$ sodass für alle $k_i \geq K_2$ und für alle $\bar{\alpha}_2 \leq \alpha \leq \bar{\alpha}_1$

$$(\theta^\delta(x^{k_i} + \alpha \Delta x^{k_i}), f(x^{k_i} + \alpha \Delta x^{k_i})) \notin \mathcal{F}_{k_i}.$$

Wähle $K \geq \min\{K_1, K_2, K_\delta\}$ sodass für alle $k_i \geq K$ gilt

$$\begin{aligned} \theta^\delta(x^{k_i}) &< \min \left\{ \theta_{inc}, \left[\frac{\nu_1 \varepsilon_2^{s_b}}{\Gamma_B} \min\{\bar{\alpha}, \bar{\alpha}_1\} \right]^{\frac{1}{s_\theta}}, 2C_\theta M_d^2 \right\} \quad \text{und} \\ \theta^\delta(x^{k_i})^2 &< \min\{\bar{\alpha}, \bar{\alpha}_1\}. \end{aligned} \tag{3.64}$$

Sei $\beta_{k_i} := \min\{\bar{\alpha}, \bar{\alpha}_1\}$ definiert, dann folgt aus den Lemmata 3.11 und 3.13 für den Fall $\bar{\alpha}_2 \leq \alpha_{k_i}^{min}$ für alle Testschrittweiten $\alpha_{k_i, l} \leq \beta_{k_i}$

$$\begin{aligned} f(x^{k_i} + \alpha_{k_i, l} \Delta x^{k_i}) - f(x^{k_i}) &\leq \eta_b \alpha_{k_i, l} \nabla f(x^{k_i})^T \Delta x^{k_i} \quad \text{und} \\ (\theta^\delta(x^{k_i} + \alpha_{k_i, l} \Delta x^{k_i}), f(x^{k_i} + \alpha_{k_i, l} \Delta x^{k_i})) &\notin \mathcal{F}_{k_i} \end{aligned}$$

Sei $\alpha_{k_i, L}$ die erste Testschrittweite, die diese Bedingungen beide erfüllt, dann gilt für alle $\alpha \geq \alpha_{k_i, L}$ wie im Beweis von [53, Lemma 10]

$$\alpha \geq \nu_1 \beta_{k_i} = \nu_1 \min\{\bar{\alpha}, \bar{\alpha}_1\} > \Gamma_B [\theta^\delta(x^{k_i})]^{s_\theta} \varepsilon_2^{-s_b}.$$

Daraus folgt nun ebenfalls die Schaltbedingung (3.31)

$$\Gamma_B [\theta^\delta(x^{k_i})]^{s_\theta} < \alpha \varepsilon_2^{s_b} = \alpha^{1-s_b} (\alpha \varepsilon_2)^{s_b} \leq \alpha^{1-s_b} [-\alpha \nabla f(x^{k_i})^T \Delta x^{k_i}]^{s_b},$$

damit ist $\alpha_{k_i, L}$ die akzeptierte Schrittweite und eine b -Schrittweite, weshalb der Filter auch in diesem Fall nicht erweitert wird.

Bleibt noch zu zeigen, dass $\bar{\alpha}_2 \leq \alpha_{k_i}^{min}$ gilt:

Für den Fall $\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d \leq \alpha_{k_i}^{min} \theta^\delta(x^{k_i})$ lässt sich $\bar{\alpha}_2$ abschätzen durch

$$\begin{aligned} \bar{\alpha}_2 &= \frac{1}{2C_\theta M_d^2} \left(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d - \sqrt{(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i}} \right) \\ &\stackrel{(V12)}{<} \frac{1}{2C_\theta M_d^2} (\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d) < \frac{\theta^\delta(x^{k_i})}{2C_\theta M_d^2} \alpha_{k_i}^{min} \stackrel{(3.64)}{<} \alpha_{k_i}^{min} \end{aligned}$$

Für den Fall $\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d > \alpha_{k_i}^{min} \theta^\delta(x^{k_i})$ gilt nach Voraussetzung 3.4 (V13) $8C_\theta M_d^2 \delta_c^{k_i} \leq (\alpha_{k_i}^{min} \theta^\delta(x^{k_i}))^2$. Daraus erhält man

$$\begin{aligned} (\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i} &\geq (\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - (\alpha_{k_i}^{min} \theta^\delta(x^{k_i}))^2 \\ &= (\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 2(\alpha_{k_i}^{min} \theta^\delta(x^{k_i}))^2 + (\alpha_{k_i}^{min} \theta^\delta(x^{k_i}))^2 \\ &\geq (\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 2(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d) \alpha_{k_i}^{min} \theta^\delta(x^{k_i}) + (\alpha_{k_i}^{min} \theta^\delta(x^{k_i}))^2 \\ &= (\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d - \alpha_{k_i}^{min} \theta^\delta(x^{k_i}))^2 \end{aligned}$$

und damit

$$\sqrt{(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i}} \geq \theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d - \alpha_{k_i}^{\min} \theta^\delta(x^{k_i})$$

Daraus erhält man mit den Bedingungen (3.64) an $\theta^\delta(x^{k_i})$

$$\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d - \sqrt{(\theta^\delta(x^{k_i}) - \delta_A^{k_i} M_d)^2 - 8C_\theta M_d^2 \delta_c^{k_i}} \leq \alpha_{k_i}^{\min} \theta^\delta(x^{k_i}) < 2C_\theta M_d^2 \alpha_{k_i}^{\min}$$

und damit $\bar{\alpha}_2 < \alpha_{k_i}^{\min}$. □

Satz 3.8 *Seien die Voraussetzungen 3.3 und 3.4 erfüllt. Dann gilt $\liminf_{k \rightarrow \infty} \chi^\delta(x^k) = 0$, d.h. wenn die Folge der Iterierten $\{x^k\}_k$ beschränkt ist, so gibt es einen Häufungspunkt dieser Folge, der KKT-Punkt des inzwischen ungestörten, gleichungsrestringierten Problems (3.53) und damit auch von (3.44) ist.*

Der Beweis verläuft wie der für [53, Theorem 2], nun aber mit Hilfe von Satz 3.7 und Lemma 3.14.

3.4.2 Globale Konvergenz für zusätzlich ungleichungsrestringierte Probleme bei gestörten Gleichungsnebenbedingungen

Im zweiten Teil wird das Innere-Punkte-Verfahren für Probleme mit Beschränkung der Optimierungsvariablen betrachtet. Um den Beweis nicht zu technisch werden zu lassen, werden hier wie in [53] nur einseitige Beschränkungen behandelt.

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{u.d.N.} \quad c^\delta(x) = 0, \quad x \geq 0 \quad (3.65)$$

Auf das Problem (3.65) wird das in Abschnitt 3.3.2.2 beschriebene Verfahren angewendet. Für den Konvergenzbeweis wird das Gleichungssystem

$$\begin{pmatrix} H_k + \mu X_k^{-2} & A^\delta(x^k) \\ A^\delta(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} \Delta x^k \\ \lambda_+^k \end{pmatrix} = - \begin{pmatrix} \nabla f(x^k) - \mu X_k^{-1} e \\ c^\delta(x^k) \end{pmatrix} \quad (3.66)$$

mit der „primalen Hessematrix“ $H_k + \mu X_k^{-2}$ betrachtet, das Gleichung (3.50) mit gestörten Nebenbedingungen entspricht. Auch hier bleibt die Argumentation gültig, wenn im Verfahren ein Gleichungssystem mit der „primal-dualen Hessematrix“ $H_k + \Sigma_k$ gelöst wird, solange die Abweichung zwischen den beiden Matrizen klein gehalten wird.

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

Für den Beweis des Verfahrens müssen die Voraussetzungen 3.2, die für den Beweis ohne Störungen notwendig waren, um Forderungen an die Störungen erweitert werden:

Voraussetzungen 3.5 Sei $\{x^k\}_k$ die Folge, die von dem auf (3.65) angewandten Filter-Verfahren erzeugt wurde, der Startpunkt $x^0 > 0$ und $\mathbf{s}_b = \mathbf{1}$. Die Feasibility Restoration Phase werde stets erfolgreich mit $x^{k+1} > 0$ beendet und der Algorithmus ende nicht an einem KKT-Punkt. Außerdem seien folgende Bedingungen erfüllt:

(V1) Es existiere eine offene Menge $C \subset \mathbb{R}^n$ mit $[x^k, x^k + \alpha_k^{\max} \Delta x^k] \subset C \quad \forall k \notin \mathcal{R}_{inc}$ sodass f und c differenzierbar sind auf C und $f, \nabla f, c$ sowie ∇c auf C beschränkt und Lipschitz-stetig sind.

(V2) Die Approximationen H_k der Hessematrix der zu (3.65) gehörigen Lagrangefunktion seien für alle $k \notin \mathcal{R}_{inc}$ gleichmäßig beschränkt.

(V3) Die Matrizen $H_k + \mu X_k^{-2}$ seien gleichmäßig positiv definit auf dem Nullraum von $A^\delta(x^k)^T$, d.h. es existiert eine Konstante $M_H^\delta > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\lambda_{\min} \left(\tilde{Z}_k^{\delta T} (H_k + \mu X_k^{-2}) \tilde{Z}_k^\delta \right) \geq M_H^\delta.$$

(V4) Es existiere eine Konstante $M_A^\delta > 0$ sodass für alle $k \notin \mathcal{R}_{inc}$ gilt

$$\sigma_{\min} (A^\delta(x^k)) \geq M_A^\delta.$$

(V5) Die Iterierten, für die die Feasibility Restoration Phase während der Berechnung einer neuen Suchrichtung gestartet wird, seien nicht beliebig nahe am zulässigen Bereich, d.h. es gibt eine Konstante $\theta_{inc} > 0$ sodass $k \notin \mathcal{R}_{inc}$ ist, wenn $\theta^\delta(x^k) \leq \theta_{inc}$ gilt.

(V6) Die Iterierten $\{x^k\}_k$ seien beschränkt.

(V7) Es existiere ein $\bar{\zeta}_s > 0$, sodass für alle $x \in \bar{X} := \overline{\{x^k\}_{k \notin \mathcal{R}}}$ die Gradienten der aktiven und fast aktiven Nebenbedingungen

$$\nabla c_1(x), \dots, \nabla c_m(x) \quad \text{und} \quad \{e_i \mid x_i \leq \bar{\zeta}_s\}$$

linear unabhängig sind.

(V8) Es existieren Konstanten $\zeta_\theta, \zeta_x > 0$ sodass in allen Iterationen, in denen die Feasibility Restoration Phase mit $\theta^\delta(x^k) \leq \zeta_\theta$ gestartet wird, eine neue Iterierte x^{k+1} berechnet wird, für die gelte

$$x_i^{k+1} \geq x_i^k \quad \text{für alle } i \text{ mit } x_i^k \leq \zeta_x$$

(V9) Für die Störung der Nebenbedingungen gelte Gleichung (3.51) und außerdem existiere eine Schranke C_δ sodass für alle $k \in \mathbb{N}$ gilt

$$0 \leq \delta_c^k, \delta_A^k \leq C_\delta < \infty.$$

Seien $C_\theta, M_d > 0$ Konstanten, sodass mit den Voraussetzungen (V1)-(V4) die Lemmata 3.2 und 3.4 erfüllt sind. Es existiere ein $K_\delta \in \mathbb{N}$, sodass für die Störungen der Nebenbedingungen und deren Ableitungen für alle $k \geq K_\delta$ gelte

(V10) Für alle $\alpha \in [0, 1]$ sei $\|c^\delta(x^k + \alpha \Delta x^k) - c(x^k + \alpha \Delta x^k)\| \leq \delta_c^k$.

(V11) Falls $\theta^\delta(x^k) > 0$ ist, dann gelte $\delta_c^k < \frac{\theta^\delta(x^k)^2}{8C_\theta M_d^2}$;
falls $\theta^\delta(x^k) = 0$ ist, dann gelte $\delta_c^k = 0$.

(V12) Falls $\theta^\delta(x^k) > 0$ ist, dann gelte

$$\delta_A^k < \frac{\theta^\delta(x^k)}{M_d} - \sqrt{8C_\theta \delta_c^k}, \quad \text{d.h.} \quad \theta^\delta(x^k) - \delta_A^k M_d > \sqrt{8C_\theta \delta_c^k M_d^2};$$

falls $\theta^\delta(x^k) = 0$ ist, dann gelte $\delta_A^k = 0$.

(V13) Falls

$$\theta^\delta(x^k) - \delta_A^k M_d > \alpha_k^{\min} \theta^\delta(x^k)$$

mit α_k^{\min} aus Gleichung (3.35) - hier jedoch mit der Zielfunktion f anstelle von B - gilt, dann sei auch

$$8C_\theta M_d^2 \delta_c^k \leq (\alpha_k^{\min} \theta^\delta(x^k))^2$$

erfüllt.

(V14) Es existiere ein $q \in (0, 1)$, sodass für alle x^k der erzeugten Folge mit $k \notin \mathcal{R}$ und alle $\zeta_s \leq \bar{\zeta}_s$ gilt

$$\sigma_{\max}(A_l^\delta(x^k) - A_l(x^k)) \leq q \sigma_{\min}(A_l(x^k)),$$

wobei $A_l(x^k)$ aus der Matrix $A(x^k)$ durch Streichung der Zeilen mit dem Index, für den $x_i^k \leq \zeta_s$ gilt, entsteht.

Die Voraussetzungen 3.5 (V1) - (V6) und (V8) entsprechen den Voraussetzungen 3.2 (V1) - (V6) und (V8), hier jedoch für den Fall mit gestörten Gleichungsnebenbedingungen. (V7) muss hier strenger formuliert werden, sodass die lineare Abhängigkeit der

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

Gradienten sowohl für alle Häufungspunkte der erzeugten Folge, als auch für Iterierte mit fast aktiven Ungleichungsnebenbedingungen gefordert wurde.

Voraussetzungen 3.5 (V9) - (V13) stimmen mit den Forderungen an die Störungen aus Voraussetzung 3.3 und 3.4 überein, während Voraussetzung 3.5 (V14) für den ungleichungsrestringierten Fall hinzugefügt wurde.

Um die Aussage von [53, Lemma 11] auch für gestörte Nebenbedingungen beweisen zu können, benötigt man einige Vorüberlegungen. Dazu seien zunächst zwei Normen für Matrizen definiert und ihr Zusammenhang mit dem maximalen Singulärwert einer Matrix erwähnt. Dies findet sich beispielsweise in [30, 43].

Definition 3.11 (Frobeniusnorm und Spektralnorm)

Für Matrizen $B = (B_{ij})_{\substack{1 \leq i \leq N \\ 1 \leq j \leq M}} \in \mathbb{R}^{N \times M}$ ist die Frobeniusnorm durch

$$\|B\|_F = \sqrt{\sum_{i=1}^N \sum_{j=1}^M B_{ij}^2}$$

und die Spektralnorm durch

$$\|B\|_2 = \sup_{x \neq 0} \frac{\|Bx\|_2}{\|x\|_2}$$

definiert, wobei die Vektornorm $\|\cdot\|_2$ die euklidischen Norm in \mathbb{R}^N bzw. \mathbb{R}^M bezeichnet.

Lemma 3.15 (Frobeniusnorm und Spektralnorm)

Sei $B \in \mathbb{R}^{N \times M}$ und $r \in \mathbb{N}$ der Rang der Matrix B . Dann gilt

$$\|B\|_2 = \sigma_{\max}(B) \tag{3.67}$$

$$\|B\|_2 \leq \|B\|_F \leq \sqrt{r} \|B\|_2 \tag{3.68}$$

Für die Abschätzung der Störung von Singulärwerten gestörter Matrizen ist in [30, Korollar 8.3.2] folgende Aussage gegeben:

Lemma 3.16 Für $B, B + E \in \mathbb{R}^{N \times M}$ mit $N \geq M$ gilt für die Singulärwerte mit $1 \leq K \leq M$

$$|\sigma_K(B + E) - \sigma_K(B)| \leq \sigma_{\max}(E) = \|E\|_2.$$

Anmerkung 3.3 Aus diesem Lemma erhält man die Stetigkeit der Singulärwerte einer Matrix in den Einträgen. Betrachtet man unter den Voraussetzungen 3.5 die im Innere-Punkte-Verfahren verwendete, ungestörte Matrix $A(x) = (\nabla c_1(x), \dots, \nabla c_m(x)) \in \mathbb{R}^{n \times m}$

in den Punkten x und y mit $\|x-y\| \leq \check{\epsilon}_1$ mit $\check{\epsilon}_1 > 0$, so erhält man aufgrund der Stetigkeit aus Voraussetzung 3.5 (V1) ein $\check{\epsilon}_2 > 0$ sodass für alle $1 \leq i \leq n$ und $1 \leq j \leq m$ gilt

$$|A_{ij}(x) - A_{ij}(y)| = \left| \frac{\partial c_i(x)}{\partial x_j} - \frac{\partial c_i(y)}{\partial x_j} \right| \leq \check{\epsilon}_2.$$

Damit folgt mit Lemma 3.16 und Lemma 3.15

$$|\sigma_K(A(x)) - \sigma_K(A(y))| \leq \|A(x) - A(y)\|_2 \leq \|A(x) - A(y)\|_F \leq \sqrt{nm} \check{\epsilon}_2.$$

Das folgende Lemma zeigt nun analog zu [53, Lemma 11], dass an hinreichend zulässigen Punkten die Suchrichtungen von fast aktiven Beschränkungen wegzeigen.

Lemma 3.17 *Seien die Voraussetzungen 3.5 erfüllt. Sei $S \subset \{1, \dots, n\}$ und $\zeta_l > 0$ gegeben. Dann existieren $\zeta_s, \zeta_\theta > 0$ sodass gilt $\Delta x_i^k > 0$ für alle $i \in S$, wenn in Iteration k keine Feasibility Restoration Phase gestartet wird und für x^k gilt*

$$x^k \in L := \{x \geq 0 \mid x_i \leq \zeta_s \ \forall i \in S, \ x_i \geq \zeta_l \ \forall i \notin S, \ \theta^\delta(x) \leq \zeta_\theta\}.$$

Beweis: Wie im Beweis von [53, Lemma 11] erfolgt zunächst die Zerlegung der primalen Variablen x . Seien in x_s die Komponenten von x enthalten, deren Indizes in S liegen, während x_l die restlichen Komponenten von x beinhalte. OBdA setzt man nun voraus, dass $x = \begin{pmatrix} x_s \\ x_l \end{pmatrix}$ gilt. Zerlegt man nun analog $\Delta x = \begin{pmatrix} \Delta x_s \\ \Delta x_l \end{pmatrix}$, $\nabla f(x) = \begin{pmatrix} \nabla_s f(x) \\ \nabla_l f(x) \end{pmatrix}$, $A^\delta(x) = \begin{pmatrix} A_s^\delta(x) \\ A_l^\delta(x) \end{pmatrix}$ und $H_k = \begin{pmatrix} H_k^{ss} & H_k^{sl} \\ H_k^{ls} & H_k^{ll} \end{pmatrix}$, so erhält man aus Gleichungssystem (3.66) nach Umformung

$$\begin{pmatrix} X_k^s H_k^{ss} X_k^s + \mu Id & X_k^s H_k^{sl} & X_k^s A_s^\delta(x^k) \\ H_k^{ls} X_k^s & H_k^{ll} + \mu (X_k^l)^{-2} & A_l^\delta(x^k) \\ A_s^\delta(x^k)^T X_k^s & A_l^\delta(x^k)^T & 0 \end{pmatrix} \begin{pmatrix} (X_k^s)^{-1} \Delta x_s^k \\ \Delta x_l^k \\ \lambda_+^k \end{pmatrix} = \begin{pmatrix} -X_k^s \nabla_s f(x^k) - \mu e_s \\ -\nabla_l f(x^k) - \mu (X_k^l)^{-1} e_l \\ -c^\delta(x^k) \end{pmatrix} \quad (3.69)$$

Sei $x \in \bar{X} \cap L$ für anfänglich gewählte ζ_s und ζ_θ mit $0 < \zeta_s \leq \bar{\zeta}_s$ und $\zeta_\theta > 0$. Aus Voraussetzung 3.5 (V7) folgt, dass die Matrix

$$\begin{pmatrix} A_s(x) & Id \\ A_l(x) & 0 \end{pmatrix} \in \mathbb{R}^{n \times (m+|S|)}$$

3.4. KONVERGENZ DES FILTER-VERFAHRENS BEI STÖRUNGEN IN DEN NEBENBEDINGUNGEN

linear unabhängige Spalten hat. Damit hat auch $A_l(x)$ linear unabhängige Spalten. Daraus folgt einerseits, dass $m \leq n - |S|$ gelten muss, und andererseits gilt

$$\sigma_{\min}(A_l(x)) > 0.$$

$\overline{X} \cap L$ ist abgeschlossen und nach Voraussetzung 3.5 (V6) beschränkt, also kompakt. Aus der Stetigkeit der Singulärwerte (Anmerkung 3.3) folgt, dass

$$\inf_{x \in \overline{X} \cap L} \{\sigma_{\min}(A_l(x))\} =: m_\sigma > 0$$

ist. Damit gilt für alle $x \in \overline{X} \cap L$

$$\sigma_{\min}(A_l(x)) = \sigma_m(A_l(x)) \geq m_\sigma.$$

Für alle Elemente der erzeugten Folge $\{x^k\}_k$ gilt $\{x^k | x^k \in L \wedge x^k \in \overline{X}\} = \{x^k | x^k \in L\}$. Aus Voraussetzung 3.5 (V14) und Lemma 3.16 folgt dann für alle $x^k \in L$

$$\begin{aligned} \sigma_{\min}(A_l(x^k)) - \sigma_{\min}(A_l^\delta(x^k)) &= \sigma_m(A_l(x^k)) - \sigma_m(A_l^\delta(x^k)) \\ &\leq |\sigma_m(A_l(x^k)) - \sigma_m(A_l^\delta(x^k))| \\ &\leq \sigma_{\max}(A_l(x^k) - A_l^\delta(x^k)) \\ &\leq q \sigma_{\min}(A_l(x^k)). \end{aligned}$$

Daraus erhält man mit

$$\sigma_{\min}(A_l^\delta(x^k)) \geq (1 - q) \sigma_{\min}(A_l(x^k)) \geq (1 - q) m_\sigma > 0$$

eine Abschätzung für den kleinsten Singulärwert von $A_l^\delta(x^k)$ für alle $x^k \in L \cap \overline{X}$.

Der restliche Beweis verläuft wie der von [53, Lemma 11]. Aus den Voraussetzungen 3.5 (V3) und (V5) folgt, nachdem ζ_θ eventuell noch verringert wurde, dass die Projektion von $H_k^{ll} + \mu(X_k^l)^{-2}$ auf den Nullraum von $A_l^\delta(x^k)^T$ für alle $x^k \in L$ gleichmäßig positiv definit ist.

Zusammen mit Voraussetzung 3.5 (V1) und (V2) sieht man, dass Gleichung (3.69)

$$\begin{aligned} \left[\begin{pmatrix} \mu Id & 0 & 0 \\ 0 & H_k^{ll} + \mu(X_k^l)^{-2} & A_l^\delta(x^k) \\ 0 & A_l^\delta(x^k)^T & 0 \end{pmatrix} + \mathcal{O}(\zeta_s) \right] \begin{pmatrix} (X_k^s)^{-1} \Delta x_s^k \\ \Delta x_l^k \\ \lambda_+^k \end{pmatrix} \\ = - \begin{pmatrix} -\mu e_s \\ \nabla_l f(x^k) - \mu(X_k^l)^{-1} e_l \\ c^\delta(x^k) \end{pmatrix} + \mathcal{O}(\zeta_s) \end{aligned}$$

für alle $x^k \in L$ erfüllt. Zudem sind in dieser Gleichung sowohl die Inverse der Matrix als auch die rechte Seite gleichmäßig beschränkt für ζ_s klein genug.

Damit ist $(X_k^s)^{-1}\Delta x_s^k = e + \mathcal{O}(\zeta_s)$ für alle $x^k \in L$. Folglich gilt $(X_k^s)^{-1}\Delta x_s^k > 0$ für klein genug gewähltes ζ_s und damit $\Delta x_s^k > 0$. \square

Satz 3.9 *Seien die Voraussetzungen 3.5 erfüllt.*

Dann existiert ein $\epsilon_x > 0$ sodass für alle k gilt $x^k \geq \epsilon_x e$.

Der Beweis funktioniert genauso wie für Satz 3.5 bzw. [53, Theorem 3] ohne Störung. Die Aussage folgt hier aus Satz 3.7, Lemma 3.17, Voraussetzung 3.5 (V8) sowie der *Fraction-to-the-boundary* Regel und der Einschränkung auf $s_b = 1$.

Kapitel 4

Numerische Ergebnisse

4.1 Einbinden von IPOPT in OPTISIM[®]

Das Programmpaket IPOPT (Version 3.10.2) wurde unter Windows[®] 7 (32-bit) mit OPTISIM[®] Version 10.8.0 gelinkt. Hierfür wurden die Compiler Microsoft Visual C++ 2008 (Version 9.0) und Intel[®] Visual Fortran Compiler (Version 11.0.074) verwendet.

Die für IPOPT notwendigen Pakete BLAS und LAPACK sind über die MKL (Math Kernel Library) von Intel[®] bereits vorhanden. Zur Lösung der linearen Gleichungssysteme mit dünnbesetzten Matrizen innerhalb von IPOPT wurde versuchsweise einerseits der Löser MA27 aus dem Harwell Subroutine Library (HSL) Archiv mit dem Vorkonditionierer MC19 eingebunden, andererseits die public domain Software MUMPS (MULTifrontal MASSively Parallel sparse direct Solver - Version 4.9.2) zusammen mit dem Vorkonditionierer METIS (Version 4.0.1).

OPTISIM[®] ist zu großen Teilen in Fortran programmiert, während IPOPT seit Version 3.0.0 in C++ implementiert wird. Im Programmpaket von IPOPT werden einige Schnittstellen in verschiedenen Programmiersprachen (C++, C und Fortran, aber auch AMPL, R, etc.) zur Anbindung an bestehenden Code mitgeliefert [34]. Diese Schnittstellen enthalten verschiedene Funktionsaufrufe zur Übermittlung von

- Problemgröße und -struktur,
- den Schranken der Optimierungsvariablen und der Nebenbedingungen,
- dem Startwert,
- den Funktionsauswertungen der Zielfunktion und der Nebenbedingungen,
- den Funktionsauswertungen der Ableitungen von Zielfunktion und Nebenbedingungen, sowie
- der Hessematrix der Lagrangefunktion, sofern zweite Ableitungen vorhanden sind.

Sind keine zweiten Ableitungen bekannt, so wird die Hessematrix von IPOPT intern mit Hilfe einer Quasi-Newton Updateregeln approximiert.

Da in OPTISIM[®] in den meisten Fällen keine zweiten Ableitungen gegeben sind, wird die Hessematrix hier durch eine Approximation nach der limited-memory BFGS-Updateregeln (vgl. hierzu [28]) ersetzt.

4.1.1 Sequentieller Ansatz

Um das Verhalten des Innere-Punkte-Verfahrens mit den seitherigen Optimierungsalgorithmen vergleichen zu können, wurde IPOPT zunächst für den sequentiellen Optimierungsansatz in OPTISIM[®] angeschlossen. Dabei wird das Optimierungsproblem (1.6) aus Abschnitt 1.3.1 betrachtet:

$$\begin{array}{ll} \min_p & \Phi(\xi(p)) \\ \text{u.d.N.} & p_{\min} \leq p \leq p_{\max} \\ & \xi_{\min} \leq \xi_J(p) \leq \xi_{\max} \\ & c_{\min} \leq c(\xi(p)) \leq c_{\max} \end{array}$$

Für diesen Ansatz wurde die von IPOPT mitgelieferte Fortran Schnittstelle verwendet. Darin wird bei den Funktionsauswertungen von Zielfunktion und Nebenbedingungen sowie deren Ableitungen angegeben, ob die entsprechende Funktion an einem neuen Punkt ausgewertet werden soll, oder ob an diesem Punkt bereits eine andere Funktion berechnet wurde. Dadurch muss das Modellgleichungssystem (1.3) nur für neue Werte der Optimierungsvariablen, die hier den Designvariablen p entsprechen, gelöst werden. Bei allen weiteren Funktionsaufrufen am selben Punkt wird dann nur die jeweils benötigte Funktion ausgewertet. Die Jacobimatrix der Nebenbedingungen muss für IPOPT in der Form einer dünnbesetzten Matrix im sogenannten *triplet format* gespeichert werden [34]. Die Speicherung einer Matrix in diesem Format besteht aus zwei ganzzahligen Feldern (Zeilenfeld und Spaltenfeld), einem reellwertigen Feld, sowie der Anzahl an Einträgen. Jedem Matrixeintrag, der nicht null ist, entspricht ein Eintrag in jedem der drei Felder. Die Position des Eintrags in der Matrix wird im Zeilen- und Spaltenfeld gespeichert, der Wert des Eintrags in dem reellwertigen Feld.

Eine Schrittweitenbeschränkung (vgl. Abschnitt 2.1.1) wurde in dieser Implementierung ermöglicht, indem bei zu großer Änderung in den Optimierungsvariablen die Auswertung der Funktionen (Zielfunktion und Nebenbedingungen) nicht erlaubt wird.

IPOPT besitzt eine Vielzahl von Optionen, die steuern, wie das Innere-Punkte-Verfahren durchgeführt werden soll. Folgende Optionen können von OPTISIM[®] aus vor dem Start der Optimierung ausgewählt werden:

1. Skalierung der Zielfunktion mit dem Wert der Zielfunktion am Startpunkt zusätzlich zur gradientenbasierten, internen Skalierung,
2. Initialisierung der dualen Variablen z^L und z^U in Abhängigkeit vom Barriereparameter μ oder mit konstanten Werten,
3. Anpassen des Barriereparameters nach der adaptiven oder der monotonen Strategie,
4. Zulassen der *Second-Order Correction* oder nicht.

4.1.2 Simultane Optimierung

Nun wird das Optimierungsproblem (2.2) aus Abschnitt 2.2.1

$$\begin{aligned} \min_x \quad & \Phi(x) \\ \text{u.d.N.} \quad & x_{\min} \leq x \leq x_{\max} \\ & c_{\min} \leq c(x) \leq c_{\max} \\ & F(x) = 0 \end{aligned}$$

mit $x = \begin{pmatrix} \xi \\ p \end{pmatrix}$, $x_{\min} = \begin{pmatrix} \xi_{\min} \\ p_{\min} \end{pmatrix}$ und $x_{\max} = \begin{pmatrix} \xi_{\max} \\ p_{\max} \end{pmatrix}$ betrachtet.

Da die im Flowsheet angegebenen Optimierungsvariablen p und die Zustandsvariablen ξ während der Optimierung nun gleichzeitig verändert werden dürfen, sind die Parametergleichungen (1.4), für diese Designvariablen eigentlich überflüssig. Da das Aus-sortieren dieser Gleichungen jedoch sehr aufwändig wäre, werden sie beibehalten. Diese Tatsache wird dann bei der Initialisierung der simultanen Optimierung berücksichtigt. Die Implementierung des simultanen Ansatzes wurde unabhängig vom Optimierungsverfahren vorgenommen. Dabei wurden folgende Punkte berücksichtigt:

Setzen der Grenzen

Für eine Designvariable p_i , die auch Optimierungsvariable ist, sind einerseits die Grenzen p_{\min_i} und p_{\max_i} , andererseits die Grenzen der zugehörige Zustandsvariable ξ_{pi} mit $\xi_{\min_{pi}}$ und $\xi_{\max_{pi}}$ gegeben. Die entsprechenden Grenzen in x_{\min} und x_{\max} werden dann so gesetzt, dass in beiden Komponenten, also für p_i und ξ_{pi} , jeweils die engeren Grenzen gelten.

„freie“ Variablen

Eine Designvariable p_i , deren zugehörige Zustandsvariable ξ_{pi} nur in der Gleichung für die Parameterquelle (1.4), jedoch weder in anderen Modellfunktionen noch in der Zielfunktion verwendet wird, wird als „frei“ bezeichnet. In diesem Fall gilt

$$\frac{d\Phi}{d\xi_{pi}} = 0 \quad \text{und} \quad \frac{dF}{d\xi_{pi}} = e_{\sigma(i)}.$$

Die beiden Variablen p_i und ξ_{pi} können theoretisch vom Optimierer beliebig gesetzt werden, solange sie beide denselben Wert haben. Um zu verhindern, dass diese Variablen verändert werden und so dem Anwender eine Abhängigkeit des Optimierungsproblems von der Designvariablen p_i vorgespiegelt wird, werden in diesem Fall die Grenzen für p_i und ξ_{pi} auf den aktuellen Wert gesetzt, sodass dieser beibehalten wird.

Auswertung der Funktionen und Ableitungen

Die Auswertung der Zielfunktion Φ und ihres Gradienten

$$\frac{d\Phi(x)}{dx} = \begin{pmatrix} \frac{\partial\Phi(\xi, p)}{\partial\xi} \\ 0 \end{pmatrix}$$

erfolgt direkt mit den aktuellen Werten von ξ .

Die Werte der Nebenbedingungen ergeben sich aus den Werten von ξ , p , $c(\xi, p)$ und $F(\xi, p)$. Die Ableitungen der Gleichungsnebenbedingungen

$$\frac{dF(x)}{dx} = \left(\frac{\partial F(\xi, p)}{\partial \xi}, \frac{\partial F(\xi, p)}{\partial p} \right)$$

sind durch die aktuellen Werte der Jacobimatrix der Modellfunktion $\frac{\partial F(\xi, p)}{\partial \xi}$ und die konstanten Werte für $\frac{\partial F(\xi, p)}{\partial p}$ aus Gleichung (1.8) gegeben. Die Ungleichungsnebenbedingungen $c(\xi, p)$ bestehen aus Summen von Zustandsvariablen, sodass deren Ableitungen konstant sind:

$$\frac{dc(x)}{dx} = \left(\frac{\partial c(\xi, p)}{\partial \xi}, \frac{\partial c(\xi, p)}{\partial p} \right) = \left(\frac{dc(\xi)}{d\xi}, 0 \right)$$

4.1.2.1 Schnittstelle zu IPOPT

Für die simultane Optimierung wurde eine Schnittstelle zwischen OPTISIM[®] und IPOPT programmiert. Dazu wurde eine C++-Klasse erzeugt, die von der abstrakten Klasse `Ipopt::TNLP` abgeleitet wurde.

Die Auswertung der Modellfunktion F sowie ihrer Jacobimatrix $\frac{\partial F}{\partial \xi}$ erfolgt immer gemeinsam und für jeden Wert der Optimierungsvariablen x nur einmal.

IPOPT erwartet die Ableitungen der Nebenbedingungen in der Form einer dünnbesetzten Matrix mit konstanter Struktur. Wird während der Optimierung eine Unstetigkeitsstelle erreicht bzw. passiert, die zur Änderung der Belegungsstruktur der Matrix $\frac{\partial F}{\partial \xi}$ führt (Punkt 3 in Abschnitt 2.1.2), so wird dies während der Berechnung der Modellfunktion F und deren Ableitungen ersichtlich.

Tritt diese Änderung der Matrixstruktur bei der Auswertung der Ableitungen der Nebenbedingungen auf, so muss der aktuelle Optimierungslauf beendet und eine neue Rechnung mit der neuen Matrixstruktur begonnen werden. Hierfür wird die Warmstartmöglichkeit von IPOPT genutzt. Dabei können die dualen Variablen z^L , z^U und λ mit den Werten aus der gerade beendeten Rechnung initialisiert werden. Die neue Rechnung wird dann über den Aufruf `ReOptimizeTNP` gestartet.

Tritt diese Änderung der Matrixstruktur hingegen bei der Auswertung der Zielfunktion, des Gradienten der Zielfunktion oder der Nebenbedingungen auf, so wird die Rechnung zunächst fortgesetzt, da diese Werte weiterhin berechnet werden können. Während der Filter-Liniensuche werden von IPOPT nur die Werte von Zielfunktion und Nebenbedingungen ausgewertet. Es kann daher sein, dass während der Liniensuche eine solche Unstetigkeitsstelle passiert wird. Dann werden diese Funktionen in einem Testpunkt $x^{k,l}$ ausgewertet, an dem die Matrixstruktur nicht mit der Struktur am Punkt x^k übereinstimmt. Falls die Jacobimatrix an dem von der Liniensuche schlussendlich akzeptierten Punkt x^{k+1} dieselbe Struktur besitzt wie in x^k , kann so ein voreiliges Abbrechen der Optimierung verhindert werden. Sonst bestünde die Gefahr wiederholt hin und her zu pendeln.

Erst nachdem der neue Punkt x^{k+1} akzeptiert wurde, werden die Ableitungen der Nebenbedingungen an diesem Punkt ausgewertet. Hat sich die Struktur der Jacobimatrix gegenüber der Jacobimatrix in x^k geändert, so wird die Optimierung neu gestartet.

Bei Unstetigkeiten, in denen sich nur die Modellfunktion, nicht aber die Struktur der Jacobimatrix ändert (Punkt 1 und 2 in Abschnitt 2.1.2), wird der Optimierungslauf nicht unterbrochen.

Die Approximation der Hessematrix der Lagrangefunktion wird beim simultanen Ansatz wieder durch ein limited-memory BFGS-Verfahren berechnet, die Initialisierung der Matrix ist dabei konstant, also ein Vielfaches der Einheitsmatrix.

Folgende Optionen von IPOPT können in OPTISIM[®] vor dem Start der Optimierung ausgewählt werden:

1. Skalierung der Zielfunktion mit dem Wert der Zielfunktion am Startpunkt oder einem festen Wert zusätzlich zur gradientenbasierten, internen Skalierung,
2. Initialisierung der dualen Variablen z^L und z^U in Abhängigkeit vom Barriereparameter μ oder mit konstanten Werten,
3. Anpassen des Barriereparameters nach der adaptiven oder der monotonen Strategie,
4. Zulassen der *Second-Order Correction* oder nicht,
5. Behandlung von fixierten Optimierungsvariablen (untere gleich obere Schranke) als Nebenbedingung oder Aussortieren dieser Variablen zu Beginn der Rechnung,
6. Skalierung der zusätzlichen Ungleichungsnebenbedingung c .

4.2 Testergebnisse

Die verschiedenen Optimierungsverfahren und -ansätze wurden im Rahmen eines Tests verglichen. Die sequentielle Optimierung wurde mit den beiden bisherigen SQP-Verfahren sowie mit IPOPT durchgeführt. Die simultane Optimierung hingegen kann nur mit IPOPT durchgeführt werden.

Die getesteten Verfahren wurden folgendermaßen bezeichnet:

- SQP1
- SQP2
- IPOPT(HSL) - nutzt den linearen Gleichungslöser MA27 mit dem Vorkonditionierer MC19 aus dem Harwell Subroutine Library (HSL) Archiv
- IPOPT(MUMPS) - nutzt den linearen Gleichungslöser MUMPS und den Vorkonditionierer METIS

Beim Einsatz von IPOPT wurden neben unterschiedliche Routinen zur Lösung des internen linearen Gleichungssystems auch verschiedene Optionen, die in der Implementierung von IPOPT gegeben sind, getestet. Die Auswahl dieser Optionen wurde durch den Parameter MOPT ermöglicht. Die Bedeutung der einzelnen Werte für MOPT sind in Tabelle 4.1 für die sequentielle und die simultane Optimierung erläutert.

MOPT	sequentielle Optimierung	simultane Optimierung
1	die Zielfunktion wird zusätzlich mit $ \Phi(x^0) $ skaliert	die Zielfunktion wird bei verletzten Nebenbedingungen mit $ \Phi(x^0) $, sonst mit 10^5 zusätzlich skaliert
2	die Initialisierung der Multiplikatoren z^U und z^L erfolgt in Abhängigkeit des Barriereparameters μ	
3	1 + 2	
4	für die Wahl des Barriereparameters μ wird die adaptive Strategie verwendet	
5	1 + 4	
8	die <i>Second-Order Correction</i> wird zugelassen	
9	1 + 8	
16	–	Optimierungsvariablen, bei denen obere und untere Schranke übereinstimmen, werden aussortiert, anstatt sie als Gleichungsnebenbedingungen zu betrachten
17	–	1 + 16

Tabelle 4.1 Liste der durch MOPT zusätzlich auswählbaren Optionen für IPOPT

Für die Tests wurde das in Abbildung 4.1 dargestellte Flowsheet betrachtet:

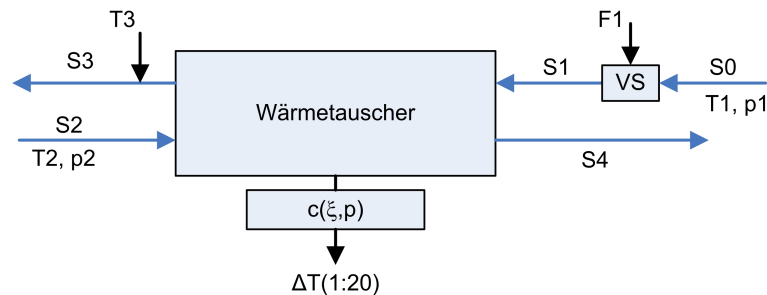


Abbildung 4.1 Diagramm des Testflowsheets

S_0 ist ein warmer Luftstrom mit konstanter Menge. Der Parameter F_1 beschreibt den Anteil von S_0 , der in den Design-Wärmetauscher als Strom S_1 eintritt. Temperatur und Druck sind fest vorgegeben. Der Luftstrom wird im Wärmetauscher durch einen kalten Strom aus reinem Stickstoff abgekühlt. Der Strom S_2 ist flüssig, wobei Menge, Druck und Temperatur fest vorgegeben sind.

Der Wärmetauscher wurde so spezifiziert, dass die Austrittstemperatur von S3 durch den Parameter $T3$ vorgegeben ist und die vom Luftstrom abgegebene Wärme vollständig vom kalten Strom aufgenommen wird. Daraus berechnet sich die Temperatur des Austrittstroms S4.

In der Optimierung sollen nun die Austrittstemperatur $T3$ und das Mengenverhältnis $F1$ so bestimmt werden, dass möglichst viel Luft auf möglichst niedrige Temperatur abgekühlt wird. Um die thermische Beanspruchung im Wärmetauscher gering zu halten, sollte die Temperaturdifferenz zwischen warmem und kaltem Strom nicht zu groß werden. Dazu wird das Temperaturprofil der Ströme entlang der übertragenen Wärmemenge betrachtet und daraus das Temperaturdifferenzprofil ΔT berechnet. Dieses wird nach oben mit 200K beschränkt. Andererseits muss ΔT nach unten mit einem positiven Wert, hier 2K, beschränkt werden, um den zweiten Hauptsatz der Thermodynamik nicht zu verletzen. Würden sich die beiden Temperaturprofile überschneiden, so müsste die Wärme vom kalten zum warmen Strom geleitet werden. Lügen die beiden Profile übereinander, so müsste der Wärmetauscher eine unendlich große Übertragungsfläche besitzen. Das betrachtete Optimierungsproblem lautet:

$$\begin{aligned} \min \quad & T3 - F1 \\ \text{u.d.N.} \quad & 0.05 \leq F1 \leq 1.0 \\ & 51.0 \leq T3 \leq 300.0 \\ & 2.0 \leq \Delta T(1:20) \leq 200.0 \end{aligned} \tag{4.1}$$

Die Ergebnisse aller Tests sind in den Tabellen 4.2 und 4.3 für jede Methode und die entsprechenden Optionen zusammengefasst.

Mit Ende wird die Endmeldung des Optimierers bezeichnet. OK bedeutet, dass die Optimierung erfolgreich beendet wurde und alle Nebenbedingungen erfüllt sind. MAXITER bedeutet, dass die maximale Anzahl an Iterationen des Optimierers (in diesem Fall 500) erreicht wurde, bevor die Optimierung erfolgreich beendet werden konnte.

Für die Zielfunktion wird jeweils der Wert am Ende der Optimierung angegeben. Opt-Iter bezeichnet die Anzahl der benötigten Optimierungsiterationen, während unter SS-Iter für die sequentielle Optimierung die Anzahl der benötigten Iterationen bei der stationären Simulation angegeben werden.

Das Ergebnis zeigt einerseits, dass die simultane Optimierung nur bei zusätzlicher Skalierung der Zielfunktion erfolgreich ist. Es ist bekannt, dass der Optimierer IPOPT nur schwer konvergiert, wenn die zweiten Ableitungen der Lagrangefunktion nur als Approximation vorliegen. Dies wird auch in dem obigen Beispiel deutlich. Wird die

Zielfunktion des Optimierungsproblems jedoch mit einem entsprechend großen Faktor multipliziert, so wird die Konvergenz um ein Vielfaches beschleunigt. In diesem Beispiel wurde als Faktor bei verletzten Nebenbedingungen der Betrag der Zielfunktion am Startpunkt und bei erfüllten Nebenbedingungen der Faktor 10^5 verwendet (vgl. Tabelle 4.1).

Alle Testläufe, in denen das Optimierungsverfahren erfolgreich beendet wurde, lieferten dasselbe Ergebnis für die Optimierungsvariablen und damit auch für die Zielfunktion und die Nebenbedingungen:

Zielfunktion: 199.0000 \rightarrow 81.6729
 T3: 200.0000 \rightarrow 82.0000
 F1: 1.0000 \rightarrow 0.3271
 $\min(\Delta T)$: 102.3472 \rightarrow 2.0000

sequentielle Optimierung:						
Methode	MOPT	Ende	Zielfunktion	Opt-Iter	SS-Iter	Zeit (s)
SQP1		OK	81.6729	13	175	7.566
SQP2		OK	81.6729	13	175	7.597
IPOPT (HSL)	0	OK	81.6729	32	181	8.065
IPOPT (HSL)	1	OK	81.6729	23	207	9.095
IPOPT (HSL)	2	OK	81.6729	82	780	29.437
IPOPT (HSL)	3	OK	81.6729	25	117	5.382
IPOPT (HSL)	4	OK	81.6729	32	181	8.096
IPOPT (HSL)	5	OK	81.6729	23	207	9.079
IPOPT (HSL)	8	OK	81.6729	31	161	7.129
IPOPT (HSL)	9	OK	81.6729	23	207	9.064
IPOPT (MUMPS)	0	OK	81.6729	32	181	8.206
IPOPT (MUMPS)	1	OK	81.6729	23	207	9.220
IPOPT (MUMPS)	2	OK	81.6729	82	780	29.859
IPOPT (MUMPS)	3	OK	81.6729	25	117	5.507
IPOPT (MUMPS)	4	OK	81.6729	32	181	8.268
IPOPT (MUMPS)	5	OK	81.6729	23	207	9.157
IPOPT (MUMPS)	8	OK	81.6729	31	161	7.301
IPOPT (MUMPS)	9	OK	81.6729	23	207	9.282

Tabelle 4.2 Ergebnisse des Tests mit sequentielle Optimierung

simultane Optimierung:						
Methode	MOPT	Ende	Zielfunktion	Opt-Iter	SS-Iter	Zeit (s)
IPOPT (HSL)	0	MAXITER	147.7681	501	0	17.145
IPOPT (HSL)	1	OK	81.6729	29	0	2.480
IPOPT (HSL)	2	MAXITER	180.1192	501	0	18.112
IPOPT (HSL)	3	OK	81.6729	28	0	2.496
IPOPT (HSL)	4	MAXITER	147.7681	501	0	17.129
IPOPT (HSL)	5	OK	81.6729	29	0	2.496
IPOPT (HSL)	8	MAXITER	147.7681	501	0	17.051
IPOPT (HSL)	9	OK	81.6729	29	0	2.652
IPOPT (HSL)	16	MAXITER	147.7681	501	0	17.082
IPOPT (HSL)	17	OK	81.6729	29	0	2.574
IPOPT (MUMPS)	0	MAXITER	147.7681	501	0	82.337
IPOPT (MUMPS)	1	OK	81.6729	29	0	8.658
IPOPT (MUMPS)	2	MAXITER	180.1192	501	0	104.427
IPOPT (MUMPS)	3	OK	81.6729	28	0	8.408
IPOPT (MUMPS)	4	MAXITER	147.7681	501	0	81.823
IPOPT (MUMPS)	5	OK	81.6729	29	0	9.079
IPOPT (MUMPS)	8	MAXITER	147.7681	501	0	82.759
IPOPT (MUMPS)	9	OK	81.6729	29	0	9.500
IPOPT (MUMPS)	16	MAXITER	147.7681	501	0	82.759
IPOPT (MUMPS)	17	OK	81.6729	29	0	8.674

Tabelle 4.3 Ergebnisse des Tests mit simultane Optimierung

Die Abbildungen 4.2 und 4.3 zeigen die Temperaturprofile des Design-Wärmetauschers vor und nach der Optimierung. Der Luftstrom ist hier schwarz und der Stickstoffstrom blau dargestellt. Der Luftstrom wird während der Optimierung so verändert, dass er schlussendlich vollständig verdampft. Dadurch erhalten die Nebenbedingungen zwei zusätzliche „Knickstellen“, an denen sie bezüglich der Enthalpie nicht stetig differenzierbar sind.

Dennoch kann das Optimierungsproblem durch die simultane Optimierung rasch gelöst werden. Hierbei können im Vergleich zur sequentiellen Optimierung viele Iterationen gespart werden. Sind beim bisherigen SQP-Verfahren beispielsweise 175 Iterationen für die stationäre Simulation nötig, so sind bei der simultanen Optimierung im besten Fall nur 28 Optimierungsiterationen erforderlich. Im Zeitvergleich stellt man fest, dass die simultane Optimierung bis zu einem Faktor drei schneller als der bisherige sequentielle Ansatz ist.

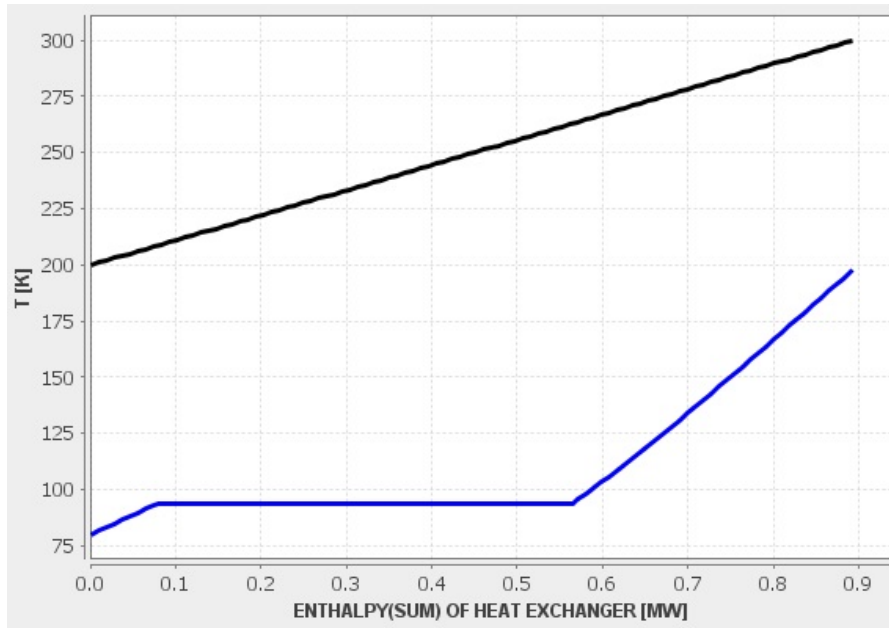


Abbildung 4.2 Temperaturprofile vor der Optimierung

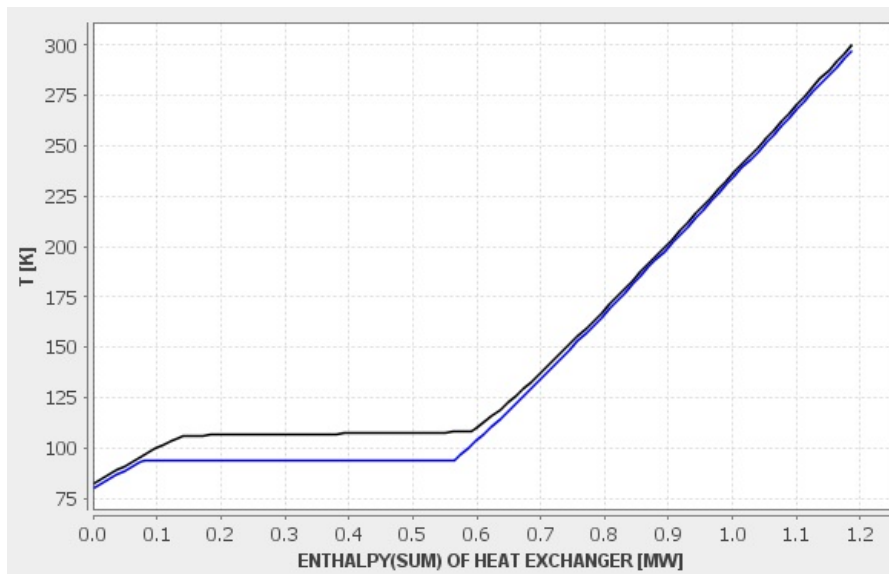


Abbildung 4.3 Temperaturprofile nach erfolgreicher Optimierung

Kapitel 5

Globale Optimierung

Wie bereits in Kapitel 2 erwähnt, ist es für die Prozessrechnung interessant, das globale Minimum des betrachteten Optimierungsproblems zu finden. Im Rahmen zweier aufeinander aufbauender Bachelorarbeiten [46] und [44] wurde die in Abschnitt 2.2.2 beschriebene Methode des mehrfachen Starts der lokalen, sequentiellen Optimierung in OPTISIM[®] umgesetzt. Dazu wurde zunächst ein Wegfindungsalgorithmus zur Bestimmung der zu den einzelnen Startwerten p_{Init} gehörenden stationären Lösungen $\xi(p_{Init})$ implementiert, bevor dieser dann zusammen mit dem bestehenden Optimierungsverfahren in OPTISIM[®] zu einem globalen Optimierungsalgorithmus gekoppelt werden konnte.

5.1 Wegfindung

Aufgrund der lokalen Konvergenz des Newtonverfahrens, kann die stationäre Lösung $\xi(p)$ für neue Werte der Designvariablen p nicht immer ohne weiteres berechnet werden. Wie bereits in Abschnitt 2.1.1 beschrieben, ist in OPTISIM[®] ein Homotopieverfahren implementiert, das die Änderungen in den Designvariablen von p_{alt} nach p_{neu} entlang der direkten Verbindungsstrecke unterteilt. Gibt es entlang dieser Verbindungsstrecke jedoch Stellen, an denen das Modellgleichungssystem (1.3) aufgrund von Unstetigkeiten oder Singularitäten (vgl. Abschnitt 2.1.2 und 2.1.4) nicht gelöst werden kann, so wird der neue Punkt in den Designvariablen p_{neu} nicht erreicht.

In [46] wurde ein Auswertungsalgorithmus entwickelt, der ausgehend von einem Punkt p_{alt} im Raum der betrachteten Designvariablen mit bekannter stationärer Lösung $\xi(p_{alt})$ die Zustandsvariablen $\xi(p_{neu})$ für einen neuen Punkt p_{neu} berechnet und dabei einzelne Punkte der Designvariablen, zu denen keine stationäre Lösung berechnet werden kann, umgeht. Anstatt die Auswertung der stationären Modellgleichungen nur auf der direkten Verbindungsstrecke vorzunehmen, werden jetzt verschiedene „Umwege“ zugelassen.

5.1.1 Gitter und Graphen

Betrachtet man in jeder Komponente die Werte von p_{alt} und p_{neu} , so lässt sich zwischen den beiden Punkten ein n_p -dimensionaler Quader im Raum der betrachteten Designvariablen aufspannen. Erlaubt man in den Komponenten, in denen sich die Werte von p_{alt} und p_{neu} unterscheiden, zusätzliche Stützstellen zwischen den beiden gegebenen Werten, so erhält man ein Gitter. Die Knoten dieses Gitters repräsentieren jeweils eine Kombination aus je einem der erlaubten Werte in den einzelnen Komponenten. Die Kanten des Gitters verlaufen parallel zu den Koordinatenachsen. Die Nachbarn eines Knotens sind all die Knoten, die über eine Kante mit dem ersten Knoten verbunden sind. Die Koordinaten zweier Nachbarn unterscheiden sich daher immer nur in einer Komponente.

Ein Gitter kann als Graph aufgefasst und darauf Algorithmen aus der Graphentheorie angewendet werden. Dazu seien zunächst einige grundlegende Begriffe für Graphen definiert (vgl. z.B. [55, 21]).

Definition 5.1 (Graphen)

Ein **Graph** ist ein Tripel, das aus einer Knotenmenge V , einer Kantenmenge U und einer Relation, die jeder **Kante** aus der Kantenmenge zwei **Knoten** aus der Knotenmenge zuordnet, besteht. Zwei Knoten heißen **Nachbarn**, falls sie die Endpunkte einer Kante aus U sind. Ein Graph heißt **endlich**, wenn die Knotenmenge U und die Kantenmenge K endlich sind.

Eine **Schlinge** ist eine Kante, deren Endpunkte gleich sind. Gibt es zwischen zwei Punkten mehrere Kanten, so nennt man diese **Mehrfachkanten**. Ein Graph heißt **einfach**, falls er keine Schlingen und keine Mehrfachkanten besitzt.

Ein **Pfad** ist eine Liste $v_0, u_1, v_1, \dots, u_k, v_k$ ($k \in \mathbb{N}$) von Knoten $v_i \in V$ und Kanten $u_i \in U$, wobei für alle $1 \leq i \leq k$ die Kante u_i die beiden Knoten v_{i-1} und v_i verbindet. Ein Pfad, in dem alle auftretenden Knoten paarweise verschieden sind, heißt **Weg**. Ein Graph heißt **zusammenhängend**, falls zu jedem Paar an Knoten aus V ein Pfad existiert, zu dem beide Knoten gehören.

Ein **gerichteter Graph** besteht aus einer Knotenmenge V , einer Kantenmenge U und einer Relation, die jeder Kante aus der Kantenmenge ein geordnetes Paar von Knoten aus der Knotenmenge zuordnet.

Ein **gewichteter Graph** ist ein Graph, für den eine zusätzliche Funktion $k_G : U \rightarrow \mathbb{R}_0^{(+)}$ existiert, die jeder Kante ein (nicht negatives) Gewicht zuordnet. Die Wegkosten $c_G(v_1, v_2)$ eines Weges von $v_1 \in V$ nach $v_2 \in V$ ergeben sich hierbei als Summe der Kantengewichte k_G der im Weg enthaltenen Kanten.

5.1.2 Der A*-Algorithmus

Der A*-Algorithmus ist ein Verfahren zur Bestimmung des kürzesten Weges zwischen zwei Knoten eines Graphen mit positiven Kantengewichten, das beispielsweise in [31] und [19] vorgestellt wird. Der A*-Algorithmus ist mit dem Dijkstra-Algorithmus [55, 19] verwandt. Während der Dijkstra-Algorithmus nur die bisherigen Wegkosten berücksichtigt und dabei gleichmäßig in alle Richtungen sucht, werden beim A*-Algorithmus zusätzlich zu den Wegkosten auch die (geschätzten) Kosten zum Ziel betrachtet [19].

Die Bewertungsfunktion $f_G : V \rightarrow \mathbb{R}_0^+$ des A*-Algorithmus setzt sich aus den bisherigen Wegkosten vom Startpunkt v_0 aus $g_G(\cdot) = c_G(v_0, \cdot)$ und einer heuristischen Funktion $h_G : V \rightarrow \mathbb{R}_0^+$, die die Kosten bis zum Ziel schätzt, zusammen. Für die Bewertung eines Knotens $v \in V$ gilt dann $f_G(v) = g_G(v) + h_G(v)$.

Der Ablauf des A*-Algorithmus wird in Algorithmus 5.1 dargestellt und kann folgendermaßen zusammengefasst werden [31, 46]: Der A*-Algorithmus verwendet zwei Listen. Liste 1 enthält bereits besuchte Knoten, Liste 2 enthält bereits bewertete Knoten, die noch zur Überprüfung (ob sie Zielknoten sind) anstehen. Zunächst wird der Startknoten in Liste 1 aufgenommen und expandiert, d.h. seine Nachbarn werden bestimmt und bewertet. Die Nachbarn werden in Liste 2 aufgenommen und nach ihrer Bewertung sortiert. Nun wird immer der Knoten aus Liste 2 mit der geringsten (besten) Bewertung in Liste 1 verschoben. Ist dieser Knoten bereits der Zielknoten, so wird die Suche beendet. Andernfalls wird dieser Knoten expandiert und alle Nachbarn werden in Liste 2 aufgenommen, falls sie dort noch nicht enthalten sind. Wird ein Nachbar bereits in Liste 2 mit einer größeren (schlechteren) Bewertung geführt, so wird seine Bewertung überschrieben.

Die Wahl der Heuristik h_G beeinflusst sowohl die Eigenschaften als auch die Geschwindigkeit des A*-Algorithmus (vgl. [31] und [19]).

Definition 5.2 (zulässige und monotone Heuristik)

Die Heuristik $h_G : V \rightarrow \mathbb{R}_0^+$ heißt **zulässig**, wenn sie die exakten Kosten h_G^* nicht überschätzt, d.h. $\forall v \in V \ h_G(v) \leq h_G^*(v)$.

Die Heuristik $h_G : V \rightarrow \mathbb{R}_0^+$ heißt **monoton**, wenn für alle Knoten $v \in V$ und für jeden Nachbarn $v' \in V$ von v gilt $h_G(v) \leq c_G(v, v') + h_G(v')$.

Algorithmus 5.1 Der A*-Algorithmus

Setze Startknoten mit Bewertung 0 in Liste 2.

while Liste 2 nicht leer **do**

 Verschiebe Knoten mit der geringsten Bewertung aus Liste 2 in Liste 1.

if Knoten ist Zielknoten **then**

 Fertig. Ziel gefunden.

end if

for Jeden Nachbarn des Knotens **do**

if Nachbar noch nicht in Liste 2 **then**

 Füge Nachbar aus Liste 2 mit Bewertung über den aktuellen Knoten ein.

 Sortiere Liste 2 anhand der Bewertungen.

else

if Bewertung über aktuellen Knoten zu diesem Nachbarn kleiner als Bewertung des Nachbarn aus Liste 2 **then**

 Aktualisiere Bewertung des Nachbarn.

 Sortiere Liste 2 anhand der Bewertungen.

end if

end if

end for

end while

Man sieht leicht, dass eine monotone Heuristik auch immer zulässig ist.

Für den A*-Algorithmus gelten folgende Eigenschaften:

- Der A*-Algorithmus terminiert auf endlichen Graphen.
- Ist die Heuristik h_G zulässig, so findet der A*-Algorithmus immer einen kostenoptimalen Pfad
- Ist die Heuristik h_G monoton, so ist der A*-Algorithmus optimal effizient, d.h. es gibt keinen anderen Algorithmus, der für die gegebene Heuristik schneller eine Lösung findet.

5.1.3 Der Wegfindungsalgorithmus in OPTISIM[®]

Das oben beschriebene Gitter, das die Wegfindung zwischen p_{alt} und p_{neu} ermöglichen soll, ist ein einfacher, zusammenhängender, endlicher, ungewichteter und ungerichteter Graph. Aufgrund der Schwierigkeiten bei der Berechnung der stationären Lösung kann es jedoch sein, dass nicht alle Knoten des Gitters besucht und nicht alle Kanten

begangen werden können. Dadurch kann es passieren, dass der Algorithmus in eine Sackgasse läuft und wieder zurück muss. Um in diesem Fall nicht wieder von vorn beginnen zu müssen, soll an geeigneten Knoten des Gitters die zu den entsprechenden Designvariablen gehörende stationäre Lösung gespeichert werden. Diese kann dann bei Bedarf wieder geladen werden, um die stationäre Simulation für die dem nächsten Knoten entsprechenden Designvariablen zu ermöglichen. War die letzte Berechnung erfolgreich, so stehen die Ergebnisse dieser Rechnung in der OPTISIM[®]-Datenbank zur Verfügung und müssen nicht geladen werden.

Der Wegfindungsalgorithmus, der in OPTISIM[®] implementiert wurde, beruht auf dem A*-Algorithmus und ist in Abbildung 5.1 schematisch dargestellt. Er wird in [46] folgendermaßen vorgestellt:

Das Gitter wird nicht vollständig angelegt, da der Speicherplatz für das gesamte Gitter sehr schnell zu groß werden würde (Zahl der Knoten wächst exponentiell mit der Dimension des Gitters) und während der Wegfindung meist nur ein Teil des Gitters durchlaufen wird. Die Knoten werden daher dynamisch erzeugt. Jeder Knoten besitzt folgende Attribute:

1. die Koordinaten; diese entsprechen nicht den exakten Werten der entsprechenden Designvariablen, sondern einer internen, ganzzahligen Nummerierung der Stützstellen
2. den Status, ob der Knoten bereits besucht wurde oder nicht,
3. die Bewertung des Knotens und
4. die Liste seiner Nachbarn.

Wird ein Knoten während der Suche expandiert, so werden alle Nachbarn, die noch nicht angelegt wurden, erzeugt. Dazu muss geprüft werden, ob und welcher Nachbar bereits existiert.

Ein Knoten wird dann als besucht markiert, wenn die stationäre Lösung für die entsprechenden Designvariablen berechnet werden konnte. Kann die Kante von einem Knoten zu seinem Nachbarn nicht begangen werden, so wird dieser Nachbar aus der Liste der Nachbarn gelöscht. Wurde ein Nachbar bereits besucht, so wird dieser Nachbar ebenfalls aus der Liste der Nachbarn gelöscht, um Endlosschleifen zu verhindern.

Alle angelegten Knoten, die noch nicht untersuchte Nachbarn haben, werden in einer Knotenliste gespeichert. Diese Knoten versprechen noch einen möglichen Weg, die Bewertung kann jedoch schlechter werden. Für jeden Knoten in der Knotenliste wird der zu den entsprechenden Designvariablen gehörende Zustandsvektor gespeichert. Wird

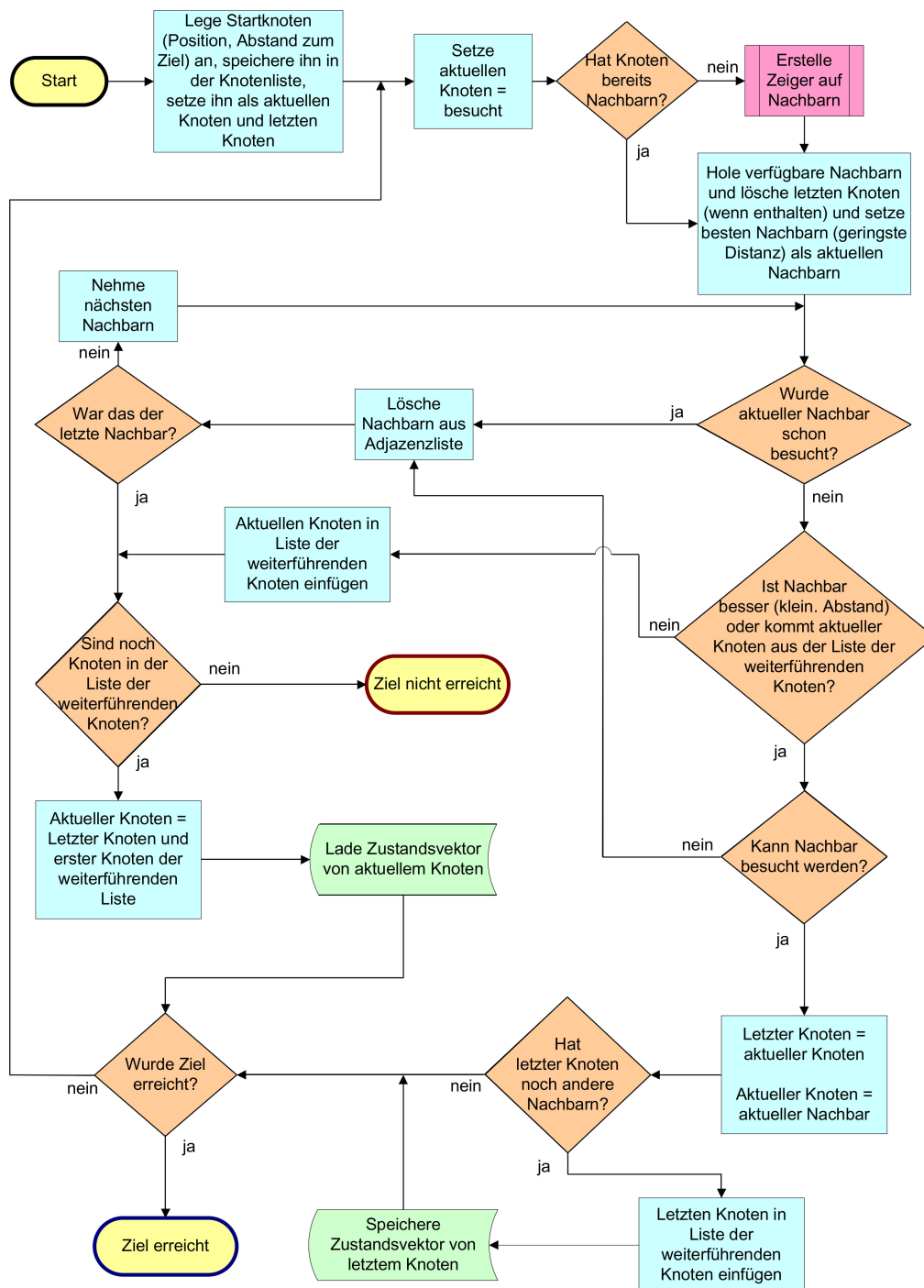


Abbildung 5.1 Schematische Darstellung des Wegfindungsalgorithmus in OPTISIM[®], aus: [46]

zu diesem Knoten später zurückgekehrt, wird der entsprechende Zustandsvektor wieder geladen. Sowohl die Knotenliste als auch die Listen der Nachbarn der einzelnen Knoten werden nach der Bewertung der in der Liste befindlichen Knoten aufsteigend sortiert.

Bei der hier gewählten Bewertung der einzelnen Knoten $f_G(v) = g_G(v) + h_G(v)$ werden die bisherigen Kosten vernachlässigt, d.h. $g_G \equiv 0$, und die geschätzten Kosten werden gleich dem quadrierten Abstand des Knotens vom Zielknoten gesetzt, d.h. $f_G(v) = h_G(v) = \|v - v_{Ziel}\|_2^2$. Zur Berechnung des Abstands zweier Knoten werden die internen, ganzzahligen Koordinaten verwendet.

Der Algorithmus beginnt mit dem Startknoten, der in die Knotenliste aufgenommen und expandiert wird. Dabei werden alle seine Nachbarn angelegt und bewertet. Es wird immer der Nachbar mit der besten Bewertung ausgewählt und es ergeben sich folgende Möglichkeiten:

- Ist dieser Nachbar noch nicht besucht worden, kann vom aktuellen Knoten aus besucht werden und hat einen geringeren Abstand zum Ziel, also eine bessere Bewertung, als der aktuelle Knoten, so wird dieser Nachbar als nächstes expandiert. Falls der Knoten noch weitere Nachbarn besitzt, so wird der Knoten in die Knotenliste aufgenommen.
- Kann der Nachbar zwar besucht werden und ist auch noch nicht besucht worden, hat jedoch eine schlechtere Bewertung als der aktuelle Knoten, so wird der aktuelle Knoten in die Knotenliste aufgenommen. Als nächster aktueller Knoten wird daraufhin der beste Knoten aus der Knotenliste genommen und aus der Liste entfernt. Der nächste Nachbar, der besucht werden kann, darf in diesem Fall auch dann als nächster aktueller Knoten gesetzt werden, wenn dieser eine schlechtere Bewertung als der aktuelle Knoten hat.
- Wurde der Nachbar schon besucht oder kann er nicht besucht werden, dann wird dieser Nachbar aus der Liste der Nachbarn des aktuellen Knotens gelöscht. Anschließend wird mit dem nächsten Nachbarn fortgefahren. Falls der aktuelle Knoten keinen Nachbarn mehr hat, wird wieder der beste Knoten aus der Knotenliste genommen und aus der Liste entfernt. In diesem Fall wurde eine Sackgasse erreicht.

Falls ein Knoten aus der leeren Knotenliste gewählt werden soll, bricht der Algorithmus ab, da er den Zielknoten nicht erreichen kann. Anderenfalls endet der Algorithmus mit dem gefunden Zielknoten.

Der hier vorgestellte Algorithmus hat folgende Eigenschaften: Erstens läuft der Algorithmus im Optimalfall, d.h. wenn alle Kanten begangen werden können, auf dem schnellsten Weg zum Ziel, da er dann immer den Nachbarn mit der besten Bewertung auswählt.

Zweitens wurde das Begehen einer Kante über das Homotopieverfahren realisiert. Dadurch steigt die Wahrscheinlichkeit, dass die Kante erfolgreich begangen werden kann. Und drittens wird bevorzugt von dem Knoten, der gerade erreicht wurde, weitergegangen. Dies bedeutet, dass im Falle einer erfolgreichen stationären Simulation für die dem Knoten entsprechenden Designvariablen das Ergebnis als Startwert für die nächste Berechnung direkt aus der OPTISIM[®]-Datenbank verwendet werden kann. Das aufwändige Lesen der durch den Wegalgorithmus gespeicherten Werte entfällt so in vielen Fällen.

5.1.4 Test des Wegfindungsalgorithmus

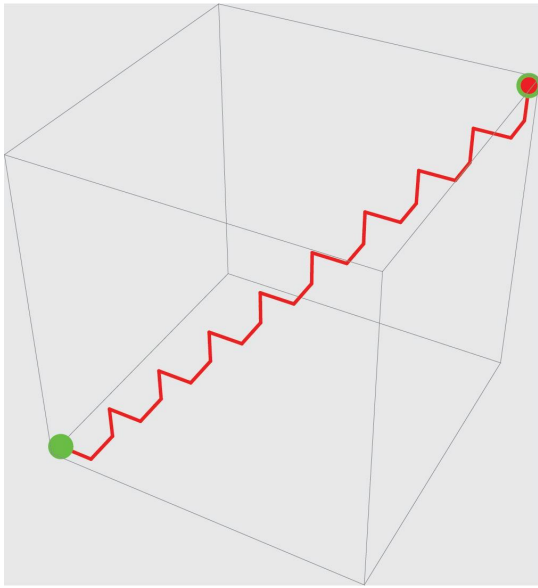


Abbildung 5.2 Wegfindung durch ein dreidimensionales Gitter, bei dem alle Kanten begehbar sind, aus: [46]

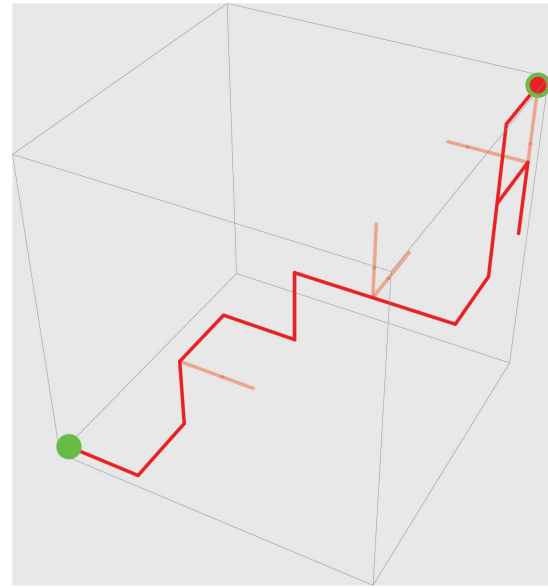


Abbildung 5.3 Wegfindung durch ein dreidimensionales Gitter, bei dem nicht alle Kanten begehbar sind, aus: [46]

Um den Wegfindungsalgorithmus unabhängig von OPTISIM[®] testen zu können, wurde in [46] eine Testroutine entwickelt, die die Konvergenzprobleme bei der stationären Simulation in OPTISIM[®] simuliert. Eine Funktion liefert in Abhängigkeit von den Koordinaten der Gitterpunkte, ob eine Kante von einem Knoten zu einem anderen Knoten begehbar ist oder nicht. Diese Funktion summiert zunächst die Koordinaten des ersten Knotens und addiert dazu die doppelte Summe der Koordinaten des zweiten Punktes. Ist diese Summe modulo einer vorgegebene Zahl $M \in \mathbb{N}$ ungleich 0, so ist die Kante begehbar, andernfalls nicht. Die Wahrscheinlichkeit der simulierten Konvergenzschwierigkeiten kann durch die Wahl der Zahl M beeinflusst werden.

Die Abbildungen 5.2 und 5.3 zeigen die Ergebnisse der Wegfindung für ein dreidimensionales Gitter von einem Eckknoten aus zum diagonal gegenüberliegenden Eckknoten. Im ersten Fall sind alle Kanten begehbar und der Algorithmus findet ohne Probleme einen kürzesten Weg. Im zweiten Fall können jedoch einige Kanten nicht begangen werden. Die nicht begehbaren Kanten, die der Algorithmus versucht zu begehen, sind in Abbildung 5.3 hell eingezeichnet. An diesem Beispiel lässt sich die Arbeitsweise der Wegfindung gut erkennen. An den ersten beiden Knoten, an denen die bevorzugte Kante nicht begangen werden kann, wird eine andere Kante gefunden, an deren Endknoten der Abstand zum Ziel dennoch kleiner ist als an dem vorherigen Knoten. Beim dritten problematischen Knoten, wird nach den ersten beiden Misserfolgen die dritte Kante versucht. Allerdings wird der Abstand zum Ziel dabei größer als der Abstand des letzten Knotens, der noch unbesuchte Nachbarn besitzt. Daher springt der Algorithmus zu diesem Knoten zurück und findet dann von dort aus zum Ziel.

5.2 Globaler Ansatz in OPTISIM[®]

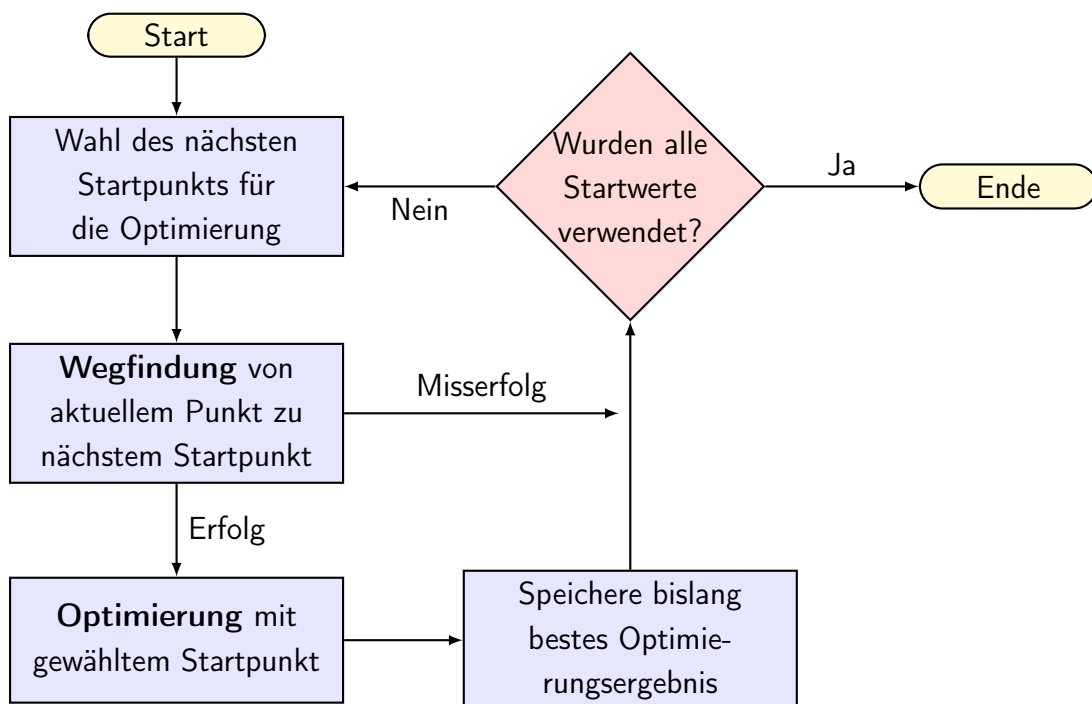


Abbildung 5.4 Schematische Darstellung des globalen Ansatzes in OPTISIM[®]

In [44] wurde ein Algorithmus implementiert, der das lokale, sequentielle Optimierungsverfahren mehrfach hintereinander von unterschiedlichen Startwerten für die

Designvariablen p aus startet. Der schematische Ablauf dieses globalen Algorithmus ist in Abbildung 5.4 dargestellt.

Zu Beginn der lokalen Optimierung muss die stationäre Lösung für den Startwert in den Designvariablen vorliegen. Diese kann mit Hilfe des Wegfindungsalgorithmus vom aktuellen Wert der Designvariablen aus berechnet werden. Ist das Erreichen des Startwertes jedoch nicht möglich, so wird der Optimierungslauf ausgelassen.

Die in [44] beschriebenen Details des Verfahrens werden in den folgenden Abschnitten erläutert.

5.2.1 Die Reihenfolge der Startwerte

Die verschiedenen Startwert p_{Init} seien durch die Menge \mathcal{P}_{Init} gegeben. Die Reihenfolge, in der diese Startwerte abgearbeitet werden, wird nicht im Vorhinein festgelegt, sondern abhängig vom Ergebnis der vorhergehenden Rechnung bestimmt. Es wird immer der Startwert aus \mathcal{P}_{Init} gewählt, der den geringsten Abstand zum aktuellen Wert der Designvariablen hat. Durch diese dynamische Wahl der Reihenfolge können unnötige „Durchquerungen“ des Raums der Designvariablen verhindert und so Rechenzeit bei der Wegfindung gespart werden. Wurde die Optimierung von einem Startwert aus gestartet, so wird dieser Startwert aus \mathcal{P}_{Init} entfernt. Das Verfahren läuft solange, bis die Menge \mathcal{P}_{Init} leer ist.

Abbildung 5.5 zeigt den Vergleich der Wege und ihrer Länge für ein zweidimensionales Beispiel mit zwei Minima. Verwendet man die Startwerte 1-4 in der Reihenfolge ihrer Nummerierung (Version A), werden Wege mit einer Gesamtlänge von 36.1 Längeneinheiten zurückgelegt. Wird immer der nächstliegende Startwert verwendet (Version B), so beträgt die Gesamtlänge der Wege nur 21.3 Längeneinheiten [44].

5.2.2 Checkpoints

Verläuft eine Optimierung oder eine Wegfindung nicht erfolgreich, so kann es sein, dass sie an einem Punkt der Designvariablen endet, für den keine stationäre Lösung gefunden wurde. Von diesem Punkt aus kann dann weder eine Wegfindung noch eine Optimierung gestartet werden. Aus diesem Grund ist es sinnvoll, wie bereits beim Wegfindungsalgorithmus, an bestimmten Punkten, sogenannten *checkpoints*, die Designvariablen mit zugehörigen Zustandsvariablen zwischenzuspeichern.

Wurde ein Startpunkt p_{Init} erfolgreich durch die Wegfindung erreicht, so wird dieser Zustand immer als *checkpoint* gespeichert. Wurde eine Optimierung erfolgreich beendet, so wird dieser Zustand $(\xi(p_{Opt}), p_{Opt})$ dann als *checkpoint* gespeichert, falls die kleinste

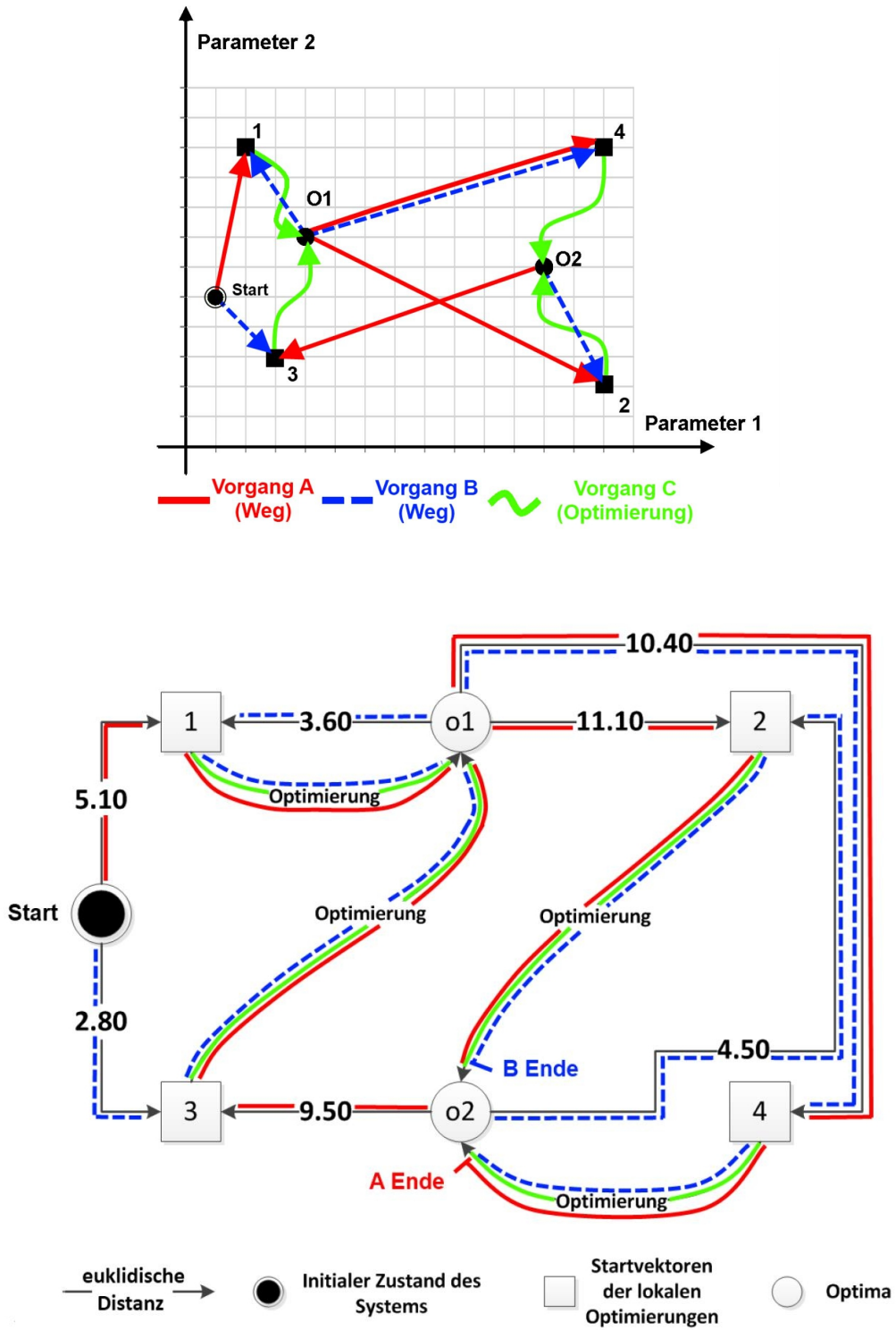


Abbildung 5.5 Vergleich der zurückgelegten Wege mit und ohne dynamische Wahl der Startpunkte, aus: [44]

Distanz von p_{Opt} zu den verbleibenden Startwerten in \mathcal{P}_{Init} kleiner ist als von dem gerade verwendeten Startwert p_{Init} , also wenn gilt

$$\min_{p \in \mathcal{P}_{Init}} \|p - p_{Opt}\| < \min_{p \in \mathcal{P}_{Init}} \|p - p_{Init}\|.$$

Wurde eine Optimierung erfolgreich beendet und deren Ergebnis als *checkpoint* gespeichert, so startet die nächste Wegfindung direkt von p_{Opt} aus. Wurde das Ergebnis nicht gespeichert oder konnte die Optimierung nicht erfolgreich beendet werden, so wird der Zustand des gerade verwendeten Startwert p_{Init} geladen und die Wegfindung beginnt dort.

Endet eine Wegfindung nicht erfolgreich, so wird die Optimierung nicht gestartet, der entsprechende Optimierungsstartwert aus \mathcal{P}_{Init} gelöscht und der letzte *checkpoint*, d.h. der aktuelle Startpunkt der Wegfindung, geladen.

5.2.3 Gitterverfeinerung

Um Rechenzeit während der Wegfindung zu sparen, werden so wenig Stützstellen wie möglich verwendet. Der Algorithmus beginnt daher immer mit einer Stützstelle pro Designvariable. Kann das Ziel so nicht erreicht werden, dann wird der *checkpoint* des Startpunktes der Wegfindung geladen, die Anzahl der Stützstellen verdoppelt und die Wegfindung erneut aufgerufen. Das Gitter wird so lange verfeinert, bis entweder die maximale Anzahl an Stützstellen überschritten oder das Ziel der Wegfindung erreicht wurde.

5.2.4 Ergebnisse

Der eben beschriebene Algorithmus wurde in [44] unter anderem an Hand des Modells einer Erdgasverflüssigungsanlage getestet. Dabei soll der Energieverbrauch der Verdichter minimiert werden, indem der Druck und die Zusammensetzung des Kühlmittels variiert werden. Das stationäre Modell besteht aus 2538 Gleichungen und Zustandsvariablen. Das Optimierungsproblem besitzt 13 Optimierungsvariablen und 108 Ungleichungsnebenbedingungen mit oberer und unterer Schranke. Es wurden fünf Startpunkte gewählt, die im Folgenden S1 - S5 genannt werden.

Das globale Verfahren wurde sowohl mit der festen Reihenfolge als auch mit der in Abschnitt 5.2.1 beschriebenen dynamischen Wahl der Reihenfolge der Startpunkte gestartet.

In Tabelle 5.1 sind die Rechenzeiten allein für die Wegfindungen, ohne die der Optimierungen, aufgelistet. Der Startpunkt S4 kann nicht erreicht werden. Der Versuch diesen Punkt mit Hilfe der Gitterverfeinerung doch noch zu erreichen, benötigt die meiste Zeit.

Feste Reihenfolge:					
Ziel der Wegfindung	S1	S2	S3	S4	S5
Zeit für Wegfindung [s]	44	24	37	15632	35
Ergebnis der Wegfindung	erreicht	erreicht	erreicht	nicht erreicht	erreicht
Gesamtzeit [s]	15771				
Gesamtzeit ohne S4 [s]	139				
Dynamische Reihenfolge:					
Ziel der Wegfindung	S2	S4	S5	S3	S1
Zeit für Wegfindung [s]	33	2751	38	16	12
Ergebnis der Wegfindung	erreicht	nicht erreicht	erreicht	erreicht	erreicht
Gesamtzeit [s]	2849				
Gesamtzeit ohne S4 [s]	99				

Tabelle 5.1 Vergleich der Rechenzeiten für die Wegfindung bei fester und dynamischer Reihenfolge der Startwerte

In beiden Fällen wird nach dem erfolglosen Versuch, S4 zu erreichen, der *checkpoint* des letzten Optimierungsergebnisses geladen und von dort aus die nächste Wegfindung gestartet. Sowohl mit als auch ohne den Versuch, S4 zu erreichen, ist die benötigte Rechenzeit bei der dynamischen Reihenfolge geringer als bei der festen Reihenfolge.

	S2	S5	S3	S1
Startwerte				
Variable 1	4.0	6.0	8.0	10.0
Variable 2	25.0	50.0	75.0	100.0
Variable 3	25.0	50.0	75.0	100.0
Variable 4	25.0	50.0	75.0	100.0
Ergebnisse				
Variable 1	5.3624	5.5431	5.4833	5.3479
Variable 2	5.3144	5.4929	8.6043	9.7579
Variable 3	24.6705	25.2744	39.0933	44.9314
Variable 4	22.9122	22.1174	36.9378	43.0971
Zielfunktion	1.5715	1.5756	1.5708	1.5700

Tabelle 5.2 Ergebnisse der Optimierung von den unterschiedlichen Startpunkten aus

Die Ergebnisse der Optimierungen von den erreichten Startpunkten S1, S2, S3 und S5 aus sind in Tabelle 5.2 dargestellt.

Literaturverzeichnis

- [1] *Aspen HYSYS*[®]. <http://www.aspentech.com/core/asp-hsys.aspx>. – AspenTech
- [2] *Aspen Plus*[®]. <http://www.aspentech.com/core/asp-plus.aspx>. – AspenTech
- [3] *CHEMCAD*. <http://www.chemstations.com/>. – Chemstations
- [4] *Computational Infrastructure for Operations Research*. <http://www.coin-or.org/>
- [5] *Dymola*. <http://www.3ds.com/products/catia/portfolio/dymola>. – Dassault Systèmes
- [6] *GNU Linear Programming Kit*. <http://www.gnu.org/software/glpk/>
- [7] *gPROMS*. <http://www.psenterprise.com/gproms/>. – Process Systems Enterprise Limited (PSE)
- [8] *IPOPT Homepage*. <https://projects.coin-or.org/Ipop>
- [9] *PRO/II*. http://iom.invensys.com/EN/Pages/SimSci-Esscor_ProcessEngSuite_PROII.aspx. – Invensys
- [10] *ProSimPlus*. <http://www.prosim.net/en/modeling/prosimplus.html>. – ProSim
- [11] *Unisim*[®] *Design*. <https://www.honeywellprocess.com/en-US/explore/products/advanced-applications/process-design/Pages/unisim-design-suite.aspx>. – Honeywell
- [12] ANDERSEN, P. ; GENOVESE, F. ; PEEREGAARD, J. : *Manual for Steady State Simulator, SEPSIM*. Lyngby, Denmark: Institut for Kemiteknik, DTH, 1991
- [13] BAEHR, H. D. ; KABELAC, S. : *Thermodynamik*. 15. Auflage. Springer-Verlag, 2012
- [14] BIEGLER, L. ; HUGHES, R. : Infeasible path optimization with sequential modular simulators. In: *AIChE Journal* 28(6) (1982), S. 994–1002

- [15] BIEGLER, L. T. ; CERVANTES, A. ; WÄCHTER, A. : Advances in simultaneous strategies for dynamic process optimization. In: *Chemical Engineering Science* 57(4) (2002), S. 575–593
- [16] BIEGLER, L. T. ; NOCEDAL, J. ; SCHMID, C. : A reduced Hessian method for large-scale constrained optimization. In: *SIAM Journal on Optimization* 5 (1995), Nr. 2, S. 314–347
- [17] BIEGLER, L. T. ; NOCEDAL, J. ; SCHMID, C. ; TERNET, D. : Numerical experience with a reduced Hessian method for large scale constrained optimization. In: *Computational Optimization and Applications* 15 (2000), S. 45–67
- [18] BIEGLER, L. T. ; WÄCHTER, A. : SQP SAND strategies that link to existing modeling systems. In: BIEGLER, L. T. (Hrsg.) ; GHATTAS, O. (Hrsg.) ; HEINKENSCHLOSS, M. (Hrsg.) ; BLOEMEN WAANDERS, B. van (Hrsg.): *Large-scale PDE-constrained optimization* Bd. 30. Springer, 2003, S. 199–217
- [19] BUNGARTZ, H.-J. ; ZIMMER, S. ; BUCHHOLZ, M. ; PFLÜGER, D. : *Modellbildung und Simulation, Eine anwendungsorientierte Einführung*. Springer-Verlag, 2009
- [20] DEUFLHARD, P. ; HOHMANN, A. : *Numerische Mathematik 1*. 2. Auflage. de Gruyter, 1993
- [21] DIESTEL, R. : *Graphentheorie*. 4. Auflage. Springer-Verlag, 2010
- [22] DURAN, M. ; GROSSMANN, I. : An outer-approximation algorithm for a class of mixed-integer nonlinear programs. In: *Mathematical Programming* 36 (1986), S. 307–339
- [23] EDGAR, T. F. ; HIMMELBLAU, D. M. ; LASDON, L. S.: *Optimization of chemical processes*. 2. Auflage. McGraw-Hill, 2001
- [24] EICH-SOELLNER, E. ; LORY, P. ; BURR, P. ; KRÖNER, A. : Stationary and dynamic flowsheeting in the chemical engineering industry. In: *Surveys on Mathematics for Industry* 7 (1997), S. 1–28
- [25] ENGL, G. ; KRÖNER, A. : Success Factors for CAPE in the Engineering Practice of a Process Plant Contractor. In: MARQUARDT, W. (Hrsg.) ; PANTELIDES, C. (Hrsg.): *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering* Bd. 21. Elsevier, 2006, S. 763–768

- [26] FLETCHER, R. ; GOULD, N. I. M. ; LEYFFER, S. ; TOINT, P. L. ; WÄCHTER, A. : Global Convergnence of a Trust-Region SQP-Filter Algorithm for General Nonlinear Programming. In: *SIAM Journal on Optimization* 13 (2002), Nr. 3, 635-659. ftp://ftp.numerical.rl.ac.uk/pub/ning/pubs/FletGoulLeyfToinWach02_siopt.pdf
- [27] FORSGREN ; GILL ; WRIGHT: Interior methods for nonlinear optimization. In: *SIAM Review* 44 (2002), Nr. 4, S. 525–597
- [28] GEIGER, C. ; KANZOW, C. : *Numerische Verfahren zur Lösung unrestringierter Optimierungsaufgaben*. Springer Verlag, 1999
- [29] GEIGER, C. ; KANZOW, C. : *Theorie und Numerik restringierter Optimierungsaufgaben*. Springer Verlag, 2002
- [30] GOLUB, G. H. ; LOAN, C. F.: *Matrix computations*. 2. Auflage. John Hopkins University Press, 1989
- [31] GÖRZ, G. ; ROLLINGER, C.-R. ; SCHNEEBERGER, J. : *Handbuch der künstlichen Intelligenz*. 4. Auflage. Oldenbourg Wissenschaftsverlag, 2003
- [32] HAFTKA, R. T.: Simultaneous Analysis and Design. In: *AIAA Journal* 23 (1985), S. 1099–1103
- [33] KARMAKAR, N. : A new polynomial-time Algorithm for linear Programming. In: *Combinatorica* 4 (1984), Nr. 4, 373-395. <http://retis.sssup.it/~bini/teaching/optim2010/karmarkar.pdf>
- [34] KAWAJIR, Y. ; LAIRD, C. ; WÄCHTER, A. : *Introduction to IPOPT: A tutorial for downloading, installing and using IPOPT*. Computational Infrastructure for Operations Research, 30. Juni 2011. <http://www.coin-or.org/download/source/Ipopt/>. – Manual zum Installationspaket IPOPT
- [35] KRÖNER, A. : *Algorithmen zur Analyse und Lösung kontinuierlich-diskreter differentiell-algebraischer Prozessmodelle in der Simulationsumgebung DIVA*. VDI Verlag Düsseldorf, 2003 (Fortschritt-Berichte VDI: Reihe 20, Nr. 369)
- [36] LINDE AG, ENGINEERING DIVISION (Hrsg.): *GMPS Benutzerhandbuch*. Linde AG, Engineering Division, 2008. – (unveröffentlicht)
- [37] MÖBIUS, H.-H. ; DÜRSELEN, W. : *Chemische Thermodynamik*. 5. Auflage. VEB Deutscher Verlag für Grundstoffindustrie, 1988

- [38] NEUMAIER, A. : Complete Search in Continuous Global Optimization and Constraint Satisfaction. Version: 2004. <http://www.mat.univie.ac.at/~neum/ms/glopt03.pdf>. In: ISERLES, A. (Hrsg.): *Acta Numerica 2004*. Cambridge University Press, 271-369
- [39] NOCEDAL, J. ; WÄCHTER, A. ; WALTZ, R. : Adaptive Barrier Update Strategies for Nonlinear Interior Methods. In: *SIAM Journal on Optimization* 19 (2008), Nr. 4, 1674-1693. <http://users.eecs.northwestern.edu/~rwaltz/articles/adaptive.pdf>
- [40] NOCEDAL, J. ; WRIGHT, S. : *Numerical Optimization*. 2. Auflage. Springer, 2006
- [41] OTTER, M. : Objektorientierte Modellierung physikalischer Systeme, Teil 1. In: *at-automatisierungstechnik* 47 (1999), S. A1-A4
- [42] POWELL, M. J. D.: A fast algorithm for nonlinearly constrained optimization calculations. In: WATSON, G. (Hrsg.): *Numerical Analysis (Proceedings of the 1977 Dundee Biennial Conference on Numerical Analysis)*, Springer-Verlag, 1978 (lecture notes in mathematics 630), S. 144-157
- [43] QUARTERONI, A. ; SACCO, R. ; SALERI, F. : *Numerische Mathematik 1*. Springer Verlag, 2002
- [44] SCHATTEL, D. : *Implementierung und Test eines globalen Optimierungsansatzes im Prozesssimulator OPTISIM[®]*, Hochschule München, Bachelorarbeit, 2011. – Linde AG
- [45] SCHMID, C. ; BIEGLER, L. T.: A simultaneous approach for flowsheet optimization with existing modelling procedures. In: *Trans. I. Chem. Eng.* 72A (1994), S. 382-388
- [46] SIVERS, I. von: *Algorithmen zur Gitterverwaltung von Designvariablen für einen globalen Optimierungsansatz im Prozesssimulator OPTISIM[®]*, Hochschule München, Bachelorarbeit, 2010. – Linde AG
- [47] SPELLUCCI, P. : A new technique for inconsistent QP problems in the SQP method. In: *Mathematical methods of operations research* 47 (1998), S. 355-400
- [48] THOMAS, I. ; KRÖNER, A. : Mixed-Integer Optimization of Distillation Column Tray Positions in Industrial Practice. In: MARQUARDT, W. (Hrsg.) ; PANTELIDES, C. (Hrsg.): *16th European Symposium on Computer Aided Process Engineering and 9th International Symposium on Process Systems Engineering* Bd. 21. Elsevier, 2006, S. 1015-1020

- [49] TONE, K. : Revisions of constraint approximations in the successive qp method for nonlinear programming problems. In: *Mathematical Programming* 26 (1983), S. 144–152
- [50] VISWANATHAN, J. ; GROSSMANN, I. E.: Optimal feed locations and number of trays for distillation columns with multiple feeds. In: *Ind. Eng. Chem. Res.* 32 (1993), S. 2942–2949
- [51] WÄCHTER, A. : *An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering*, Carnegie Mellon University, Diss., 2002. <http://users.iems.northwestern.edu/~andreasw/pubs/thesis.pdf>
- [52] WÄCHTER, A. ; BIEGLER, L. T.: Line search filter methods for nonlinear programming: Local convergence. In: *SIAM Journal on Optimization* 16 (2005), Nr. 1, S. 32–48
- [53] WÄCHTER, A. ; BIEGLER, L. T.: Line search filter methods for nonlinear programming: Motivation and global convergence. In: *SIAM Journal on Optimization* 16 (2005), S. 1–31
- [54] WÄCHTER, A. ; BIEGLER, L. T.: On the Implementation of an Interior-Point Filter Line-Search Algorithm for Large-Scale Nonlinear Programming. In: *Mathematical Programming* 106 (2006), Nr. 1, S. 25–57
- [55] WEST, D. B.: *Introduction to graph theory*. 2. Auflage. Prentice-Hall, 2001

Alle hier angeführten Links sind Stand 18. März 2013

Erklärung

Hiermit versichere ich, STEPHANIA HOKENMAIER, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe. Die Arbeit wurde in dieser oder ähnlicher Form noch keiner Prüfungskommission vorgelegt.

(STEPHANIA HOKENMAIER)